



INSTITUT TEKNOLOGI BANDUNG

PROGRAM STUDI TEKNIK ELEKTRO

JALAN GANESHA NO. 10 Gedung Labtek V Lantai 2 (022)2508135-36, (022)2500940
BANDUNG 40132

Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

Judul Dokumen

TUGAS AKHIR TEKNIK ELEKTRO:
*Perancangan Sistem Kecerdasan dan Keamanan
pada Perangkat Internet-of-Things Rumah Cerdas*

Jenis Dokumen

IMPLEMENTASI

Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB

Nomor Dokumen

B400-02-TA1617.01.056

Nomor Revisi

02

Nama File

B400

Tanggal Penerbitan

3 Mei 2017

Unit Penerbit

Prodi Teknik Elektro - ITB

Jumlah Halaman

410

(termasuk lembar sampul ini)

Data Pengusul

Ditulis Oleh	Nama Tanggal	Bryan Tandiawan 03 Mei 2017	Jabatan Tanda Tangan	Anggota Kelompok
	Nama Tanggal	Billy Austen Manangkalangi 03 Mei 2017	Jabatan Tanda Tangan	Anggota Kelompok
	Nama Tanggal	Revie Marthensa 03 Mei 2017	Jabatan Tanda Tangan	Anggota Kelompok
Disetujui Oleh	Nama Tanggal	Trio Adiono, S.T., M.T., Ph.D. 03 Mei 2017	Jabatan Tanda Tangan	Pembimbing

DAFTAR ISI

DAFTAR ISI.....	2
CATATAN SEJARAH PERBAIKAN DOKUMEN	4
1. PENGANTAR	5
1.1 RINGKASAN ISI DOKUMEN	5
1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN.....	5
1.3 REFERENSI.....	6
1.4 DAFTAR GAMBAR.....	7
1.5 DAFTAR TABEL	9
2. RINGKASAN PENJELASAN SISTEM SECARA KESELURUHAN.....	11
2.1 PENDAHULUAN.....	11
3. IMPLEMENTASI REMOTE CONTROL.....	13
3.1 IMPLEMENTASI HARDWARE REMOTE CONTROL.....	13
3.2 IMPLEMENTASI PROGRAM STM32 UNTUK REMOTE CONTROL.....	15
4. IMPLEMENTASI PROTOKOL KOMUNIKASI SUBSISTEM INDOOR	20
4.1 IMPLEMENTASI PROTOKOL PADA DEVICE SENSOR SUHU DAN KELEMBAPAN.....	20
4.2 IMPLEMENTASI PROTOKOL PADA LAMPU RGB.....	24
4.3 IMPLEMENTASI PROTOKOL PADA TIRAI.....	27
4.4 IMPLEMENTASI PROTOKOL PADA PINTU	31
4.5 IMPLEMENTASI PROTOKOL PADA KIPAS	34
4.6 IMPLEMENTASI PROTOKOL PADA SAKLAR.....	36
5. IMPLEMENTASI SERVER DAN RASPBERRY PI HOST	39
5.1 IMPLEMENTASI BASIS DATA.....	39
5.1.1 <i>Implementasi Basis Data pada Server</i>	39
5.1.2 <i>Implementasi Basis Data pada Host</i>	41
5.1.3 <i>Pengelolaan Basis Data dalam Bahasa Python</i>	48
5.2 IMPLEMENTASI PROGRAM PADA SERVER DAN HOST	49
5.2.1 <i>Impelementasi Environment RabbitMQ pada Program Python</i>	49

<i>5.2.2 Implementasi Sistem Keamanan Jaringan dalam Bahasa Python.....</i>	<i>50</i>
<i>5.2.3 Implementasi Program Python pada Server.....</i>	<i>51</i>
<i>5.2.4 Implementasi Program Python pada Host (Raspberry Pi).....</i>	<i>74</i>
6. IMPLEMENTASI ANDROID MOBILE APPLICATION	116
<i>6.1 HALAMAN UTAMA (LOGINACTIVITY.JAVA).....</i>	<i>120</i>
<i>6.2 HALAMAN SIGN IN (SIGNINACTIVITY.JAVA)</i>	<i>124</i>
<i>6.3 FORGET PASSWORD ACTIVITY (FORGETACTIVITY.JAVA)</i>	<i>134</i>
<i>6.4 HALAMAN SIGN UP (SIGNUPACTIVITY.JAVA)</i>	<i>138</i>
<i>6.5 HALAMAN MENU UTAMA (MAINACTIVITY.JAVA)</i>	<i>146</i>
<i>6.6 HALAMAN DEVICE (SWITCHACTIVITY.JAVA)</i>	<i>209</i>
<i>6.7 HALAMAN ROOM (ROOMACTIVITY.JAVA)</i>	<i>249</i>
<i>6.8 HALAMAN ADD DEVICE (ADDDEVICE1ACTIVITY.JAVA, ADDDEVICEACTIVITY.JAVA)</i>	<i>250</i>
<i>6.9 HALAMAN ADD ROOM (EDITROOMACTIVITY.JAVA, ADDDEVICEROOM.JAVA).....</i>	<i>261</i>
<i>6.10 HALAMAN ADD SCENARIO (ADDSCENARIOACTIVITY.JAVA)</i>	<i>278</i>
<i>6.11 HALAMAN NOTIFIKASI DAN PESAN (NOTIFICATIONACTIVITY.JAVA, MESSAGEACTIVITY.JAVA)</i>	<i>299</i>
<i>6.12 HALAMAN ANALISIS DATA (DATAACTIVITY.JAVA).....</i>	<i>309</i>
<i>6.13 HALAMAN MEMBER (CREATENEWACTIVITY2.JAVA)</i>	<i>317</i>
<i>6.14 HALAMAN ADD MEMBER (ADDMEMBERACITIVTY.JAVA).....</i>	<i>328</i>
<i>6.15 HALAMAN GETTINGSTARTED (GETTINGSTARTEDACTIVITY.JAVA)</i>	<i>336</i>
<i>6.16 HALAMAN GET INVITED (INVITEACTIVITY1.JAVA).....</i>	<i>339</i>
<i>6.17 HALAMAN SETTING (SETTINGACTIVITY.JAVA).....</i>	<i>347</i>
<i>6.18 HALAMAN USER (USERACTIVITY.JAVA)</i>	<i>359</i>
<i>6.19 HALAMAN INFORMASI RUMAH (CREATENEWACTIVITY1.JAVA).....</i>	<i>373</i>
<i>6.20 HALAMAN PRIVILEGE (PRIVILEGEACTIVITY.JAVA).....</i>	<i>393</i>
<i>6.21 HALAMAN ABOUT (ABOUTACTIVITY.JAVA)</i>	<i>398</i>
<i>6.22 HALAMAN HELP (PAGERACTIVITY.JAVA)</i>	<i>400</i>

Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	PENYUNTING	PERBAIKAN
2	1 Mei	Bryan	Penambahan penjelasan <i>device IR Remote</i> dan perubahan protokol komunikasi <i>hardware</i>
		Billy	Tambahan penjelasan <i>layout</i> aplikasi dan <i>screenshot</i> tampilan
		Revie	Pemindahan <i>source code</i> seluruh program ke penjelasan (bukan di lampiran) dan tambahan penjelasan kode

Implementasi Proyek Perancangan Sistem Kecerdasan dan Keamanan pada Perangkat Internet-of-Things Rumah Cerdas

1. Pengantar

1.1 RINGKASAN ISI DOKUMEN

Pada dokumen ini, dijabarkan beberapa hal penting mengenai implementasi proyek *Perancangan Sistem Kecerdasan dan Keamanan pada Perangkat Internet-of-Things Rumah Cerdas*. Beberapa hal penting tersebut meliputi:

- Bab Pengantar berisi ringkasan isi dokumen B400, tujuan penulisan dan aplikasi/kegunaan dokumen, referensi, dan daftar singkatan
- Bab penjelasan sistem secara keseluruhan yang berisi penjelasan ruang lingkup kerja dan pengantar untuk penjelasan mendetil berikutnya
- Bab Implementasi *Infrared Remote Control*
- Bab Implementasi Protokol Komunikasi Subsistem *Indoor* secara keseluruhan
- Bab implementasi Server dan *Host* yang berisi penjelasan implementasi basis data, protokol komunikasi, sistem keamanan jaringan, serta program-program pelayan data yang dijalankan di server dan *host*.
- Bab Implementasi *Android Mobile Application* yang berisi penjelasan implementasi aplikasi Android

Bab Pengantar disusun untuk memberi gambaran umum dari proyek ini dan membantu pembaca untuk memahami bab-bab selanjutnya. Bab Implementasi menjelaskan bagaimana proyek *Perancangan Sistem Kecerdasan dan Keamanan pada Perangkat Internet-of-Things Rumah Cerdas* dilakukan secara rinci dan mendalam. Dokumen B400 ini merupakan penerapan dan pemilihan desain dari dokumen B300 sebelumnya.

1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN

Tujuan penulisan dokumen ini adalah sebagai dokumentasi Tugas Akhir, yaitu proyek *Perancangan Sistem Kecerdasan dan Keamanan pada Perangkat Internet-of-Things Rumah Cerdas*. Dokumen B400 ini dapat dijadikan sebagai acuan dalam mengimplementasikan

sistem rumah cerdas dan dapat memberikan gambaran dan menjelaskan sistem rumah cerdas berbasis IoT secara spesifik dan mendetail.

1.3 REFERENSI

1. M.Y. Fathany and T. Adiono, Wireless protocol design for smart home on mesh wireless sensor network, “Int. Symp. on Intelligent Signal Processing and Communication System”, Bali, November 2015.
2. T. Adiono, R.V.W. Putra, M.Y. Fathany, M.A. Wibisono, W. Adijarto, “Smart home platform based on optimized wireless sensor network protocol and scalable architecture,” Proc. of 9th Int. Conf. On Telecommunication Systems Services and Applications, pp.1-5, November 2015.
3. <https://www.rabbitmq.com/tutorials/>
4. <https://www.cloudamqp.com/blog/2014-10-28-rabbitmq-on-android.html>
5. <https://developer.android.com/>
6. <https://www.mysql.com/>
7. <https://stackoverflow.com>

1.4 DAFTAR GAMBAR

Gambar 1 Ruang lingkup implementasi.....	12
Gambar 2 Diagram Blok Sistem Keseluruhan	13
Gambar 3 Skematik <i>remote control IrDA-based</i>	14
Gambar 4 PCB diagram <i>remote control IrDA-based</i>	14
Gambar 5 Hasil <i>remote control IrDA0based</i>	15
Gambar 6 Implementasi Tabel Users	40
Gambar 7 Impelementasi Tabel aeskeylist	41
Gambar 8 Implementasi Tabel info.....	42
Gambar 9 Implementasi Tabel devices	43
Gambar 10 Implementasi Tabel rooms	44
Gambar 11 Implementasi Tabel scenarios	44
Gambar 12 Implementasi Tabel dataMMYYYY.....	46
Gambar 13 Implementasi Tabel users.....	47
Gambar 14 Implementasi Tabel Isi Skenario.....	48
Gambar 15 Antrian Data ke Server	52
Gambar 16 Pie chart OS market share	117
Gambar 17 Logo Aplikasi MINDS: Smart House	118
Gambar 18 Flowchart Hubungan antar <i>Activity</i> dalam Aplikasi Android MINDS	119
Gambar 19 Halaman Utama Aplikasi (LoginActivity.java)	120
Gambar 20 Halaman Sign In (SignInActivity.java).....	124
Gambar 21 Halaman Forget Password.....	134
Gambar 22 Halaman Sign Up (SignUpActivity.java).....	138
Gambar 23 Dashboard.....	146
Gambar 24 Dashboard Room.....	146
Gambar 25 Dashboard Scenario.....	146
Gambar 26 Popup Menu Trash pada Room.....	146
Gambar 27 Popup Menu Trash pada Scenario.....	146
Gambar 28 Dashboard Add Button.....	146
Gambar 29 Dashboard locked.....	147
Gambar 30 Navigation Drawer	147
Gambar 31 Menu Curtain.....	211

Gambar 32 Menu Door	211
Gambar 33 Menu Fan.....	211
Gambar 34 Menu Lamp	211
Gambar 35 Dialog pada Menu Lamp.....	211
Gambar 36 Menu Switch	211
Gambar 37 Menu Temperature	212
Gambar 38 Halaman Room (RoomActivity.java)	249
Gambar 39 Halaman Add Device (AddDevice1Activity.java).....	251
Gambar 40 Halaman Nama Device (AddDeviceActivity.java).....	251
Gambar 41 Halaman Add Room (EditRoomActivity.java).....	262
Gambar 42 Halaman Add Device pada Room (AddDeviceRoom.java).....	262
Gambar 43 Halaman Add Scenario (AddScenarioActivity.java)	279
Gambar 44 Halaman Add Device (AddDeviceRoom.java).....	279
Gambar 45 Halaman Notifikasi (NotificationActivity.java).....	300
Gambar 46 Halaman Pesan (MessageActivity.java).....	300
Gambar 47 Halaman Analisis Data (DataActivity.java).....	310
Gambar 48 Halaman Member (CreateNewActivity2.java)	318
Gambar 49 Halaman Add Member (AddMemberActivity.java)	328
Gambar 50 Halaman Getting Started (GettingStartedActivity.java)	336
Gambar 51 Halaman Get Invited (InviteActivity1.java).....	339
Gambar 52 Halaman Setting (SettingActivity.java)	347
Gambar 53 Halaman User (UserActivity.java)	359
Gambar 54 Halaman Create New (CreateNewActivity1.java)	374
Gambar 55 Halaman Privilege Setting.....	394
Gambar 56 Halaman About (AboutActivity.java)	398
Gambar 57 Halaman Pager 1	400
Gambar 58 Halaman Pager 2	400
Gambar 59 Halaman Pager 3	400

1.5 DAFTAR TABEL

Tabel 1 Kolom-kolom Tabel users di <i>database general_data</i>	39
Tabel 2 Kolom-kolom Tabel aeskeylist di <i>database general_data</i>	40
Tabel 3 Kolom-kolom Tabel info di <i>database homeXXXXXX</i>	41
Tabel 4 Kolom-kolom Tabel devices di <i>database homeXXXXXX</i>	42
Tabel 5 Kolom-kolom Tabel rooms di <i>database homeXXXXXX</i>	43
Tabel 6 Kolom-kolom Tabel scenarios di <i>database homeXXXXXX</i>	44
Tabel 7 Kolom-kolom Tabel dataMMYYYY di <i>database homeXXXXXX</i>	44
Tabel 8 Fungsi pada server_rsa.py	57
Tabel 9 Pesan yang Dilayani oleh server_rsa.py	58
Tabel 10 Fungsi pada server_aes.py	70
Tabel 11 Pesan yang Dilayani server_aes.py	71
Tabel 12 Antrian Data ke <i>Host</i>	78
Tabel 13 Fungsi pada raspi_rsa.py	80
Tabel 14 Pesan yang Dilayani raspi_rsa.py	80
Tabel 15 Fungsi pada raspi_aes.py	84
Tabel 16 Pesan yang Dilayani raspi_aes.py Bagian 1	87
Tabel 17 Pesan yang Dilayani raspi_aes.py Bagian 2	89
Tabel 18 Pesan yang Dilayani raspi_aes.py Bagian 3	106
Tabel 19 Pesan yang Dilayani raspi_aes.py Bagian 4	115
Tabel 20 Data JSON untuk Sign In	133
Tabel 21 Data JSON Forget Password.....	138
Tabel 22 Data JSON Sign Up	145
Tabel 23 Data JSON untuk Loading Data dari Raspberry Pi.....	178
Tabel 24 Data JSON untuk <i>Delete</i> dan <i>Edit Room</i>	187
Tabel 25 Data JSON untuk <i>Delete</i> dan <i>Edit Scenario</i>	196
Tabel 26 Data JSON Pengaktifan Skenario	197
Tabel 27 Data JSON Perintah Pengendalian <i>Devices</i>	248
Tabel 28 Data JSON <i>Delete Devices</i>	248
Tabel 29 Data JSON <i>Add</i> atau <i>Edit Room</i>	269
Tabel 30 Data JSON untuk <i>Add</i> dan <i>Edit Scenario</i>	289
Tabel 31 Data JSON Permintaan Data Penggunaan <i>Device</i>	317

Tabel 32 Tabel JSON House Member	328
Tabel 33 Tabel JSON Add Member	335
Tabel 34 Tabel JSON pada get invited	347
Tabel 35 JSON untuk <i>edit profile</i>	373
Tabel 36 Tabel JSON pada saat request data rumah.....	393
Tabel 37 Tabel JSON pada saat edit informasi rumah.....	393
Tabel 38 Tabel JSON Privilege Setting	398

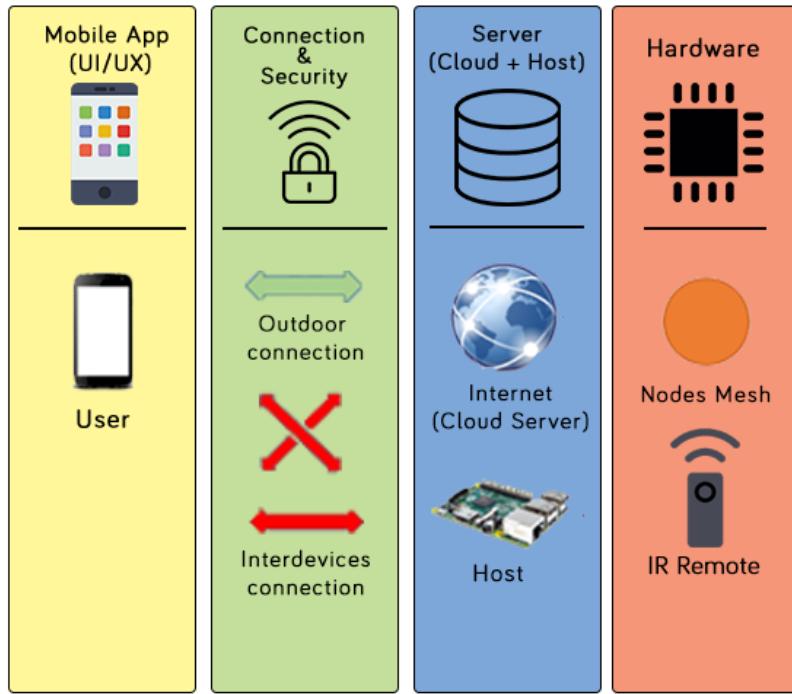
2. Ringkasan Penjelasan Sistem Secara Keseluruhan

2.1 Pendahuluan

Seperti yang telah dijelaskan pada dokumen terdahulu, proyek yang dikerjakan ini merupakan kelanjutan dari proyek yang telah dikerjakan oleh tim dari Pusat Mikro Elektronika (PME). Proyek yang telah dikerjakan oleh tim tersebut dibatasi pada perancangan sistem di bagian *indoor*. Beberapa hal yang telah dilakukan dalam proyek sebelumnya adalah sebagai berikut.

- Perancangan berbagai jenis *nodes* yang telah diintegrasikan dengan *end device*, yaitu *switch, curtain, door, thermometer, fan, RGB Lamp*.
- Perancangan skema komunikasi antar *nodes* tersebut menggunakan protokol Zigbee di bawah koordinasi *host* berupa Raspberry Pi 3.
- Pada proyek tersebut, user bisa mengakses keseluruhan sistem *indoor* ini melalui sebuah aplikasi android, namun masih menggunakan jaringan Bluetooth, belum menggunakan jaringan internet. Perintah-perintah seperti menyalakan lampu, mematikan kipas, membuka pintu, dan lain-lain sudah dapat dijalankan dengan baik.

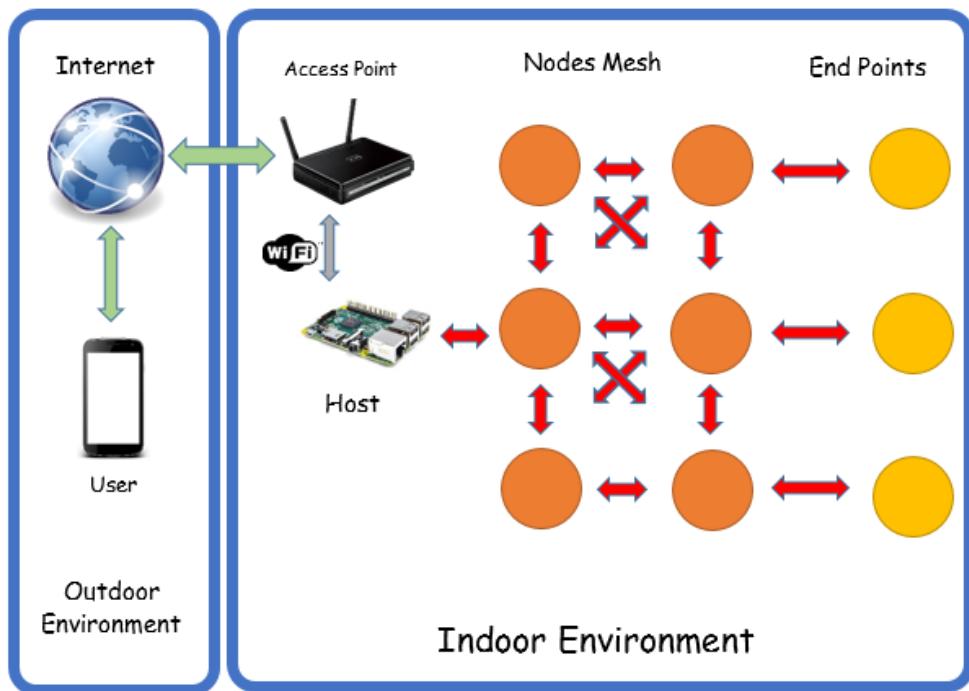
Proyek tugas akhir ini diharapkan dapat melanjutkan proyek yang telah dikerjakan di atas. Adapun hal yang menjadi ruang lingkup penggerjaan dalam dokumen ini dibagi menjadi 4 bagian implementasi yaitu diilustrasikan dan dijelaskan di bawah:



Gambar 1 Ruang lingkup implementasi

- Perancangan *cloud server* yang digunakan sebagai tempat penyimpanan data keseluruhan sistem. Selain itu, server juga berfungsi sebagai pelayan *client-client* yang ada serta mengatur lalu lintas data yang dikirimkan antar perangkat. (*Server*)
- Perancangan aplikasi android sebagai garda terdepan yang langsung berhubungan dengan user. Aplikasi ini diharapkan bersifat *user friendly* dan pengguna bisa melakukan akses terhadap fitur-fitur dengan mudah tanpa mengalami kesulitan atau kebingungan. (*Mobile App UI/UX*)
- Penambahan kecerdasan ke dalam sistem. Kecerdasan yang diharapkan dapat ditambahkan adalah adanya *time-based scenario*, *autolock* rumah berdasarkan lokasi (GPS), dan *scenario* berdasarkan gerakan menggunakan accelerometer. (*Hardware*)
- Penambahan sistem keamanan pada jaringan. Dengan demikian, semua data yang dialirkan (baik pada bagian *indoor* maupun *outdoor* harus dalam keadaan terenkripsi). Selain itu, diharapkan juga ada skema autentikasi setiap kali ada *user baru* atau *node baru* bergabung dengan sistem. (*Network & Security*).

- Penambahan *hardware* berupa *Infrared Remote*. Hal ini dimaksudkan agar dapat mengendalikan *device* yang membutuhkan sinyal *infrared* misalnya *Air Conditioner* dan TV.



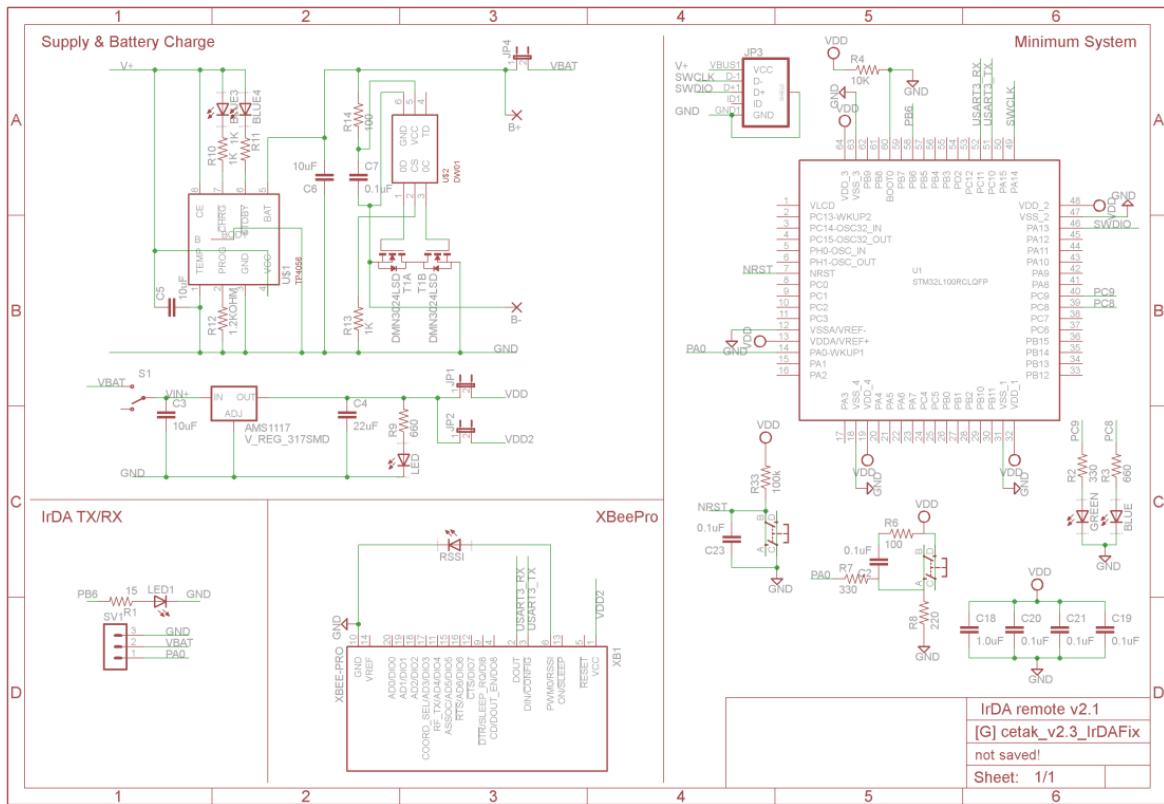
Gambar 2 Diagram Blok Sistem Keseluruhan

3. Implementasi Remote Control

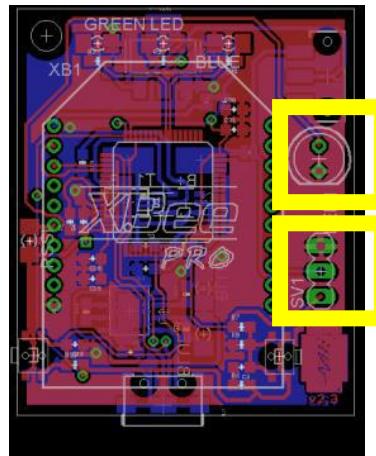
3.1 Implementasi Hardware Remote Control

Implementasi *remote* ini dilakukan dengan memodifikasi desain PCB pemantauan suhu dan kelembapan dengan menghilangkan sensor DHT11. Karena tujuan dari perancangan perangka ini adalah menggantikan fungsi remote konvensional yang umumnya berbasis IrDA, maka selanjutnya dilakukan penambahan *driver transmitter* maupun *receiver* IrDA. Selain *transmitter* dan *receiver*, *remote* akan dipasang sebuah button yang akan dipakai untuk mengubah status *remote* dari mode *send* menjadi mode *receive*. Pada implementasi *remote*, hanya IR *transmitter* yang dipasang, sedangkan *receiver* belum dikarenakan

perlunya riset lebih lanjut mengenai integrasi bahasa pemrograman python oleh Raspberry Pi dengan C oleh STM32.



Gambar 3 Skematik remote control IrDA-based



IrDA Tx

(IR LED)

IrDA Rx

(TSOP 1838)

Gambar 4 PCB diagram remote control IrDA-based



Gambar 5 Hasil *remote control* IrDA0based

3.2 Implementasi Program STM32 untuk Remote Control

Langkah awal untuk dapat mengimplementasikan *remote* adalah dengan mengatur pewaktuan STM32L100 untuk bekerja dengan *internal clock* 16MHz. Hal ini dilakukan dengan mengatur sistem *clock* untuk menghasilkan keluaran 16MHz pada file spreadsheet yang dapat diunduh pada website STM32. Source code system_stm32l1xx.c yang dibuat secara otomatis dari file spreadsheet kemudian di salin pada file system_stm32l1xx.c dalam folder proyek. Langkah kemudian adalah merancang sistem pewaktuan yang dapat menghasilkan pulsa PWM dengan frekuensi 38 KHz sebagai frekuensi *carrier* untuk dapat meniru data IR yang berupa pulsa HIGH dan LOW secara bergantian. Pada implementasi ini harus dirancang PWM yang dapat membangkitkan lebar pulsa setidaknya mendekati periode 38 kHz yaitu dengan periode 26,315 us (diganti pada `TIM_TimeBaseInitStruct.TIM_Period`). Sehingga dari *internal clock* 16MHz akan dicacah menjadi 1MHz sebagai referensi pewaktuan yang dibangun. Sehingga sistem pewaktuan (TIM4) dapat memiliki orde pewaktuan per 1 us.

```

void PWM_Configuration()
{
    GPIO_InitTypeDef GPIO_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseInitStruct;
    TIM_OCInitTypeDef TIM_OCInitStruct;

    // Enable HSI Clock
    RCC_HSICmd(ENABLE);

    // Wait till HSI is ready
    while (RCC_GetFlagStatus(RCC_FLAG_HSIRDY) == RESET) {}
    RCC_SYSCLKConfig(RCC_SYSCLKSource_HSI);

    // SysTick 1 msec interrupts (1000 = 1ms, 100 = 10 ms ...)
    if(SysTick_Config(SystemCoreClock / 1000))
    {
        // capture error
        while(1);
    }

    // Enable clock for GPIOB
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOB,ENABLE);
    // Enable clock for TIM4
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4,ENABLE);

    GPIO_StructInit(&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;//PB6=output of PWM
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
    GPIO_Init(GPIOB,&GPIO_InitStructure);
    GPIO_PinAFConfig(GPIOB,GPIO_PinSource6,GPIO_AF_TIM4);

    TIM_TimeBaseStructInit(&TIM_TimeBaseInitStruct);
    TIM_TimeBaseInitStruct.TIM_ClockDivision = TIM_CKD_DIV4;
    TIM_TimeBaseInitStruct.TIM_Period = 26 - 1; //bangkitin di
frekuensi berapa, 26 artinya 1/38kHz
    TIM_TimeBaseInitStruct.TIM_Prescaler = 16 - 1;
    TIM_TimeBaseInit(TIM4,&TIM_TimeBaseInitStruct);

    TIM_OCStructInit(&TIM_OCInitStruct);
    TIM_OCInitStruct.TIM_OutputState = TIM_OutputState_Enable;
    TIM_OCInitStruct.TIM_OCMode = TIM_OCMode_PWM1;
    TIM_OCInitStruct.TIM_Pulse = 0;

    TIM_OC1Init(TIM4,&TIM_OCInitStruct);
    TIM_Cmd(TIM4,ENABLE);
}

```

Tahap berikutnya adalah merancang perintah kontrol untuk mengirimkan data melalui IrDA. Data hasil rekaman IR akan ditumpangkan pada frekuensi *carrier* 38KHz dan akan dikirim ke perangkat yang memiliki IR *receiver*. Sehingga dibuatlah fungsi perintah pengiriman tersebut seperti di bawah.

```

int ac_log[]={3535, 3386, 891, 2516, 890, 780, 887, 781, 888, 781, 887,
2516, 891, 2515, 895, 2512, 891, 2515, 892, 2515, 892
, 779, 888, 780, 888, 780, 888, 2516, 891, 2516, 890,
2516, 890, 781, 887, 2516, 891, 2514, 891, 781, 887, 779, 889
, 779, 888, 780, 888, 780, 888, 779, 888, 2516, 890, 2516, 890,
888, 780, 889, 779, 888, 779, 889, 780, 3496, 3385, 892
, 2515, 893, 779, 887, 780, 888, 779, 888, 2515, 892, 2513, 893,
2514, 892, 2515, 891, 2516, 891, 781, 887, 781, 887, 781, 887
, 2517, 890, 2518, 890, 2517, 890, 2517, 889, 780, 888, 2516, 890,
2516, 891, 781, 887, 780, 888, 780, 888, 780, 887, 780, 887
, 781, 886, 2516, 890, 2518, 888, 781, 886, 781, 887, 781, 886, 782,
886, 782, 3496, 3387, 891, 3387, 3536, 782, 890, 782, 886
, 2516, 886, 782, 890, 2517, 886, 2517, 889, 2517, 889, 2519, 888,
782, 889, 782, 886, 2519, 886, 782, 888, 2518, 886, 2518, 889
, 2518, 889, 2519, 889, 782, 890, 2519, 886, 2517, 888, 782, 890,
2518, 887, 2518, 888, 782, 889, 781, 886, 782, 887, 2517, 886
, 2518, 890, 782, 889, 2518, 886, 2517, 889, 781, 890, 782, 886,
3386, 3495, 781, 889, 782, 887, 2518, 886, 781, 890, 2518, 886
, 2518, 889, 2519, 889, 2518, 887, 781, 889, 782, 886, 2518, 886,
782, 888, 2518, 885, 2517, 889, 2516, 889, 2517, 891, 782, 890
, 2519, 885, 2520, 889, 782, 888, 2517, 886, 2516, 889, 782, 890,
782, 885, 782, 886, 2518, 886, 2517, 889, 782, 890, 2518, 886
, 2518, 889, 782, 890, 782, 886, 3388, 3495, 3388};

void pwr_ac()
{
    int it, itung=0;
    for(it=0;it<= ((sizeof(ac_log)/sizeof(int))/2 + 1);it++)
    {
        TIM4->CCR1=8; //8 untuk HIGH
        Delay_us(ac_log[itung]);
        itung++;
        TIM4->CCR1=0; //0 untuk LOW
        Delay_us(ac_log[itung]);
        itung++;
    }
}

```

Paket data yang dikirimkan (ac_log[]) didapat dari pembacaan IrDA *receiver* dengan menggunakan modifikasi *library* NEC_decode_m.c.

Untuk dapat menjalankan fungsi pwr_ac(), dilakukan pengecekan protocol terlebih dahulu agar tidak dapat diakses oleh orang lain. Protokol yang dibuat berupa HOMEIDDEVICEADDRESS yang akan diterima oleh STM32 tiap byte. Pada contoh ini, HOMEID yang diimplementasi adalah 123456 dan DEVICEADDRESS nya adalah 40A88C1D. Setelah seluruh byte yang diterima host sama dengan protocol tersebut, maka variable *flag* diset menjadi 1 yang berarti sinyal IR siap dikirim oleh program utama. Implementasi pembacaan protocol ini dilakukan dengan menggunakan interrupt handler yang ditunjukkan pada source code di bawah ini.

```

void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) // read interrupt
    {
        USART3->SR &= ~USART_FLAG_RXNE; // clear interrupt
    }
    DataUsart3 = USART3->DR;

    switch(StateReceive)
    {
        /* case 1 - 5 check header HOMEID*/
        case 0 :
            if(DataUsart3=='1')StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2')StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3')StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
            if(DataUsart3=='4')StateReceive=4;
            else StateReceive=0;
            break;

        case 4 :
            if(DataUsart3=='5')StateReceive=5;
            else StateReceive=0;
            break;

        case 5 :
            if(DataUsart3=='6')StateReceive=6;
            else StateReceive=0;
            break;

        /* check Device Address*/
        //IR Address 40A88C1D
        case 6 :
            if(DataUsart3=='4')StateReceive=7;
            else StateReceive=0;
            break;
    }
}

```

```

        case 7 :
            if(DataUsart3=='0')StateReceive=8;
            else StateReceive=0;
            break;
        case 8 :
            if(DataUsart3=='A')StateReceive=9;
            else StateReceive=0;
            break;
        case 9 :
            if(DataUsart3=='8')StateReceive=10;
            else StateReceive=0;
            break;
        case 10 :
            if(DataUsart3=='8')StateReceive=11;
            else StateReceive=0;
            break;
        case 11:
            if(DataUsart3=='C')StateReceive=12;
            else StateReceive=0;
            break;
        case 12 :
            if(DataUsart3=='1')StateReceive=13;
            else StateReceive=0;
            break;
        case 13 :
            if(DataUsart3=='D')
            {
                StateReceive=0;
                flag = 1;
            }
            break;
    }
}

```

Pada program utama, source code yang dibuat ditunjukkan pada gambar di bawah. Pada program utama, langkah awal ialah dengan memanggil beberapa fungsi pengaturan STM32, yakni fungsi `USART3_Configure()`, `PWM_Configuration()`, `Counter_Config()`, `Exti_Config()`. Fungsi `USART3_Configure()` berfungsi untuk mengatur keluaran pin RX/TX pada USART3. Fungsi `PWM_Configuration()` digunakan untuk mengatur agar keluaran sinyal berupa PWM 38 kHz. Fungsi `Counter_Config()` dipanggil untuk mengatur timer yang digunakan. Fungsi `Exti_Config()` digunakan untuk mengaktifkan sistem interrupt agar dapat masuk pada fungsi `USART3_IRQHandler()`. Setelah itu, dilakukan perulangan mengecek nilai flag (1 atau 0). Bila flag = 0, maka pengecekan akan berlangsung terus menerus dan sinyal IR tidak akan dikirim. Apabila flag = 1, maka remote akan mengirim sinyal IR, lalu mengatur ulang nilai flag menjadi 0. Pada implementasi remote ini, data yang dikirim masih

terbatas untuk satu buah produk saja, dan sinyal yang dikirimkan juga masih terbatas untuk menghidupkan/mematikan AC atau TV. Setelah source code selesai dibuat, dilakukan kompilasi terhadap proyek lalu file *.hex diupload ke modul remote menggunakan software uVision atau TrueStudio + ST-Link Utility, menggunakan serial downloader ST-Link v2.

```
int main(void)
{
    /* TODO - Add your application code here */
    USART3_Configure();
    PWM_Configuration();
    Counter_Config();
    Exti_Config();

    /* Infinite loop */
    while (1)
    {
        if (flag == 1)
        {
            pwr_ac();
            flag =0;
        }
    }
}
```

4. Implementasi Protokol Komunikasi Subsistem Indoor

Untuk menambah keamanan *device*, protocol tiap device dirancang sedemikian rupa menjadi HomeID + Alamat Device + perintah. Hal ini dilakukan dengan mengubah fungsi USART3_IRQHandler dan merancang cara kerja pengecekan protocol.

4.1 Implementasi Protokol pada Device Sensor Suhu dan Kelembapan

Pada implementasi ini, protokol yang digunakan adalah 12345640A88C29H untuk kelembapan dan 12345640A88C29S untuk suhu. Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk memberikan data hasil suhu atau kelembapan.

```

void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) {                                // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE;                         // clear interrupt
    }
    DataUsart3 = USART3->DR;
    //GPIO_ToggleBits(GPIOC, GPIO_Pin_9); // green led

    /* case 1 - 5 check header*/
    switch (StateReceive)
    {
        case 0 :
            if(DataUsart3=='1') StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2') StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3') StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
            if(DataUsart3=='4') StateReceive=4;
            else StateReceive=0;
            break;

        case 4 :
            if(DataUsart3=='5') StateReceive=5;
            else StateReceive=0;
            break;

        case 5 :
            if(DataUsart3=='6') StateReceive=6;
            else StateReceive=0;
            break;

        /* check Device Address*/
        //Sensor Address 40A88C29
        case 6 :
            if(DataUsart3=='4') StateReceive=7;
            else StateReceive=0;
            break;

        case 7 :
            if(DataUsart3=='0') StateReceive=8;
            else StateReceive=0;
    }
}

```

```

        break;
    case 8 :
        if(DataUsart3=='A') StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8') StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8') StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='C') StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='2') StateReceive=13;
        else StateReceive=0;
        break;
    case 13 :
        if(DataUsart3=='9') StateReceive=14;
        else StateReceive=0;
        break;
    case 14 :
        if (DataUsart3=='H')
        {
            GPIO_ToggleBits(GPIOC, GPIO_Pin_9);
            flagDHT=1;
        }
        else if (DataUsart3=='S')
        {
            flagDHT=2;
        }
        GPIO_ToggleBits(GPIOC, GPIO_Pin_8); //blue
        StateReceive = 0;
        break;
    }
}

```

```

int main(void)
{
    /* Add your application code here
     */
    // GPIO Init
    GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
    RCC_AHBPeriphClockCmd( RCC_AHBPeriph_GPIOC, ENABLE); // Enable
    GPIOC clock
    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7 | GPIO_Pin_8 |
    GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    // Delay Init
    SystemCoreClockSetHSI();
    SystemCoreClockUpdate();
    SysTick_Config(SystemCoreClock / 1000); // SysTick 1 msec
    interrupts

    // USART Init
    USART2_Configure();

/* Infinite loop */

    /* IO DHT11 */
    // Output for DHT11 Trigger
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_OD;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
    GPIO_HIGH(GPIOB,GPIO_Pin_1);

    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_OD;
    GPIO_Init(GPIOC, &GPIO_InitStructure);
    GPIO_HIGH(GPIOC,GPIO_Pin_2);
    // Input Port
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    while (1)
    {
        if (flagDHT==1)
        {
            DHT11_RawRead(buf_dht);
            res_dht = DHT11_Humidity(buf_dht);
            sprintf(strDisp, "H : %02d \r", buf_dht[0]);
            kirim2_string(strDisp);
            flagDHT=0;
        }
    }
}

```

```

        }
        Delay(500);
    else if (flagDHT==2)
    {
        DHT11_RawRead(buf_dht);
        res_dht = DHT11_Temperature(buf_dht);
        sprintf(strDisp, "S : %02d \r", buf_dht[2]);
        kirim2_string(strDisp);
        flagDHT=0;
    }
    Delay(500);

}

```

4.2 Implementasi Protokol pada lampu RGB

Pada implementasi ini, protokol yang digunakan adalah 12345640A88BB2 + nilai R (0-255) + nilai G (0-255) + nilai B(0 -255). Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk memberikan warna lampu sesuai masukkan nilai R,G dan B.

```

void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) {                                // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE;                         // clear interrupt
    }

    DataUsart3 = USART3->DR;

    switch(StateReceive) //StateReceive set to 0
initialization
    {
        /* case 1 - 5 check header*/
        case 0 :
            if(DataUsart3=='1') StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2') StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3') StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
    }
}

```

```

        if(DataUsart3=='4') StateReceive=4;
        else StateReceive=0;
        break;
    case 4 :
        if(DataUsart3=='5') StateReceive=5;
        else StateReceive=0;
        break;

    case 5 :
        if(DataUsart3=='6') StateReceive=6;
        else StateReceive=0;
        break;

        /* check Device Address*/
        //RGB Address 40A88BB2
    case 6 :
        if(DataUsart3=='4') StateReceive=7;
        else StateReceive=0;
        break;

    case 7 :
        if(DataUsart3=='0') StateReceive=8;
        else StateReceive=0;
        break;
    break;
    case 8 :
        if(DataUsart3=='A') StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8') StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8') StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='B') StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='B') StateReceive=13;
        else StateReceive=0;
        break;
    case 13 :
        if(DataUsart3=='2') StateReceive=14;
        else StateReceive=0;
        break;
        // Command R,G,B value
    case 14 :
        R = DataUsart3;
        StateReceive=15;
        break;
    break;

    case 15 :
        G = DataUsart3;

```

```

        StateReceive=16;
        break;
    case 16 :
        B=DataUsart3;
        StateReceive=0;
        GPIO_ToggleBits(GPIOC, GPIO_Pin_8); //blue
        break;
    }
}

```

```

int main(void)
{
    /* Add your application code here*/
    // GPIO Init
    GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // Enable
    GPIOC clock
    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 |
    GPIO_Pin_4 | GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    // Delay Init
    SystemCoreClockSetHSI();
    SystemCoreClockUpdate();
    SysTick_Config(SystemCoreClock / 1000); // SysTick 1 msec
    interrupts

    // USART Init
    USART3_Configure();
    PWM_Configuration();
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
    GPIO_Init(GPIOB, &GPIO_InitStructure);

    while (1)
    {

        R_fix= (R*1000)/255;
        G_fix= (G*1000)/255;
        B_fix= (B*1000)/255;

        TIM3->CCR1= R_fix;
        TIM3->CCR3= G_fix;
        TIM3->CCR2= B_fix;
    }
}

```

4.3 Implementasi Protokol pada Tirai

Pada implementasi ini, protokol yang digunakan adalah 12345640A88C2C + perintah korden (0x00 atau 0x01). Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk memberikan pergerakan korden sesuai perintah.

```
void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) {                                // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE;                         // clear interrupt
    }

    DataUsart3 = USART3->DR;
    GPIO_ToggleBits(GPIOC, GPIO_Pin_8); //blue
    switch(StateReceive)
    {
        /* case 1 - 5 check header*/
        case 0 :
            if(DataUsart3=='1')StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2')StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3')StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
            if(DataUsart3=='4')StateReceive=4;
            else StateReceive=0;
            break;

        case 4 :
            if(DataUsart3=='5')StateReceive=5;
            else StateReceive=0;
            break;

        case 5 :
            if(DataUsart3=='6')StateReceive=6;
            else StateReceive=0;
            break;

        /* check Device Address*/
        //Korden Address 40A88BC1
        case 6 :
            if(DataUsart3=='4')StateReceive=7;
    }
}
```

```

        else StateReceive=0;
        break;

    case 7 :
        if(DataUsart3=='0')StateReceive=8;
        else StateReceive=0;
        break;
    case 8 :
        if(DataUsart3=='A')StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8')StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8')StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='C')StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='2')StateReceive=13;
        else StateReceive=0;
        break;
    case 13 :
        if(DataUsart3=='C')StateReceive=14;
        else StateReceive=0;
        break;
    // Command korden value
    case 14 :
        if(DataUsart3==0x00) flag=0;
        else if(DataUsart3==0x01) flag=1;
        else flag=3;
        StateReceive=0;
        break;
    }
}

```

```

int main(void)
{
    /* Add your application code here
     */
    // GPIO Init
    GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // Enable
    GPIOC clock

    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_7 |
    GPIO_Pin_8 | GPIO_Pin_9 | GPIO_Pin_4 | GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;

```

```

GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOC, &GPIO_InitStructure);

RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOB, ENABLE); // Enable GPIOC clock
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_12 | GPIO_Pin_14;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOB, &GPIO_InitStructure);
// Delay Init
SystemCoreClockSetHSI();
SystemCoreClockUpdate();
SysTick_Config(SystemCoreClock / 1000); // SysTick 1 msec
interrupts

// USART Init
USART3_Configure();

//ADC_Configuration();
Counter_Config();
Exti_Config();
/* Infinite loop */

// Input Port
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);

while (1)
{
    if((flag==0)&&(laststatus==1)){
        for(loop=0;loop<580;loop++){
            GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
            GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
            Delay(1);
            GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
            GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
            Delay(1);
            GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
            GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
            Delay(1);
            GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
            GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_SET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
            Delay(1);
            GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
            GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_SET);
            GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
        }
    }
}

```

```

        GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
        Delay(1);
        GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_SET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
        Delay(1);
        GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
        Delay(1);
        GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
        Delay(1);
        laststatus=0;
    }
    flag=3;
}
if((flag==1)&&(laststatus==0)){
for(loop=0;loop<580;loop++){
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_RESET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_SET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
    Delay(1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_RESET);
    GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
    GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
    Delay(1);
}
}

```

```

        GPIO_WriteBit(GPIOC, GPIO_Pin_6, Bit_SET);
        GPIO_WriteBit(GPIOC, GPIO_Pin_7, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_14, Bit_RESET);
        GPIO_WriteBit(GPIOB, GPIO_Pin_12, Bit_SET);
        Delay(1);
        laststatus=1;
    }
    flag=3;
}
}
}

```

4.4 Implementasi Protokol pada Pintu

Pada implementasi ini, protokol yang digunakan adalah 12345640A88C2C + perintah pintu (0x00 untuk mengunci atau 0x01 untuk membuka). Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk memberikan pergerakkan pintu sesuai perintah.

```

void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) {                                // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE;                         // clear interrupt
    }

    DataUsart3 = USART3->DR;
    switch(StateReceive)
    {
        /* case 1 - 5 check header*/
        case 0 :
            if(DataUsart3=='1') StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2') StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3') StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
            if(DataUsart3=='4') StateReceive=4;
            else StateReceive=0;
            break;

        case 4 :
            if(DataUsart3=='5') StateReceive=5;
            break;
    }
}

```

```

        else StateReceive=0;
        break;

    case 5 :
        if(DataUsart3=='6')StateReceive=6;
        else StateReceive=0;
        break;
    /* check Device Address*/
    //Door Address 40A88C2C
    case 6 :
        if(DataUsart3=='4')StateReceive=7;
        else StateReceive=0;
        break;

    case 7 :
        if(DataUsart3=='0')StateReceive=8;
        else StateReceive=0;
        break;
    break;
    case 8 :
        if(DataUsart3=='A')StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8')StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8')StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='C')StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='2')StateReceive=13;
        else StateReceive=0;
        break;
    case 13 :
        if(DataUsart3=='8')StateReceive=14;
        else StateReceive=0;
        break;
    // Command Door value
    case 14 :
        rawlock=DataUsart3;
        StateReceive=0;
        GPIO_ToggleBits(GPIOC, GPIO_Pin_8);
        break;
    }
}

```

```

int main(void)
{
    */
    // GPIO Init

```

```

GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // Enable
GPIOC clock
// GPIO Configuration
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 |
GPIO_Pin_4 | GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOC, &GPIO_InitStructure);

RCC_AHBPeriphClockCmd( RCC_AHBPeriph_GPIOB, ENABLE); // Enable
GPIOB clock
// GPIO Configuration
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
GPIO_Init(GPIOB, &GPIO_InitStructure);
// Delay Init
SystemCoreClockSetHSI();
SystemCoreClockUpdate();
SysTick_Config(SystemCoreClock / 1000); // SysTick 1 msec
interrupts

// USART Init
USART3_Configure();
Counter_Config();
Exti_Config();

// Input Port
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);

while (1)
{
    if(rawlock==0x00)
    {
        GPIO_LOW(GPIOB,GPIO_Pin_0);
        GPIO_HIGH(GPIOC,GPIO_Pin_8); //blue
        GPIO_LOW(GPIOC,GPIO_Pin_9); //green
    }
    else if(rawlock==0x01)
    {
        GPIO_HIGH(GPIOB,GPIO_Pin_0);
        GPIO_LOW(GPIOC,GPIO_Pin_8); //blue
        GPIO_HIGH(GPIOC,GPIO_Pin_9); //green
    }
}
}

```

4.5 Implementasi Protokol pada Kipas

Pada implementasi ini, protokol yang digunakan adalah 12345640A88C39 + perintah kecepatan kipas (0-255). Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk memberikan pergerakan kipas sesuai perintah.

```
void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;

    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) {                                // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE;                         // clear interrupt
    }

    DataUsart3 = USART3->DR;
    GPIO_ToggleBits(GPIOC, GPIO_Pin_8); //blue

    switch(StateReceive)
    {
        /* case 1 - 5 check header*/
        case 0 :
            if(DataUsart3=='1')StateReceive=1;
            else StateReceive=0;
            break;

        case 1 :
            if(DataUsart3=='2')StateReceive=2;
            else StateReceive=0;
            break;

        case 2 :
            if(DataUsart3=='3')StateReceive=3;
            else StateReceive=0;
            break;

        case 3 :
            if(DataUsart3=='4')StateReceive=4;
            else StateReceive=0;
            break;

        case 4 :
            if(DataUsart3=='5')StateReceive=5;
            else StateReceive=0;
            break;

        case 5 :
            if(DataUsart3=='6')StateReceive=6;
            else StateReceive=0;
            break;

        /* check Device Address*/
        //Door Address 40A88C39
        case 6 :
```

```

        if(DataUsart3=='4') StateReceive=7;
        else StateReceive=0;
        break;

    case 7 :
        if(DataUsart3=='0') StateReceive=8;
        else StateReceive=0;
        break;
    case 8 :
        if(DataUsart3=='A') StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8') StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8') StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='C') StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='3') StateReceive=13;
        else StateReceive=0;
        break;
    case 13 :
        if(DataUsart3=='9') StateReceive=14;
        else StateReceive=0;
        break;
    // Command Fan value
    case 14 :
        rawkec=DataUsart3;
        StateReceive=0;
        GPIO_ToggleBits(GPIOC, GPIO_Pin_8);
        break;
    }
}

```

```

int main(void)
{
    /* Add your application code here
     */
    // GPIO Init
    GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // Enable
    GPIOC clock
    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 |
    GPIO_Pin_4 | GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;

```

```

GPIO_Init(GPIOC, &GPIO_InitStructure);
// Delay Init
SystemCoreClockSetHSI();
SystemCoreClockUpdate();
SysTick_Config(SystemCoreClock / 1000); // SysTick 1 msec
interrupts
// USART Init
USART3_Configure();
PWM_Configuration();

Counter_Config();
Exti_Config();
/* Infinite loop */
// Input Port
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
GPIO_Init(GPIOB, &GPIO_InitStructure);

while (1)
{
    if(rawkec==0) TIM3->CCR3=0;
    else
    {
        kec=(rawkec/255)*500+500;
        TIM3->CCR3=kec;
    }
}
}

```

4.6 Implementasi Protokol pada Saklar

Pada implementasi ini, protokol yang digunakan adalah 12345640A88C4C + perintah saklar (0x00 untuk *off* atau 0x01 untuk *on*). Setelah mengubah protokol pada fungsi USART3_IRQHandler, dilakukan pemrosesan terhadap masukkan pada program utama di dalam while(1) untuk menyalakan/mematikan saklar sesuai perintah.

```

void USART3_IRQHandler(void)
{
    volatile unsigned int IIR;
    IIR = USART3->SR;
    if (IIR & USART_FLAG_RXNE) { // read interrupt
        USART3->SR &= ~USART_FLAG_RXNE; // clear interrupt
    }

    DataUsart3 = USART3->DR;
    switch(StateReceive)
    {
        /* case 1 - 5 check header*/
        case 0 :
            if(DataUsart3=='1') StateReceive=1;
            else StateReceive=0;
    }
}

```

```

        break;

    case 1 :
        if(DataUsart3=='2') StateReceive=2;
        else StateReceive=0;
        break;

    case 2 :
        if(DataUsart3=='3') StateReceive=3;
        else StateReceive=0;
        break;

    case 3 :
        if(DataUsart3=='4') StateReceive=4;
        else StateReceive=0;
        break;

    case 4 :
        if(DataUsart3=='5') StateReceive=5;
        else StateReceive=0;
        break;

    case 5 :
        if(DataUsart3=='6') StateReceive=6;
        else StateReceive=0;
        break;

    /* check Device Address*/
    //Sekering Address 40A88C4C
    case 6 :
        if(DataUsart3=='4') StateReceive=7;
        else StateReceive=0;
        break;

    case 7 :
        if(DataUsart3=='0') StateReceive=8;
        else StateReceive=0;
        break;
    case 8 :
        if(DataUsart3=='A') StateReceive=9;
        else StateReceive=0;
        break;
    case 9 :
        if(DataUsart3=='8') StateReceive=10;
        else StateReceive=0;
        break;
    case 10 :
        if(DataUsart3=='8') StateReceive=11;
        else StateReceive=0;
        break;
    case 11:
        if(DataUsart3=='C') StateReceive=12;
        else StateReceive=0;
        break;
    case 12 :
        if(DataUsart3=='4') StateReceive=13;
        else StateReceive=0;
        break;

```

```

        case 13 :
            if(DataUsart3=='C') StateReceive=14;
            else StateReceive=0;
            break;
        // Command Fan value
        case 14 :
            rawswitch=DataUsart3;
            StateReceive=0;
            //sprintf(buf, "%d ", rawlock);
            //kirim3_string(buf);
            GPIO_ToggleBits(GPIOC, GPIO_Pin_8); //blue
            break;
    }
}

```

```

int main(void)
{
    /* Add your application code here
     */
    // GPIO Init
    GPIO_InitTypeDef GPIO_InitStructure; // Initialization Structure
    RCC_AHBPeriphClockCmd(RCC_AHBPeriph_GPIOC, ENABLE); // Enable
    GPIOC clock
    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_9 |
    GPIO_Pin_4 | GPIO_Pin_3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOC, &GPIO_InitStructure);

    RCC_AHBPeriphClockCmd( RCC_AHBPeriph_GPIOB, ENABLE); // Enable
    GPIOB clock
    // GPIO Configuration
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
    GPIO_Init(GPIOB, &GPIO_InitStructure);
    //GPIO_HIGH(GPIOB,GPIO_Pin_0);

    // Delay Init
    SystemCoreClockSetHSI();
    SystemCoreClockUpdate();
    SysTick_Config(SystemCoreClock / 1000);           // SysTick 1 msec
    interrupts

    // USART Init
    USART3_Configure();
    Counter_Config();
    Exti_Config();
/* Infinite loop */
    // Input Port
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_40MHz;
}

```

```

GPIO_Init(GPIOB, &GPIO_InitStructure);

while (1)
{
    if(rawswitch==0x00) GPIO_LOW(GPIOC,GPIO_Pin_4);
    else if(rawswitch==0x01) GPIO_HIGH(GPIOC,GPIO_Pin_4);
}

```

5. Implementasi Server dan Raspberry Pi Host

5.1 Implementasi Basis Data

Pada dokumen B300, telah dijelaskan bahwa basis data (baik di server maupun di host) diimplementasikan dengan *environment* MySQL dengan pertimbangan yang telah dijelaskan pada dokumen tersebut.

5.1.1 Implementasi Basis Data pada Server

Seperti yang telah dirancang pada B300, basis data pada server berisi data-data seluruh pengguna sistem rumah cerdas. Data tersebut disimpan dalam suatu database bernama general_data. Database general_data diimplementasikan dengan menggunakan MySQL dengan memasukan kode berikut.

```
CREATE DATABASE general_data;
```

Database ini berisi dua buah tabel, yaitu:

- Tabel users: tabel yang berisi data-data profil pengguna
- Tabel aeskeylist: tabel yang berisi daftar kunci enkripsi AES untuk masing-masing pengguna

Kolom-kolom tabel users adalah sebagai berikut.

Tabel 1 Kolom-kolom Tabel users di database general_data

Nama Kolom	Tipe Data	Isi
email	VARCHAR(50)	Email pengguna
password	VARCHAR(20)	Password pengguna

homeid	VARCHAR(12)	ID rumah di mana pengguna didaftarkan
name	VARCHAR(20)	Nama pengguna
picture	MEDIUMTEXT	Gambar profil pengguna yang telah mengalami Base64 <i>encoding</i>
privilege	VARCHAR(11)	Jenis privilege pengguna (ADMIN atau Guest)

Tabel di atas diimplementasikan pada MySQL dengan memasukan perintah sebagai berikut.

```
CREATE TABLE users (email VARCHAR(50), password VARCHAR(20), homeid
VARCHAR(12), name VARCHAR(20), picture MEDIUMTEXT, privilege
VARCHAR(11));
```

Dengan demikian, tabel users terbentuk.

```
mysql> describe users;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| email | varchar(50) | NO   |   | NULL    |       |
| password | varchar(20) | YES  |   | NULL    |       |
| homeid | varchar(12) | YES  |   | NULL    |       |
| name  | varchar(20) | YES  |   | NULL    |       |
| picture | mediumtext | YES  |   | NULL    |       |
| privilege | varchar(11) | YES  |   | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 6 Implementasi Tabel Users

Kolom-kolom tabel aeskeylist adalah sebagai berikut.

Tabel 2 Kolom-kolom Tabel aeskeylist di database general_data

Nama Kolom	Tipe Data	Isi
phoneid	VARCHAR(16)	ID ponsel Android pengguna
aeskey	VARCHAR(16)	Kunci AES untuk ponsel yang bersangkutan

Tabel tersebut diimplementasikan dengan kode SQL sebagai berikut.

```
CREATE TABLE aeskeylist (phoneid VARCHAR(16), aeskey VARCHAR(16));
```

Dengan demikian, tabel aeskeylist terbentuk.

```

mysql> describe aeskeylist;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| phoneid | varchar(16) | YES |   | NULL |   |
| aeskey | varchar(16) | YES |   | NULL |   |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

Gambar 7 Impelementasi Tabel aeskeylist

5.1.2 Implementasi Basis Data pada Host

Host memiliki database yang menampung data-data yang berhubungan dengan perangkat-perangkat di dalam rumah yang bersangkutan tempat *host* diletakan. Untuk menyimpan semua data tersebut, terdapat sebuah basis data bernama homeXXXXXX, dengan XXXXXX adalah 6 digit ID rumah (home ID). Untuk membentuk database tersebut, diberikan perintah kode SQL sebagai berikut (misal home ID yang digunakan adalah “123456”).

```
CREATE DATABASE home123456;
```

Di dalam database ini, terdapat 6 tabel yang dijelaskan sebagai berikut.

- info: berisi informasi rumah
- devices: berisi data seluruh perangkat yang ada di rumah
- rooms: berisi data-data kategori ruangan di dalam rumah
- scenarios: berisi data-data skenario di dalam rumah
- dataMMYYYY: berisi durasi penggunaan setiap perangkat di dalam rumah (MMYYYY merupakan bulan dan tahun dari data tersebut, misalnya bulan April 2017 berarti data042017)
- users: berisi data-data pengguna yang terdaftar pada rumah tersebut
- Tabel-tabel dengan nama skenario-skenario yang terdaftar pada rumah: berisi informasi mengenai bagaimana status setiap perangkat ketika skenario ini dijalankan.

Tabel pertama adalah tabel info. Bentuk tabel info adalah sebagai berikut.

Tabel 3 Kolom-kolom Tabel info di database homeXXXXXX

Nama Kolom	Tipe Data	Isi
name	VARCHAR(20), PRIMARY KEY	Nama rumah
latitude	DOUBLE	Koordinat posisi latitude (lintang) rumah
longitude	DOUBLE	Koordinat posisi longitude (bujur) rumah

Tabel tersebut diimplementasikan dengan kode sebagai berikut.

```
CREATE TABLE info (name VARCHAR(20) PRIMARY KEY, latitude DOUBLE,  
longitude DOUBLE);
```

Dengan demikian, tabel akan terbentuk sebagai berikut.

```
mysql> describe info;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra   |
+-----+-----+-----+-----+-----+
| name  | varchar(20) | NO   | PRI  | NULL    |          |
| latitude | double | YES  |       | NULL    |          |
| longitude | double | YES  |       | NULL    |          |
| lockstatus | varchar(6) | YES  |       | NULL    |          |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Gambar 8 Implementasi Tabel info

Tabel berikutnya adalah tabel devices dengan bentuk sebagai berikut.

Tabel 4 Kolom-kolom Tabel devices di database homeXXXXXX

Nama Kolom	Tipe Data	Isi
name	VARCHAR(20)	Nama perangkat
address	VARCHAR(15), PRIMARY KEY	Alamat XBee perangkat
type	VARCHAR(11)	Tipe perangkat (sakelar, lampa, tirai, dsb.)
status	VARCHAR(8)	Status dari perangkat tersebut (ON, OFF, 50%, dsb.)

room	VARCHAR(20)	Ruangan di mana perangkat tersebut diletakan
privilege	VARCHAR(5)	Privilege device, apakah device bisa diakses oleh ADMIN saja atau oleh ADMIN dan Guest

Tabel di atas diimplementasikan dengan perintah kode SQL sebagai berikut.

```
CREATE TABLE devices (name VARCHAR(20), address VARCHAR(8) PRIMARY KEY,
type VARCHAR(11), status VARCHAR(8), room VARCHAR(20), privilege
VARCHAR(5));
```

Dengan demikian, tabel devices terbentuk.

```
mysql> describe devices;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | varchar(20) | YES |   | NULL    |       |
| address | varchar(8) | NO  | PRI | NULL    |       |
| type  | varchar(11) | YES |   | NULL    |       |
| status | varchar(8) | YES |   | NULL    |       |
| room   | varchar(30) | YES |   | NULL    |       |
| privilege | varchar(5) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Gambar 9 Implementasi Tabel devices

Tabel berikutnya adalah tabel rooms dengan bentuk sebagai berikut.

Tabel 5 Kolom-kolom Tabel rooms di database homeXXXXXX

Nama Kolom	Tipe Data	Isi
name	VARCHAR(20), PRIMARY KEY	Nama ruangan

Tabel di atas diimplementasikan dengan kode SQL sebagai berikut.

```
CREATE TABLE rooms (name VARCHAR(20));
```

Maka terbentuklah tabel rooms.

```
mysql> describe rooms;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | varchar(40) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Gambar 10 Implementasi Tabel rooms

Tabel berikutnya adalah tabel scenarios. Bentuk tabel scenarios adalah sebagai berikut.

Tabel 6 Kolom-kolom Tabel scenarios di database homeXXXXXX

Nama Kolom	Tipe Data	Isi
name	VARCHAR(20), PRIMARY KEY	Nama skenario
time	VARCHAR(4)	Waktu pengaktifan skenario bila pengguna ingin mengaktifkan skenario ini secara rutin setiap hari (bila skenario tidak dijalankan secara rutin, maka kolom ini diisi ‘none’)

Tabel tersebut diimplementasikan dengan kode SQL sebagai berikut.

```
CREATE TABLE rooms (name VARCHAR(20), time VARCHAR(4));
```

Dengan demikian, terbentuklah tabel scenarios.

```
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| name  | varchar(20) | YES |   | NULL    |       |
| time  | varchar(4)  | YES |   | NULL    |       |
| privilege | varchar(5) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Gambar 11 Implementasi Tabel scenarios

Tabel berikutnya adalah tabel dataMMYYYY, yaitu tabel yang menampung data penggunaan setiap perangkat dalam rumah. Bentuk tabel tersebut adalah sebagai berikut.

Tabel 7 Kolom-kolom Tabel dataMMYYYY di database homeXXXXXX

Nama Kolom	Tipe Data	Isi
address	VARCHAR(8), PRIMARY KEY	Alamat XBee perangkat

lasttime	VARCHAR(14)	Waktu terakhir (jam, menit, detik, hari, bulan, tahun) ketika perangkat terakhir dinyalakan atau diakifikan
d01 sampai d31 (d adalah date)	VARCHAR(7)	Durasi penggunaan device sampai dengan waktu ketika database diakses (durasi dalam satuan menit dan 1 digit di belakang koma) untuk tanggal tertentu

Tabel tersebut diimplementasikan dalam kode SQL sebagai berikut (misalnya bulan dan tahun yang digunakan adalah 042017).

```
CREATE TABLE data042017 (address VARCHAR(8), lasttime VARCHAR(14), d01
VARCHAR(7), d02 VARCHAR(7), d03 VARCHAR(7), d04 VARCHAR(7), d05
VARCHAR(7), d06 VARCHAR(7), d07 VARCHAR(7), d08 VARCHAR(7), d09
VARCHAR(7), d10 VARCHAR(7), d11 VARCHAR(7), d12 VARCHAR(7), d13
VARCHAR(7), d14 VARCHAR(7), d15 VARCHAR(7), d16 VARCHAR(7), d17
VARCHAR(7), d18 VARCHAR(7), d19 VARCHAR(7), d20 VARCHAR(7), d21
VARCHAR(7), d22 VARCHAR(7), d23 VARCHAR(7), d24 VARCHAR(7), d25
VARCHAR(7), d26 VARCHAR(7), d27 VARCHAR(7), d28 VARCHAR(7), d29
VARCHAR(7), d30 VARCHAR(7), d31 VARCHAR(7));
```

Dengan demikian, terbentuklah tabel data042017.

Field	Type	Null	Key	Default	Extra
address	varchar(8)	NO	PRI	NULL	
lasttime	varchar(14)	YES		NULL	
d01	varchar(7)	YES		NULL	
d02	varchar(7)	YES		NULL	
d03	varchar(7)	YES		NULL	
d04	varchar(7)	YES		NULL	
d05	varchar(7)	YES		NULL	
d06	varchar(7)	YES		NULL	
d07	varchar(7)	YES		NULL	
d08	varchar(7)	YES		NULL	
d09	varchar(7)	YES		NULL	
d10	varchar(7)	YES		NULL	
d11	varchar(7)	YES		NULL	
d12	varchar(7)	YES		NULL	
d13	varchar(7)	YES		NULL	
d14	varchar(7)	YES		NULL	
d15	varchar(7)	YES		NULL	
d16	varchar(7)	YES		NULL	
d17	varchar(7)	YES		NULL	
d18	varchar(7)	YES		NULL	
d19	varchar(7)	YES		NULL	
d20	varchar(7)	YES		NULL	
d21	varchar(7)	YES		NULL	
d22	varchar(7)	YES		NULL	
d23	varchar(7)	YES		NULL	
d24	varchar(7)	YES		NULL	
d25	varchar(7)	YES		NULL	
d26	varchar(7)	YES		NULL	
d27	varchar(7)	YES		NULL	
d28	varchar(7)	YES		NULL	
d29	varchar(7)	YES		NULL	
d30	varchar(7)	YES		NULL	
d31	varchar(7)	YES		NULL	

33 rows in set (0.00 sec)

Gambar 12 Implementasi Tabel dataMMYYYY

Tabel berikutnya dalam database homeXXXXXX pada host adalah tabel users. Tabel ini berbentuk sebagai berikut.

Nama Kolom	Tipe Data	Isi
phoneid	VARCHAR(8), PRIMARY KEY	ID pengguna
Email	VARCHAR(50)	Email pengguna
privilege	VARCHAR(14)	Privilege pengguna (ADMIN atau Guest)

aeskey	VARCHAR(16)	Kunci enkripsi AES untuk pengguna (akan dijelaskan lebih lanjut pada bagian kemananan jaringan)
--------	-------------	---

Tabel diimplementasikan dengan kode SQL sebagai berikut.

```
CREATE TABLE users (phoneid VARCHAR(16), email VARCHAR(50), privilege
VARCHAR(5), aeskey VARCHAR(16));
```

Dengan demikian, terbentuklah tabel users sebagai berikut.

```
mysql> describe users;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| phoneid | varchar(16) | NO | PRI | NULL | |
| email | varchar(50) | YES | | NULL | |
| privilege | varchar(5) | YES | | NULL | |
| aeskey | varchar(16) | YES | | NULL | |
+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Gambar 13 Implementasi Tabel users

Selain keenam tabel di atas, terdapat tabel-tabel tambahan untuk menyimpan informasi masing-masing skenario. Nama tabel tersebut disesuaikan dengan nama skenario. Tabel ini berfungsi untuk mengetahui perlakuan apa saja yang perlu dilakukan oleh masing-masing perangkat ketika skenario dijalankan. Bentuk tabel-tabel ini sama persis dengan bentuk tabel devices karena keduanya sama-sama berisi informasi perangkat dan statusnya. Contoh tabel ini adalah skenario lock_scen sebagai berikut.

```
mysql> describe lock_scen;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| name | varchar(20) | YES | | NULL | |
| address | varchar(20) | YES | | NULL | |
| type | varchar(11) | YES | | NULL | |
| status | varchar(9) | YES | | NULL | |
| room | varchar(20) | YES | | NULL | |
| privilege | varchar(5) | YES | | NULL | |
+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

mysql> select * from lock_scen;
+-----+-----+-----+-----+-----+
| name | address | type | status | room | privilege |
+-----+-----+-----+-----+-----+
| switch1 | 40A88C4C | switch | 100 | none | ADMIN |
| fan1 | 40A88C39 | fan | 0 | none | ADMIN |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Gambar 14 Implementasi Tabel Isi Skenario

Pada dokumen B300 juga telah dijelaskan bahwa database pada masing-masing *host* di setiap rumah akan selalu juga disimpan di server pusat sebagai *back up*. Dengan demikian, seluruh tabel yang dijelaskan di atas juga sebetulnya dibuat di server pusat.

5.1.3 Pengelolaan Basis Data dalam Bahasa Python

Program yang dijalankan pada server dan *host* diimplementasikan dalam bahasa Python (akan dijelaskan kemudian). Dengan demikian, pengelolaan basis data pun harus dilakukan dalam bahasa ini. Untuk melakukan pengelolaan basis data, digunakan suatu *library* bernama mysql.connector.

Ada beberapa jenis operasi pengelolaan basis data, yaitu:

- Operasi pertama adalah menambah isi tabel. Kode yang digunakan untuk menambah isi tabel adalah sebagai berikut.

```
INSERT INTO table_name VALUES (value1, value2, value3);
```

Value diisi dengan nilai-nilai yang hendak diisi pada tiap kolom.

- Operasi kedua adalah menghapus isi tabel. Kode yang digunakan untuk menambah isi tabel adalah sebagai berikut.

```
DELETE FROM table_name WHERE column_x = value_x;
```

Nilai column_x diisi dengan nama salah satu kolom yang hendak dijadikan patokan.

Nilai value_x diisi dengan nilai dari column_x yang barisnya hendak dihapus.

- Operasi ketiga adalah mengubah isi tabel. Kode yang digunakan untuk menambah isi tabel adalah sebagai berikut.

```
UPDATE table_name SET column_x = value_x WHERE column_y = value_y;
```

Column_x diisi dengan kolom yang hendak diubah isinya. Value_x adalah nilai baru dari column_x yang diinginkan. Column_y merupakan kolom patokan. Value_y merupakan nilai dari kolom patokan di mana barisnya hendak diubah nilainya.

- Operasi keempat adalah membaca dan mengambil isi tabel. Kode yang digunakan untuk menambah isi tabel adalah sebagai berikut.

```
SELECT column_1, column_2 WHERE column_x = value_x;
```

Column_1 dan column_2 merupakan kolom yang nilainya hendak diambil.

Column_x merupakan kolom patokan. Value_x merupakan nilai dari kolom_x yang hendak jadi patokan.

5.2 Implementasi Program pada Server dan Host

Pada dokument B300, telah dijelaskan bahwa program pada server dan *host* akan diimplementasikan menggunakan bahasa Python. Alasan utamanya adalah karena diperlukan penyesuaian dengan program untuk mengendalikan modul komunikasi Zigbee (komunikasi *indoor*). Modul XBee Pro memiliki library yang mudah digunakan pada bahasa Python. Dengan demikian, bahasa ini juga yang dipilih untuk komunikasi *outdoor*.

5.2.1 Implementasi Environment RabbitMQ pada Program Python

Seluruh elemen dalam subsistem *outdoor* berkomunikasi dengan protokol AMQP pada *environment* RabbitMQ. RabbitMQ memfasilitasi fitur *queueing* terhadap pesan yang dialirkan di dalam jaringan. RabbitMQ diinstalasi di server pusat. Setiap perangkat komunikasi bisa terlibat di jaringan ini dengan mengetahui username dan password dari akun RabbitMQ.

Library yang digunakan untuk terhubung dengan RabbitMQ adalah *library* pika. Dengan menggunakan *library* ini, sebuah *client* dapat terhubung ke RabbitMQ di suatu server dengan kode sebagai berikut.

```
credentials = pika.PlainCredentials('username', 'password')
parameters =
pika.ConnectionParameters('167.205.24.10', 5672, 'minds132', credentials)
connection = pika.BlockingConnection(parameters)
channel = connection.channel()
```

Penjelasan kode di atas adalah sebagai berikut.

- Pada baris pertama, *credentials* diisi dengan username dan password RabbitMQ pada server yang bersangkutan.
- Baris kedua merupakan pengisian parameter-parameter koneksi, yaitu alamat IP, port, nama virtual host RabbitMQ, serta *credentials* yang telah dimasukan pada baris pertama. Alamat IP disesuaikan dengan IP server. Port untuk RabbitMQ secara *default* adalah 5672. Nama *virtual host* disesuaikan dengan pengaturan pada server.
- Baris ketiga merupakan perintah untuk membuat koneksi ke RabbitMQ dengan parameter yang telah dideklarasikan pada baris kedua.
- Baris keempat merupakan baris untuk membuat *channel* atau saluran data ke RabbitMQ di server.

Setelah suatu *client* terkoneksi, maka *client* dapat melakukan *publish* maupun *subscribe* pesan. Kode untuk melakukan *publish* adalah sebagai berikut.

```
channel.basic_publish(exchange=' ',  
                      routing_key='queue_name',  
                      body='Hello World!')
```

Pada kode di atas, terdapat istilah *exchange*, *routing_key*, dan *body*. Untuk kasus-kasus pada sistem ini, *exchange* tidak digunakan. *Routing key* merupakan nama antrian (*queue*) yang menjadi tujuan dari pesan yang hendak dikirim. Setelah itu, *body* merupakan isi pesan yang hendak dikirimkan.

Selanjutnya adalah kode untuk melakukan *subscribe*. Untuk melakukan *subscribe*, suatu *client* harus mendefinisikan, ia ingin melakukan *subscribe* terhadap *queue* apa. Bila *queue* belum ada atau belum dibentuk di server, maka *queue* tersebut perlu dibuat terlebih dahulu.

```
channel.queue_declare(queue='queue_name')
```

Setelah *queue* dideklarasi, barulah *client* dapat melakukan *subscribe* ke *queue* tersebut dengan kode sebagai berikut.

```
def callback(ch, method, properties, body):  
    # diisi dengan algoritma yang akan dijalankan ketika suatu pesan  
    # diterima  
  
    channel.basic_consume(callback,  
                          queue='queue_name',  
                          no_ack=True)  
  
    channel.start_consuming()
```

Pada kode di atas, terdapat fungsi *callback*. Fungsi inilah yang akan dijalankan ketika suatu pesan data ke antrian. Isi pesan terkandung di dalam *body* yang terletak pada argumen fungsi. Isi fungsi *callback* sangat dibebaskan kepada *programmer*.

5.2.2 Implementasi Sistem Keamanan Jaringan dalam Bahasa Python

Dalam dokumen B300 telah dijelaskan bahwa jaringan komunikasi dalam sistem diamankan dengan dua buah metode enkripsi, yaitu AES dan RSA. Untuk mengimplementasikan keduanya, digunakan *library* PyCrypto. Implementasi enkripsi dan dekripsi AES adalah sebagai berikut.

```
def AESencrypt(input_stream, key):  
    iv = 'RandomInitVector'  
    cipher = AES.new(key, AES.MODE_CFB, iv)  
    return base64.b64encode(iv + cipher.encrypt(input_stream))
```

```

def AESdecrypt(input_stream, key) :
    iv = 'RandomInitVector'
    cipher = AES.new(key, AES.MODE_CFB, iv)
    return (cipher.decrypt(base64.b64decode(input_stream))[16:])

```

Pada fungsi di atas, key merupakan kunci enkripsi bertipe string. Karena enkripsi yang digunakan adalah AES 128 bit, maka panjang string key adalah 16 karakter (16 byte = 128 bit). IV merupakan *initial vector*, yaitu nilai awal dari string *cipher* atau hasil enkripsi. Hasil enkripsi di-*encode* menggunakan *base64 encoding* karena hasil enkripsi adalah data byte array. Dengan di-*encode* menggunakan base64, maka hasil enkripsi bertipe string.

Untuk mengimplementasikan enkripsi dan dekripsi RSA, digunakan kode sebagai berikut.

```

RSAprivKey = RSA.importKey(base64.b64decode('String_Private_Key'))
RSAPubKey = RSA.importKey(base64.b64decode('String_Public_Key'))

def RSAencrypt(input_stream) :
    global RSAPubKey
    return
        base64.b64encode(RSAPubKey.encrypt(input_stream, 'x')[0])

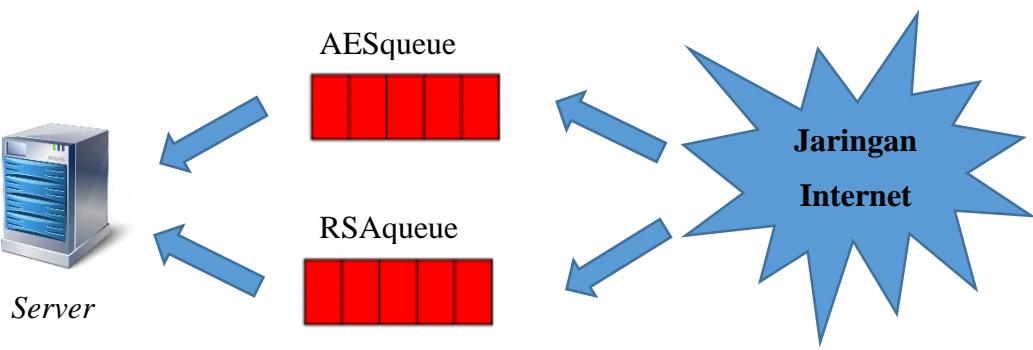
def RSAdecrypt (input_stream):
    global RSAprivKey
    return RSAprivKey.decrypt(base64.b64decode(input_stream))

```

Pada kode di atas, String_Private_Key dan String_Public_Key diisi dengan string kunci publik dan privat RSA. String kunci perlu di-*decode* dulu menggunakan base64 karena memang dalam keadaan terkode base64. Sama seperti pada AES, hasil enkripsi harus di-*encode* menggunakan base64. Dengan demikian, input fungsi dekripsi perlu di-*decode* terlebih dahulu.

5.2.3 Implementasi Program Python pada Server

Karena terdapat dua macam enkripsi yang diterapkan dalam sistem, maka server harus melakukan *subscribe* terhadap dua antrian, yaitu antrian data yang dienkripsi dengan AES dan antrian data yang dienkripsi dengan RSA (telah dijelaskan pada B300).



Gambar 15 Antrian Data ke Server

Dengan demikian, terdapat dua program Python yang harus dibuat, yaitu program untuk mengeksekusi data dari RSAqueue dan AESqueue. Data yang mengantri di antrian RSAqueue dieksekusi menggunakan program server_rsa.py. Berikut isi kode program tersebut.

```
#server_rsa.py

1  import mysql.connector
2  import sys
3  import gc
4  import pika
5  import json
6  from random import randint
7  from Crypto.PublicKey import RSA
8  from Crypto.Cipher import AES
9  import base64
10 import smtplib
11 from datetime import datetime,timedelta
12
13 #KUNCI ENKRIPSI DAN DEKRIPSI RSA
14 RSAprivKey =
RSA.importKey(base64.b64decode('MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwg
SjAgEAAoIBAQCQd0CaamKDmTB5ulbre/SLcrlzEl4I8O4hXpVJNF8SBEXvm0xLcAPGRk1
hJS18dKSx+PFd3nboTUga/WqBzdqlVfgjobdBSrKwf/o2miG2gB7zk1StpD7CGYBzqTBg
v1A1g78bcjNqAN1oLwC1Dwgnol7NocN43k7J3i5uqXW/rkgIB/Rrd2ouBMWe0GsOUJaQgn
d+7bPaFlAa3di4/xjnAkTdlekJzStx7qsx173ziRCrU+wU1zCXGHfUqSzH2AH5BWvcng6
i3Hr8vu2rZxChm/BUkUn6C1a9JHBFxj6wsg+Mxi4zL46RDaHy5kgm+jH48HMMeJ0k/ERTI
od65AgMBAAECggEAdm69MjosfxQz8Gc54L1y4ms/7x0z0eEIEBqlCm4vqsJt+NgtLhcCj5
4U05IIVAyWg6yJZxYmeU50fkrjT+uwm5kpPLe7ijA1P0R/DocXEA5iPRz/9/7LXGmVtMug
Um2vmcKPMD1d1OlwV0FxjejDcvv9IYhUFTgKhOd5dJWoHQeNzBMXuecYvfecW1c2TgW1FqV
HyGhTEfztCxyafH2UtbtNW3neq60IfB5gnTmB3tuUFhZ8KgmSgVBRBalFB71pu9miJ7sjs
5GZJYJiQvoReSOgJ16OcyexnubeU0d4YaJhzmtHOZywiNfkA9cHexHmxy1qzQs8uXdF5VK
qDGtO+nQKBgQDN28fb7TmIt7yHYNE9hA7CNetiho3UWrusbPkJZgkDYX6zL/IFjUwNURq
aXOGeCEGBbjx9TpIGUAbl1G1fxnZRQzbpp8PKU0HL7+qQrXu15q8+z7GwOHLov7W1kCFdc
r11o6YHRZxj/nHNH018hPC0k9kt070XSOCoUTMPQxouwKBgQCzp1tiaYmq0tfK/sYRk8t9
bKZ5dRNHLJYE1QCdjkBj67B0e/HCck20KWFHg11sZd3HTEfcWCNzUPXHy98vYuxQf4+uCWP
hrlwTSPShPlMFCPCYYe7M363aPR1dhU2aH/XHA2H+No9GvaQL5dMC2ZQPgLo7d4yLSbto
KtLQWMjJGwKBgCVMJr/gEUaF+g1/82KxOmuxxIv6s51GNRueMrupXBFCKeQw6KZ175kWbs
Zd6MJ6TXFr+9WwQc5D/Z7ND6wEpzc9+FsjPF4SC5/JjjQ8WmewrhuxQb5t286tb4K50uNz
XhyAjUFeM8zLizPefLY9T/tCY4QHIoT/wwlN8ssWXIM1AoGBAJ9aHBnvyPfnqxdY4zFmKg
```

```

InpIZpTV89chfAXtpjl7DGzJHWksCfmAGsE3wmZNzy8Niqv8Mgulfa5L8CAyNL1W7wCM8IMtiEDPe63Y2xQbbRiW0AFBjO18wANbOTeY0J2RzkLEfjfjOb7tJ+xu3ZNpaFtSrYQKawEPY1veamSfrvAoGAGR1YL3sOj7L32o9ybbupV8q35U463Arc9RhHqA/fn436s0+VPYe+e43tiFk8P+hQ8GAYJtEh8rVmGGqHiQ9SsXWSmBEpvDBf5mnC47J8x3PUBcU4YXK9HaziOI8mUuyJxJSVlOgNe1cELCrMw4mmmy35SZRnEM6VjYytihpabP/Q='"))
15     RSApubKey =
RSA.importKey(base64.b64decode('MIIBIjANBgkqhkiG9w0BAQEFAOCQ8AMIIIBCgKCAQEAKHdAmmpig5kwebpW63v0i3K5d8xJeCPDuIV6VSTRfEgRF75tMS3ADxkZNYsUtfHksfjxxd526E1IGv1qgc3apVX4I6G3QUqyln/6NpohtoAe85CpUk6Q+whmA6kwYL9QNYO/G3IzagDdaC8AtQ8IJ6JezahDeN5Oyd4ubql1v65ICAf0a3dqLgTFntBrD1CWkJ3fu2z2hZQGt3Yo+P8Y5wJE3ZXpPc0rce6rMZe984kQq1P11Ncwlxh31Kksx9gB+QVr3J4ootx6/L7tq2cQoZvwVJFJ+gpWvSRwRcY+sLIPjMYuMy+OkQ2h8uZIJvox+PBzJhCdJPxEbSKHeuQIDAQAB'))
16
17     #INITIAL VECTOR UNTUK AES
18     iv = 'RandomInitVector'
19
20     #FUNGSI UNTUK MENGIRIM EMAIL KETIKA ADA USER LUPA PASSWORD
21     def send_email_forgetpwd(to,newpwd) :
22         gmail_user = 'mindssmarthome@gmail.com'
23         gmail_password = 'BRBisthebest'
24
25         #judul pesan
26         subject = 'Request New Password'
27         body = "Your new password :\n\n" + newpwd
28
29         #membentuk pesan
30         email_text = """Subject: %s
31
32             %s"""\ % (subject, body)
33
34         #mengirim pesan
35         try:
36             server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
37             server.ehlo()
38             server.login(gmail_user, gmail_password)
39             print ('bro')
40             server.sendmail(gmail_user, to, email_text)
41             server.close()
42             print ('Email sent!')
43         except:
44             print ('Something went wrong... ')
45
46     #FUNSI UNTUK MEMVERIFIKASI DATA USER KETIKA ADA YANG SIGN IN
47     def findUserSI (data,email,password):
48         for row in data:
49             temp = {}
50             if (row[0] == email and row[1] == password):
#correct authentication
51                 if (len(row[2]) > 6): #belum memasukan kode
unik
52                     temp['homeid'] = 'none'
53                     temp['name'] = row[3]
54                     temp['picture'] = row[4]
55                     temp['privilege'] = row[5]
56                     return (temp)
57             else : #lolos
58                 temp['homeid'] = row[2]

```

```

59                     temp['name'] = row[3]
60                     temp['picture'] = row[4]
61                     temp['privilege'] = row[5]
62                     return (temp)
63             temp['homeid'] = '0'
64             return (temp) #wrong username or password
65
66 #FUNGSI UNTUK MEMVERIFIKASI USER KETIKA ADA YANG SIGN UP
67 def findUserSU (data,email):
68     for row in data:
69         if (row[0] == email):    #check if email already
registered
70             return ('0')           #SU not successful
71     return ('1') #SU successful
72
73 #FUNGSI UNTUK ENKRIPSI RSA
74 def RSAencrypt (input) :
75     global RSApubKey
76     return base64.b64encode(RSApubKey.encrypt(input,'x')[0])
77
78 #FUNGSI UNTUK DEKRIPSI RSA
79 def RSAdecrypt (input) :
80     global RSAprivKey
81     return RSAprivKey.decrypt(base64.b64decode(input))
82
83 #FUNGSI UNTUK ENKRIPSI AES
84 def AESencrypt (input,key) :
85     global iv
86     cipher = AES.new(key, AES.MODE_CFB, iv)
87     return (base64.b64encode(iv + cipher.encrypt(input)))
88
89 #FUNGSI UNTUK DEKRIPSI AES
90 def AESdecrypt (input,key) :
91     global iv
92     cipher = AES.new(key, AES.MODE_CFB, iv)
93     return (cipher.decrypt(base64.b64decode(input))[16:])
94
95 #CONNECT TO RABBITMQ
96 credentials = pika.PlainCredentials('mindsTA','minds')
97 parameters =
pika.ConnectionParameters('localhost',5672,'mindsTA132',credentials)
98 connection_rsa = pika.BlockingConnection(parameters)
99 channel_rsa = connection_rsa.channel()
100
101 print(' [*] Waiting for messages. To exit press CTRL+C')
102
103 #FUNGSI YANG DIJALANKAN KETIKA ADA DATA MASUK KE ANTRIAN
RSAqueue
104 def callbackRSA (ch, method, props, body):
105     global RSAprivKey
106     global RSApubKey
107
108     #menerima data, lalu menerjemahkan data JSON
109     sent = json.loads(RSAdecrypt(body))
110     print ("[x] Received: " + sent)
111
112     #membagi kategori pesan berdasarkan header

```

```

113     if (sent['head'] == 'SI') : #header sign in
114         email = sent['email']
115         password = sent['password']
116
117     try :
118         #akses database
119         cnx = mysql.connector.connect(user='brb',
120                                         password='28031995', host='localhost', database='general_data')
121         cursor = cnx.cursor()
122         cursor.execute("SELECT * FROM users")
123         data = cursor.fetchall()
124         hasil = {}
125         #memverifikasi email dan password user
126         hasil = findUserSI(data,email,password)
127
128         #jawaban '0' berarti salah email atau salah
129         #password
130         if (hasil['homeid'] == '0'):
131             #jawaban dikirim
132             ch.basic_publish(exchange='',
133
134             routing_key=props.reply_to,
135
136             properties=pika.BasicProperties(correlation_id =
137
138                                         props.correlation_id),
139
140             body=AESencrypt(json.dumps(hasil),sent['aeskey']))
141
142             #jawaban bukan '0' berarti email dan password
143             benar
144             else:
145                 #melakukan update AES key di tabel
146                 aeskeylist
147                 cursor.execute("SELECT EXISTS (SELECT
148                 phoneid FROM aeskeylist WHERE phoneid = %s)",(props.correlation_id,))
149                 if (cursor.fetchone()[0]) :
150                     cursor.execute("UPDATE aeskeylist
151                     SET aeskey = %s WHERE phoneid = %s ", (sent['aeskey'],
152                     props.correlation_id))
153                     else :
154                         cursor.execute("INSERT INTO
155                         aeskeylist values (%s,%s)",(props.correlation_id,sent['aeskey']))
156                         cnx.commit()
157                         #mengirim balasan ke user
158                         ch.basic_publish(exchange='',
159
160                         routing_key=props.reply_to,
161
162                         properties=pika.BasicProperties(correlation_id =
163
164                                         props.correlation_id),
165
166                         body=AESencrypt(json.dumps(hasil),sent['aeskey']))
167
168             except mysql.connector.Error as err:

```

```

153             print (err)
154             ch.basic_publish(exchange=' ', 
155
155 routing_key=props.reply_to,
156
156 properties=pika.BasicProperties(correlation_id = \
157
157                         props.correlation_id),
158
158                     body="0")
159
159
160     elif (sent['head'] == 'MU') : #manajemen user
161         if (sent['type'] == 'forgetpwd') :
162             #generate password baru (6 digit)
163             newpwd = str(randint (100000,999999))
164             #akses database
165             cnx = mysql.connector.connect(user='brb',
password='28031995', host='localhost', database='general_data')
166             cursor = cnx.cursor()
167             #cek apakah email requester terdaftar
168             cursor.execute("SELECT EXISTS (SELECT email
FROM users WHERE email = %s)", (sent['email'],))
169
169             #jika email ditemukan
170             if (cursor.fetchone()[0]) :
171                 #ubah password di database
172                 cursor.execute("UPDATE users SET
password = %s WHERE email = %s", (newpwd,sent['email']))
173                 cnx.commit()
174                 #kirim password baru ke email user
175
175 send_email_forgetpwd(sent['email'],newpwd)
176
176                     #kirim balasan ke ponsel user
177                     ch.basic_publish(exchange=' ',
178
178 routing_key=props.reply_to,
179
179 properties=pika.BasicProperties(correlation_id = \
180
180                         props.correlation_id),
181
181                     body=AESencrypt ('1',sent['aeskey']))
182                     #jika email tidak ditemukan, kirim balasan '0'
183                     else :
184                         ch.basic_publish(exchange=' ',
185
185 routing_key=props.reply_to,
186
186 properties=pika.BasicProperties(correlation_id = \
187
187                         props.correlation_id),
188
188                     body=AESencrypt ('0',sent['aeskey']))
189                     #tutup koneksi ke database
190                     cursor.close()
191                     cnx.close()
192                     gc.collect()
193
194

```

```

195     elif (sent['head'] == 'SU') : #sign up
196         #akses database
197         cnx = mysql.connector.connect(user='brb',
password='28031995', host='localhost', database='general_data')
198         cursor = cnx.cursor()
199         cursor.execute("SELECT * FROM users")
200         data = cursor.fetchall()
201         #cek apakah email belum pernah didaftarkan
sebelumnya
202         hasil = findUserSU(data,sent['email'])
203
204         #hasil = '1' berarti email belum pernah didaftarkan
205         if (hasil == '1') :
206             #tambahkan user baru ke database
207             cursor.execute("INSERT INTO users (email,
password, homeid, name, picture, privilege) VALUES
(%s,%s,%s,%s,%s)", (sent['email'],sent['password'],
'none',sent['name'], 'none','none'))
208             cursor.execute("INSERT INTO aeskeylist
(phoneid, aeskey) VALUES (%s,%s)",
(props.correlation_id,sent['aeskey']))
209             cnx.commit()
210             #kirim balasan ke ponsel user
211             ch.basic_publish(exchange='',
212                             routing_key=props.reply_to,
213
properties=pika.BasicProperties(correlation_id = \
214
                                props.correlation_id,
215
                                body=RSAencrypt(hasil))
216             # tutup koneksi ke database
217             cursor.close()
218             cnx.close()
219             gc.collect()
220 ##### AKHIR DARI FUNGSI CALLBACK #####
221
222 #server consume data ke RSAqueue
223 channel_rsa.basic_consume(callbackRSA,
queue='RSAqueue',no_ack=True)
224 channel_rsa.start_consuming()
225 #####
226 ##### AKHIR DARI server_rsa.py #####

```

Program server_rsa.py memiliki beberapa function yang memiliki *objective* sebagai berikut:

Tabel 8 Fungsi pada server_rsa.py

Nama Fungsi	Objective Fungsi	Data Return
send_email_for getpwd (to,newpwd)	Mengirim email berisi password baru kepada <i>user</i> yang lupa password	Tidak ada

findUserSI (data, email, password)	Memverifikasi apakah email dan password untuk <i>sign in</i> valid dan sesuai dengan database	<ul style="list-style-type: none"> • Return ‘0’ jika email atau password salah • Return informasi <i>user</i> jika email dan password tepat
findUserSU (data, email, password)	Memverifikasi apakah email yang akan didaftarkan (<i>sign up</i>) valid (belum pernah didaftarkan sebelumnya)	<ul style="list-style-type: none"> • Return ‘1’ bila email valid • Return ‘0’ bila tidak valid
RSAencrypt (input)	Melakukan enkripsi RSA terhadap input	Hasil enkripsi
RSAdecrypt (input)	Melakukan dekripsi RSA terhadap input	Hasil dekripsi
AESencrypt (input,key)	Melakukan enkripsi AES terhadap input dengan kunci key	Hasil enkripsi
AESdecrypt (input,key)	Melakukan dekripsi AES terhadap input dengan kunci key	Hasil dekripsi

Bagian *main program* dari server_rsa.py berisi algoritma yang akan dijalankan ketika suatu pesan datang. Setiap pesan pasti didekripsi dulu menggunakan RSA. Kemudian, pesan JSON dipecah dulu sehingga diperoleh bagian-bagian dalamnya. Algoritma dipilih berdasarkan isi *header* pesan yang datang. Bagian ini mengimplementasikan sebagian dari yang telah dijelaskan pada **tabel xxx** di dokumen B300. Berikut adalah bagian-bagian yang diimplementasikan dalam program ini.

Tabel 9 Pesan yang Dilayani oleh server_rsa.py

Header	Pengirim	Tipe	Fungsi	Isi Paket Permintaan	Aksi Server
SI (Sign In)	Ponsel	-	Permintaan untuk sign in ke akun tertentu	<ul style="list-style-type: none"> • Email pengguna • Password 	<p>Server mengecek email dan password ke database, apakah sesuai.</p> <p>Jika email dan password sesuai dan pengguna sudah terdaftar di suatu rumah,</p>

					<p>server menjawab:</p> <ul style="list-style-type: none"> • Home ID • Nama pengguna • Gambar profil • Privilege pengguna (ADMIN atau Guest) • User ID <p>Jika email dan password tidak sesuai, server menjawab:</p> <ul style="list-style-type: none"> • Jawaban berupa karakter ‘0’ <p>Jika email dan password sesuai, tapi pengguna belum terdaftar di suatu rumah tertentu, server menjawab:</p> <ul style="list-style-type: none"> • Jawaban berupa string ‘none’
SU (Sign Up)	Ponsel	-	Permintaan untuk sign up atau mendaftarkan akun tertentu	<ul style="list-style-type: none"> • Email pengguna • Password • Nama pengguna 	<p>Server mengecek apakah email sudah terdaftar.</p> <p>Jika email belum pernah didaftarkan sebelumnya:</p> <ul style="list-style-type: none"> • Jawaban berupa karakter ‘1’ <p>Jika email sudah terdaftar:</p>

					• Jawaban berupa karakter ‘0’
MU (Manajemen Users)	Forgetpwd	Permintaan reset password dari pengguna yang lupa password	• Email peminta	Server menggenerasikan 6 digit angka acak, lalu mengubah password peminta di database menjadi 6 angka tersebut. Selain itu, 6 angka itu juga dikirimkan ke email peminta.	

Program kedua adalah server_aes.py. Program ini berisi algoritma-algoritma yang akan dijalankan untuk memproses data yang datang ke antrian AESqueue. Berikut adalah kode program.

```
#server_aes.py
1 import mysql.connector
2 import sys
3 import gc
4 import pika
5 import json
6 from random import randint
7 from Crypto.Cipher import AES
8 from Crypto.PublicKey import RSA
9 import base64
10 import smtplib
11 from datetime import datetime
12
13 #INITIAL VECTOR UNTUK AES
14 iv = 'RandomInitVector'
15
16 #KUNCI RSA
17 RSAprivKey =
RSA.importKey(base64.b64decode ('MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwgg
SjAgEAAoIBAQCD0CaamKDmTB5ulbre/SIcrc13zE14I8O4hXpVJNF8SBExvm0xLcAPGRk1
hJS18dKSx+PFd3nboTUga/WqBzdqlVfgjobdBSrKwf/o2miG2gB7zkK1STpD7CGYBzqTBg
v1A1g78bcjNqAN1oLwC1Dwgno17NocN43k7J3i5uqXW/rkgIB/Rrd2ouBMWe0GsOUJaQgn
d+7bPaFlAa3di4/xjnAkTdle9zStx7qsx173ziRCrU+WU1zCXGHfUqSzH2AH5BWvcng6
i3Hr8vu2rZxChm/BUkUn6C1a9JHBFxj6wsg+Mxi4zL46RDaHy5kgm+jH48HMmEJ0k/ERTI
od65AgMBAECggEAdm69MjosfxQz8Gc54L1y4ms/7x0z0eEIEBqlCm4vqsJt+NgtLhcCj5
4U05IIVAyWg6yJZxYmeU50fkrjT+uwm5kpPLe7ijA1P0R/DocXEA5iPRz/9/7LXGmVtMug
Um2vmcKPMD1d10lw0fXejDcvv9IYhUFTgKhOd5dJWoHQeNzBMXuecYvfecW1c2TgW1FqV
HyGhTEfztCxyafH2UtbtNW3neq601fB5gnTmB3tuUFhZ8KgmSgVBRBalFB71pu9miJ7sjs
5GZJYJiQvoReSoGj160cyexnubeU0d4YaJhzmtHOZywiNfkA9cHexHmxy1qzQs8uXdF5VK
qDGtO+nQKBgQDN28fb7TmIt7yHYNE9hA7CNetih03UWrusbPkJZgkDYX6zL/IFjUwNURq
aXOGeCEGBjx9TpGUAb11G1fxnZRQzbpp8PKU0HL7+qQrXu15q8+z7GwoHLLov7W1kCFdc
r11o6YHRZxj/nHNH018hPC0k9kT070XSOCoUTMPQxouwKBgQCzp1tiaYmq0tfK/sYRk8t9
```

```

bKZ5dRNHLJYE1QCdkBJ67B0e/HCck20KWFHgl1sZd3HTEfcWCNzUPXHy98vYuxQf4+uCWP
hrlwTSPShPlMFCPCYYe7M363aPR1dhU2aN/HXA2H+No9GvaQL5d6MC2ZQPgLo7d4yLSbto
KtLQWMjJGwKBgCVMJr/gEUaF+g1/82KxOmuxxIv6s51GNRueMrupXBFcKeQw6KZ175kWbs
Zd6MJ6TXFr+9WwQc5D/Z7ND6wEpzc9+FsjPF4SC5/JjjQ8WmewrhuxQb5t286tb4K5OuNz
XhyAjUFeM8zLizPefLY9T/tCY4QHIOt/wwlN8ssWXIM1AoGBAJ9aHBnvyPfnqxdY4zFmKg
InpIZpTV89chfAXtpj17DGzJHWksCfmAGsE3wmZNzy8Niqv8Mgu1fa5L8CAyNL1W7wCM8I
MtiedPe63Y2xQbbRiW0AFbj018wANbOTeY0J2RzkLEfjfjOb7tJ+xu3ZNPafSrYQKawEP
Y1veamSfrvAoGAGR1YL3sOj7L32o9ybbupV8q35U463Arc9RhHqA/fn436s0+VPYe+e43t
iFk8P+hQ8GAYJtEh8rVmGGqHiQ9SSxWSmBEPvDBf5mnC47J8x3PUBcU4YXK9HaziOI8mUu
yJxJSV1OgNe1cELCrMw4mmmy35SRnEM6VjYytihpabP/Q='))

18     RSApubKey =
RSA.importKey(base64.b64decode('MIIBIjANBgkqhkiG9w0BAQEFAOCQ8AMIIIBCg
KCAQEAkHdAmmpig5kwebpW63v0i3K5d8xJeCPDuIV6VSTRfEgRF75tMS3ADxkZNYsUtfHS
ksfjxxd526E1IGv1qgc3apVX4I6G3QUqyln/6NpohToAe85CpUk6Q+whmAc6kwYL9QNYO/
G3IzagDdaC8AtQ8IJ6JezaHDeN5Oyd4ubql1v65ICAf0a3dqLgTFntBrD1CWkIJ3fu2z2h
ZQGt3Yo+P8Y5wJE3ZXpPc0rce6rMZe984kQq1Pl1Ncwlxh31Kksx9gb+QVr3J4Ootx6/L7
tq2cQoZvwVJFJ+gpWvSRwRcY+sLIPjMYuMy+OkQ2h8uZIJvox+PBzJhCdJPxEbSKHeuQID
AQAB'))

19
20 #FUNGSI UNTUK MENGIRIM EMAIL INVITASI
21 def send_email_invitation(to,homename, sender, code) :
22     gmail_user = 'mindssmarthome@gmail.com'
23     gmail_password = 'BRBisthebest'
24
25     subject = 'Invitation'
26     body = "You are invited to '" + homename + "' by " +
sender + "\n\nClick 'Get Invited' in your app and enter this code to
join:\n\n" + code
27
28     email_text = """Subject: %s
29
30     %s"""\n % (subject, body)
31
32     try:
33         server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
34         server.ehlo()
35         server.login(gmail_user, gmail_password)
36         server.sendmail(gmail_user, to, email_text)
37         server.close()
38         print ('Email sent!')
39     except:
40         print ('Something went wrong...')

41
42 #FUNGSI ENKRIPSI RSA
43 def RSAencrypt (input) :
44     global RSApubKey
45     return base64.b64encode(RSApubKey.encrypt(input,'x')[0])
46
47 #FUNGSI UNTUK MEMPEROLEH AES KEY SUATU PONSEL
48 def getkeyfromdatabase (phoneid) :
49     cnx = mysql.connector.connect(user='brb',
password='28031995', host='localhost', database='general_data')
50     cursor = cnx.cursor()
51     cursor.execute("SELECT aeskey FROM aeskeylist WHERE
phoneid = '" + phoneid + "'")
52     return (cursor.fetchone()[0])
53
54 #FUNGSI UNTUK ENKRIPSI AES

```

```

55     def AESencrypt(input,key):
56         global iv
57         cipher = AES.new(key, AES.MODE_CFB, iv)
58         return base64.b64encode(iv + cipher.encrypt(input))
59
60 #FUNGSI DEKRIPSI AES
61 def AESdecrypt(input,key) :
62     global iv
63     cipher = AES.new(key, AES.MODE_CFB, iv)
64     return (cipher.decrypt(base64.b64decode(input))[16:])
65
66 #FUNGSI UNTUK MEMVERIFIKASI PROSES REGISTRASI USER BARU DI RUMAH
67 def findUserRegister (data,email):
68     for row in data:
69         if (row[0] == email): #check if already registered
70             return ('0')
71     return ('1') #SU successful
72
73 #FUNGSI UNTUK MEMVERIFIKASI PROSES KONFIRMASI USER BARU DI RUMAH
74 def findUserConfirm (data,email,code):
75     for row in data:
76         if (row[0] == email and row[2][0:6] == code): #check
if the code is correct
77             res = row[2][6:]
78             return (res)
79     return ('0') #Confirm Unsuccessful
80
81 #FUNGSI UNTUK MEMERIKSA APAKAH USER SEORANG ADMIN ATAU BUKAN
82 def checkAdmin (data, email) :
83     for row in data:
84         if (row[1] == email and row[2] == 'ADMIN'): #check
if the user is ADMIN
85             return ('0')
86     return ('2') # not ADMIN
87
88
89 #CONNECT TO RABBITMQ
90 credentials = pika.PlainCredentials('mindsTA','minds')
91 parameters =
pika.ConnectionParameters('localhost',5672,'mindsTA132',credentials)

92 connection_aes = pika.BlockingConnection(parameters)
93 channel_aes = connection_aes.channel()
94
95 print(' [*] Waiting for messages. To exit press CTRL+C')
96
97 def callbackAES (ch, method, props, body):
98
99     #memperoleh kunci untuk pengirim
100    key = getkeyfromdatabase(props.correlation_id)
101    #memecah pesan JSON
102    sent = json.loads(AESdecrypt(body,key))
103    print("[x] Received : " + AESdecrypt(body,key))
104
105    if (sent['head'] == 'SO') : #sign out
106        #akses database

```

```

107         cnx = mysql.connector.connect(user='brb',
108                                         password='28031995', host='localhost', database='general_data')
109         cursor = cnx.cursor()
110         #menghapus key dari database
111         cursor.execute("DELETE FROM aeskeylist WHERE phoneid
= '" + props.correlation_id + "'")
112         cnx.commit()
113         cursor.close()
114         cnx.close()
115         gc.collect()

116     if (sent['head'] == 'MU') : #manajemen user
117         if (sent['type'] == 'list'):
118             cnx = mysql.connector.connect(user='brb',
119                                         password='28031995', host='localhost', database='general_data')
120             cursor = cnx.cursor()
121             #mengambil seluruh user di dalam rumah yang
bersangkutan
122             cursor.execute("SELECT name,email,privilege
FROM users WHERE homeid = " + sent['homeid'])
123             data = cursor.fetchall()
124             #mengirim balasan
125             ch.basic_publish(exchange='',
routing_key=props.reply_to,
126
properties=pika.BasicProperties(correlation_id =
127                                         props.correlation_id),
128

body=AESencrypt(json.dumps(data),key))
129             print (json.dumps(data))

130
131             cursor.close()
132             cnx.close()
133             gc.collect()

134
135     elif (sent['type'] == 'register') :
136         cnx = mysql.connector.connect(user='brb',
password='28031995', host='localhost', database='general_data')
137         cursor = cnx.cursor()
138         #mengambil seluruh anggota rumah
139         cursor.execute("SELECT name,email,privilege
FROM users WHERE homeid = " + sent['homeid'])
140         data = cursor.fetchall()
141         #memeriksa apakah pengirim ADMIN
142         check = checkAdmin (data,sent['sender'])
143         #jika admin
144         if (check != '2') :
145             cursor.execute("SELECT * FROM users")
146             data = cursor.fetchall()
147             #cek apakah email terundang memang valid
(belum diinvite kemanapun)
148             hasil = findUserRegister
(data,sent['email'])


```

```

149                                     #jika valid
150         if (hasil == '0') :
151             #generate kode unik
152             code = str(randint
153                                         (100000, 999999))
154             #kode unik disisipkan di homeid
155             dan privilege
156             cursor.execute("UPDATE users SET
157             homeid = %s, privilege = %s WHERE email = %s ",
158             ((code)+sent['homeid'], (code)+sent['privilege'], sent['email']))
159             cnx.commit()
160             cnx =
161             mysql.connector.connect(user='brb', password='28031995',
162             host='localhost', database='home'+sent['homeid'])
163             cursor = cnx.cursor()
164             cursor.execute("SELECT name FROM
165             info")
166             #kirim email
167             send_email_invitation(sent['email'],cursor.fetchone()[0],sent['s
168             ender'],code)
169             #kirim balasan kode unik
170             ch.basic_publish(exchange='',
171             routing_key=props.reply_to,
172             properties=pika.BasicProperties(correlation_id =
173
174             props.correlation_id),
175
176             body=AESencrypt(code,key))
177             #jika email tidak valid (sudah terdaftar
178             di rumah lain)
179             else :
180                 ch.basic_publish(exchange='',
181
182             routing_key=props.reply_to,
183             properties=pika.BasicProperties(correlation_id =
184
185             props.correlation_id),
186
187             body=AESencrypt(hasil,key))
188             #jika pengirim bukan admin
189             else :
190                 ch.basic_publish(exchange='',
191
192             routing_key=props.reply_to,
193             properties=pika.BasicProperties(correlation_id =
194
195             props.correlation_id),

```

```

181                                     body=AESencrypt(check,key))
182
183             cursor.close()
184             cnx.close()
185             gc.collect()
186
187         elif (sent['type'] == 'confirm') :
188             cnx = mysql.connector.connect(user='brb',
189                                         password='28031995', host='localhost', database='general_data')
190             cursor = cnx.cursor()
191             cursor.execute("SELECT * FROM users")
192             data = cursor.fetchall()
193             hasil = {}
194             #periksa apakah kode benar
195             hasil['homeid'] =
196             findUserConfirm(data,sent['email'],sent['code'])
197
198             #KODE BERHASIL TERKONFIRMASI
199             if (hasil['homeid'] != '0') :
200                 cursor.execute("UPDATE users SET homeid
201 = %s WHERE email = %s", (hasil['homeid'],sent['email']))
202                 cursor.execute("SELECT privilege FROM
203 users WHERE email = '" + sent['email'] + "'")
204                 data = cursor.fetchone()
205                 hasil['privilege'] = data[0][6:]
206
207                 cursor.execute("UPDATE users SET
208 privilege = %s WHERE email = %s", (hasil['privilege'],sent['email']))
209                 cnx.commit()
210
211                 ch.basic_publish(exchange='',
212                                 routing_key=props.reply_to,
213
214                                 properties=pika.BasicProperties(correlation_id =
215                                         props.correlation_id),
216
217                                         body=AESencrypt(json.dumps(hasil),key))
218
219             cursor.close()
220             cnx.close()
221             gc.collect()
222
223         elif (sent['type'] == 'editprof') :
224             cnx = mysql.connector.connect(user='brb',
225                                         password='28031995', host='localhost', database='general_data')
226             cursor = cnx.cursor()
227             #ubah data dalam database
228             cursor.execute ("UPDATE users SET picture
229 = %s, name = %s, password = %s WHERE email = %s",
230             (sent['picture'],sent['name'],sent['password'], sent['email']))
231             cnx.commit()
232             cursor.close()
233             cnx.close()
234             gc.collect()

```

```

224
225         elif (sent['type'] == 'deleteuser') :
226             cnx = mysql.connector.connect(user='brb',
227                                         password='28031995', host='localhost', database='general_data')
228             cursor = cnx.cursor()
229             #ubah homeid dan privilege user yang dihapus
dari rumah
230             cursor.execute ("UPDATE users SET homeid =
231                             'none', privilege = 'none' WHERE email = %s", (sent['email'],))
232             cnx.commit()
233             cursor.close()
234             cnx.close()
235             gc.collect()
236
237         elif (sent['type'] == 'editaeskey') :
238             cnx = mysql.connector.connect(user='brb',
239                                         password='28031995', host='localhost', database='general_data')
240             cursor = cnx.cursor()
241             #ubah aeskey
242             cursor.execute ("UPDATE aeskeylist SET aeskey
= %s WHERE phoneid = %s", (sent['aeskey'],props.correlation_id))
243             cnx.commit()
244             cursor.close()
245             cnx.close()
246             gc.collect()
247
248         elif (sent['head'] == 'MH') : #manajemen house
249             if (sent['type'] == 'edit') :
250                 try :
251                     cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
252                     cursor = cnx.cursor()
253                     #ubah nama dan koordinat rumah
254                     cursor.execute("UPDATE info SET name
= %s, latitude = %s, longitude = %s",
(sent['name'],sent['lat'],sent['lng']))
255                     cnx.commit()
256                     cursor.close()
257                     cnx.close()
258                     gc.collect()
259
260             except mysql.connector.Error as err :
261                 print (err)
262
263         elif (sent['head'] == 'MD') : #manajemen device
264             if (sent['type'] == 'adddevice') :
265                 try:
266                     cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
267                     cursor = cnx.cursor()
268                     #masukan device ke database
if (sent['type'] == 'lamp') :
cursor.execute("INSERT INTO
devices VALUES (%s,%s,%s,%s,%s)",
(sent['name'],sent['address'],sent['devtype'],'ff000000','none'))
elif (sent['type'] == 'lamp') :

```



```

312                                     cursor.execute("UPDATE devices SET
room = %s WHERE name = %s", (sent['name'], element[0]))
313                                         cnx.commit()
314                                         cursor.close()
315                                         cnx.close()
316                                         gc.collect()
317             except mysql.connector.Error as err :
318                 print (err)
319
320             elif (sent['type'] == 'deleteroom') :
321                 try :
322                     cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
323                                         cursor = cnx.cursor()
324                                         #hapus nama room dari tabel rooms
cursor.execute("DELETE FROM rooms WHERE
name = %s", (sent['name'],))
325                                         #ubah isi tabel devices
cursor.execute("UPDATE devices SET room
= %s WHERE room = %s", ('none', sent['name']))
326                                         cnx.commit()
327                                         cursor.close()
328                                         cnx.close()
329                                         gc.collect()
330             except mysql.connector.Error as err :
331                 print (err)
332
333             elif (sent['type'] == 'editroom') :
334                 try :
335                     cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
336                                         cursor = cnx.cursor()
337                                         #ubah nama room pada tabel rooms
cursor.execute("UPDATE rooms SET name
= %s WHERE name = %s", (sent['name'],sent['oldname']))
338                                         #ubah nama room pada tabel devices
cursor.execute("UPDATE devices SET room
= %s WHERE room = %s", ('none',sent['oldname']))
339                                         for element in sent['devices'] :
340                                         cursor.execute("UPDATE devices SET
room = %s WHERE name = %s", (sent['name'], element[0]))
341                                         cnx.commit()
342                                         cursor.close()
343                                         cnx.close()
344                                         gc.collect()
345             except mysql.connector.Error as err :
346                 print (err)
347
348             elif (sent['type'] == 'addscenario') :
349                 try:
350                     cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
351                                         cursor = cnx.cursor()
352                                         #membuat tabel untuk menyimpan info
perangkat dalam skenario

```

```

357                     cursor.execute("CREATE TABLE " +
sent['name'] + " (name VARCHAR(20), address VARCHAR(20), type
VARCHAR(11), status VARCHAR(9))")
358                         #menambah nama skenario ke tabel
skenario
359                     cursor.execute("INSERT INTO scenarios
(name,time) VALUES (%s,%s)", (sent['name'],sent['time']))
360                         #mengisi tabel baru
361                         for element in sent['devices'] :
362                             cursor.execute("INSERT INTO " +
sent['name'] + "(name,address,type,status) VALUES (%s,%s,%s,%s)",
(element[0],element[1],element[2],element[3]))
363                             cnx.commit()
364                             cursor.close()
365                             cnx.close()
366                             gc.collect()
367                         except mysql.connector.Error as err :
368                             print (err)
369
370                     elif (sent['type'] == 'deletescenario') :
371                         try :
372                             cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
373                             cursor = cnx.cursor()
374                             #menghapus nama skenario dari tabel
scenarios
375                             cursor.execute("DELETE FROM scenarios
WHERE name = %s", (sent['name'],))
376                             #menghapus tabel skenario
377                             cursor.execute("DROP TABLE " +
sent['name'])
378                             cnx.commit()
379                             cursor.close()
380                             cnx.close()
381                             gc.collect()
382                         except mysql.connector.Error as err :
383                             print (err)
384
385                     elif (sent['type'] == 'editscenario') :
386                         try :
387                             cnx =
mysql.connector.connect(user='brb', password='28031995',
host='localhost', database='home'+sent['homeid'])
388                             cursor = cnx.cursor()
389
390                             #delete old
391                             cursor.execute("DELETE FROM scenarios
WHERE name = %s", (sent['oldname'],))
392                             cursor.execute("DROP TABLE " +
sent['oldname'])
393                             cnx.commit()
394
395                             #create new
396                             cursor.execute("CREATE TABLE " +
sent['name'] + " (name VARCHAR(20), address VARCHAR(20), type
VARCHAR(11), status VARCHAR(9))")

```

```

398                     cursor.execute("INSERT INTO scenarios
399             (name,time) VALUES (%s,%s)", (sent['name'],sent['time']))
400             for element in sent['devices'] :
401                 cursor.execute("INSERT INTO " +
402                     sent['name'] + "(name,address,type,status) VALUES (%s,%s,%s,%s)",
403                     (element[0],element[1],element[2],element[3]))
404             cnx.commit()
405
406             cursor.close()
407             cnx.close()
408             gc.collect()
409
410         except mysql.connector.Error as err1 :
411             print (err1)
412
413         elif (sent['head'] == 'UP') :
414             try :
415                 cnx = mysql.connector.connect(user='brb',
416                     password='28031995', host='localhost', database='home'+sent['homeid'])
417                 cursor = cnx.cursor()
418                 #update isi tabel sesuai dengan kiriman
419                 for row in sent['data'] :
420                     cursor.execute("UPDATE " + sent['table']
421                         + " SET " + sent['column'] + " = " + row[1] + " WHERE address = " +
422                         row[0])
423                     cnx.commit()
424                     cursor.close()
425                     cnx.close()
426                     gc.collect()
427             except:
428                 e = sys.exc_info()[0]
429                 print(e)
430
431 ##### AKHIR DARI CALLBACK #####
432
433     #SERVER CONSUME DATA DARI ANTRIAN AESqueue
434     channel_aes.basic_consume(callbackAES,
435     queue='AESqueue',no_ack=True)
436     channel_aes.start_consuming()
437
438 ##### AKHIR DARI server aes.py #####

```

Pada program di atas, terdapat beberapa fungsi yaitu:

Tabel 10 Fungsi pada server_aes.py

Nama Fungsi	Objective Fungsi	Data Return
send_email_invitation(to,code)	Mengirim email berisi kode unik untuk <i>user</i> yang diundang ke suatu rumah	Tidak ada
getkeyfromdatabase (phoneid)	Memperoleh key untuk suatu user dengan phone ID tertentu	<ul style="list-style-type: none"> • Return key
findUserRegister (data, email, password)	Memverifikasi apakah email yang akan diundang ke rumah (<i>register</i>) valid (tidak terdaftar pada rumah lain)	<ul style="list-style-type: none"> • Return '0' bila email valid • Return '1' bila tidak valid

findUserConfirm (data, email, code)	Memeriksa apakah kode yang dimasukan oleh <i>confirm user</i> tepat	<ul style="list-style-type: none"> • Return ID rumah jika benar • Return ‘0’ jika salah
checkAdmin (data, email)	Memeriksa apakah suatu user dengan email input merupakan ADMIN atau Guest	<ul style="list-style-type: none"> • Return ‘0’ bila admin • Return ‘2’ bila guest
RSAencrypt (input)	Melakukan enkripsi RSA terhadap input	Hasil enkripsi
RSAdecrypt (input)	Melakukan dekripsi RSA terhadap input	Hasil dekripsi
AESencrypt (input,key)	Melakukan enkripsi AES terhadap input dengan kunci key	Hasil enkripsi
AESdecrypt (input,key)	Melakukan dekripsi AES terhadap input dengan kunci key	Hasil dekripsi

Program di atas mengimplementasikan skema yang telah dijelaskan pada B300, yaitu sebagai berikut.

Tabel 11 Pesan yang Dilayani server_aes.py

Header	Pengirim	Tipe	Fungsi	Isi Paket Permintaan	Aksi Server
SO (Sign Out)	Ponsel	-	Permintaan Sign Out	<ul style="list-style-type: none"> • Phone ID 	Server menghapus phone ID dari tabel aeskeylist.
MU (Manajemen Users)	Ponsel	List	Permintaan daftar seluruh akun yang terdaftar untuk suatu rumah	<ul style="list-style-type: none"> • Home ID 	Server mengambil daftar seluruh <i>user</i> yang terdaftar di rumah pengirim, lalu dikirimkan sebagai jawaban.
		Register	Pendaftaran atau pengundangan suatu pengguna ke dalam suatu rumah oleh ADMIN rumah	<ul style="list-style-type: none"> • Home ID • Email terundang 	Server mengecek apakah email terundang belum terdaftar di rumah manapun. Bila email terundang belum terdaftar di rumah lain, server menjawab: <ul style="list-style-type: none"> • Enam digit kode yang harus dimasukan di

					ponsel terundang supaya bisa bergabung dengan rumah (digenerasi secara acak oleh server) Bila email terundang sudah terdaftar di rumah lain: <ul style="list-style-type: none">• Jawaban berupa karakter '1'
		Confirm	Konfirmasi dari akun terundang yang didaftarkan melalui 'Register'	<ul style="list-style-type: none">• Enam digit kode konfirmasi	Bila kode benar, server menjawab: <ul style="list-style-type: none">• Home ID Bila kode salah, server menjawab: <ul style="list-style-type: none">• Jawaban berupa karakter '0'
		Editprof	Permintaan edit profile dari suatu akun	<ul style="list-style-type: none">• Email peminta	Server mengubah data profil pada tabel general_data.users
		Deleteuser	Permintaan pengeluaran satu akun dari rumah	<ul style="list-style-type: none">• Email yang hendak dikeluar kan dari rumah	Server mengubah Home ID email tersebut menjadi 'none' dan privilege-nya juga menjadi 'none'.
		Editaeskey	Permintaan mengubah kunci AES karena sudah expired	<ul style="list-style-type: none">• Aes key yang baru	Server mengubah aeskey user bersangkutan pada tabel aeskeylist
MH (Manaja men Home)	Ponsel	edit	Permintaan edit profil rumah	<ul style="list-style-type: none">• Home ID• Nama rumah baru• Koordinat rumah baru	Server mengubah nama dan koordinat rumah di tabel homeXXXXXX.info.
MD (Manajemen Device)	Ponsel	adddevice	Permintaan menambah perangkat baru	<ul style="list-style-type: none">• Home ID• Nama Perangkat	Server menyimpan nama, address, dan tipe perangkat ke database

				<ul style="list-style-type: none"> • Address • Tipe 	homeXXXXXX.devices.
		deletedevice	Permintaan menghapus perangkat	<ul style="list-style-type: none"> • Home ID • Address perangkat yang ingin dihapus 	Server menghapus perangkat dengan address yang sesuai.
		editdevice	Permintaan edit nama perangkat	<ul style="list-style-type: none"> • Home ID • Address perangkat • Nama baru 	Server mengganti nama perangkat yang sesuai menjadi nama yang baru
		addrroom	Permintaan menambah room	<ul style="list-style-type: none"> • Home ID • Nama room • Daftar perangkat yang terkandung pada room itu 	Server menambahkan nama room ke homeXXXXXX.rooms, lalu mengubah kolom room pada homeXXXXXX.devices menjadi nama room tersebut untuk perangkat yang terdapat pada daftar
		deleteroom	Permintaan menghapus room	<ul style="list-style-type: none"> • Home ID • Nama room 	Server menghapus nama room dari homeXXXXXX.rooms dan mengganti isi kolom room dengan nama room bersangkutan menjadi ‘none’
		editroom	Permintaan edit nama atau isi room	<ul style="list-style-type: none"> • Home ID • Nama room baru • Daftar perangkat baru 	Server mengganti nama pada homeXXXXXX.rooms menjadi nama baru. Server juga mengganti kolom room supaya sesuai dengan daftar perangkat yang dikirimkan.
		addscenario	Permintaan menambah skenario	<ul style="list-style-type: none"> • Home ID 	Server menambahkan nama dan waktu skenario ke tabel

				<ul style="list-style-type: none"> • Nama scenario • Waktu scenario • Daftar perangkat dalam skenario tersebut 	homeXXXXXX.scenarios. Server juga membuat tabel bernama sesuai dengan nama skenario yang isinya adalah daftar perangkat skenario tersebut beserta statusnya.
		deletescenario	Permintaan menghapus skenario	<ul style="list-style-type: none"> • Home ID • Nama scenario 	Server menghapus nama scenario dari tabel homeXXXXXX.scenarios, lalu menghapus tabel bernama skenario tersebut dari basis data.
		editscenario	Permintaan edit skenario	<ul style="list-style-type: none"> • Home ID • Nama scenario • Nama baru • Waktu baru • Daftar perangkat baru 	Server mengubah nama dan waktu skenario di tabel scenarios, lalu mengubah isi tabel dengan nama skenario tersebut dengan daftar perangkat yang baru.
UP (Update)	Host	-	<i>Update</i> data analitik	<ul style="list-style-type: none"> • Home ID • Lama penggunaan seluruh perangkat di rumah dalam 1 hari 	Server mengubah isi tabel dataMMYYYY untuk tanggal saat itu dengan data dari host.

5.2.4 Implementasi Program Python pada Host (Raspberry Pi)

Pada implementasi server local di dalam rumah, digunakan Raspberry Pi 3.0 dengan pertimbangan kelengkapan modul *on-board-networking*, kemampuan *multitasking*,

konsumsi daya yang kecil, serta dapat bekerja secara terus menerus. Ketika terpasang dalam sistem rumah, host server Raspberry Pi ini akan menyimpan informasi yang berhubungan dengan seluruh *device* yang terpasang pada sistem rumah serta status *real-time* dari setiap *device* pada sistem. Penyimpanan informasi-informasi tersebut disimpan berupa table pada server MySQL yang terpasang pada Raspberry Pi (Penjelasan basis data telah dijelaskan pada subbab 3.1).

Secara umum, host server Raspberry Pi berfungsi menjalankan 2 buah aksi, yakni menerima pesan kiriman dari ponsel pengguna (dan memberikan balasan jika perlu), serta memberikan perintah kepada perangkat-perangkat di dalam rumah sesuai dengan keadaan yang diinginkan.

Untuk dapat menjalankan aksi yang pertama, Raspberry Pi cukup menghubungkan dirinya ke jaringan RabbitMQ yang ada di server pusat dan ikut melakukan *subscribe* terhadap antrian tertentu. Implementasi programnya sama persis dengan yang ada pada server pusat (telah dijelaskan pada subbagian 3.2.1). Untuk menjalankan aksi yang kedua, Raspberry Pi harus dapat mengendalikan modul XBee Pro dari program Python. Dengan demikian, program yang dijalankan di Raspberry Pi harus mengandung bagian yang dapat mengendalikan modul XBee dan berkomunikasi dengan modul XBee lain yang ada pada *home devices*.

Hal pertama yang perlu dimasukan ke dalam program adalah bagian untuk mengakses port USB pada Raspberry Pi. Modul XBee Pro pada Raspberry Pi terhubung melalui port USB. Dengan demikian, bagian ini menjadi sangat penting. Implementasinya pada program Python adalah sebagai berikut.

```
#KONEKSI KE PORT USB UNTUK MENGAKSES XBEE PRO
while True:
    try :
        XB_Port = serial.Serial("/dev/ttyUSB0", baudrate=9600,
        timeout=3.0)
        print "XBEE is connected to ttyUSB0"
        break
    except :
        XB_Port = serial.Serial("/dev/ttyUSB1", baudrate=9600,
        timeout=3.0)
        print "XBEE is connected to ttyUSB1"
        break
```

Direktori dev/ttyUSB0 merupakan direktori untuk mengakses port USB0, dan dev/ttyUSB1 merupakan direktori untuk mengakses port USB1. Modul XBee hanya dihubungkan pada salah satu dari kedua direktori ini. Baudrate yang digunakan adalah 9600

sesuai dengan pengaturan pada modul XBee. Timeout 3 detik digunakan untuk membatasi batas waktu percobaan koneksi bila seandainya koneksi gagal. Setelah 3 detik berlalu, maka program akan mencoba kembali membangun koneksi ke port yang bersangkutan.

Bagian kedua yang perlu diimplementasikan dalam program adalah definisi kelas *host*. Kelas ini merupakan kelas untuk mendefinisikan bahwa Raspberry Pi ini merupakan *host* sebagai pusat kendali seluruh *device* yang terhubung dalam jaringan Zigbee dalam rumah. Isi kode Python dari program ini adalah sebagai berikut.

```
#KELAS Host
class Host:
    #fungsi inisialisasi
    def __init__(self, address):
        self.address = address
    #fungsi untuk mendapatkan current destination
    def get_destination(self):
        response = "AAA"
        while response != "OK\r":
            XB_Port.write(bytes("+++\r"))
            response = XB_Port.read(3)
        XB_Port.write(bytes("ATDL\r"))
        self.destination = XB_Port.read(9)
        XB_Port.write(bytes("ATCN\r"))
        response = XB_Port.read(3)
    #fungsi untuk mengubah destinasi host
    def change_destination(self, destination):
        while self.destination != destination:
            response = "AAA"

            while response != "OK\r":
                XB_Port.write("+++\r")
                response = XB_Port.read(3)
                # print ("Enter command : %s" %(response))

            response = "AAA"

            while response != "OK\r":

                XB_Port.write(bytes("ATDL%s" %(destination)))
                response = XB_Port.read(3)
                # print ("Change destination
to : %s" %(response))
                XB_Port.write(bytes("ATDL\r"))
                self.destination = XB_Port.read(9)
            response = "AAA"
            while response != "OK\r":
                XB_Port.write(bytes("ATCN\r"))
                response = XB_Port.read(3)
                # print ("Quit Command Mode : %s" %(response))
```

Kelas ini memiliki beberapa fungsi, yaitu:

- `_init_` : fungsi untuk inisialisasi alamat XBee milik *host*

- `get_destination`: fungsi untuk memperoleh *current destination* ke mana *host* sedang terhubung. Perlu diketahui bahwa pada satu saat tertentu *host* hanya dapat terhubung (*paired*) dengan salah satu modul XBee saja di dalam rumah. Fungsi ini dipanggil untuk mengetahui modul XBee mana yang sedang *paired* dengan *host*.
- `change_destination`: fungsi untuk mengganti destinasi XBee pada *host*. Fungsi ini dipanggil ketika modul XBee pada *host* harus di-*pair* dengan modul XBee *node* yang lain yang bukan *current destination*.

Berikutnya, dengan menggunakan fungsi dan kelas yang telah didefinisikan di atas, *host* dapat memberikan perintah kepada suatu *node* tertentu dalam rumah. Isi perintah harus mengimplementasikan desain protokol perintah yang telah dirancang pada B300, yaitu sebagai berikut.

ID Rumah (6 karakter)	Alamat XBee (8 karakter)	Isi Perintah

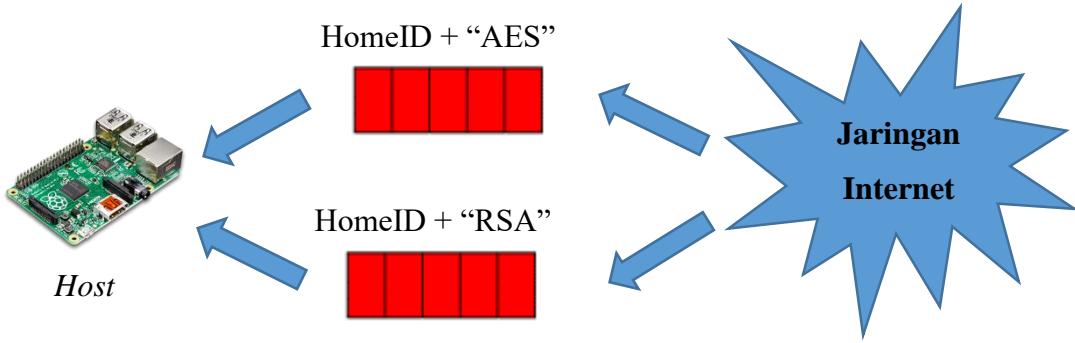
Dengan demikian, kode untuk mengimplementasikan perintah ke *node* adalah sebagai berikut.

```
#sending xbee signal
Master = Host(host_address)
Master.get_destination()
Master.change_destination(node_address)

header_xbee = bytes(homeid + node_address)
XB_Port.write(header_xbee)
XB_Port.write(isi_perintah)
```

Bagian yang diberi warna kuning adalah bagian yang perlu diisi dan disesuaikan dengan keadaan.

Selanjutnya adalah penjelasan keseluruhan program yang dijalankan pada Raspberry Pi. Sama seperti pada server pusat, program pada Raspberry Pi pun terdiri dari dua bagian, yaitu program yang memproses data yang dienkripsi dengan AES dan RSA. Untuk itu, antrian data pun dibagi menjadi dua bagian, yaitu antrian xxxxxAES dan xxxxxRSA, dengan xxxxxx merupakan 6 digit ID rumah. Program yang memproses data dari antrian AES adalah raspi_aes.py, dan program yang memproses data dari antrian RSA adalah raspi_rsa.py.



Tabel 12 Antrian Data ke Host

Kode program raspi_rsa.py adalah sebagai berikut.

```

1 import pika
2 import sys
3 import json
4 import mysql.connector
5 import gc
6 from Crypto.PublicKey import RSA
7 import base64
8
9 #DEKLARASI VARIABEL GLOBAL
10
11 #HOME ID
12 homeid = '123456'
13
14 #KUNCI RSA
15 RSAprivKey =
RSA.importKey(base64.b64decode('MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggs
SjAgEAAoIBAQCQd0CaamKDmTB5ulbre/SLcrl3zE14I804hXpVJNF8SBEXvm0xLcAPGRk1
hJS18dKSx+PFd3nboTUga/WqBzdqlVfgjobdBSrKwf/o2miG2gB7zkK1STpD7CGYBzqTBg
v1A1g78bcjNqAN1oLwC1Dwgno17NocN43k7J3i5uqXW/rkgIB/Rrd2ouBMWe0GsOUJaQgn
d+7bPaF1Aa3di4/xjnAkTdle9zStx7qsx173ziRCrU+WU1zCXGHfUqSzH2AH5BWvcng6
i3Hr8vu2rZxChm/BUkUn6C1a9JHBFxj6wsg+Mxi4zL46RDaHy5kgm+jH48HMmEJ0k/ERTI
od65AgMBAECggEAdm69MJosfxQz8Gc54L1y4ms/7x0z0eEIEBqlCm4vqsJt+NgtLhcCj5
4U05IIVAyWg6yJZxYmeU50fkrjT+uwm5kpPLe7ijA1P0R/DocXEA5iPRz/9/7LXGmVtMug
Um2vmcKPMD1d1Olwv0fXejDcvv9IYhUFTgKhOd5dJW0HQeNzBMXuecYvfecW1c2TgW1FqV
HyGhTEfztCxyafH2UtbtNW3neq60IfB5gnTmB3tuUFhZ8KgmSgVBRBalFB71pu9miJ7js
5GZJYJiQvoReSOgJ16OcyexnubeU0d4YaJhzmtHOZywiNfkA9cHexHmxylqzQs8uXdf5VK
qDGtO+nQKBgQDN28fb7TmIt7yHYNE9hA7CNetiho3UWrusbPkJZgkDYX6zL/IFjUwNURq
aXOGeCEGBbjx9TpIGUAbl1G1fxnZRQzbpp8PKU0HL7+qQrXu15q8+z7GwOHLlov7W1kCFdc
r11o6YHRZxj/nHNH018hPC0k9kt070XSOCouTMPQxouwKBgQCzp1tiaYmq0tfK/sYRk8t9
bKZ5dRNHLJYE1QCdkBJ67B0e/HCck20KWFHgl1sZd3HTEfcWCNzUPXHy98vYuxQf4+uCwp
hrlwTSPShP1MFPCYYe7M363aPRldhu2aN/HXA2H+No9GvaQL5d6MC2ZQPgLo7d4yLSbto
KtLQWMjJGwKBgCVMJr/gEUaF+g1/82KxOmuxxIv6s51GNRueMrupXBFcKeQw6KZ175kWbs
Zd6MJ6TXFr+9WwQc5D/Z7ND6wEpzc9+FsjPF4SC5/JjjQ8WmewrhuxQb5t286tb4K50uNz
XhyAjUFeM8zLizPefLY9T/tCY4QH1oT/wwlN8ssWXIM1AOGBAJ9aHBnvyPfnqxdY4zFmKg
InpIZptV89chfAXtpj17DGzJHWksCfmAGsE3wmZNzy8Niqv8Mgu1fa5L8CAyNL1W7wCM8I
MtIEDPe63Y2xQbbRiW0AFBj018wANbOteY0J2RzkLEfjfjOb7tJ+xu3ZnpaFtSrYQKawEP
Y1veamSfrvAoGAGR1YL3sOj7L32o9ybbupV8q35U463Arc9RhHqA/fn436s0+VPYe+e43t
ifk8P+hQ8GAYJtEh8rVmGGqHiQ9SsXWSmBEPvDBf5mnC47J8x3PUBcU4YXK9HaziOI8mUu
yJxJSV1OgNe1cELCrMw4mmmy35SZRnEM6VjYtyihpabP/Q='))
16 RSApubKey =
RSA.importKey(base64.b64decode('MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIIBCg

```

```

KCAQEAkHdAmmpig5kwebpW63v0i3K5d8xJeCPDuIV6VSTRfEgRF75tMS3ADxkZNYSUtfHS
ksfjxXd526E1IGv1qgc3apVX4I6G3QUqyln/6NpohtoAe85CpUk6Q+whmAc6kwYL9QNYO/
G3IzagDdaC8AtQ8IJ6JezahDeN5Oyd4ubql1v65ICAf0a3dqLgTFntBrDlCWkiJ3fu2z2h
ZQGt3Yo+P8Y5wJE3ZXpPc0rce6rMZe984kQq1P11Ncwlxh31Kksx9gB+QVr3J4Ootx6/L7
tq2cQoZvwVJFJ+gpWvSRwRcY+sLIPjMYuMy+OkQ2h8uZIJvox+PBzJhCdJPxEbSKHeuQID
AQAB"))
17
18     #KONEKSI KE RABBITMQ
19     credentials = pika.PlainCredentials('mindsTA', 'minds')
20     parameters =
pika.ConnectionParameters('167.205.24.10', 5672, 'mindsTA132', credentials)
21     connection = pika.BlockingConnection(parameters)
22     channel = connection.channel()
23
24     #DEKLARASI ANTRIAN
25     queue_name = homeid+"RSA"
26     channel.queue_declare(queue=queue_name)
27
28     #FUNGSI UNTUK ENCRYPT RSA
29     def RSAencrypt(input_stream) :
30         global RSApubKey
31         return
base64.b64encode(RSApubKey.encrypt(input_stream,'x')[0])
32
33     #FUNGSI UNTUK DECRYPT RSA
34     def RSAdecrypt (input_stream):
35         global RSAprivKey
36         return
RSAprivKey.decrypt(base64.b64decode(input_stream))
37
38     print(' [*] Waiting for logs. To exit press CTRL+C')
39
40     #FUNGSI YANG AKAN DIJALANKAN KETIKA ADA DATA MASUK KE ANTRIAN
41     def callback(ch, method, props, body):
42         print("[x] Received: " + RSAdecrypt(body))
43         try :
44             #memecah data JSON dari hasil dekripsi
45             sent = json.loads(RSAdecrypt(body))
46             #hanya ada 1 header yang ditangani pada program ini
47             if (sent['head'] == 'SI') :
48                 #terhubung ke database
49                 cnx = mysql.connector.connect(user='root',
password='28031995', host='localhost', database='home'+homeid)
50                 cursor = cnx.cursor()
51                 #memeriksa apakah user pernah terdaftar
52                 #props.correlation_id berisi phone id user
53                 cursor.execute("SELECT EXISTS (SELECT phoneid
FROM users WHERE phoneid = %s)", (props.correlation_id,))
54                 if (cursor.fetchone()[0]):                      #jika
pernah terdaftar
55                     #cukup update isi aeskey
56                     cursor.execute("UPDATE users SET
aeskey = %s WHERE phoneid = %s", (sent['aeskey'],props.correlation_id))
57                     else :                         #jika belum terdaftar
58                         #tambahkan baris baru
59                         cursor.execute("INSERT INTO users
VALUES

```

```

(%s,%s,%s,%s)",(props.correlation_id,sent['email'],sent['privilege'],sent['aeskey']))
60                      cnx.commit()
61                      cursor.close()
62                      cnx.close()
63                      gc.collect()
64      except:
65          print(sys.exc_info()[0])
66
67
68
69      channel.basic_consume(callback,
70                             queue=queue_name, no_ack=True)
71
72      channel.start_consuming()

```

Pada raspi_rsa.py, terdapat beberapa fungsi berikut.

Tabel 13 Fungsi pada raspi_rsa.py

Nama Fungsi	Objective Fungsi	Data Return
RSAencrypt (input)	Melakukan enkripsi RSA terhadap input	Hasil enkripsi
RSAdecrypt (input)	Melakukan dekripsi RSA terhadap input	Hasil dekripsi

Kode di atas terdiri dari beberapa bagian penting, yaitu:

- Baris 9 sampai 16 merupakan deklarasi variabel global, yakni ID rumah (*home ID*) dan kunci RSA.
- Baris 18 sampai 22 merupakan bagian untuk membangun koneksi ke RabbitMQ di server. IP, port, dan *credential* dimasukan sesuai dengan yang ditetapkan.
- Baris 24 sampai 26 merupakan deklarasi antrian. Karena ini adalah program raspi_rsa.py, maka antrian yang harus dideklarasikan adalah ID rumah + “RSA”.
- Baris 28 sampai 36 adalah definisi fungsi (telah dijelaskan pada tabel di atas).
- Baris 41 sampai 65 berisi definisi fungsi *callback* yang akan dijalankan setiap kali ada data masuk ke antrian. Respon program terhadap data yang masuk bergantung pada isi data. Data pertama kali didekripsi terlebih dahulu menggunakan RSA, lalu data hasil dekripsi seharusnya berupa data JSON. Data JSON tersebut dipecah (baris 45) dan disimpan pada variabel sent. Pada program ini, hanya terdapat 1 jenis header yang perlu ditangani, yaitu SI. Bagian ini merupakan implementasi dari pelayanan *host* terhadap data berikut.

Tabel 14 Pesan yang Dilayani raspi_rsa.py

Header	Pengirim	Tipe	Fungsi	Isi Paket Permintaan	Aksi Server
SI	Ponsel	-	Permintaan sign in dan pengubahan kunci enkripsi	<ul style="list-style-type: none"> • Email pengirim • Privilege pengirim • Kunci enkripsi baru 	Host menyimpan kunci enkripsi yang baru untuk user yang bersangkutan. Tidak ada balasan ke ponsel.

- Baris 69 sampai 72 adalah perintah untuk mulai melaksanakan *data consuming* terhadap antrian yang telah dideklarasikan.

Program berikutnya adalah program raspi_aes.py. Karena program ini sangat panjang, maka pembahasan program ini akan dibagi menjadi beberapa bagian. Berikut adalah bagian pertama kode program tersebut.

```

1 import pika
2 import sys
3 import json
4 import mysql.connector
5 import gc
6 import os
7 from subprocess import Popen, PIPE, check_call
8 import time
9 from datetime import datetime, timedelta
10 from Crypto.Cipher import AES
11 from Crypto.PublicKey import RSA
12 from xbee import XBee, ZigBee
13 import usb
14 import serial
15 import base64
16 import unicodedata
17 import math
18
19 #DEKLARASI VARIABEL GLOBAL
20 homeid = '123456'      #ID RUMAH
21 iv = 'RandomInitVector' #INITIAL VECTOR UNTUK AES
22
23 #KUNCI RSA
24 RSAprivKey =
RSA.importKey(base64.b64decode('MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwg
SjAgEAAoIBAQCQd0CaamKDmTB5ulbre/SLcrl3zEl4I804hXpVJNF8SBEXvm0xLcAPGRk1
hJS18dKSx+PFd3nboTUga/WqBzdqlVfgjobdBSrKwf/o2miG2gB7zkK1STpD7CGYBzqTBg
v1A1g78bcjNqAN1oLwC1Dwgno17NocN43k7J3i5uqXW/rkgIB/Rrd2ouBMWe0GsOUJaQgn
d+7bPaF1Aa3di4/xjnAkTdle9zStx7qsx173ziRCrU+WU1zCXGHfUqSzH2AH5BWvcng6
i3Hr8vu2rZxChm/BUkUn6C1a9JHBFxj6wsg+Mxi4zL46RDaHy5kgm+jH48HMmEJ0k/ERTI
od65AgMBAAECggEAdm69MJosfxQz8Gc54L1y4ms/7x0z0eEIEBqlCm4vqsJt+NgtLhcCj5
4U05IIIVAyWg6yJZxYmeU50fkrjT+uwm5kpPLe7ijA1P0R/DocXEA5iPRz/9/7lXGmVtMug
Um2vmcKPMD1d10lw0fxejDcvv9IYhUFTgKhOd5dJWoHQeNzBMXuecYvfecW1c2TgW1FqV
HyGhTEfztCxyafH2UtbtNW3neq601fB5gnTmB3tuUFhZ8KgmSgVBRBalFB71pu9miJ7sjs
5GZJYJiQvoReSOgJ16OcyexnubeU0d4YaJhzmtHOZywiNfkA9cHexHmxylqzQs8uXdF5VK
qDGtO+nQKBgQDN28fb7TmIt7yHYNE9hA7CNetih03UWrusbPkJZgkDYX6zL/IFjUwNURq
aXOGeCEGBjx9TpIGUAb11G1fxnZRQzbpp8PKU0HL7+qOrXu15q8+z7GwOHLov7W1kCFdc

```

```

r11o6YHRZxj/nHNH018hPC0k9kT070XSOC0UTMPQxouwKBgQCzp1tiaYmq0tfK/sYRk8t9
bKZ5dRNHLJYE1QCDkBJ67B0e/HCck20KWFHg1sZd3HTEfcWCNzUPXHy98vYuxQf4+uCWP
hrlwTSPShP1MFCPCYYe7M363aPRldhU2aN/HXA2H+No9GvaQL5d6MC2ZQPgLo7d4yLSbto
KtLQWMjJGwKBgCVMJr/gEUaF+g1/82KxOmuxxIv6s51GNRueMrupXBFCKeQw6KZ175kWbs
Zd6MJ6TXFr+9WwQc5D/Z7ND6wEpzc9+FsjPF4SC5/JjjQ8WmewrhuxQb5t286tb4K50uNz
XhyAjUFeM8zLizPefLY9T/tCY4QH1oT/wwlN8ssWXIM1AoGBAJ9ahBnvyPfnqxdY4zFmKg
InpIZpTV89chfAXtpj17DGzJHWksCfmAGsE3wmZNzy8Niqv8Mgu1fa5L8CAyNL1W7wCM8I
MtiedPe63Y2xQbbRiW0AFbj018wANbOTeY0J2RzkLEfjfjOb7tJ+xu3ZNPafSrYQKawEP
YlveamSfrvAoGAGR1YL3sOj7L32o9ybbupV8q35U463Arc9RhHqA/fn436s0+VPYe+e43t
ifk8P+hQ8GAYJtEh8rVmGGqHiQ9SsXWSmBEpvDBf5mnC47J8x3PUBcU4YXK9HaziOI8mUu
yJxJSV1OgNe1cELCrMw4mmmy35SRnEM6VjYytihpabP/Q='))

25     RSApubKey =
RSA.importKey(base64.b64decode('MIIBIjANBgkqhkiG9w0BAQEFAOCAQ8AMIBCg
KCAQEAkHdAmmpig5kwebpW63v0i3K5d8xJeCPDuIV6VSTRfEgRF75tMS3ADxkZNYsUtfHS
ksfjxXd526E1IGv1qgc3apVX4I6G3QUqyln/6NpohtoAe85CpUk6Q+whmAc6kwYL9QNYO/
G3IzagDdaC8AtQ8IJ6JezaHDeN50yd4ubql1v65ICAf0a3dqLgTFntBrD1CWkIJ3fu2z2h
ZQGt3Yo+P8Y5wJE3ZXpPc0rce6rMZe984kQq1P11Ncwlxh31Kksx9gB+QVr3J4ootx6/L7
tq2cQoZvwVJFJ+gpWvSRwRcY+sLIPjMYuMy+OkQ2h8uZIJvox+PBzJhCdJPxEbSKHeuQID
AQAB'))

26
27     #KONEKSI KE RABBITMQ
28     credentials = pika.PlainCredentials('mindsTA', 'minds')
29     parameters =
pika.ConnectionParameters('167.205.24.10', 5672, 'mindsTA132', credentials)
30     connection = pika.BlockingConnection(parameters)
31     channel = connection.channel()

32
33     #KONEKSI KE PORT USB UNTUK MENGAKSES XBEE
34     while True:
35         try :
36             baudrate=9600, timeout=3.0
37             XB_Port = serial.Serial("/dev/ttyUSB0",
38             print "XBEE is connected to ttyUSB0"
39             break
39         except :
40             baudrate=9600, timeout=3.0
41             XB_Port = serial.Serial("/dev/ttyUSB1",
42             print "XBEE is connected to ttyUSB1"
42             break
43
44     #FUNGSI ENKRIPSI AES
45     def AESencrypt(input_stream, key):
46         global iv
47         cipher = AES.new(key, AES.MODE_CFB, iv)
48         return base64.b64encode(iv + cipher.encrypt(input_stream))
49
50     #FUNGSI DEKRIPSI AES
51     def AESdecrypt(input_stream, key) :
52         global iv
53         cipher = AES.new(key, AES.MODE_CFB, iv)
54         return
(cipher.decrypt(base64.b64decode(input_stream))[16:])
55
56     #FUNGSI UNTUK MENDAPATKAN KUNCI AES DARI DATABASE
57     def getkeyfromdatabase (phoneid):
58         cnx = mysql.connector.connect(user='root', password=
'28031995', host = 'localhost', database='home'+homeid)

```

```

59             cursor = cnx.cursor()
60             cursor.execute("SELECT aeskey FROM users WHERE
phoneid = %s", (phoneid,))
61             return (cursor.fetchone()[0])
62
63     #FUNGSI UNTUK MENGHAPUS SUATU ELEMEN DI DALAM TIPE DATA
DICTIONARY (AKAN DIGUNAKAN PADA DELETE ATAU EDIT SCENARIO)
64     def removekey(d, key):
65         r = dict(d)
66         del r[key]
67         return r
68
69     #DEFINISI KELAS Host
70     class Host:
71         #fungsi inisialisasi
72         def __init__(self, address):
73             self.address = address
74         #fungsi untuk mendapatkan current destination XBEE pada
host
75         def get_destination(self):
76             response = "AAA"
77             while response != "OK\r":
78                 XB_Port.write(bytes("+++\r"))
79                 response = XB_Port.read(3)
80             XB_Port.write(bytes("ATDL\r"))
81             self.destination = XB_Port.read(9)
82             XB_Port.write(bytes("ATCN\r"))
83             response = XB_Port.read(3)
84
85         #fungsi untuk mengganti destinasi XBEE host
86         def change_destination(self, destination):
87             while self.destination != destination:
88                 response = "AAA"
89
90                 while response != "OK\r":
91                     XB_Port.write("+++\r")
92                     response = XB_Port.read(3)
93                     #      print ("Enter
command : %s" %(response))
94
95                     response = "AAA"
96
97                 while response != "OK\r":
98
99                     XB_Port.write(bytes("ATDL%s" %(destination)))
100                     response = XB_Port.read(3)
101                     #      print ("Change destination
to : %s" %(response))
102                     XB_Port.write(bytes("ATDL\r"))
103                     self.destination = XB_Port.read(9)
104                     response = "AAA"
105                     while response != "OK\r":
106                         XB_Port.write(bytes("ATCN\r"))
107                         response = XB_Port.read(3)
108                         #      print ("Quit Command
Mode : %s" %(response))
109
#DEKLARASI VARIABEL HOST BESERTA ALAMAT XBEE-NYA

```

```

110 Master = Host("40A88C29\r")
111
112 #DEKLARASI ANTRIAN AES
113 queue_name = homeid+"AES"
114 channel.queue_declare(queue=queue_name)
115
116 #DEKLARASI VARIABEL DICTIONARY (UNTUK SCENARIO)
117 bank = {}
118
119 print(' [*] Waiting for logs. To exit press CTRL+C')

```

Bagian pertama ini berisi deklarasi variabel global, koneksi ke XBee melalui USB, koneksi ke RabbitMQ, definisi kelas Host, dan definisi fungsi. Koneksi ke RabbitMQ, USB, dan definisi kelas host telah dijelaskan pada subbagian 3.2.1. Pada raspi_aes.py, terdapat beberapa fungsi yang dijelaskan sebagai berikut.

Tabel 15 Fungsi pada raspi_aes.py

Nama Fungsi	Objective Fungsi	Data Return
getkeyfromdat abase (phoneid)	Memperoleh key untuk suatu user dengan phone ID tertentu	• Return key
AESencrypt (input,key)	Melakukan enkripsi AES terhadap input dengan kunci key	Hasil enkripsi
AESdecrypt (input,key)	Melakukan dekripsi AES terhadap input dengan kunci key	Hasil dekripsi
Removekey (d, key)	Menghapus suatu elemen di dalam data bertipe <i>dictionary</i> (fungsi ini dipanggil dalam proses edit scenario)	<i>Dictionary</i> yang telah dikurangi elemennya

Bagian berikutnya adalah sebagai berikut.

```

121 #FUNGSI CALLBACK YANG AKAN DIJALANKAN KETIKA ADA DATA MASUK KE
122 #ANTRIAN AES
123 def callback(ch, method, props, body):
124
125     #DEKLARASI VARIABEL GLOBAL
126     global bank
127     global Master
128     global XB_Port
129
130     try :
131         #memperoleh kunci aes berdasarkan phone id
132         #props.correlation_id berisi phone id
133         key = getkeyfromdatabase(props.correlation_id)
134         print ("[x] received: " + AESdecrypt(body,key)+"\n")
135         #memecah data JSON
136         sent = json.loads(AESdecrypt(body,key))
137
138         #SIGN OUT
139         if (sent['homeid'] == homeid and sent['head'] ==
'SO') :

```

```

140                                     cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
141                                         cursor = cnx.cursor()
142                                         #hapus user dari tabel users
143                                         cursor.execute("DELETE FROM users
WHERE phoneid = %s", (props.correlation_id,))
144                                         cnx.commit()
145                                         cursor.close()
146                                         cnx.close()
147                                         gc.collect()
148                                         except mysql.connector.Error as err :
149                                         print (err)
150
151                                         # LOAD DATA
152                                         elif (sent['homeid'] == homeid and sent['head'] ==
'LO') :
153                                         try :
154                                         #akses ke databasae
155                                         cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
156                                         cursor = cnx.cursor()
157                                         #memeriksa apakah user telah dihapus
dari rumah
158                                         cursor.execute("SELECT email FROM users
WHERE phoneid = %s", (props.correlation_id,))
159                                         deletedemail = cursor.fetchone()[0]
160
161                                         #jika user telah didelete oleh ADMIN
tuan rumah
162                                         if (deletedemail[0:8] == 'todelete') :
163                                         #hapus dari database
164                                         cursor.execute("DELETE FROM users
WHERE email = %s", (deletedemail,))
165                                         cnx.commit()
166                                         #kirim balasan 'deleted'
167                                         ch.basic_publish(exchange='',
168                                         routing_key=props.reply_to,
169                                         properties=pika.BasicProperties(correlation_id =
170                                         props.correlation_id,
171                                         body=AESencrypt('deleted',key))
172                                         #NOT DELETED :
173                                         else :
174                                         #jika key sudah kadaluarsa, update
key
175                                         if (sent['type'] == 'updatekey') :
176                                         cursor.execute("UPDATE users
SET aeskey = %s WHERE phoneid
= %s", (sent['aeskey'],props.correlation_id))
177                                         cnx.commit()
178
179                                         #membentuk pesan balasan
180                                         message = {}

```

```

181                         message['head'] = 'LO'
182
183                         #membentuk isi devices dan
184                         scenario berdasarkan privilege peminta data
185                         if (sent['privilege'] ==
186                             'Guest') :
187                             cursor.execute("SELECT *"
188                             FROM devices WHERE privilege = 'Guest'")
189                         cursor.fetchall()
190
191                         cursor.execute("SELECT *"
192                             FROM scenarios WHERE privilege = 'Guest'")
193                         cursor.fetchall()
194
195                         #membentuk isi room
196                         cursor.execute("SELECT * FROM
197                             rooms")
198                         cursor.fetchall()
199
200                         #membentuk isi notifikasi
201                         cursor.execute("SELECT * FROM
202                             notifications")
203                         cursor.fetchall()
204
205                         #membentuk isi lockstatus dan
206                         homename
207                         lockstatus FROM info")
208
209                         cursor.execute("SELECT name,
210                             lockstatus")
211                         data = cursor.fetchone()
212                         message['homename'] = data[0]
213                         message['lockstatus'] = data[1]
214
215                         #mengirim balasan
216                         ch.basic_publish(exchange='',
217                             routing_key=props.reply_to,
218                             properties=pika.BasicProperties(correlation_id =
219                                 props.correlation_id),
220                                 body=AESencrypt((json.dumps(message)),key))
221                         cursor.close()
222                         cnx.close()
223                         gc.collect()
224
225                         except mysql.connector.Error as err :

```

219

print (err)

Bagian ini memulai definisi fungsi *callback*, yaitu fungsi yang akan dipanggil ketika ada data masuk ke antrian. Potongan di atas melayani pesan dengan header ‘SO’ dan ‘LO’.

Tabel 16 Pesan yang Dilayani raspi_aes.py Bagian 1

Header	Pengirim	Tipe	Fungsi	Isi Paket Permintaan	Aksi Server
SO	Ponsel	-	Permintaan sign out	-	Host menghapus user dengan phone ID pada <i>correlation ID</i> dari tabel homeXXXXXX.users
LO (Manajamen Home)	Ponsel	-	Permintaan data perangkat, room, dan scenario	• Home ID	Host mengakses tabel homeXXXXXX.info, kemudian mengambil nama dan koordinat rumah, lalu dikirimkan sebagai jawaban.

Potongan berikutnya adalah sebagai berikut.

```

222             # MANAJEMEN USERS
223         elif (sent['homeid'] == homeid and sent['head'] ==
224             'MU') :
224                 if (sent['type'] == 'adduser') :
225                     try:
226                         cnx =
226                         mysql.connector.connect(user='root', password='28031995',
227                         host='localhost', database='home'+homeid)
227                         cursor = cnx.cursor()
228                         #membentuk pesan broadcast
229                         message = {}
230                         message['head'] = 'MU'
231                         message['notifsender'] =
231                         sent['notifsender']
232                         message['notifmessage'] =
232                         sent['notifmessage']
233                         cursor.execute("SELECT * FROM
233                         notifications")
234                         message['notifications'] =
234                         cursor.fetchall()
235                         #broadcast the message
235                         cursor.execute("SELECT phoneid,
236                         aeskey, privilege FROM users")
236                         data = cursor.fetchall()
236                         for row in data :
237                             ch.basic_publish(exchange=' ',
238                             routing_key=row[0],
239

```

```

241     body=AESencrypt(json.dumps(message),row[1]))
242             cursor.close()
243             cnx.close()
244             gc.collect()
245         except mysql.connector.Error as err :
246             print (err)
247         if (sent['type'] == 'deleteuser') :
248             try :
249                 cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
250                     cursor = cnx.cursor()
251                     # form the message
252                     message = {}
253                     message['head'] = 'MU'
254                     message['notifsender'] =
sent['notifsender']
255                     message['notifmessage'] =
256                     cursor.execute("SELECT * FROM
notifications")
257                     message['notifications'] =
cursor.fetchall()
258                     # broadcast the message
259                     cursor.execute("SELECT phoneid,
aeskey, privilege FROM users")
260                     data = cursor.fetchall()
261                     for row in data :
262
ch.basic_publish(exchange='',
263
routing_key=row[0],
264
body=AESencrypt(json.dumps(message),row[1]))
265             # change data in database
266             cursor.execute("UPDATE users SET
email = %s WHERE email = %s",('todelete'+sent['email'],sent['email']))
267             cnx.commit()
268             cursor.close()
269             cnx.close()
270             gc.collect()
271         except mysql.connector.Error as err :
272             print (err)
273
# MANAJEMEN HOME
274         elif (sent['homeid'] == homeid and sent['head'] ==
'MH') :
275             if (sent['type'] == 'list') :
276                 try :
277                     cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
278                     cursor = cnx.cursor()
279                     cursor.execute("SELECT *
FROM info")
280                     data = cursor.fetchone()
281
282

```

```

283                                     message = {}
284                                     message['name'] = data[0]
285                                     message['lat'] =
286                                     message['lng'] =
287                                     print (message['name'],
288                                     data[1], data[2])
289                                     ch.basic_publish(exchange='',
290                                     routing_key=props.reply_to,
291                                     properties=pika.BasicProperties(correlation_id = \
292                                     props.correlation_id),
293                                     body=AESencrypt(json.dumps(message),key))
294                                     cursor.close()
295                                     cnx.close()
296                                     gc.collect()
297                                     except mysql.connector.Error as
err :
298                                         print (err)
299                                         elif (sent['type'] == 'edit') :
300                                             try :
301                                                 cnx =
302                                                 mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
303                                                 cursor = cnx.cursor()
304                                                 cursor.execute("UPDATE info
305                                                 SET name = %s, latitude=%s,
longitude=%s", (sent['name'],sent['lat'],sent['lng']))
306                                                 cnx.commit()
307                                                 cursor.close()
308                                                 cnx.close()
309                                                 gc.collect()
310                                         except mysql.connector.Error as
err :
311                                         print (err)

```

Potongan ini melayani pesan dengan *header* MU dan MH. Bentuk pelayanan *host* adalah sebagai berikut.

Tabel 17 Pesan yang Dilayani raspi_aes.py Bagian 2

MH (Manajamen Home)	Ponsel	list	Permintaan data identitas rumah	•	Home ID	Host mengakses tabel homeXXXXXX.info, kemudian mengambil nama dan koordinat rumah, lalu
--------------------------------------	---------------	-------------	--	----------	----------------	--

dikirimkan sebagai jawaban.					
		edit	Permintaan edit profil rumah	<ul style="list-style-type: none"> • Home ID • Nama rumah baru • Koordinat rumah baru 	Host mengubah nama dan koordinat rumah di tabel homeXXXXXX.info.
MU (Manajemen User)	Server	adduser	Permintaan penambahan anggota rumah	<ul style="list-style-type: none"> • Home ID • Email pengguna • Privilege 	Host menyimpan data pengguna baru di homeXXXXXX.users .
	Ponsel	deleteuser	Permintaan penghapusan anggota rumah	<ul style="list-style-type: none"> • Home ID • Email pengguna 	Host menghapus user tersebut dari basis data.

Potongan berikutnya adalah sebagai berikut.

```
312 # MANAJEMEN DEVICES

313         elif (sent['homeid'] == homeid and sent['head'] ==
314             'MD') :
315             if (sent['type'] == 'adddevice') :
316                 try:
317                     cnx =
318                     mysql.connector.connect(user='root', password='28031995',
319                     host='localhost', database='home'+homeid)
320                     cursor = cnx.cursor()
321                     if (sent['type'] ==
322                         'lamp') :
323                         cursor.execute("INSERT INTO devices VALUES (%s,%s,%s,%s,%s,%s)",
324                             (sent['name'],sent['address'],sent['devtype'],'ff000000','none','ADMIN'))
325                     else :
326                         cursor.execute("INSERT INTO devices VALUES (%s,%s,%s,%s,%s,%s)",
327                             (sent['name'],sent['address'],sent['devtype'],'0','none','ADMIN'))
328                     cursor.execute("INSERT INTO data" + datetime.now().strftime('%m%Y') + " VALUES ('" +
329                     sent['address'] +
330                     "', 'none', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0',
331                     '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0', '0')")+
332                     cnx.commit()
333                     # form the message
334                     message = {}
335                     message['head'] = 'RE'
336                     message['type'] = 'devices'
337                     message['notifsender'] =
338                     sent['notifsender']
```

```

329                         message['notifmessage'] =
sent['notifmessage']
330                         cursor.execute("SELECT *
FROM notifications")
331                         cursor.fetchall()
332                         # broadcast the message
333                         message['notifications'] =
334                         cursor.execute("SELECT
phoneid, aeskey, privilege FROM users")
335                         data = cursor.fetchall()
336                         for row in data :
337                             if (row[2] ==
'Guest') :
338                                 Select only Guests' Devices
339                                 cursor.execute("SELECT * FROM devices WHERE privilege =
'Guest' ")
340                                 if (row[2] ==
'ADMIN') :
341                                     Select all devices
342                                     cursor.execute("SELECT * FROM devices")
343                                     message['devices'] = cursor.fetchall()
344                                     ch.basic_publish(exchange=' ',
345                                     routing_key=row[0],
346                                     body=AESEncrypt(json.dumps(message),row[1]))
347                                     cursor.close()
348                                     cnx.close()
349                                     gc.collect()
350
351
352                         except mysql.connector.Error as err :
353                             print (err)
354
355                         elif (sent['type'] == 'deletedevice') :
356                             try :
357                                 cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
358                                 cursor = cnx.cursor()
359                                 cursor.execute("DELETE
FROM devices WHERE address = %s", (sent['address'],))
360                                 cnx.commit() # form the
message
361                                 message = {}
362                                 message['head'] = 'RE'
363                                 message['type'] =
364                                 message['notifsender'] =
sent['notifsender']

```

```

365     message['notifmessage'] = sent['notifmessage']
366                                         cursor.execute("SELECT
367 * FROM notifications")
368     message['notifications'] = cursor.fetchall()
369                                         # broadcast the
370 message
371                                         cursor.execute("SELECT
372 phoneid, aeskey, privilege FROM users")
373                                         data =
374 cursor.fetchall()
375                                         for row in data :
376                                         if (row[2]
377 == 'Guest') :
378     # Select only Guests' Devices
379     cursor.execute("SELECT * FROM devices WHERE privilege =
380 'Guest'")
381                                         if (row[2]
382 == 'ADMIN') :
383     # Select all devices
384     cursor.execute("SELECT * FROM devices")
385     message['devices'] = cursor.fetchall()
386     ch.basic_publish(exchange='',
387 routing_key=row[0],
388 body=AESencrypt(json.dumps(message),row[1]))
389                                         cursor.close()
390                                         cnx.close()
391                                         gc.collect()
392
393                                         except mysql.connector.Error as
394 err :
395                                         print (err)
396
397                                         elif (sent['type'] == 'editdevice') :
398                                         try :
399                                         cnx =
400 mysql.connector.connect(user='root', password='28031995',
401 host='localhost', database='home'+homeid)
402                                         cursor = cnx.cursor()
403                                         cursor.execute("UPDATE
404 devices SET name = %s WHERE address = %s",
405 (sent['name'],sent['address']))
406                                         cnx.commit()
407                                         # form the message
408                                         message = {}
409                                         message['head'] = 'RE'
410                                         message['type'] =
411 'devices'

```

```

400                                         message['notifsender']
401     = sent['notifsender']
402
403         message['notifmessage'] = sent['notifmessage']
404                                         cursor.execute("SELECT
405 * FROM notifications")
406
407         message['notifications'] = cursor.fetchall()
408                                         # broadcast the
409 message
410                                         cursor.execute("SELECT
411 phoneid, aeskey, privilege FROM users")
412
413         cursor.fetchall()
414
415 == 'Guest' ) :
416
417     # Select only Guests' Devices
418
419     cursor.execute("SELECT * FROM devices WHERE privilege =
420 'Guest'")
421                                         if (row[2]
422 == 'ADMIN') :
423
424     # Select all devices
425
426     cursor.execute("SELECT * FROM devices")
427
428     message['devices'] = cursor.fetchall()
429
430     ch.basic_publish(exchange=' ',
431
432     routing_key=row[0],
433
434     body=AESencrypt(json.dumps(message),row[1]))
435
436                                         cursor.close()
437                                         cnx.close()
438                                         gc.collect()
439
440                                         except mysql.connector.Error as
441 err :
442
443                                         print (err)
444
445                                         elif (sent['type'] == 'addrroom') :
446                                         try :
447
448                                         cnx =
449 mysql.connector.connect(user='root', password='28031995',
450 host='localhost', database='home'+homeid)
451                                         cursor = cnx.cursor()
452                                         cursor.execute("INSERT
453 INTO rooms (name) VALUES (%s)", (sent['name'],))
454                                         for element in
455 sent['devices'] :
456
457                                         cursor.execute("UPDATE devices SET room =
458 %s WHERE name = %s",
459 (sent['name'], element[0]))

```

```

433                                     cnx.commit()
434
435                                         # form the message
436                                         message = {}
437                                         message['head'] = 'RE'
438                                         message['type'] =
439                                         'rooms'
440                                         cursor.execute("SELECT
* FROM rooms")
440                                         message['rooms'] =
441                                         cursor.fetchall()
441                                         message['notifsender'] =
442                                         sent['notifsender']
442                                         message['notifmessage'] = sent['notifmessage']
443                                         cursor.execute("SELECT
* FROM notifications")
444                                         message['notifications'] = cursor.fetchall()
445                                         # broadcast the
445                                         message
446                                         cursor.execute("SELECT
phoneid,aeskey,privilege FROM users")
447                                         data =
448                                         cursor.fetchall()
449                                         for row in data :
450                                         if (row[2]
450 == 'Guest') :
450                                         # Select only Guests' Devices
451                                         cursor.execute("SELECT * FROM devices WHERE privilege =
451 'Guest'")
452                                         if (row[2]
452 == 'ADMIN') :
453                                         # Select all devices
454                                         cursor.execute("SELECT * FROM devices")
455                                         message['devices'] = cursor.fetchall()
456                                         ch.basic_publish(exchange=' ',
457                                         routing_key=row[0],
458                                         body=AESencrypt(json.dumps(message),row[1]))
459                                         print
459 ("[x] Sent to " + row[0])
460                                         cursor.close()
461                                         cnx.close()
462                                         gc.collect()
463
464                                         except mysql.connector.Error as
465                                         err :
466                                         print (err)
467

```

```

468                 elif (sent['type'] == 'deleteroom') :
469                     try :
470                         cnx =
471                         mysql.connector.connect(user='root', password='28031995',
472                         host='localhost', database='home'+homeid)
473                         cursor = cnx.cursor()
474                         cursor.execute("DELETE
475                         devices SET room = %s WHERE room = %s", ('none', sent['name']))
476                         cursor.execute("UPDATE
477                         rooms SET name = %s WHERE name = %s", (sent['name'], sent['name']))
478                         cnx.commit()
479                         # form the message
480                         message = {}
481                         message['head'] = 'RE'
482                         message['type'] =
483                         'rooms'
484                         cursor.execute("SELECT
485                         * FROM rooms")
486                         cursor.fetchall()
487                         message['notifsender'] =
488                         sent['notifsender']
489                         message['notifmessage'] = sent['notifmessage']
490                         cursor.execute("SELECT
491                         * FROM notifications")
492                         cursor.fetchall()
493                         message['notifications'] = cursor.fetchall()
494                         # broadcast the
495                         message
496                         cursor.execute("SELECT
497                         phoneid, aeskey, privilege FROM users")
498                         cursor.fetchall()
499                         data =
500                         for row in data :
501                             if (row[2]
502                             == 'Guest') :
503                                 # Select only Guests' Devices
504                                 cursor.execute("SELECT * FROM devices WHERE privilege =
505 'Guest'")
506                                 if (row[2]
507                                 == 'ADMIN') :
508                                     # Select all devices
509                                     cursor.execute("SELECT * FROM devices")
510                                     message['devices'] = cursor.fetchall()
511                                     ch.basic_publish(exchange=' ',
512                                     routing_key=row[0],
513                                     body=AESencrypt(json.dumps(message), row[1]))
514                                     print(row[0], row[1], row[2], json.dumps(message))

```

```

500                                         cursor.close()
501                                         cnx.close()
502                                         gc.collect()
503
504                                         except mysql.connector.Error as
err :
505                                         print (err)
506
507                                         elif (sent['type'] == 'editroom') :
508                                         try :
509                                         cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
510                                         cursor = cnx.cursor()
511                                         cursor.execute("UPDATE
rooms SET name = %s WHERE name = %s", (sent['name'],sent['oldname']))
512                                         cursor.execute("UPDATE
devices SET room = %s WHERE room = %s", ('none',sent['oldname']))
513                                         for element in
sent['devices'] :
514                                         cursor.execute("UPDATE devices SET room = %s WHERE name = %s",
(sent['name'], element[0]))
515                                         cnx.commit()
516                                         # form the message
517                                         message = {}
518                                         message['head'] = 'RE'
519                                         message['type'] =
'rooms'
520                                         cursor.execute("SELECT
* FROM rooms")
521                                         message['rooms'] =
cursor.fetchall()
522                                         message['notifsender'] =
sent['notifsender']
523                                         message['notifmessage'] = sent['notifmessage']
524                                         cursor.execute("SELECT
* FROM notifications")
525                                         message['notifications'] = cursor.fetchall()
526                                         # broadcast the
message
527                                         cursor.execute("SELECT
phoneid,aeskey,privilege FROM users")
528                                         data =
cursor.fetchall()
529                                         for row in data :
530                                         if (row[2]
== 'Guest') :
531                                         # Select only Guests' Devices
532                                         cursor.execute("SELECT * FROM devices WHERE privilege =
'Guest'")
533                                         if (row[2]
== 'ADMIN') :

```

```

534     # Select all devices
535     cursor.execute("SELECT * FROM devices")
536     message['devices'] = cursor.fetchall()
537     ch.basic_publish(exchange=' ',
538         routing_key=row[0],
539         body=AESencrypt(json.dumps(message),row[1]))
540                                         cursor.close()
541                                         cnx.close()
542                                         gc.collect()
543
544             except mysql.connector.Error as
err :
545                     print (err)
546
547             elif (sent['type'] == 'addscenario') :
548                     print (sent)
549                     try :
550                         #terhubung ke database
551                         cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
552                                         cursor = cnx.cursor()
553                                         cursor.execute("CREATE
TABLE " + sent['name'] + " (name VARCHAR(20), address VARCHAR(20),
type VARCHAR(11), status VARCHAR(9), room VARCHAR(20), privilege
VARCHAR(5))")
554                                         privilege = 'Guest'
555                                         for element in
sent['devices'] :
556
                    cursor.execute("INSERT INTO " + sent['name'] + " VALUES
(%s,%s,%s,%s,%s)",
(element[0],element[1],element[2],element[3],element[4],element[5]))
557                                         if
(element[5] == 'ADMIN') :
558
                    privilege = 'ADMIN'
559                                         #memasukan name
skenario ke tabel scenarios
560                                         cursor.execute("INSERT
INTO scenarios VALUES (%s,%s,%s)",
(sent['name'],sent['time'],privilege))
561                                         cnx.commit()
562
563                                         #memformat waktu
564                                         if (sent['time'][0] ==
'0') :
565
                    time_hr =
sent['time'][1]
566                                         else :
567                                         time_hr =
sent['time'][0] + sent['time'][1]

```

```

568                                         if (sent['time'][2] ==
569                                         time_mn =
'0') :
570                                         time_mn =
571                                         else :
572                                         time_mn =
573                                         sent['time'][2] + sent['time'][3]
574                                         #menulis file python
575                                         baru untuk dijalankan secara rutin
576                                         if (sent['time'] != 'none') :
577                                         f = open
578                                         (sent['name']+'.py', 'w')
579                                         f.write("import pika\n")
580                                         f.write("import json\n")
581                                         f.write("import pika\n")
582                                         f.write("from datetime import datetime\n")
583                                         f.write("from threading import Timer\n")
584                                         f.write("import time\n\n")
585                                         f.write("homeid = '123456'\n")
586                                         f.write("def recursive() :\n")
587                                         f.write("\tchannel.basic_publish(exchange='amq.topic',
588                                         routing_key='123456RE', body=joined)\n")
589                                         f.write("\tttime.sleep(2)\n")
590                                         f.write("\tx = datetime.today()\n")
591                                         f.write("\ty = x.replace(day = x.day+1, hour = " + time_hr + ",
592                                         minute = " + time_mn + ", second = 0)\n")
593                                         f.write("\tttime.sleep(2)\n")
594                                         f.write("\tdelta = y-x\n")
595                                         f.write("\tt = Timer(delta.seconds, recursive,())\n")
596                                         f.write("\tt.start()\n\n")
597                                         f.write("tosend = {}\n")
598                                         f.write("tosend['homeid'] = homeid\n")
599                                         f.write("tosend['head'] = 'SC'\n")
600                                         f.write("tosend['name'] = '" + sent['name'] + "'\n")

```

```

598     f.write("joined = json.dumps(tosend) \n\n")
599
600     f.write("credentials =
pika.PlainCredentials('revie','revie')\n")
601
602     f.write("parameters =
pika.ConnectionParameters('192.168.43.237',5672,'host_test',credential
s)\n")
603
604     f.write("connection = pika.BlockingConnection(parameters)\n")
605
606     f.write("channel = connection.channel()\n\n")
607
608     f.write("if (y < x) :\n")
609
610     f.write("\ty = y.replace(day = x.day +1, hour = " + time_hr + ", minute = " + time_mn
+ ", second = 0)\n")
611
612     f.write("delta = y-x\n")
613
614
615 [sent['name'] +'.py']
616
617 bank[sent['name']] = Popen([sys.executable or 'python'] + args,
618 stdut=PIPE)
619
620 print(sent['name'] + " = Popen([sys.executable or 'python'] +
621 args, stdut=PIPE)")
622
623 # form the message
624 message = {}
625 message['head'] = 'RE'
626 message['type'] =
627 'scenarios'
628 cursor.execute("SELECT
629 * FROM scenarios")
630
631 cursor.fetchall()
632
633 = sent['notifsender']
634
635 message['notifmessage'] = sent['notifmessage']
636
637 cursor.execute("SELECT
638 * FROM notifications")

```

```

627     message['notifications'] = cursor.fetchall()
628                                         # broadcast the
message
629                                         cursor.execute("SELECT
phoneid, aeskey FROM users")
630                                         data =
cursor.fetchall()
631                                         for row in data :
632
ch.basic_publish(exchange=' ',
633     routing_key=row[0],
634     body=AESencrypt(json.dumps(message),row[1]))
635                                         cursor.close()
636                                         cnx.close()
637                                         gc.collect()
638
639     except mysql.connector.Error as
err :
640         print (err)
641
642     elif (sent['type'] == 'deletescenario') :
643         try :
644             #terhubung ke database
645             cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
646             cursor = cnx.cursor()
647             #menghapus skenario
dari database
648             cursor.execute("DELETE
FROM scenarios WHERE name = %s", (sent['name'],))
649             cursor.execute("DROP
TABLE " + sent['name'])
650             cnx.commit()
651             #menghentikan program
python skenario (bila ada
652             filename =
sent['name']+'.py'
653             try :
654                 bank[sent['name']].terminate()
655             bank =
removekey(bank,sent['name'])
656             os.remove(filename)
657             except :
658                 e =
659                 print(e)
660
661             sys.exc_info()[0]
662
663             # form the message
message = {}
664             message['head'] = 'RE'
message['type'] =
'scenarios'

```

```

665                                         cursor.execute("SELECT
* FROM scenarios")
666                                         message['scenarios'] =
cursor.fetchall()
667                                         message['notifsender'] =
sent['notifsender']
668                                         message['notifmessage'] = sent['notifmessage']
669                                         cursor.execute("SELECT
* FROM notifications")
670                                         message['notifications'] = cursor.fetchall()
671                                         # broadcast the
message
672                                         cursor.execute("SELECT
phoneid, aeskey FROM users")
673                                         data =
cursor.fetchall()
674                                         for row in data :
675                                         ch.basic_publish(exchange='',
676                                         routing_key=row[0],
677                                         body=AESencrypt(json.dumps(message),row[1]))
678                                         cursor.close()
679                                         cnx.close()
680                                         gc.collect()
681
682                                         except mysql.connector.Error as
err1 :
683                                         print (err1)
684
685                                         except OSError as err2 :
686                                         print (err2)
687
688                                         elif (sent['type'] == 'editscenario_req') :
689                                         try :
690                                         #akses ke database
691                                         cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
692                                         cursor = cnx.cursor()
693                                         #mengambil data dari
database
694                                         cursor.execute("SELECT
* FROM " + sent['name'])
695                                         data =
cursor.fetchall()
696                                         tosend = {}
697                                         tosend['devices'] =
(data)
698                                         cursor.execute("SELECT
* FROM scenarios WHERE name = %s", (sent['name'],))
699                                         data =
cursor.fetchall()
700                                         tosend['time'] =
data[0][1]

```

```

701                                         message =
702 json.dumps(tosend)                                #mengirim pesan
703
704         ch.basic_publish(exchange=' ',           ,
705                           routing_key=props.reply_to,
706                           properties=pika.BasicProperties(correlation_id = \
707
708
709                                         props.correlation_id),
707
710                                         body=AESencrypt(message, key))
708
711                                         cursor.close()
710                                         cnx.close()
711                                         gc.collect()
712                                         except mysql.connector.Error as
err :
713                                         print (err)
714
715                                         elif (sent['type'] == 'editscenario') :
716                                         try :
717
718                                         cnx =
719 mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
720                                         cursor = cnx.cursor()
721                                         #delete old
721                                         cursor.execute("DELETE
FROM scenarios WHERE name = %s", (sent['oldname'],))
722                                         cursor.execute("DROP
TABLE " + sent['oldname'])
723                                         cnx.commit()
724                                         filename =
724                                         sent['oldname']+'.py'
725                                         try :
726
726                                         bank[sent['oldname']].terminate()
727                                         bank =
727                                         removekey(bank,sent['name'])
728
728                                         os.remove(filename)
729
730                                         except :
731                                         e =
731                                         sys.exc_info()[0]
732                                         print(e)
733
734                                         #create new
735                                         cursor.execute("CREATE
TABLE " + sent['name'] + " (name VARCHAR(20), address VARCHAR(20),
type VARCHAR(11), status VARCHAR(9), room VARCHAR(20), privilege
VARCHAR(5)) ")

```

```

736                                         privilege = 'Guest'
737                                         for element in
738     sent['devices'] :
739
740         cursor.execute("INSERT INTO " + sent['name'] + " VALUES
741 (%s,%s,%s,%s,%s,%s)",
742 (element[0],element[1],element[2],element[3],element[4],element[5]))           if
743
744     (element[5] == 'ADMIN') :
745
746         privilege = 'ADMIN'                                         cursor.execute("INSERT
747 INTO scenarios VALUES (%s,%s,%s)",
748 (sent['name'],sent['time'],privilege))                                         cnx.commit()
749
750
751     if (sent['time'][0] == '0') :                                         time_hr =
752
753     sent['time'][1]                                         else :
754
755     sent['time'][0] + sent['time'][1]                                         time_hr =
756
757     if (sent['time'][2] == '0') :                                         time_mn =
758
759     sent['time'][3]                                         else :
760
761     sent['time'][2] + sent['time'][3]                                         time_mn =
762
763
764     if (sent['time'] != 'none') :                                         f = open
765
766     (sent['name']+'.py', 'w')
767
768     f.write("import pika\n")
769
770     f.write("import json\n")
771
772     f.write("import pika\n")
773
774     f.write("from datetime import datetime\n")
775
776     f.write("from threading import Timer\n")
777
778     f.write("import time\n\n")
779
780
781     f.write("homeid = '123456'\n")
782
783     f.write("def recursive() :\n")
784
785     f.write("\tchannel.basic_publish(exchange='amq.topic',
786 routing_key='123456RE', body=joined)\n")
787
788     f.write("\tttime.sleep(2)\n")

```

```

767     f.write("\tx = datetime.today()\n")
768     f.write("\ty = x.replace(day = x.day+1, hour = " + time_hr + ",
769     minute = " + time_mn + ", second = 0)\n")
769     f.write("\tttime.sleep(2)\n")
770     f.write("\tdelta = y-x\n")
771     f.write("\tt = Timer(delta.seconds, recursive,())\n")
772     f.write("\tt.start()\n\n")
773     f.write("tosend = {}")
774     f.write("tosend['homeid'] = homeid\n")
775     f.write("tosend['head'] = 'SC'\n")
776     f.write("tosend['name'] = '" + sent['name'] + "'\n")
777     f.write("joined = json.dumps(tosend)\n\n")
778
779     f.write("credentials =
780 pika.PlainCredentials('revie','revie')\n")
780     f.write("parameters =
781 pika.ConnectionParameters('192.168.43.237',5672,'host_test',credentials)\n")
781     f.write("connection = pika.BlockingConnection(parameters)\n")
782     f.write("channel = connection.channel()\n\n")
783
784                                     f.write("x
784 = datetime.today()\n")
785                                     f.write("y
785 = x.replace(day = x.day, hour = " + time_hr + ", minute = " + time_mn
786 + ", second = 0)\n")
786     f.write("if (y < x) :\n")
787     f.write("\ty = y.replace(day = x.day +1, hour = " + time_hr + ",
787     minute = " + time_mn + ", second = 0)\n")
788     f.write("delta = y-x\n")
789
790                                     f.write("t
790 = Timer(delta.seconds, recursive,())\n")
791     f.write("t.start()")
792
793                                     f.close()
793 args =
794 [sent['name'] +'.py']
794 #exec("global a")

```

```

795     bank[sent['name']] = Popen([sys.executable or 'python'] + args,
796                               stdout=PIPE)
796     print(sent['name'] + " = Popen([sys.executable or 'python'] +
797 args, stdout=PIPE)")
797
798
799                                         # form the message
800 message = {}
801 message['head'] = 'RE'
802 message['type'] =
802 'scenarios'
803                                         cursor.execute("SELECT
803 * FROM scenarios")
804 cursor.fetchall()
805 = sent['notifsender']
806
806 message['notifmessage'] = sent['notifmessage']
807                                         cursor.execute("SELECT
807 * FROM notifications")
808
808 message['notifications'] = cursor.fetchall()
809                                         # broadcast the
809 message
810                                         cursor.execute("SELECT
810 phoneid, aeskey FROM users")
811
811 cursor.fetchall()
812                                         for row in data :
813
813 ch.basic_publish(exchange='',
814 routing_key=row[0],
815
815 body=AESencrypt(json.dumps(message),row[1]))
816                                         cursor.close()
817                                         cnx.close()
818                                         gc.collect()
819
820                                         except mysql.connector.Error as
820 err1 :
821                                         print (err1)
822
823                                         except OSError as err2 :
824                                         print (err2)
825
826                                         elif (sent['type'] == 'editprivilege') :
827                                         try :
828
829                                         cnx =
829 mysql.connector.connect(user='root', password='28031995',
829 host='localhost', database='home'+homeid)
830                                         cursor = cnx.cursor()
831
832                                         print ("here")

```

```

833
834                                     cursor.execute("UPDATE
devices SET privilege = 'ADMIN' ")
835                                     for element in
836 sent['checked'] :
837
838                                     cursor.execute("UPDATE devices SET privilege = 'Guest' WHERE
name = %s", (element[0],))
839                                     cnx.commit()
840                                     # form the message
841                                     message = {}
842                                     message['head'] = 'RE'
843                                     message['type'] =
844                                     # broadcast the
845                                     cursor.execute("SELECT
846                                     data =
847                                     for row in data :
848                                     if (row[2]
849                                     == 'Guest') :
850                                     # Select only Guests' Devices
851                                     cursor.execute("SELECT * FROM devices WHERE privilege =
852                                     'Guest' ")
853                                     if (row[2]
854                                     == 'ADMIN') :
855                                     # Select all devices
856                                     cursor.execute("SELECT * FROM devices")
857                                     message['devices'] = cursor.fetchall()
858                                     ch.basic_publish(exchange=' ',
859                                     routing_key=row[0],
860                                     body=AESencrypt(json.dumps(message),row[1]))
861                                     cursor.close()
862                                     cnx.close()
863                                     gc.collect()
864                                     except mysql.connector.Error as
err1 :
865                                     print (err1)

```

Bagian ini melayani data dengan *header* ‘MD’ (manajemen *devices*).

Tabel 18 Pesan yang Dilayani raspi_aes.py Bagian 3

MD	Pons el	adddevice	Perminta an	•	Home ID	Host menyimpan nama, address, dan
----	---------	-----------	-------------	---	---------	-----------------------------------

(Manajemen Device)		menambah perangkat baru	<ul style="list-style-type: none"> • Nama Perangkat • Address • Tipe • Notif message • Notif sender 	tipe perangkat ke database homeXXXXXX.devices.
		deletedevice	<p>Permintaan menghapus perangkat</p> <ul style="list-style-type: none"> • Home ID • Address perangkat yang ingin dihapus • Notif message • Notif sender 	Host menghapus perangkat dengan address yang sesuai.
		editdevice	<p>Permintaan edit nama perangkat</p> <ul style="list-style-type: none"> • Home ID • Address perangkat • Nama baru • Notif message • Notif sender 	Host mengganti nama perangkat yang sesuai menjadi nama yang baru
		addrroom	<p>Permintaan menambah room</p> <ul style="list-style-type: none"> • Home ID • Nama room • Daftar perangkat yang terkandung pada room itu • Notif message • Notif sender 	Host menambahkan nama room ke homeXXXXXX.rooms , lalu mengubah kolom room pada homeXXXXXX.devices menjadi nama room tersebut untuk perangkat yang terdapat pada daftar

		deleteroom	Permintaan menghapus room	<ul style="list-style-type: none"> • Home ID • Nama room • Notif message • Notif sender 	Host menghapus nama room dari homeXXXXXX.rooms dan mengganti isi kolom room dengan nama room bersangkutan menjadi ‘none’
		editroom	Permintaan edit nama atau isi room	<ul style="list-style-type: none"> • Home ID • Nama room baru • Daftar perangkat baru • Notif message • Notif sender 	Host mengganti nama pada homeXXXXXX.rooms menjadi nama baru. Host juga mengganti kolom room supaya sesuai dengan daftar perangkat yang dikirimkan.
		addscenario	Permintaan menambah skenario	<ul style="list-style-type: none"> • Home ID • Nama scenario • Waktu scenario • Daftar perangkat dalam skenario tersebut • Notif message • Notif sender 	Host menambahkan nama dan waktu skenario ke tabel homeXXXXXX.scenarios. Host juga membuat tabel bernama sesuai dengan nama skenario yang isinya adalah daftar perangkat skenario tersebut beserta statusnya.
		deletescenario	Permintaan menghapus skenario	<ul style="list-style-type: none"> • Home ID • Nama scenario • Notif message • Notif sender 	Host menghapus nama scenario dari tabel homeXXXXXX.scenarios, lalu menghapus tabel bernama skenario tersebut dari basis data.
		editscenario	Permintaan edit skenario	<ul style="list-style-type: none"> • Home ID 	Host mengubah nama dan waktu skenario di tabel scenarios, lalu

				<ul style="list-style-type: none"> • Nama scenario • Nama baru • Waktu baru • Daftar perangkat baru • Notif message • Notif sender 	mengubah isi tabel dengan nama skenario tersebut dengan daftar perangkat yang baru.
		editprivilege	Permintaan edit privilege perangkat	<ul style="list-style-type: none"> • Home ID • Daftar privilege baru 	Host mengubah seluruh privilege perangkat di tabel homeXXXXXX.devices.

Potongan berikutnya adalah sebagai berikut.

```
# COMMAND TO DEVICES
866             elif (sent['homeid'] == homeid and sent['head'] ==
'CO') :
867
868                 try :
869                     cnx =
mysql.connector.connect(user='root', password='28031995',
host='localhost', database='home'+homeid)
870                     cursor = cnx.cursor()
871
872                     #CHECK LOCK STATUS
873                     cursor.execute("SELECT
lockstatus FROM info")
874
875                     #BILA SEDANG TERKUNCI,
#KIRIMKAN WARNING NOTIFICATION
876                     if (cursor.fetchone()[0] ==
'lock') :
877                         message = {}
878                         message['head'] =
'WA'
879                         message
880                         cursor.execute("SELECT phoneid, aeskey FROM users")
881                         data =
cursor.fetchall()
882                         ch.basic_publish(exchange='',
883                                         routing_key=row[0],
```

```

884         body=AESencrypt(json.dumps(message),row[1]))
885
886             else :
887
888             cursor.execute("UPDATE devices SET status = %s WHERE address
= %s", (sent['command'], sent['address']))
889             cursor.execute("SELECT lasttime, d" +
datetime.now().strftime('%d') + " from " + 'data' +
datetime.now().strftime('%m%Y') + " WHERE address = '" +
sent['address'] + "")+
890             print ("SELECT
lasttime, d" + datetime.now().strftime('%d') + " from " + 'data' +
datetime.now().strftime('%m%Y') + " WHERE address = '" +
sent['address'] + "")+
891             data=cursor.fetchone()
892             if (data[0] != 'none' and (sent['command'] == '0' or sent['command'] ==
'ff000000')) :
893             lasttime = datetime.strptime(data[0],"%d%m%Y%H%M%S")
894             delta = float((datetime.now()-lasttime).seconds)/60
895             print (delta)
896             new =
"%.1f" % (float(data[1]) + delta)
897             cursor.execute("UPDATE " + 'data' +
datetime.now().strftime('%m%Y') + " SET d" +
datetime.now().strftime('%d') + " = " + new + ", lasttime = 'none'
WHERE address = '" + sent['address'] + "")+
898             elif (data[0] ==
'none' and not(sent['command'] == '0' or sent['command'] ==
'ff000000')) :
899             cursor.execute("UPDATE " + 'data' +
datetime.now().strftime('%m%Y') + " SET lasttime = " +
datetime.now().strftime('%d%m%Y%H%M%S') + " WHERE address = '" +
sent['address'] + "")+
900             cnx.commit()
901
902
903             #sending xbee
signal
904             Master.get_destination()
905             Master.change_destination(sent['address']+`\r`)
906
907             if (sent['type'] ==
'switch') :
908             header_xbee = bytes(homeid + sent['address'])
```

```

909     XB_Port.write(header_xbee)
910
911     if (sent['command'] == '100') :
912         XB_Port.write(chr(1))
913     elif (sent['command'] == '0') :
914         XB_Port.write(chr(0))
915     elif (sent['type'] == 'lamp') :
916         r = int(sent['command'][2:4],16)
917         g = int(sent['command'][4:6],16)
918         b = int(sent['command'][6:8],16)
919         print (r,g,b)
920
921         header_xbee = bytes(homeid + sent['address'])
922         XB_Port.write(header_xbee)
923         XB_Port.write(chr(r))
924         XB_Port.write(chr(g))
925         XB_Port.write(chr(b))
926         elif (sent['type'] == 'fan') :
927             intensity = int(int(sent['command'])*2.55)
928             header_xbee = bytes(homeid + sent['address'])
929             XB_Port.write(header_xbee)
930             XB_Port.write(chr(intensity))
931             elif (sent['type'] == 'door') :
932                 header_xbee = bytes(homeid + sent['address'])
933                 XB_Port.write(header_xbee)
934                 if (sent['command'] == '100') :
935                     XB_Port.write(chr(1))
936                 elif (sent['command'] == '0') :
937                     XB_Port.write(chr(0))
938                 elif (sent['type'] == 'curtain') :
939                     header_xbee = bytes(homeid + sent['address'])

```

```

938     XB_Port.write(header_xbee)
939     intensity = int(int(sent['command'])*2.55)
940
941     if
942     (intensity > 0) :
943         XB_Port.write(chr(1))
944
945     else :
946
947         XB_Port.write(chr(0))
948
949     # broadcast the
950     message to update device status in all phones
951     cursor.execute("SELECT phoneid, aeskey FROM users")
952     data =
953     cursor.fetchall()
954     for row in
955     data :
956         ch.basic_publish(exchange='',
957             routing_key=row[0],
958             body=AESencrypt(json.dumps(sent),row[1]))
959
960
961
962         # SCENARIO COMMAND
963         elif (sent['homeid'] == homeid and sent['head'] ==
964         'SC') :
965             try :
966                 print (sent['name'])
967                 cnx =
968                 mysql.connector.connect(user='root', password='28031995',
969                 host='localhost', database='home'+homeid)
970                 cursor = cnx.cursor()
971                 cursor.execute("SELECT *
972                 from " + sent['name'])
973                 data = cursor.fetchall()
974                 tosend = {}
975                 #mengirimkan perintah CO
976                 for element in data :
977                     tosend['homeid'] =
978                     tosend['head'] =
979                     'CO'

```

```

975     tosend['address'] = element[1]
976     tosend['command'] = element[3]
977     tosend['type'] =
978     element[2]
979     ch.basic_publish(exchange=' ',
980                     routing_key=sent['homeid']+AES',
981                     properties=pika.BasicProperties(correlation_id = \
982
983                                     props.correlation_id),
984
985                     body=AESencrypt(json.dumps(tosend),key))
986
987             if (sent['name'] ==
988                 'lock_scen') :
989                 cursor.execute("UPDATE info SET lockstatus = 'lock'")
990
991             if (sent['name'] ==
992                 'unlock_scen') :
993                 cursor.execute("UPDATE info SET lockstatus = 'unlock'")
994
995                 cnx.commit()
996                 cursor.close()
997                 cnx.close()
998                 gc.collect()
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
20210
20211
20212
20213
20214
20215
20216
20217
20218
20219
20220
20221
20222
20223
20224
20225
20226
20227
20228
20229
202210
202211
202212
202213
202214
202215
202216
202217
202218
202219
202220
202221
202222
202223
202224
202225
202226
202227
202228
202229
2022210
2022211
2022212
2022213
2022214
2022215
2022216
2022217
2022218
2022219
2022220
2022221
2022222
2022223
2022224
2022225
2022226
2022227
2022228
2022229
20222210
20222211
20222212
20222213
20222214
20222215
20222216
20222217
20222218
20222219
20222220
20222221
20222222
20222223
20222224
20222225
20222226
20222227
20222228
20222229
202222210
202222211
202222212
202222213
202222214
202222215
202222216
202222217
202222218
202222219
202222220
202222221
202222222
202222223
202222224
202222225
202222226
202222227
202222228
202222229
2022222210
2022222211
2022222212
2022222213
2022222214
2022222215
2022222216
2022222217
2022222218
2022222219
2022222220
2022222221
2022222222
2022222223
2022222224
2022222225
2022222226
2022222227
2022222228
2022222229
20222222210
20222222211
20222222212
20222222213
20222222214
20222222215
20222222216
20222222217
20222222218
20222222219
20222222220
20222222221
20222222222
20222222223
20222222224
20222222225
20222222226
20222222227
20222222228
20222222229
202222222210
202222222211
202222222212
202222222213
202222222214
202222222215
202222222216
202222222217
202222222218
202222222219
202222222220
202222222221
202222222222
202222222223
202222222224
202222222225
202222222226
202222222227
202222222228
202222222229
2022222222210
2022222222211
2022222222212
2022222222213
2022222222214
2022222222215
2022222222216
2022222222217
2022222222218
2022222222219
2022222222220
2022222222221
2022222222222
2022222222223
2022222222224
2022222222225
2022222222226
2022222222227
2022222222228
2022222222229
20222222222210
20222222222211
20222222222212
20222222222213
20222222222214
20222222222215
20222222222216
20222222222217
20222222222218
20222222222219
20222222222220
20222222222221
20222222222222
20222222222223
20222222222224
20222222222225
20222222222226
20222222222227
20222222222228
20222222222229
202222222222210
202222222222211
202222222222212
202222222222213
202222222222214
202222222222215
202222222222216
202222222222217
202222222222218
202222222222219
202222222222220
202222222222221
202222222222222
202222222222223
202222222222224
202222222222225
202222222222226
202222222222227
202222222222228
202222222222229
2022222222222210
2022222222222211
2022222222222212
2022222222222213
2022222222222214
2022222222222215
2022222222222216
2022222222222217
2022222222222218
2022222222222219
2022222222222220
2022222222222221
2022222222222222
2022222222222223
2022222222222224
2022222222222225
2022222222222226
2022222222222227
2022222222222228
2022222222222229
20222222222222210
20222222222222211
20222222222222212
20222222222222213
20222222222222214
20222222222222215
20222222222222216
20222222222222217
20222222222222218
20222222222222219
20222222222222220
20222222222222221
20222222222222222
20222222222222223
20222222222222224
20222222222222225
20222222222222226
20222222222222227
20222222222222228
20222222222222229
202222222222222210
202222222222222211
202222222222222212
202222222222222213
202222222222222214
202222222222222215
202222222222222216
202222222222222217
202222222222222218
202222222222222219
202222222222222220
202222222222222221
202222222222222222
202222222222222223
202222222222222224
202222222222222225
202222222222222226
202222222222222227
202222222222222228
202222222222222229
2022222222222222210
2022222222222222211
2022222222222222212
2022222222222222213
2022222222222222214
2022222222222222215
2022222222222222216
2022222222222222217
2022222222222222218
2022222222222222219
2022222222222222220
2022222222222222221
2022222222222222222
2022222222222222223
2022222222222222224
2022222222222222225
2022222222222222226
2022222222222222227
2022222222222222228
2022222222222222229
20222222222222222210
20222222222222222211
20222222222222222212
20222222222222222213
20222222222222222214
20222222222222222215
20222222222222222216
20222222222222222217
20222222222222222218
20222222222222222219
20222222222222222220
20222222222222222221
20222222222222222222
20222222222222222223
20222222222222222224
20222222222222222225
20222222222222222226
20222222222222222227
20222222222222222228
20222222222222222229
202222222222222222210
202222222222222222211
202222222222222222212
202222222222222222213
202222222222222222214
202222222222222222215
202222222222222222216
202222222222222222217
202222222222222222218
202222222222222222219
202222222222222222220
202222222222222222221
202222222222222222222
202222222222222222223
202222222222222222224
202222222222222222225
202222222222222222226
202222222222222222227
202222222222222222228
202222222222222222229
2022222222222222222210
2022222222222222222211
2022222222222222222212
2022222222222222222213
2022222222222222222214
2022222222222222222215
2022222222222222222216
2022222222222222222217
2022222222222222222218
2022222222222222222219
2022222222222222222220
2022222222222222222221
2022222222222222222222
2022222222222222222223
2022222222222222222224
2022222222222222222225
2022222222222222222226
2022222222222222222227
2022222222222222222228
2022222222222222222229
20222222222222222222210
20222222222222222222211
20222222222222222222212
20222222222222222222213
2
```

```

1010         routing_key=props.reply_to,
1011         properties=pika.BasicProperties(correlation_id = \
1012
1013
1014     body=AESencrypt(json.dumps(tosend),key))
1015             except mysql.connector.Error as err2 :
1016                 print (err2)
1017
1018             elif (sent['homeid'] == homeid and sent['head'] ==
1019                 'LC') :
1020                 print (sent['lat']),
1021                 cnx =
1022                 mysql.connector.connect(user='root', password='28031995',
1023                 host='localhost', database='home'+homeid)
1024                 cursor = cnx.cursor()
1025                 cursor.execute("SELECT
1026                 latitude, longitude FROM info")
1027                 data = cursor.fetchone()
1028                 #memperoleh nilai latitude
1029                 dan longitude dalam float
1030                 latf = float(sent['lat'])
1031                 lngf = float(sent['lng'])
1032                 #jika lat dan lng bernilai
1033                 0, artinya GPS ponsel pengguna sedang mati
1034                 if (latf == 0.0 and lngf ==
1035                 0.0) :
1036                     #asumsikan ada di
1037                     dalam rumah
1038                     cursor.execute("UPDATE
1039                     users SET location = 'in' WHERE phoneid = %s",
1040                     (props.correlation_id,))
1041                     #jika posisi pengguna sudah
1042                     berada di luar radius batas (kurang lebih 100 meter)
1043                     elif (sqrt(math.pow(latf-
1044                     data[0],2) + math.pow(lngf-data[1],2)) > 0.001) :
1045                     cursor.execute("UPDATE
1046                     users SET location = 'out' WHERE phoneid = %s",
1047                     (props.correlation_id,))
1048                     #cek apakah saat itu masih
1049                     ada user di dalam radius
1050                     cursor.execute("SELECT EXIST
1051                     (SELECT location FROM users WHERE location = 'in')")
1052                     #jika tidak ada
1053                     if
1054                     (not(cursor.fetchone()[0])):
1055                         tosend['homeid'] =
1056                         homeid
1057                         tosend['head'] = 'SC'
1058                         tosend['name'] =
1059                         'lock scen'

```

```

1040     ch.basic_publish(exchange=' ',
1041         routing_key=sent['homeid']+ 'AES' ,
1042         properties=pika.BasicProperties(correlation_id = \
1043
1044             props.correlation_id),
1045
1046
1047     except :
1048         print(sys.exc_info()[0])
1049
1050     channel.basic_consume(callback,
1051                             queue=queue_name, no_ack=True)
1052
1053     channel.start_consuming()

```

Potongan ini melayani data dengan *header* ‘CO’, ‘SC’, ‘DA’, dan ‘LC’.

Tabel 19 Pesan yang Dilayani raspi_aes.py Bagian 4

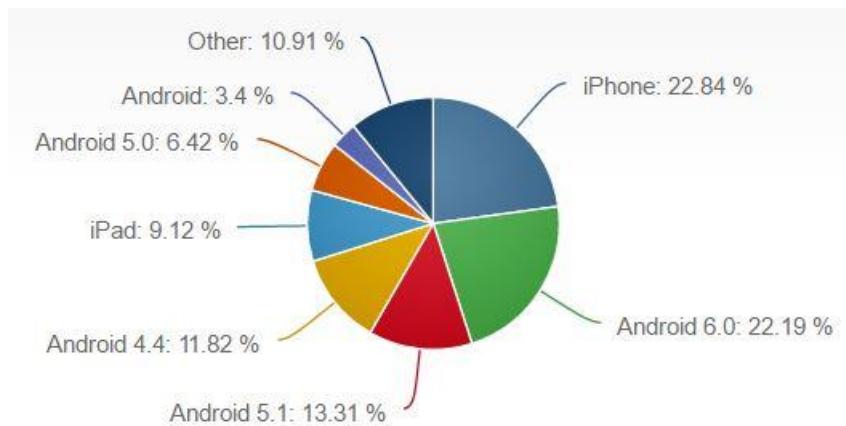
Header	Pengirim	Tip	Fungsi	Isi paket permintaan	Aksi Host
CO (Command)	Ponsel	-	Perintah terhadap perangkat (menyalakan, matikan, dll)	<ul style="list-style-type: none"> • Home ID • Alamat perangkat yang ingin diberi perintah • Isi perintah 	Host mengubah status perangkat pada tabel homeXXXXXX.device s. Host juga mengirimkan pesan ke perangkat melalui jaringan Zigbee (bentuk pesan sesuai dengan yang telah dibahas pada subbab 4.2).
NO	Ponsel	-	Perintah untuk menyebarkan notifikasi	<ul style="list-style-type: none"> • Home ID • Pengirim notifikasi • Isi pesan notifikasi 	Host menyebarkan isi notifikasi ke seluruh user yang terdaftar pada rumah. Host juga menyimpan isi notifikasi ke tabel notifikasi.
SC	Ponsel	-	Perintah untuk menjalankan suatu skenario	<ul style="list-style-type: none"> • Home ID • Nama scenario 	Host melihat isi tabel scenario. Lalu, satu per satu menjalankan aksi seperti pada header CO sesuai dengan isi tabel scenario tersebut.

DA	Ponsel	-	Permintaan data penggunaan perangkat	<ul style="list-style-type: none"> • Home ID • Alamat perangkat 	Host melihat isi tabel data, kemudian mengirimkan data penggunaan perangkat yang dimaksud sesuai dengan alamat yang diberikan ke ponsel peminta.
LC	Ponsel	-	Pengiriman lokasi <i>user</i>	<ul style="list-style-type: none"> • Posisi <i>latitude</i> • Posisi <i>longitude</i> 	Host menghitung jarak <i>user</i> dengan posisi rumah. Bila semua <i>user</i> sudah berada di luar radius maksimum, maka host akan mengaktifkan skenario <i>lock</i> .

6. Implementasi Android Mobile Application

Untuk menghubungkan pengguna dengan perangkat rumah cerdas, digunakan aplikasi Android. Perancangan aplikasi ini didasarkan pada rancangan yang ada pada dokumen B300, namun dengan beberapa pembaruan untuk membuat aplikasi berjalan dengan lebih efisien, baik dari sisi layoutnya maupun dari sisi hubungan antar halamannya.

Aplikasi diimplementasikan menggunakan Android SDK dengan API 16: Android 4.1 (Jelly Bean) sebagai sistem minimum aplikasi dengan jumlah pengguna 95.2% dari total pengguna Android. Hal ini dilakukan dengan pertimbangan banyaknya pengguna yang masih menggunakan sistem operasi Jelly Bean. Untuk kedepannya, akan dibuat aplikasi Rumah Cerdas dalam sistem operasi IOS dan juga Windows Phone agar lebih memudahkan pengguna dalam menggunakan sistem Rumah Cerdas.



Gambar 16 Pie chart OS market share

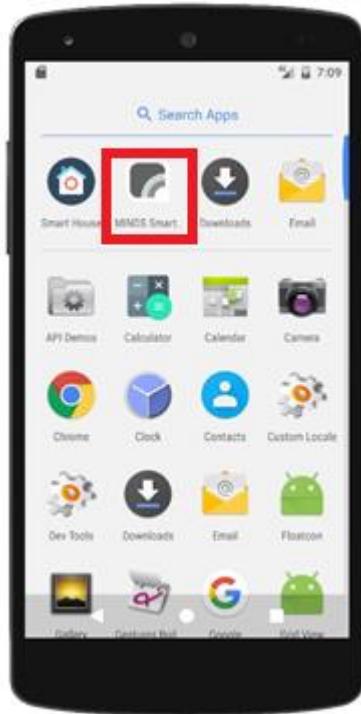
Aplikasi Rumah Cerdas ini diberi nama MINDS yang merupakan singkatan dari *Meshed and Internet Networked Devices System*. MINDS dibuat dengan menggunakan Android Studio 2.2 karena komunitas pengguna Android Studio lebih besar dibanding dengan komunitas pengguna Eclipse. Pada Android Studio, layout dapat dengan mudah dibuat dengan adanya fasilitas *drag-and-drop* dan pada Android Studio juga terdapat fitur Android Simulator sehingga memudahkan pada saat uji coba aplikasi. Digunakan juga desain android 25.1.0 dengan pertimbangan banyaknya fitur yang dapat diakses jika menggunakan desain tersebut.

Dalam perancangan aplikasi Android, dikenal istilah *Activity*. *Activity* secara sederhana dapat diartikan sebagai halaman dalam aplikasi. Satu *activity* pasti minimal terdiri dari satu buah berkas *front end* (tampilan) dan satu buah berkas *back end*. Dalam implementasi proyek ini, bahasa yang digunakan untuk merancang tampilan adalah XML, sedangkan bahasa yang digunakan untuk merancang *back end* adalah Java. Kedua bahasa ini telah ditentukan oleh Android Studio sebagai bahasa standar.

Aplikasi Rumah Cerdas dibuat menggunakan berbagai *Activity* yang dihubungkan dengan menggunakan *Intent*. *Activity* dirancang berdasarkan rancangan yang telah dijabarkan pada dokumen B300, namun dengan pembaruan dan penyesuaian supaya aplikasi berjalan dengan lebih efisien, baik dari sisi tampilan maupun dari sisi hubungan antar *Activity*. *Trigger* yang digunakan untuk menjalankan *Intent* dapat berupa berbagai macam, mulai dari *Button* seperti pada *FirstActivity*, *SignInActivity*, dan *SignUpActivity*, *ImageView* seperti pada gambar “>” dalam *PagerActivity*, *TextView* seperti pada tulisan “*add member*” dalam *CreateNewActivity2*, *GridView* seperti pada menu *device tab, room*

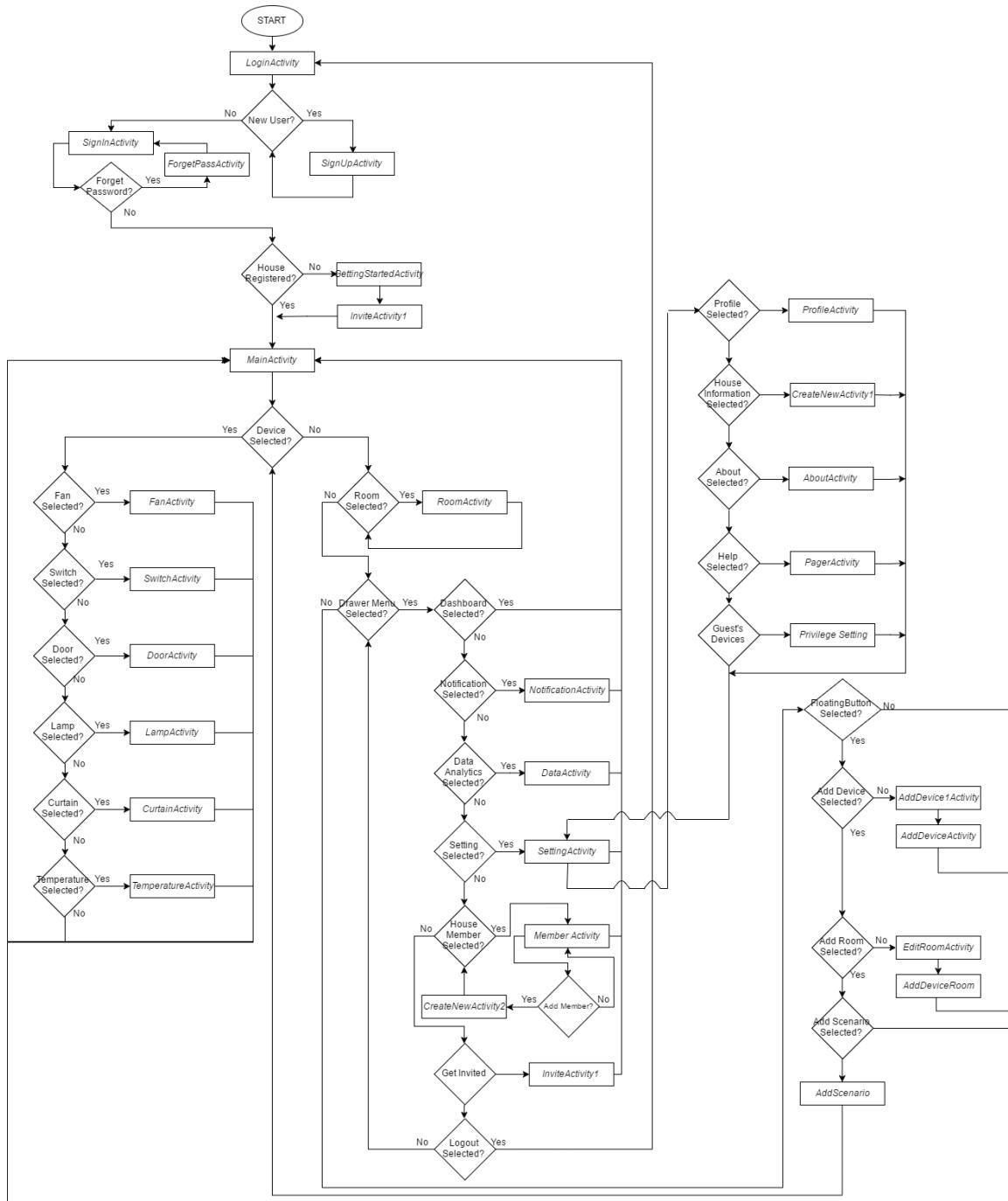
tab maupun *scenario tab* pada *MainActivity* dan *ListView* seperti pada bagian *RoomActivity*. Untuk menampilkan tampilan sebuah *Activity*, dibutuhkan sebuah *layout file* yang memiliki *format xml* yang mengatur sebagian besar hal yang dibutuhkan untuk menampilkan *Activity* tersebut, mulai dari komponen, *widget*, cara peletakan, ukuran, jarak antar komponen hingga *style* dari setiap komponen.

Font yang digunakan untuk *TextView*, *EditText* pada aplikasi ini adalah “Quicksand-Bold.otf” sedangkan untuk *Button* adalah “Quicksand-Regular.otf”. *Font* tersebut dipilih karena tepian dari setiap karakter tidak terlalu tajam sehingga menimbulkan kesan yang ringan dan tidak formal. Penggantian *font* pada setiap objek pada *Activity* dilakukan dengan mengganti *TypeFace* dari setiap objek tersebut. Warna pada setiap *Activity* didesain tidak terlalu mencolok dengan sedikit permainan warna pada bagian yang penting. *Icon* aplikasi MINDS menggunakan logo MINDS dengan nama aplikasi “MINDS Smart House” seperti pada gambar 8.



Gambar 17 Logo Aplikasi MINDS: Smart House

Berikut adalah *activity flowchart* pada aplikasi MINDS.



Gambar 18 Flowchart Hubungan antar *Activity* dalam Aplikasi Android MINDS

6.1 Halaman Utama (LoginActivity.java)



Gambar 19 Halaman Utama Aplikasi (LoginActivity.java)

Kode XML halaman ini adalah sebagai berikut (activity_login.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_login"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.example.billyausten.newfinal.LoginActivity"
    android:weightSum="1"
    android:background="@drawable/backblackr">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">
        <ImageView
            android:layout_width="300dp"
            android:layout_height="200dp"
            android:src="@drawable/mindsawal"
            android:layout_gravity="center"
            android:layout_marginTop="80dp"
            android:layout_marginBottom="30dp"/>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Sign In"
            android:layout_marginTop="40dp"
            android:textSize="30dp"
            android:id="@+id/bsignin"
            android:background="@drawable/signin"/>
        <TextView
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/tvor"
        android:gravity="center"
        android:text="-----or-----"
        android:textSize="20dp"
        android:layout_marginTop="5dp"
        android:layout_weight="0.00" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Sign Up"
        android:layout_marginTop="5dp"
        android:textSize="30dp"
        android:id="@+id-bsignup"
        android:background="@drawable/signin"/>
</LinearLayout>
</RelativeLayout>
```

Pada halaman pertama pada saat MINDS dijalankan, ditampilkan *ImageView* berupa logo dari aplikasi Rumah Cerdas dan nama aplikasi Rumah Cerdas, dua buah *Button* yaitu “SIGN IN” dan “SIGN UP” dan juga sebuah *TextView* untuk memberitahu pengguna untuk memilih salah satu diantara kedua buah *Button*. Pada halaman ini, hal pertama yang dilakukan saat *Activity* dibuka adalah pengecekan apakah sebelumnya pengguna sudah pernah *login* pada sesi sebelumnya dan belum melakukan *logout*. Jika iya, maka tampilan aplikasi akan langsung melompat ke *MainActivity.java*. Jika tidak, maka *Activity* inilah yang akan tetap dipertahankan di layar.

Pada *Activity* ini, hanya digunakan *Intent* ketika *Button* ditekan yang berfungsi untuk mengubah tampilan dari satu *Activity* ke *Activity* lain. *Button* “SIGN IN” diberikan fungsi *SetOnClickListener* agar dapat melakukan *trigger* terhadap *Intent* sehingga dapat digunakan pengguna untuk berpindah ke *SignInActivity.java* sedangkan *Button* “SIGN UP” menggunakan prinsip yang sama untuk berpindah ke *SignUpActivity.java*.

Kode java halaman ini adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.graphics.Typeface;
7 import android.os.StrictMode;
8 import android.support.v7.app.AppCompatActivity;
9 import android.os.Bundle;
10 import android.view.View;
11 import android.view.WindowManager;
12 import android.widget.Button;
```

```

13 import android.widget.TextView;
14 import android.widget.Toast;
15
16 import com.rabbitmq.client.AMQP;
17 import com.rabbitmq.client.Channel;
18 import com.rabbitmq.client.Connection;
19 import com.rabbitmq.client.ConnectionFactory;
20 import com.rabbitmq.client.DefaultConsumer;
21 import com.rabbitmq.client.Envelope;
22
23 import org.json.simple.JSONObject;
24
25 import java.io.IOException;
26 import java.io.StringWriter;
27 import java.util.UUID;
28 import java.util.concurrent.ArrayBlockingQueue;
29 import java.util.concurrent.BlockingQueue;
30
31 public class LoginActivity extends AppCompatActivity {
32
33     @Override
34     protected void onCreate(Bundle savedInstanceState) {
35         super.onCreate(savedInstanceState);
36         setContentView(R.layout.activity_login);
37
38         StrictMode.ThreadPolicy policy = new
39         StrictMode.ThreadPolicy.Builder().permitAll().build();
40         StrictMode.setThreadPolicy(policy);
41
42         //INI ADALAH ACTIVITY YANG AKAN TERBUKA PERTAMA KALI
43
44         //MEMERIKSA APAKAH HOMEID SUDAH TERDEFINISI
45         //BILA SUDAH, MAKA HALAMAN INI TIDAK AKAN TERBUKA, NAMUN
46         //LANGSUNG MELOMPAT KE MAINACTIVITY (DASHBOARD)
47         try {
48             SharedPreferences sharedPref =
49                 getSharedPreferences("Authentication", MODE_PRIVATE);
50             Intent mainIntent = new
51                 Intent(LoginActivity.this,MainActivity.class);
52             if (sharedPref.getString("homeid","", "").length() == 6)
53             {
54                 //memindahkan activity ke MainActivity
55                 LoginActivity.this.startActivity(mainIntent);
56             }
57         } catch (Exception e) {
58
59             //PENGATURAN FONT
60             Typeface font_signin =
61                 Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
62             Typeface font_or = Typeface.createFromAsset(getAssets(),
63                 "fonts/Quicksand-BoldItalic.otf");
64
65             //BUTTON UNTUK MASUK KE HALAMAN SIGN IN
66             Button bsignin = (Button) findViewById(R.id.bsignin);
67             bsignin.setTypeface(font_signin);
68             bsignin.setOnClickListener(new View.OnClickListener() {

```

```

64
65             @Override
66             public void onClick(View v) {
67                 Intent afterloginIntent = new
Intent(LoginActivity.this, SignInActivity.class);
68
LoginActivity.this.startActivity(afterloginIntent);
69             }
70
71         });
72
73         //TEXTVIEW TULISAN "OR"
74         TextView tvor = (TextView) findViewById(R.id.tvor);
75         tvor.setTypeface(font_or);
76
77         //BUTTON UNTUK MASUK KE HALAMAN SIGN UP
78         Button bsignup = (Button) findViewById(R.id.signup);
79         bsignup.setTypeface(font_signin);
80         bsignup.setOnClickListener(new View.OnClickListener() {
81
82             @Override
83             public void onClick(View v) {
84                 Intent afterloginIntent = new
Intent(LoginActivity.this, SignUpActivity.class);
85
LoginActivity.this.startActivity(afterloginIntent);
86             }
87
88         });
89
90     }
91
92     public void onBackPressed() {
93         moveTaskToBack(true);
94     }
95
96 }

```

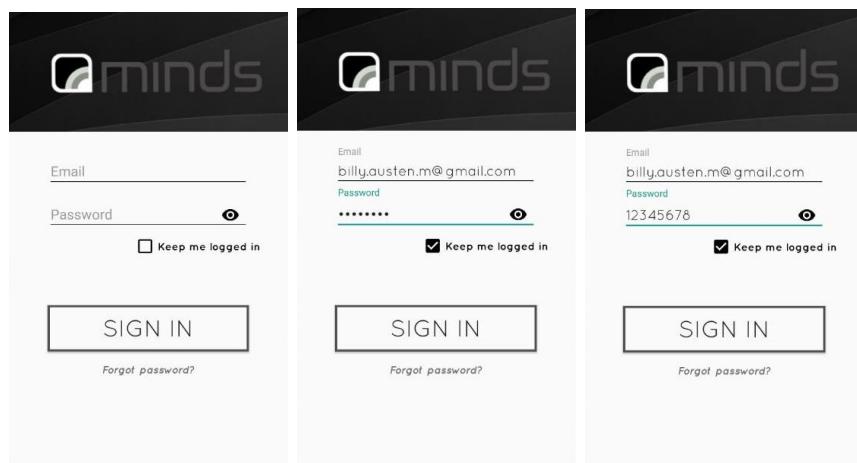
Kode Java dalam suatu *activity* pasti minimal memiliki suatu bagian fungsi bernama *OnCreate*. Fungsi inilah yang akan dijalankan ketika *activity* dibuka. Pada halaman pertama ini, ada beberapa bagian penting dari *OnCreate*.

- Baris 45 sampai 54 merupakan bagian untuk memeriksa apakah di dalam memori ponsel telah tersimpan data user. Bila data user sudah pernah tersimpan di dalam memori, maka pembentukan halaman ini tidak akan dilanjutkan, melainkan langsung pindah ke halaman utama (*MainActivity*). Data user diperiksa dengan melihat apakah ada informasi *home ID* di dalam memori.
- Bagian-bagian berikutnya dari halaman ini hanyalah berupa pengaturan tombol-tombol. Ketika tombol SIGN IN ditekan, maka halaman akan berpindah ke halaman

Sign In. Ketika tombol SIGN UP ditekan, maka halaman akan berpindah ke halaman Sign Up.

- Pada baris ke 92, terdapat fungsi OnBackPressed. Fungsi ini berisi algoritma yang akan dijalankan ketika tombol *back* pada ponsel ditekan. Isi dari fungsi pada *activity* ini adalah moveTaskToBack (true), yang berarti aplikasi langsung diberakhir ketika tombol *back* ditekan.

6.2 Halaman Sign In (*SignInActivity.java*)



Gambar 20 Halaman Sign In (*SignInActivity.java*)

Kode XML halaman ini adalah sebagai berikut (*activity_signin.xml*).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_sign_in"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.example.billyausten.newfinal.SignInActivity"
    android:descendantFocusability="beforeDescendants"
    android:focusableInTouchMode="true">
    <LinearLayout
        android:layout_width="401dp"
        android:layout_height="160dp"
        android:background="@drawable/mindsawallr"></LinearLayout>
    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp">
```

```

<EditText
    android:layout_width="260dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:inputType="textEmailAddress"
    android:id="@+id/etEmail"
    android:hint="Email"
    android:backgroundTint="#000000"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"/>
</android.support.design.widget.TextInputLayout>
<android.support.design.widget.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:passwordToggleContentDescription="Test"

    app:passwordToggleDrawable="@drawable/ic_remove_red_eye_black_24dp"
    app:passwordToggleEnabled="true"
    app:passwordToggleTint="#000000"
    android:backgroundTint="#000000">
<EditText
    android:layout_width="260dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="15dp"
    android:layout_gravity="center"
    android:inputType="textPassword"
    android:id="@+id/etPassword"
    android:hint="Password"
    android:backgroundTint="#000000"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"/>
</android.support.design.widget.TextInputLayout>
<CheckBox
    android:text="Keep me logged in"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/cbremember"
    android:layout_marginLeft="160dp"
    android:textColorLink="@android:color/background_dark"
    android:textColorHighlight="@android:color/background_dark"
    android:buttonTint="#000000" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Sign In"
    android:layout_marginTop="60dp"
    android:textSize="30dp"
    android:id="@+id/bsignin"
    android:background="@drawable/signin" />

<TextView
    android:layout_width="match_parent"

```

```
        android:layout_height="wrap_content"
        android:id="@+id/tvForgot"
        android:gravity="center"
        android:text="Forgot password?"
        android:layout_marginTop="15dp"
        android:textSize="14dp" />
    </LinearLayout>
```

Halaman ini hanya dapat terbuka ketika pengguna menekan *Button* “SIGN IN” pada *LoginActivity.java*. Perlu diingat kembali bahwa halaman ini tidak akan muncul bila pengguna pernah *login* pada session sebelumnya dan *account* masih diingat oleh aplikasi (*remembered*).

Halaman ini merupakan pintu gerbang pengguna untuk menggunakan peralatan Rumah Cerdas. Pengguna diminta untuk memasukan *email* dan *password* sebagai identitas pengguna. Selain menggunakan *email* dan *password* yang tersimpan dalam *database* MINDS.

Pengguna dapat memasukan identitasnya melalui *EditText* “Email” dan “Password”. *InputType* yang digunakan pada *EditText* “Email” adalah “textEmailAddress” sehingga pada saat pengguna menekan *EditText*, *keyboard* yang muncul adalah *keyboard* khusus yang akan memudahkan pengguna dalam menuliskan *email* mereka. Penggunaan *InputType* tersebut juga membuat tombol *enter* pada *keyboard* pengguna berubah menjadi *next*, sehingga ketika pengguna telah selesai menuliskan *email* mereka, pengguna langsung dapat mengisi *EditText* “Password” tanpa harus melakukan *tapping* pada bagian *EditText* tersebut. *InputType* yang digunakan pada *EditText* “Password” adalah “textPassword”, dengan menggunakan *InputType* tersebut, *password* yang dimasukan oleh pengguna akan terlihat seperti lingkaran hitam sehingga tidak dapat dibaca oleh orang lain, tetapi dengan menggunakan fitur ini, pengguna sangat rentan untuk melakukan kesalahan dalam mengetikkan *password* mereka, oleh karena itu diberikan *toggle* berbentuk mata yang diberikan kode “PasswordToggleEnabled” dengan nilai *True* untuk menampilkan *password* yang sudah diketik sehingga dapat terbaca. Pada *SignInActivity.java* terdapat *setOnKeyListener* dan *KeyEvent.ACTION_DOWN* yang berfungsi untuk membaca tombol yang ditekan pada *keyboard*. *Switch case* dilakukan pada bagian ini sebagai fungsi dari pengambilan keputusan sehingga Case *KeyEvent.KEYCODE_ENTER* berfungsi untuk membaca ketika *keyboard enter* ditekan. Jika pengguna masih salah dalam memasukan *email* maupun *password*, akan terdapat peringatan berupa *Toast* dengan pesan “Wrong

Email or Password". Terdapat juga *TextView* "forgot password?" yang dapat digunakan untuk mengirimkan *password* MINDS pengguna ke alamat email pengguna ketika pengguna lupa akan *password*-nya. Terdapat juga *Checkbox* "Keep me logged in" agar pengguna tetap terdaftar dalam aplikasi walaupun aplikasi sudah ditutup. Pengguna hanya dapat *logout* dari aplikasi MINDS ketika pengguna melakukan *tapping* pada menu "Logout" yang akan dijelaskan pada *MainActivity.java*. Pada *file xml*, ditambahkan kode *android:focusableInTouchMode="true"* untuk menghindari fokus terhadap *EditText* secara otomatis ketika pengguna memasuki *Activity* baru yang mengandung *EditText*. Fokus terhadap *EditText* dapat menyebabkan *keyboard* akan otomatis tampil pada layar *smartphone* walau *EditText* tidak ditekan. Kode *android:focusableInTouchMode="true"* digunakan pada setiap *Activity* yang mengandung *EditText*.

Kode Java halaman ini adalah sebagai berikut.

```
1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.graphics.Typeface;
7 import android.os.StrictMode;
8 import android.support.v7.app.AppCompatActivity;
9 import android.os.Bundle;
10 import android.text.SpannableString;
11 import android.text.style.UnderlineSpan;
12 import android.view.KeyEvent;
13 import android.view.View;
14 import android.view.WindowManager;
15 import android.widget.Button;
16 import android.widget.CheckBox;
17 import android.widget.CompoundButton;
18 import android.widget.EditText;
19 import android.widget.TextView;
20 import android.widget.Toast;
21 import android.provider.Settings.Secure;
22
23 import com.rabbitmq.client.AMQP;
24 import com.rabbitmq.client.Channel;
25 import com.rabbitmq.client.Connection;
26 import com.rabbitmq.clientConnectionFactory;
27 import com.rabbitmq.client.DefaultConsumer;
28 import com.rabbitmq.client.Envelope;
29 import com.rabbitmq.client.*;
30
31 import org.json.simple.JSONObject;
32 import org.json.simple.parser.JSONParser;
33
34 import java.io.IOException;
35 import java.io.StringWriter;
36 import java.lang.reflect.Array;
```

```

37 import java.text.SimpleDateFormat;
38 import java.util.ArrayList;
39 import java.util.Arrays;
40 import java.util.Random;
41 import java.util.UUID;
42 import java.util.concurrent.ArrayBlockingQueue;
43 import java.util.concurrent.BlockingQueue;
44 import java.util.Date;
45 import java.util.Calendar;
46
47 import javax.crypto.Cipher;
48 import javax.crypto.KeyGenerator;
49 import javax.crypto.SecretKey;
50 import javax.crypto.spec.IvParameterSpec;
51 import javax.crypto.spec.SecretKeySpec;
52
53 import android.util.Base64;
54
55
56 public class SignInActivity extends AppCompatActivity {
57
58     public static String AESkey;
59     public static ConnectionFactory factory;
60     public static Connection connection;
61     public static Channel channel;
62     Sharedpreferences sharedPref;
63     Sharedpreferences.Editor prefEditor;
64     public static KeyGenerator keyGenerator;
65
66     @Override
67     protected void onCreate(Bundle savedInstanceState) {
68         super.onCreate(savedInstanceState);
69         setContentView(R.layout.activity_sign_in);
70
71         StrictMode.ThreadPolicy policy = new
72         StrictMode.ThreadPolicy.Builder().permitAll().build();
73         StrictMode.setThreadPolicy(policy);
74
75         //CONNECTION SET UP
76         try {
77             factory = new ConnectionFactory();
78             factory.setUri(MainActivity.server_uri);
79             connection = factory.newConnection();
80             channel = connection.createChannel();
81         } catch (Exception e) {
82             Toast.makeText(getApplicationContext(),"Cannot connect
83             to Server",Toast.LENGTH_SHORT).show();
84         }
85
86         //KEY GENERATOR SET UP
87         try {
88             keyGenerator = KeyGenerator.getInstance("AES");
89             keyGenerator.init(128);
90             AESkey = generateAESkey();
91             MainActivity.iv = new
92             IvParameterSpec(MainActivity.iv_string.getBytes());
93             MainActivity.skeySpec = new
94             SecretKeySpec(AESkey.getBytes("UTF-8"), "AES");

```

```

91         MainActivity.aescipher =
Cipher.getInstance("AES/CFB8/NoPadding");
92     } catch (Exception e) {}
93
94     //BUKA SALURAN KE MEMORY
95     sharedPref =
getSharedPreferences("Authentication", MODE_PRIVATE);
96     prefEditor = sharedPref.edit();
97
98     Button bsignin = (Button) findViewById(R.id.bsignin);
99     Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
100    bsignin.setTypeface(font_signin);
101
102    Typeface font_or = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-BoldItalic.otf");
103
104    final EditText etEmail = (EditText)
findViewById(R.id.etEmail);
105    etEmail.setTypeface(font_signin);
106
107    final EditText etPassword = (EditText)
findViewById(R.id.etPassword);
108    etPassword.setTypeface(font_signin);
109
110    TextView tvforgot = (TextView) findViewById(R.id.tvforgot);
111    tvforgot.setTypeface(font_or);
112
113    final CheckBox cbremember = (CheckBox)
findViewById(R.id.cbremember);
114    Typeface font_cb = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Bold.otf");
115    cbremember.setTypeface(font_cb);
116
117    tvforgot.setOnClickListener(new View.OnClickListener() {
118        @Override
119        public void onClick(View v) {
120            Intent forgetIntent = new
Intent(SignInActivity.this, ForgetActivity.class);
121            SignInActivity.this.startActivity(forgetIntent);
122
123        }
124    );
125
126    bsignin.setOnClickListener(new View.OnClickListener() {
127
128        @Override
129        public void onClick(View v) {
130
131            String response = rpc
(etEmail.getText().toString(), etPassword.getText().toString());
132
133            JSONParser parser = new JSONParser();
134            JSONObject jsonObject = new JSONObject();
135            String homeid;
136            String usname;
137            String profpic;
138            String privilege;

```

```

139             try {
140                 jsonObject = (JSONObject)
parser.parse(response);
141             } catch (Exception e) {}
143             homeid = (String)jsonObject.get("homeid");
145             if ((homeid.equals("0"))) {
147                 Toast.makeText(getApplicationContext(),"Invalid
Identity or Password", Toast.LENGTH_SHORT).show();
148             } else if ((homeid.equals("none"))) {
149
150                 uname = (String)jsonObject.get("name");
151                 profpic = (String)jsonObject.get("picture");
152
prefEditor.putString("email",etEmail.getText().toString());
153
prefEditor.putString("password",etPassword.getText().toString());
154
prefEditor.putString("phoneid",Secure.getString(getApplicationContext(),S
ecure.ANDROID_ID));
155                 prefEditor.putString("uname",uname);
156                 prefEditor.putString("profpic",profpic);
157                 prefEditor.putString("aeskey",AESkey);
158                 prefEditor.putBoolean("remember",
cbremember.isChecked());
159                 Calendar calendar = Calendar.getInstance();
160                 calendar.setTime(new Date());
161                 calendar.add(Calendar.DATE,1);
162                 prefEditor.putString("aeskeyexp", (new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss")).format(calendar.getTime()));
163                 prefEditor.apply();
164
//TUTUP KONEKSI
165             try {
166                 channel.close();
168                 connection.close();
169             } catch(Exception e) {}
170
171             //PINDAH ACTIVITY KE MAINACTIVITY
172             Intent mainIntent = new
Intent(SignInActivity.this,MainActivity.class);
173                 SignInActivity.this.startActivity(mainIntent);
174             } else {
175
176
177                 SharedPreferences sharedPref =
getSharedPreferences("Authentication",MODE_PRIVATE);
178                 SharedPreferences.Editor prefEditor =
sharedPref.edit();
179                 uname = (String)jsonObject.get("name");
180                 profpic = (String)jsonObject.get("picture");
181                 privilege =
(String)jsonObject.get("privilege");
182
183                 //MENGIRIM PESAN SIGN IN KEPADA HOST
184             try {

```

```

185                         AMQP.BasicProperties props = new
AMQP.BasicProperties
186                                 .Builder()
187                                 .correlationId(Secure.getString(get
ContentResolver(), Secure.ANDROID_ID))
188                                 .build();
189
190                         JSONObject obj = new JSONObject();
191                         obj.put("head", "SI");
192                         obj.put("aeskey", AESkey);
193
194                         obj.put("email", etEmail.getText().toString());
195                         obj.put("privilege", privilege);
196                         StringWriter toSent = new StringWriter();
197                         obj.writeJSONString(toSent);
198                         String message = toSent.toString();
199                         channel.basicPublish("", homeid+"RSA",
props, MainActivity.RSAencrypt(message).getBytes("UTF-8"));
200
201                     } catch (Exception e) {
202
Toast.makeText(getApplicationContext(),"gagal
kirim",Toast.LENGTH_SHORT).show();
203
204                     //MENYIMPAN SELURUH INFORMASI DI DALAM MEMORY
205
prefEditor.putString("email",etEmail.getText().toString());
206                     prefEditor.putString("homeid",homeid);
207
prefEditor.putString("password",etPassword.getText().toString());
208                     prefEditor.putString("username",username);
209                     prefEditor.putString("profpic",profpic);
210
prefEditor.putString("phoneid",Secure.getString(getContentResolver(), s
ecure.ANDROID_ID));
211                     prefEditor.putString("privilege",privilege);
212                     prefEditor.putBoolean("remember",
cbremember.isChecked());
213                     prefEditor.putString("aeskey", AESkey);
214                     Calendar calendar = Calendar.getInstance();
215                     calendar.setTime(new Date());
216                     calendar.add(Calendar.DATE,1);
217                     prefEditor.putString("aeskeyexp", (new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss")).format(calendar.getTime()));
218                     prefEditor.apply();
219
220                     //TUTUP KONEKSI
221                     try {
222                         channel.close();
223                         connection.close();
224                     } catch(Exception e) {}
225
226
227                     //PINDAH ACTIVITY KE MAINACTIVITY
228                     Intent mainIntent = new
Intent(SignInActivity.this,MainActivity.class);
229                     SignInActivity.this.startActivity(mainIntent);

```

```

230         }
231     }
232
233 }
234
235 }
236
237 public String rpc (String email, String password) {
238     try {
239         String replyQueueName =
channel.queueDeclare().getQueue ();
240
AMQP.BasicProperties props = new AMQP.BasicProperties
242             .Builder ()
243             .correlationId(Secure.getString(getApplicationContext(),
244             Secure.ANDROID_ID))
245             .replyTo(replyQueueName)
246             .build ();
247
JSONObject obj = new JSONObject ();
248
249     obj.put("head", "SI");
250     obj.put("email", email);
251     obj.put("password", password);
252     obj.put("aeskey", AESkey);
253
StringWriter toSent = new StringWriter ();
obj.writeJSONString(toSent);
String message = toSent.toString ();
channel.basicPublish("", "RSAqueue", props,
MainActivity.RSAencrypt(message).getBytes ("UTF-8"));
258
259     final BlockingQueue<String> response = new
ArrayBlockingQueue<>(1);
260
261     channel.basicConsume(replyQueueName, true, new
DefaultConsumer(channel) {
262         @Override
263         public void handleDelivery(String consumerTag,
Envelope envelope, AMQP.BasicProperties properties, byte[] body)
throws IOException {
264             if
(properties.getCorrelationId ().equals (Secure.getString (getApplicationContext(),
Secure.ANDROID_ID))) {
265                 response.offer (new String (body, "UTF-8"));
266
267             }
268         }
269     });
270     String res = MainActivity.AESdecrypt (response.take ());
271     return res;
272
273 } catch (Exception e) {
274     Toast.makeText(getApplicationContext (), "Can't connect
to the server\nCheck your connection", Toast.LENGTH_SHORT).show ();
275     return null;
276 }
277 }
```

```

278
279     public static String generateAESkey () {
280         try {
281             SecretKey secretKey = keyGenerator.generateKey();
282             return (Base64.encodeToString(secretKey.getEncoded(),
283                                         Base64.DEFAULT).substring(0,16));
284         } catch (Exception e) {
285             return null;
286         }
287     }
288
289     public void onBackPressed() {
290         super.onBackPressed();
291         try {
292             connection.close();
293         } catch (Exception e) {}
294     }
295
296     public void onDestroy() {
297         super.onDestroy();
298         try {
299             connection.close();
300         } catch (Exception e) {}
301     }

```

Ada beberapa bagian penting dalam kode di atas.

- Baris 75 sampai 82 merupakan bagian untuk *set up* koneksi ke RabbitMQ.
- Baris 84 sampai 92 merupakan bagian untuk generasi kunci AES. Kunci AES ini nantinya akan dikirimkan ke *server* dan *host* sebagai kunci enkripsi selama 1 hari.
- Baris 117 sampai 124 merupakan pengaturan tombol forgot password. Ketika tombol ini ditekan, maka halaman akan berpindah ke halaman ForgotPassword.
- Baris 126 sampai 233 merupakan bagian pengaturan tombol SIGN IN. Ketika *Button* “SIGN IN” ditekan, MINDS akan melakukan verifikasi terhadap *email* dan *password* pengguna di database. Untuk melakukan itu, MINDS mengirimkan pesan kepada server dengan skema RPC dalam bentuk pesan JSON. Pesan yang dikirimkan adalah:

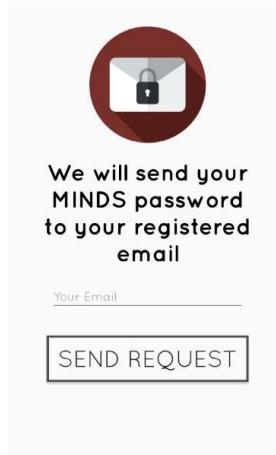
Tabel 20 Data JSON untuk Sign In

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
SI	-	<ul style="list-style-type: none"> • Email • Password • Kunci AES 	Server	RSA	<p>Jika email dan password sesuai dan pengguna sudah terdaftar di suatu rumah:</p> <ul style="list-style-type: none"> • Home ID • Nama pengguna • Gambar profil

					<ul style="list-style-type: none"> • Privilege pengguna (ADMIN atau Guest) • User ID <p>Jika email dan password tidak sesuai:</p> <ul style="list-style-type: none"> • Jawaban berupa karakter ‘0’ <p>Jika email dan password sesuai, tapi pengguna belum terdaftar di suatu rumah tertentu:</p> <p>Jawaban berupa string ‘none’</p>
--	--	--	--	--	---

- Dalam kode di atas juga terdapat fungsi rpc, yaitu fungsi untuk mengirimkan pesan SIGN IN ke server sekaligus menunggu jawabannya.
- Fungsi berikutnya adalah generateAESkey, yaitu fungsi untuk melakukan generasi kunci AES.

6.3 Forget Password Activity (*ForgetActivity.java*)



Gambar 21 Halaman Forget Password

Kode XML halaman ini adalah sebagai berikut (activity_forget.xml).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_forget"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="com.example.billyausten.newfinal.ForgetActivity"
```

```

        android:descendantFocusability="beforeDescendants"
        android:focusableInTouchMode="true">
    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:layout_gravity="center"
        android:layout_marginTop="20dp"
        android:src="@drawable/email"/>
    <TextView
        android:layout_width="300dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:textAlignment="center"
        android:text="We will send your MINDS password to your
registered email"
        android:textColor="#000000"
        android:textSize="30dp"
        android:layout_marginTop="20dp"
        android:id="@+id/tvforget"/>
    <EditText
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:id="@+id/etEmail"
        android:textSize="16dp"
        android:inputType="textEmailAddress"
        android:layout_marginTop="20dp"
        android:textColor="#000000"
        android:hint="Your Email" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Send Request"
        android:layout_marginTop="30dp"
        android:textSize="30dp"
        android:id="@+id/bsend"
        android:background="@drawable/signin"/>
</LinearLayout>

```

Dengan menekan *TextView* “Forget Password?” pada *SignInActivity.java* pengguna akan diarahkan pada *Activity* ini. *ForgetActivity.java* berfungsi untuk menerima input *email* pengguna yang lupa terhadap *password* MINDS-nya dan mengirim pesan tersebut kepada server ketika pengguna menekan *Button* “Send Request”. Setelah server menerima pesan tersebut, server akan mengirimkan *password* MINDS pengguna tersebut melalui *email* yang telah pengguna masukan melalui *EditText* yang tersedia. Ketika *email* yang dikirimkan tidak terdaftar pada server MINDS, server akan mengirimkan *email* yang menyatakan bahwa *email* yang dikirimkan tidak terdaftar pada server MINDS.

Kode java dari *activity* ini adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.graphics.Typeface;

```

```

4 import android.provider.Settings;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.WindowManager;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import com.rabbitmq.client.AMQP;
15 import com.rabbitmq.client.Channel;
16 import com.rabbitmq.client.Connection;
17 import com.rabbitmq.client.ConnectionFactory;
18 import com.rabbitmq.client.DefaultConsumer;
19 import com.rabbitmq.client.Envelope;
20
21 import org.json.simple.JSONObject;
22
23 import java.io.IOException;
24 import java.io.StringWriter;
25 import java.util.UUID;
26 import java.util.concurrent.ArrayBlockingQueue;
27 import java.util.concurrent.BlockingQueue;
28
29 import javax.crypto.Cipher;
30 import javax.crypto.spec.IvParameterSpec;
31 import javax.crypto.spec.SecretKeySpec;
32
33 public class ForgetActivity extends AppCompatActivity {
34
35     @Override
36     protected void onCreate(Bundle savedInstanceState) {
37         super.onCreate(savedInstanceState);
38         setContentView(R.layout.activity_forget);
39
40         //PENGATURAN FONT
41         Typeface font_signin =
42             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
43         Typeface font_title = Typeface.createFromAsset(getAssets(),
44             "fonts/Quicksand-Bold.otf");
45         Typeface font_or = Typeface.createFromAsset(getAssets(),
46             "fonts/Quicksand-BoldItalic.otf");
47
48         //TEXTVIEW KETERANGAN
49         TextView tvforget = (TextView) findViewById(R.id.tvforget);
50         tvforget.setTypeface(font_title);
51
52         //BUTTON UNTUK REQUEST RESET PASSWORD
53         Button bsend = (Button) findViewById(R.id.bsend);
54         bsend.setTypeface(font_signin);
55
56         //EDITTEXT UNTUK MENGISI EMAIL
57         final EditText etemail = (EditText)

```



```

99         Toast.makeText(getApplicationContext(),"The
email is not registered",Toast.LENGTH_SHORT).show();
100     }
101
102     } catch (Exception e) {
103         Toast.makeText(getApplicationContext(),"Can't
connect to the server\nCheck your
connection",Toast.LENGTH_SHORT).show();
104     }
105 }
106 }
107 }
108}

```

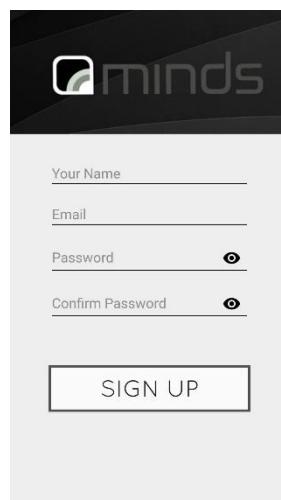
Bagian terpenting dari kode di atas adalah baris ke 58 sampai akhir, yaitu pengaturan penekanan tombol SEND. Ketika SEND ditekan, maka aplikasi akan membentuk data JSON yang berbentuk sebagai berikut.

Tabel 21 Data JSON Forget Password

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	forgetpwd	• Email	Server	RSA	'1' jika email terdaftar '0' jika email tidak terdaftar

Bila aplikasi menerima jawaban ‘0’, maka akan muncul pesan Toast “The email is not registered”. Bila menerima jawaban ‘1’, maka aplikasi akan menampilkan pesan Toast (“Sent”) dan password baru akan dikirimkan oleh server ke email pengguna.

6.4 Halaman Sign Up (*SignUpActivity.java*)



Gambar 22 Halaman Sign Up (*SignUpActivity.java*)

Kode XML halaman ini adalah sebagai berikut (activity_signup.xml).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_sign_up"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.example.billyausten.newfinal.SignUpActivity"
    android:descendantFocusability="beforeDescendants"
    android:focusableInTouchMode="true">

    <LinearLayout
        android:layout_width="401dp"
        android:layout_height="160dp"
        android:background="@drawable/mindsawallr"></LinearLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp">
        <EditText
            android:layout_marginTop="15dp"
            android:layout_width="260dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:inputType="textCapWords"
            android:id="@+id/etName"
            android:hint="Your Name"
            android:backgroundTint="#000000"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true"/>
    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        >
        <EditText
            android:layout_width="260dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:inputType="textEmailAddress"
            android:id="@+id/etEmail"
            android:hint="Email"
            android:backgroundTint="#000000"
            android:layout_alignParentLeft="true"
```

```

        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true"/>
    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:id="@+id/TILpass"
        android:layout_height="wrap_content"
        app:passwordToggleContentDescription="Test"

        app:passwordToggleDrawable="@drawable/ic_remove_red_eye_black_24dp"
        app:passwordToggleEnabled="true"
        app:passwordToggleTint="#000000"
        android:backgroundTint="#000000">

        <EditText
            android:layout_width="260dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:layout_gravity="center"
            android:inputType="textPassword"
            android:id="@+id/etPassword"
            android:hint="Password"
            android:backgroundTint="#000000"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true"/>
    </android.support.design.widget.TextInputLayout>

    <android.support.design.widget.TextInputLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:passwordToggleContentDescription="Test"

        app:passwordToggleDrawable="@drawable/ic_remove_red_eye_black_24dp"
        app:passwordToggleEnabled="true"
        app:passwordToggleTint="#000000"
        android:backgroundTint="#000000">

        <EditText
            android:layout_width="260dp"
            android:layout_height="wrap_content"
            android:layout_marginTop="15dp"
            android:layout_gravity="center"
            android:inputType="textPassword"
            android:id="@+id/etPasswordCon"
            android:hint="Confirm Password"
            android:backgroundTint="#000000"
            android:layout_alignParentLeft="true"
            android:layout_alignParentStart="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentEnd="true" />
    </android.support.design.widget.TextInputLayout>

    <Button
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Sign Up"
        android:layout_marginTop="55dp"
        android:textSize="30dp"
        android:id="@+id/bsignup"
        android:background="@drawable/signin"/>

    </LinearLayout>

```

Halaman ini digunakan untuk mendaftarkan *email* serta *password* yang digunakan untuk aplikasi MINDS. Untuk mendaftar, pengguna perlu memasukan data untuk proses identifikasi, data tersebut berupa nama, *email* dan *password* serta konfirmasi dari *password* yang sudah dimasukan sebelumnya untuk memastikan bahwa *password* yang diketikan benar sehingga ketika *password* dan konfirmasi *password* yang diketikan berbeda, akan terdapat *Toast* yang memperingatkan pengguna bahwa *password* dan konfirmasi *password* berbeda. Selain dengan menggunakan konfirmasi *password*, terdapat juga *toggle* yang dapat menampilkan *password* seperti yang sudah dijelaskan pada bagian sebelumnya untuk memastikan bahwa *password* yang dimasukan benar. Dengan menekan *Button* “SIGN UP” pengguna dapat mendaftarkan diri dengan menggunakan data yang sebelumnya sudah diisi. Ketika berhasil mendaftar, pengguna segera dialihkan menuju *SignInActivity.java* untuk memasukan *email* dan *password* untuk keperluan *Sign In*.

Pada *Activity* ini, terdapat empat buah *EditText* sebagai *input* untuk mendaftarkan diri kedalam server. *EditText* “Your Name” menggunakan *InputType* “textCapWords” agar ketika *input* dimasukkan, seluruh huruf pertama pada setiap kata merupakan huruf besar. *EditText* “Email” menggunakan *InputType* yang sama dengan *EditText* “Email” pada *SignInActivity.java* dan pada kedua *EditText* “Password” dan *EditText* “Confirm Password”, *InputType* yang digunakan adalah “textPassword” dengan tujuan yang sudah disampaikan sebelumnya, yaitu menutupi setiap karakter *password* agar tidak terbaca orang lain.

Kode Java activity ini adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.graphics.Typeface;
7 import android.os.StrictMode;
8 import android.provider.Settings;
9 import android.support.design.widget.TextInputLayout;
10 import android.support.v7.app.AppCompatActivity;
11 import android.os.Bundle;
12 import android.text.TextPaint;
13 import android.view.View;

```

```

14 import android.view.WindowManager;
15 import android.widget.Button;
16 import android.widget.CheckBox;
17 import android.widget.EditText;
18 import android.widget.TextView;
19 import android.widget.Toast;
20 import android.provider.Settings.Secure;
21
22 import com.rabbitmq.client.AMQP;
23 import com.rabbitmq.client.Channel;
24 import com.rabbitmq.client.Connection;
25 import com.rabbitmq.client.ConnectionFactory;
26 import com.rabbitmq.client.DefaultConsumer;
27 import com.rabbitmq.client.Envelope;
28
29 import org.json.simple.JSONObject;
30
31 import java.io.IOException;
32 import java.io.StringWriter;
33 import java.lang.reflect.Field;
34 import java.text.SimpleDateFormat;
35 import java.util.Calendar;
36 import java.util.Date;
37 import java.util.Random;
38 import java.util.UUID;
39 import java.util.concurrent.ArrayBlockingQueue;
40 import java.util.concurrent.BlockingQueue;
41
42 public class SignUpActivity extends AppCompatActivity {
43
44     String AESkey;
45
46     @Override
47     protected void onCreate(Bundle savedInstanceState) {
48         super.onCreate(savedInstanceState);
49         setContentView(R.layout.activity_sign_up);
50
51         StrictMode.ThreadPolicy policy = new
52         StrictMode.ThreadPolicy.Builder().permitAll().build();
53         StrictMode.setThreadPolicy(policy);
54
55         //PENGATURAN FONT
56         Typeface font_signin =
57         Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
58
59         //EDITTEXT UNTUK MEMASUKAN NAMA USER BARU
60         final EditText etName = (EditText)
61         findViewById(R.id.etName);
62         etName.setTypeface(font_signin);
63
64         //EDITTEXT UNTUK MEMASUKAN EMAIL
65         final EditText etEmail = (EditText)
66         findViewById(R.id.etEmail);
67         etEmail.setTypeface(font_signin);
68
69         //EDITTEXT UNTUK MEMASUKAN PASSWORD
70         final EditText etPassword = (EditText)
71         findViewById(R.id.etPassword);

```

```

67         etPassword.setTypeface(font_signin);
68
69         //EDITTEXT UNTUK MEMASUKAN PASSWORD KONFIRMASI
70         final EditText etPasswordCon = (EditText)
findViewById(R.id.etPasswordCon);
71         etPasswordCon.setTypeface(font_signin);
72
73         //BUTTON UNTUK MELAKSANAKAN PROSES SIGN UP
74         Button bsignup = (Button) findViewBy
Id(R.id.bsignup);
75         bsignup.setTypeface(font_signin);
76         bsignup.setOnClickListener(new View.OnClickListener() {
77             @Override
78             public void onClick(View v) {
79                 //memeriksa apakah password dan password konfirmasi
sudah sesuai
80                 if
(etPassword.getText().toString().equals(etPasswordCon.getText().toStr
ing())) {
81                     //memeriksa apakah panjang nama user antara 1
sampai 20 karakter
82                     if (etName.length() > 0 && etName.length() <=
20) {
83                         String response =
rpc(etEmail.getText().toString(), etName.getText().toString(),
etPassword.getText().toString());
84                         //jika response '1', maka sign up berhasil,
email valid dan blm pernah didaftarkan sebelumnya
85                         if (response.equals("1")) {
86                             //menyimpan informasi dalam memori
SharedPreferences sharedPref =
getSharedPreferences("Authentication", MODE_PRIVATE);
88                             SharedPreferences.Editor prefEditor =
sharedPref.edit();
89                             prefEditor.putString("email",
etEmail.getText().toString());
90                             prefEditor.putString("password",
etPassword.getText().toString());
91                             prefEditor.putString("aeskey", AESkey);
92                             prefEditor.putString("usname",
etName.getText().toString());
93                             prefEditor.putString("phoneid",
Secure.getString(getApplicationContext(), Settings.Secure.ANDROID_ID));
94                             prefEditor.putString("profpic","none");
95                             prefEditor.apply();
96
97                             //masuk ke MainActivity
98                             Intent mainIntent = new
Intent(SignUpActivity.this, MainActivity.class);
99                             SignUpActivity.this.startActivity(mainIntent);
100
101
102                         } else if (response.equals("0")) {
103                             //jika response '0', maka email tidak
valid
104                             Toast.makeText(SignUpActivity.this,
"This email has been registered",
105                                         Toast.LENGTH_SHORT).show();

```

```

106
107
108
109
110
111
112
113
114
115
116
117
118 //FUNGSI UNTUK MENGIRIM PESAN SIGN UP DENGAN SKEMA RPC
119 public String rpc (String email, String name, String password)
{
120     try {
121         //SETUP KONEKSI
122         ConnectionFactory factory = new ConnectionFactory();
123         factory.setUri(MainActivity.server_uri);
124         Connection connection = factory.newConnection();
125         Channel channel = connection.createChannel();
126
127         String replyQueueName =
128         channel.queueDeclare().getQueue();
129
130         //properti pesan
131         AMQP.BasicProperties props = new AMQP.BasicProperties
132             .Builder()
133             .correlationId(Secure.getString(getApplicationContext(),
134             Settings.Secure.ANDROID_ID))
135             .replyTo(replyQueueName)
136             .build();
137
138         //pesan JSON
139         JSONObject obj = new JSONObject();
140         obj.put("head", "SU");
141         obj.put("email", email);
142         obj.put("name", name);
143         obj.put("password", password);
144         AESkey = generateAESkey();
145         obj.put("aeskey", AESkey);
146         StringWriter toSent = new StringWriter();
147         obj.writeJSONString(toSent);
148         String message = toSent.toString();
149         channel.basicPublish("", "RSAqueue", props,
150             MainActivity.RSAencrypt(message).getBytes("UTF-8"));
151
152         //menunggu balasan
153         final BlockingQueue<String> response = new
154         ArrayBlockingQueue<>(2);
155         channel.basicConsume(replyQueueName, true, new
156         DefaultConsumer(channel) {
157             @Override
158             public void handleDelivery(String consumerTag,
159             Envelope envelope, AMQP.BasicProperties properties, byte[] body)
160             throws IOException {

```

```

154             if
155                 (properties.getCorrelationId().equals(Secure.getString(getApplicationContext(), Settings.Secure.ANDROID_ID))) {
156                     response.offer(new String(body, "UTF-8"));
157                 }
158             });
159             //mengambil balasan
160             String res = MainActivity.RSAdecrypt(response.take());
161             channel.close();
162             connection.close();
163             return res;
164         } catch (Exception e) {
165             Toast.makeText(getApplicationContext(),"Can't connect
to the server\nCheck your connection",Toast.LENGTH_SHORT).show();
166             return null;
167         }
168     }
169 }
170
171 //FUNGSI UNTUK GENERATE AES KEY
172 public String generateAESkey () {
173     char[] chars =
"abcdefghijklmnopqrstuvwxyz123456".toCharArray();
174     StringBuilder sb = new StringBuilder();
175     Random random = new Random();
176     for (int i = 0; i < 16; i++) {
177         char c = chars[random.nextInt(chars.length)];
178         sb.append(c);
179     }
180     return sb.toString();
181 }
182

```

Bagian terpenting dari kode di atas adalah baris ke 76 sampai akhir. Ketika *Button* “SIGN UP” ditekan, maka aplikasi akan melakukan hal berikut.

- Memeriksa apakah password dan konfirmasinya sudah sama. Bila tidak sama, maka akan muncul pesan *Toast* “Passwords are not the same”
- Bila sudah sama, maka MINDS akan mengirimkan pesan JSON kepada server dengan skema RPC berupa data JSON sebagai berikut.

Tabel 22 Data JSON Sign Up

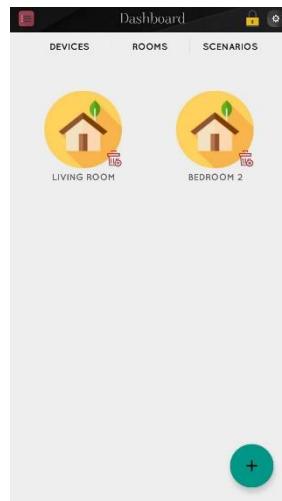
Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
SU (Sign Up)	-	<ul style="list-style-type: none"> • Nama <i>user</i> • Email • Password • Kunci AES 	Server	RSA	<p>Jika email valid (belum pernah didaftarkan sebelumnya):</p> <ul style="list-style-type: none"> • Jawaban berupa karakter ‘1’ <p>Jika email tidak valid (sudah terdaftar):</p> <p>Jawaban berupa karakter ‘0’</p>

- Ketika balasan ‘1’ diterima, maka MINDS akan memindahkan *Activity* ke *MainActivity.java*. Bila balasan ‘0’ diterima, maka akan muncul pesan *Toast* “Email has been registered”.

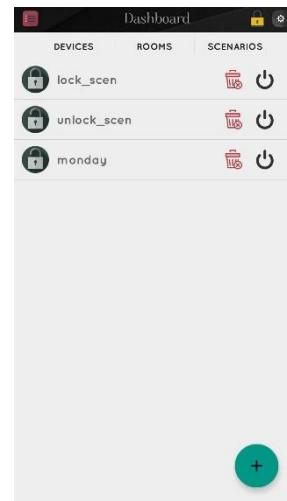
6.5 Halaman Menu Utama (*MainActivity.java*)



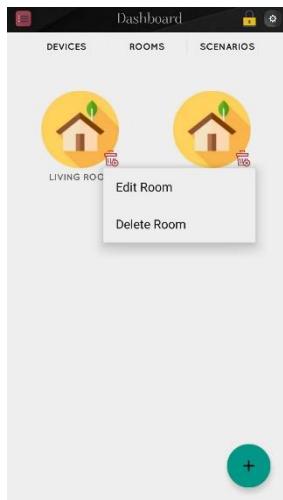
Gambar 23 Dashboard



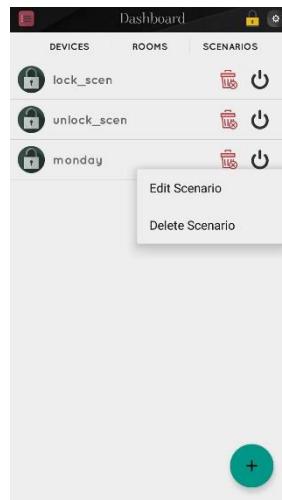
Gambar 24 Dashboard Room



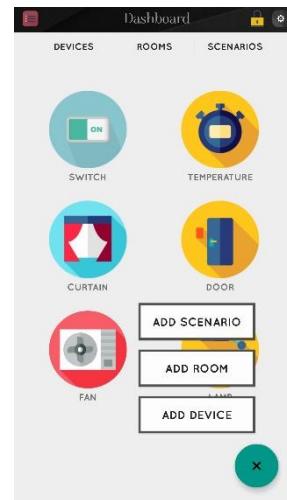
Gambar 25 Dashboard Scenario



Gambar 26 Popup Menu Trash pada Room



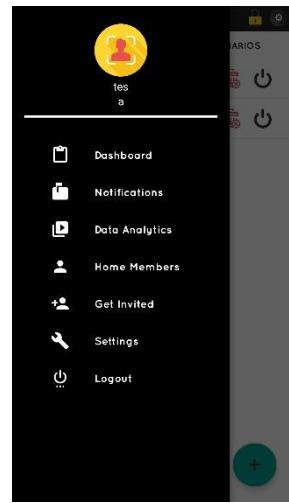
Gambar 27 Popup Menu Trash pada Scenario



Gambar 28 Dashboard Add Button



Gambar 29 Dashboard locked



Gambar 30 Navigation Drawer

Kode XML utama *activity* ini adalah sebagai berikut (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.DrawerLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:id="@+id/drawer_layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        tools:openDrawer="start">

    <include
        layout="@layout/app_bar_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <android.support.design.widget.NavigationView
        android:id="@+id/nav_view"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:fitsSystemWindows="true"
        app:headerLayout="@layout/nav_header_main"
        app:menu="@menu/activity_main_drawer"
        android:backgroundTint="#000000"
        app:itemTextColor="#FFFFFF"
        app:itemIconTint="#FFFFFF"
        android:theme="@style/NavigationViewPadding"/>

</android.support.v4.widget.DrawerLayout>
```

Kode XML untuk membentuk Tab adalah sebagai berikut (content_main.xml).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
```

```

xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/content_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="0dp"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.example.billyausten.newfinal.MainActivity"
    tools:showIn="@layout/app_bar_main">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <ImageView
                android:layout_width="30dp"
                android:layout_height="30dp"
                android:src="@drawable/burgermenur"
                android:id="@+id/burgermenu"
                android:layout_marginRight="30dp"
                android:layout_gravity="center"/>

            <TextView
                android:layout_width="236dp"
                android:layout_height="wrap_content"
                android:text="Dashboard"
                android:id="@+id/activitytitle"
                android:textAlignment="center"
                android:layout_gravity="center"
                android:textSize="18dp"
                android:textColor="#AAAAAA"/>

            <RelativeLayout
                android:layout_width="22dp"
                android:layout_height="22dp"
                android:layout_gravity="center">

                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"
                    android:src="@drawable/locked"
                    android:id="@+id/ivlocked"
                    android:visibility="visible"/>

                <ImageView
                    android:layout_width="match_parent"
                    android:layout_height="match_parent"

```

```

        android:id="@+id/ivunlocked"
        android:src="@drawable/unlocked"
        android:visibility="invisible"/>
    
```

```

</RelativeLayout>

<ImageView
    android:layout_width="22dp"
    android:layout_height="22dp"
    android:src="@drawable/settingmenur"
    android:id="@+id/settingmenu"
    android:layout_marginLeft="8dp"
    android:layout_gravity="center"/>

</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginTop="33dp"
    android:id="@+id/l1main">

<TabHost
    android:id="@+id/tabHost"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:layout_alignParentBottom="true"
    android:layout_alignParentTop="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <TabWidget
                android:id="@+id/tabs"
                android:layout_width="match_parent"
                android:layout_height="35dp"
                android:layout_gravity="center" />

        </LinearLayout>
    </LinearLayout>

```

```

<FrameLayout
    android:id="@+id/tabcontent"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

```

```

<LinearLayout
    android:id="@+id/tab2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <GridView
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/gridview2"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:columnWidth="150dp"
        android:numColumns="auto_fit"
        android:verticalSpacing="24dp"
        android:horizontalSpacing="0dp"
        android:paddingTop="40dp"
        android:paddingBottom="40dp"
        android:clipToPadding="false"
        android:scrollbars="none"
        android:stretchMode="spacingWidthUniform"
    />

</LinearLayout>

<LinearLayout
    android:id="@+id/tab3"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/listView">

    </ListView>

</LinearLayout>

<LinearLayout
    android:id="@+id/tab1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <GridView
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/gridview"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:gravity="center"
        android:columnWidth="150dp"
        android:numColumns="auto_fit"
        android:verticalSpacing="24dp"
        android:horizontalSpacing="0dp"
        android:paddingTop="40dp"

```

```

        android:paddingBottom="40dp"
        android:clipToPadding="false"
        android:scrollbars="none"
        android:stretchMode="spacingWidthUniform"
    />

    </LinearLayout>
</FrameLayout>
</LinearLayout>
</TabHost>

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="33dp"
    android:id="@+id/l1lockback"
    android:background="@drawable/backlock"
    android:orientation="vertical"
    android:visibility="invisible">

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="33dp"
    android:id="@+id/l1lock"
    android:orientation="vertical"
    android:visibility="invisible">

<ImageView
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_gravity="center"
    android:src="@drawable/lockiconr"
    android:layout_marginTop="100dp"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:id="@+id/tvlocked"
    android:textSize="25dp"
    android:text="Your device is being locked,"
    android:layout_marginTop="20dp"
    android:textColor="#000000"/>

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAlignment="center"
    android:id="@+id/tvlocked2"
    android:textSize="25dp"
    android:layout_marginTop="5dp"
    android:text="Please unlock your device"
    android:textColor="#000000"/>

```

```
</LinearLayout>  
</RelativeLayout>
```

Kode XML untuk membentuk *navigation drawer* adalah sebagai berikut (nav_header_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
        xmlns:app="http://schemas.android.com/apk/res-auto"  
        android:layout_width="match_parent"  
        android:layout_height="@dimen/nav_header_height"  
        android:background="#000000"  
        android:gravity="bottom"  
        android:orientation="vertical"  
        android:paddingBottom="@dimen/activity_vertical_margin"  
        android:paddingLeft="@dimen/activity_horizontal_margin"  
        android:paddingRight="@dimen/activity_horizontal_margin"  
        android:paddingTop="@dimen/activity_vertical_margin"  
        android:theme="@style/ThemeOverlay.AppCompat.Dark">  
    <ImageView  
        android:id="@+id/ivProfilePict"  
        android:layout_width="68dp"  
        android:layout_height="68dp"  
        android:layout_marginTop="10dp"  
        android:layout_gravity="center_horizontal"  
        android:src="@drawable/billy"/>  
  
    <TextView  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:paddingTop="3dp"  
        android:text="Billy Austen"  
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
        android:id="@+id/userName"/>  
  
    <TextView  
        android:id="@+id/userEmail"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:paddingTop="0dp"  
        android:paddingBottom="10dp"  
        android:text="billy.austen.m@gmail.com"  
        android:textAppearance="@style/TextAppearance.AppCompat.Body1"  
    />  
  
    <ImageView  
        android:layout_width="match_parent"  
        android:layout_height="3dp"  
        android:background="@drawable/pembatasdrawer"  
        android:layout_gravity="bottom|end"/>  
  
</LinearLayout>
```

MainActivity.java merupakan *Activity* utama dalam aplikasi ini karena dengan mengakses halaman ini, pengguna dapat menggunakan berbagai fitur utama Rumah Cerdas. Pengguna dapat mengakses seluruh *device* rumah cerdas yang dikelompokan berdasarkan jenis *device*, lokasi *device* maupun skenario aktivasi *device*.

Halaman ini terdiri dari tiga buah *tab*. Tab pertama adalah *devices*, berisi tombol-tombol untuk pengelompokan perangkat berdasarkan tipe perangkat. Ada 6 tipe perangkat, yaitu *switch*, *curtain*, *fan*, *lamp*, *temperature*, dan *door*. Ketika salah satu dari tombol tersebut ditekan, maka halaman akan berpindah ke halaman *device* (dijelaskan pada subbab 4.6). Tab kedua adalah *rooms*, yaitu berisi tombol pengelompokan perangkat berdasarkan ruangan. Pembagian ruangan dibebaskan kepada *user* untuk mengatur. Tab ketiga merupakan daftar *scenario*. Pada halaman ini *user* dapat melakukan manajemen *scenario* dan mengaktifkan suatu skenario tertentu.

Pada bagian atas, terdapat beberapa tombol. Di bagian kiri terdapat tombol burger untuk mengeluarkan *navigation drawer*. Di bagian kanan terdapat tombol kunci dan tombol *setting*. Tombol kunci digunakan untuk mengaktifkan fitur *lock*. Tombol *setting* digunakan untuk memindahkan halaman ke *setting*.

Navigation drawer berisi beberapa menu berupa tombol yang digunakan untuk memindahkan halaman ke masing-masing fungsi menu. Fungsi masing-masing menu akan dijelaskan di bagian penjelasan kode java.

Karena ini merupakan halaman utama, maka kode program untuk halaman ini cukup panjang sehingga perlu dibagi menjadi beberapa bagian. Bagian pertama kode java activity ini adalah sebagai berikut.

```
1 package com.example.billyausten.newfinal;
2
3 import android.app.ActivityManager;
4 import android.app.PendingIntent;
5 import android.content.ActivityNotFoundException;
6 import android.content.Context;
7 import android.content.Intent;
8 import android.graphics.Typeface;
9 import android.os.Bundle;
10 import android.os.Environment;
11 import android.os.Handler;
12 import android.os.Message;
13 import android.os.StrictMode;
14 import android.provider.Settings;
15 import android.support.design.widget.FloatingActionButton;
16 import android.support.design.widget.Snackbar;
```

```

17 import android.support.v7.widget.LinearLayoutCompat;
18 import android.support.v7.widget.PopupMenu;
19 import android.text.Spannable;
20 import android.text.SpannableString;
21 import android.util.Log;
22 import android.view.SubMenu;
23 import android.view.View;
24 import android.support.design.widget.NavigationView;
25 import android.support.v4.view.GravityCompat;
26 import android.support.v4.widget.DrawerLayout;
27 import android.support.v7.app.ActionBarDrawerToggle;
28 import android.support.v7.app.AppCompatActivity;
29 import android.support.v7.widget.Toolbar;
30 import android.view.Menu;
31 import android.view.MenuItem;
32 import android.view.animation.AccelerateInterpolator;
33 import android.view.animation.Animation;
34 import android.view.animation.AnimationUtils;
35 import android.view.animation.TranslateAnimation;
36 import android.widget.AdapterView;
37 import android.widget.ArrayAdapter;
38 import android.widget.BaseAdapter;
39 import android.widget.Button;
40 import android.widget.GridView;
41 import android.widget.LinearLayout;
42 import android.widget.ListView;
43 import android.widget.RelativeLayout;
44 import android.widget.TabHost;
45 import android.widget.TabWidget;
46 import android.widget.TextView;
47 import android.content.SharedPreferences;
48 import android.app.AlertDialog;
49 import android.content.DialogInterface;
50 import java.io.File;
51 import java.io.FileNotFoundException;
52 import java.io.IOException;
53 import java.io.StringWriter;
54 import java.net.URISyntaxException;
55 import java.security.KeyManagementException;
56 import java.security.NoSuchAlgorithmException;
57 import java.text.SimpleDateFormat;
58 import java.util.ArrayList;
59 import java.util.Calendar;
60 import java.util.UUID;
61 import java.util.concurrent.ArrayBlockingQueue;
62 import java.util.concurrent.BlockingQueue;
63 import java.util.Date;
64 import android.app.Activity;
65 import android.content.Intent;
66 import android.graphics.Bitmap;
67 import android.graphics.BitmapFactory;
68 import android.net.Uri;
69 import android.os.Bundle;
70 import android.view.View;
71 import android.widget.Button;
72 import android.widget.ImageView;
73 import android.widget.TextView;
74 import android.widget.Toast;

```

```

75 import com.rabbitmq.client.AMQP;
76 import com.rabbitmq.client.Channel;
77 import com.rabbitmq.client.Connection;
78 import com.rabbitmq.client.ConnectionFactory;
79 import com.rabbitmq.client.DefaultConsumer;
80 import com.rabbitmq.client.Envelope;
81 import com.rabbitmq.client.QueueingConsumer;
82 import com.rabbitmq.client.Consumer;
83 import org.json.simple.parser.JSONParser;
84 import org.json.simple.JSONObject;
85 import java.security.KeyPairGenerator;
86 import javax.crypto.Cipher;
87 import java.security.KeyPair;
88 import java.security.PrivateKey;
89 import java.security.PublicKey;
90 import android.util.Base64;
91 import java.security.spec.PKCS8EncodedKeySpec;
92 import java.security.spec.X509EncodedKeySpec;
93 import java.security.KeyFactory;
94 import javax.crypto.KeyGenerator;
95 import javax.crypto.SecretKey;
96 import javax.crypto.spec.SecretKeySpec;
97 import javax.crypto.spec.IvParameterSpec;
98 import java.util.Arrays;
99 import java.util.Random;
100
101
102 public class MainActivity extends AppCompatActivity
103     implements NavigationView.OnNavigationItemSelectedListener
{
104
105     boolean isOpen = false;
106     public static Animation
FabOpen,FabOpen1,FabOpen2,FabClose,FabRClockwise,FabRCClockwise,
Fablocked, Fabunlocked, Fabblocked, Fabbunklocked;
107
108     //Array list utama yang menyimpan daftar seluruh device
109     public static ArrayList<ArrayList<String>> mainList = new
ArrayList<>();
110
111     //url server
112     public static String server_uri =
"amqp://mindsTA:minds@167.205.24.10:5672/mindsTA132";
113
114     //informasi user
115     public static String email;
116     public static String homeid;
117     public static String homename;
118     String password;
119     public static String usname;
120     public static String phoneid;
121     public static String profpic;
122     public static String privilege;
123     public static boolean CurrentStatLock;
124     public static boolean CurrentStatNotif;
125     public static String AESkeyString;
126     public static String iv_string = "RandomInitVector";
127

```

```

128     //variabel untuk transmisi data
129     ConnectionFactory factory = new ConnectionFactory();
130     public static Connection connection;
131     public static Channel channel;
132     Thread subscribeThread;
133     JSONParser parser = new JSONParser();
134
135     //tampilan pada TAB 2 (rooms)
136     public static GridView gridView2;
137     public static MenuAdapter2 menuAdapter2;
138     public static ArrayList<String> menus2 = new ArrayList<>();
139
140     //floating button
141     Button AddDevice;
142     Button AddRoom;
143     Button AddScenario;
144     public static FloatingActionButton fab;
145
146     //tampilan pada TAB 3 (scenarios)
147     public static ListView listView3;
148     public static ListScenario list_scenario;
149     public static ArrayList<String> menus3 = new
ArrayList<String>();
150
151     //variabel untuk profpic
152     Uri image_uri;
153     public static ImageView ivprofpic;
154
155     //variabel untuk AES
156     public static String privkey_s =
"MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggsjAgEAAoIBAQCQd0CaamKDmTB5ulbre
/SLcrl3zE14I804hXpVJNF8SBEXvm0xLcAPGRk1hJS18dKSx+PFd3nboTUga/WqBzdqlVf
gjobdBsRkWf/o2miG2gB7zkK1STpD7CGYBzqTBgv1A1g78bcjNqAN1oLwC1Dwgnol7NocN
43k7J3i5uqXW/rkgIB/Rrd2ouBMWe0GsOUJaQgnd+7bPaF1Aa3dij4/xjnAkTdlekJzStx
7qsx173ziRCrU+WU1zCXGHfUqSzH2AH5BWvcng6i3Hr8vu2rZxChm/BUkUn6C1a9JHBFxj
6wsg+Mxi4zL46RDaHy5kgm+jH48HMmEJ0k/ERtIod65AgMBAAEcggEAdm69MJosfxQz8Gc
54L1y4ms/7x0z0eEIEBqlCm4vqsJt+NgtLhcCj54U05IIVAyWg6yJZxYmeU50fkrtT+uwm
5kpPLe7ija1P0R/DocXEA5iPRz/9/7LXGmVtMugUm2vmcKPMD1d1OlwV0fXejDcvv9IYhU
FTgKhOd5dJW0HqeNzBMXuecYvfecW1c2TgW1FqvHyGhTEfztCxyafH2UtbtNW3neq60IfB
5gnTmB3tuUFHz8KgmSgVBRBalfB7lpu9miJ7sjs5GZJYJiQvoReSOgj160cyexnubeU0d4
YaJhzmtHOZywiNfkA9cHexHmxylqzQs8uXdF5VKqdGtO+nQKBgQDN28fb7TmIty7yHYNE9
hA7CNetih03UWrusbPkJZgkDYX6zL/IFjUwNURqaXOGeCEGBjx9TpiguAb11G1fxnZRQzb
pp8PKU0HL7+qRxu15q8+z7GwOHLLov7W1kCFdcrl1o6YHRZxj/nHNH018hPC0k9kT070X
SOCoUTMPQxouwKBgQCzp1tiaYmq0tfK/sYRk8t9bKZ5dRNHLJYE1QcdkBJ67B0e/HCck20
KWFHgllsZd3HTEfcWCNzUPXHy98vYuxQf4+uCWphrlwTSPShPlMFCPCYYe7M363aPR1dhU
2aN/HXA2H+No9GvaQL5d6MC2ZQPgLo7d4yLSbtoKtLQWMjJGwKBgCVMJr/gEUaF+g1/82K
xOmuxxIv6s51GNRueMrupXBFcKeQw6KZ175kWbsZd6MJ6TXFr+9WwQc5D/Z7ND6wEpzc9+
FsJPf4SC5/JjjQ8WmewrhuxQb5t286tb4K50uNzXhyAjUFeM8zLizPefLY9T/tCY4QHIoT
/wwlN8ssWXIM1AoGBAJ9aHBnvyPfnqxdY4zFmKgInpIZpTV89chfAxtpj17DGzJHWksCfm
AGsE3wmZNzy8Niqv8Mgu1fa5L8CAyNL1W7wCM8IMtiEDPe63Y2xQbbRiW0AFBj018wAnbO
TeY0J2RzkLEfjfjOb7tJ+xu3ZNpaFtSrYQKawEPY1veamSfrvAoGAGR1YL3sOj7L32o9yb
bupV8q35U463Arc9RhHqA/fn436s0+VPYee43tiFk8P+hQ8GAYJtEh8rVmGGqHiQ9SsXW
SmBEpvDBf5mnC47J8x3PUBcU4YXK9HaziOI8mUuyJxJSV10gNe1cELCrMw4mmy35SZRnEM
6VjYytihpabP/Q=";
157     public static String pubkey_s =
"MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEakHdAmmpig5kwebpW63v0i3K5d
8xJeCPDuIV6VSTRfEgRF75tMS3ADxkZNYsUtfHSksfjxXd526E1IGv1qgc3apvx4I6G3QU
qyln/6NpohtoAe85CpUk6Q+whmA6kwYL9QNYO/G3IzagDdaC8AtQ8IJ6JezaHDeN5Oyd4

```

```

ubql1v65ICAF0a3dqLgTFntBrDlCWkIJ3fu2z2hZQGt3Yo+P8Y5wJE3ZXpPc0rce6rMZe9
84kQq1P11Ncw1xh31Kksx9gB+QVr3J4Ootx6/L7tq2cQoZvwVJFJ+gpWwSRwRcY+sLIPjM
YuMy+OkQ2h8uZIJvox+PBzJhCdJPxEbSKHeuQIDAQAB";
158     public static PublicKey pubkey = loadPublicKey(pubkey_s);
159     public static PrivateKey privkey = loadPrivateKey(privkey_s);
160
161     //variabel untuk service
162     public static SensorService mSensorService;
163     public static Intent mServiceIntent;
164
165     //shared preference (penyimpanan)
166     SharedPreferences sharedPref;
167     SharedPreferences.Editor prefEditor;
168
169     //variabel enkripsi AES
170     public static IvParameterSpec iv;
171     public static SecretKeySpec skeySpec;
172     public static Cipher aescipher;
173
174     //variabel tampilan
175     public static ImageView ivlock;
176     public static ImageView ivunlock;
177     public static LinearLayout llback;
178     public static LinearLayout lllock;
179     public static ImageView burgermenu;
180     public static ImageView settingmenu;
181     public static TabWidget tw;
182     public static GridView gridView;
183     public static TabHost host;
184     public static DrawerLayout drawer;
185
186
187     @Override
188     protected void onCreate(Bundle savedInstanceState) {
189         super.onCreate(savedInstanceState);
190         setContentView(R.layout.activity_main);
191
192         StrictMode.ThreadPolicy policy = new
193         StrictMode.ThreadPolicy.Builder().permitAll().build();
194         StrictMode.setThreadPolicy(policy);
195
196         //PENGATURAN FONT
197         Typeface font_activity =
198         Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
199
200         //JUDUL ACTIVITY (TITLE BAR)
201         TextView activitytitle = (TextView)
202         findViewById(R.id.activitytitle);
203         activitytitle.setTypeface(font_activity);
204
205         //MENGAMBIL NILAI-NILAI IDENTITAS USER YANG TERSIMPAN
206         //DALAM MEMORI (SHARED PREFERENCE)
207         sharedPref =
208         getSharedPreferences("Authentication", MODE_PRIVATE);
209         prefEditor = sharedPref.edit();
210         email = sharedPref.getString("email","");
211         password = sharedPref.getString("password","");
212         homeid = sharedPref.getString("homeid","");

```

```

208     homename = sharedPref.getString("homename","");
209     uname = sharedPref.getString("uname","");
210     profpic = sharedPref.getString("profpic","none");
211     privilege = sharedPref.getString("privilege","");
212     phoneid = sharedPref.getString("phoneid","");
213     CurrentStatNotif = sharedPref.getBoolean("NotifStat",
214         true);
215     CurrentStatLock = sharedPref.getBoolean("LockStat", true);
216     AESkeyString = sharedPref.getString("aeskey", "");
217
218     //SETUP AES KEY
219     try {
220         iv = new IvParameterSpec(iv_string.getBytes());
221         skeySpec = new SecretKeySpec(AESkeyString.getBytes("UTF-8"), "AES");
222         aescipher = Cipher.getInstance("AES/CFB8/NoPadding");
223     } catch (Exception e) {
224
225     }
226
227     //SETUP RABBITMQ CONNECTION
228     try {
229         factory.setAutomaticRecoveryEnabled(false);
230         factory.setUri(server_uri);
231         connection = factory.newConnection();
232         channel = connection.createChannel();
233     } catch (Exception e) {
234         Toast.makeText(getApplicationContext(),"Failed to
Connect",
235             Toast.LENGTH_SHORT).show();
236     }

```

Bagian pertama ini berisi *library* yang digunakan dalam kode, deklarasi variabel, dan bagian persiapan. Bagian persiapan terdiri dari:

- Pengaturan font
- Proses *loading* nilai-nilai yang ada di dalam memori. Nilai yang dimaksud adalah profil pengguna, misalnya email, password, ID rumah, ID ponsel, dll.
- *Set up* enkriptor dan dekriptor AES berdasarkan kunci yang telah diperoleh dari memori
- Membangun koneksi ke RabbitMQ server

Bagian kedua adalah sebagai berikut.

```

237     //SETUP LAYOUT
238     Toolbar toolbar = (Toolbar)
findViewById(R.id.action_settings);
239     setSupportActionBar(toolbar);
240
241     host = (TabHost) findViewById(R.id.tabHost);
242     host.setup();
243
244     //Tab 1 (DEVICES)
245     TabHost.TabSpec spec = host.newTabSpec("Devices");

```

```

246         spec.setContent(R.id.tab1);
247         spec.setIndicator("Devices");
248         host.addTab(spec);
249         tw = (TabWidget)host.findViewById(android.R.id.tabs);
250         View tabView1 = tw.getChildTabViewAt(0);
251         TextView tv1 =
252             (TextView)tabView1.findViewById(android.R.id.title);
253             Typeface font_title =
254             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
255             tv1.setTypeface(font_title);
256             tv1.setTextSize(12);
257
258             //Gridview untuk tipe-tipe device
259             gridView = (GridView) findViewById(R.id.gridview);
260             MenuAdapter menuAdapter = new MenuAdapter(this,menus);
261             gridView.setAdapter(menuAdapter);
262
263             gridView.setOnItemClickListener(new
264                 AdapterView.OnItemClickListener() {
265                     @Override
266                     public void onItemClick(AdapterView<?> parent, View
267                     view,
268                         int position, long id) {
269                         Intent deviceIntent = new
270                             Intent(MainActivity.this,SwitchActivity.class);
271                         deviceIntent.putExtra("base","type");
272
273                         switch (position) {
274                             case 0:
275                                 deviceIntent.putExtra("group","switch");
276
277                         MainActivity.this.startActivity(deviceIntent);
278                             break;
279                         case 1:
280
281                         deviceIntent.putExtra("group","temperature");
282
283                         MainActivity.this.startActivity(deviceIntent);
284                             break;
285                         case 2:
286                                 deviceIntent.putExtra("group","curtain");
287
288                         MainActivity.this.startActivity(deviceIntent);
289                             break;
290                         case 3:
291                                 deviceIntent.putExtra("group","door");
292
293                         MainActivity.this.startActivity(deviceIntent);
294                             break;
295                         case 4:
296                                 deviceIntent.putExtra("group","fan");
297
298                         MainActivity.this.startActivity(deviceIntent);
299                             break;
300                         case 5:
301                                 deviceIntent.putExtra("group","lamp");
302
303                         MainActivity.this.startActivity(deviceIntent);
304

```

```

292                     break;
293                 }
294             }
295         );
296     );
297
298     //Tab 2 (ROOMS)
299     spec = host.newTabSpec("Rooms");
300     spec.setContent(R.id.tab2);
301     spec.setIndicator("Rooms");
302     host.addTab(spec);
303     View tabView2 = tw.getChildTabViewAt(1);
304     TextView tv2 =
305     (TextView)tabView2.findViewById(android.R.id.title);
306     tv2.setTypeface(font_title);
307     tv2.setTextSize(12);
308
309     gridView2 = (GridView) findViewById(R.id.gridview2);
310     menuAdapter2 = new MenuAdapter2(this,menus2);
311     gridView2.setAdapter(menuAdapter2);
312
313     gridView2.setOnItemClickListener(new
314     AdapterView.OnItemClickListener() {
315         @Override
316         public void onItemClick(AdapterView<?> parent, View
317         view,
318             int position, long id) {
319
320             Intent deviceIntent = new
321             Intent(MainActivity.this,SwitchActivity.class);
322             deviceIntent.putExtra("base","room");
323             deviceIntent.putExtra("group",
324             menus2.get(position));
325             MainActivity.this.startActivity(deviceIntent);
326         }
327     });
328
329     //Tab 3 (SCENARIOS)
330     spec = host.newTabSpec("Scenarios");
331     spec.setContent(R.id.tab3);
332     spec.setIndicator("Scenarios");
333     host.addTab(spec);
334     View tabView3 = tw.getChildTabViewAt(2);
335     TextView tv3 =
336     (TextView)tabView3.findViewById(android.R.id.title);
337     tv3.setTypeface(font_title);
338     tv3.setTextSize(12);
339
340     list_scenario = new ListScenario(this, menus3);
341
342     listView3 = (ListView) findViewById(R.id.listView);
343     listView3.setAdapter(list_scenario);
344
345     host.setOnTabChangedListener(new
346     AnimatedTabHostListener(host));

```

Bagian kedua ini merupakan bagian persiapan untuk pengaturan tampilan, terutama pengaturan *tab*. Halaman ini terdiri dari tiga tab seperti terlihat pada gambar.

- Tab pertama adalah tab *devices*. Pada saat pengguna membuka *tab* ini, pengguna dapat melihat enam buah menu yang dikelompokan berdasarkan jenis *device* yaitu “SWITCH”, “TEMPERATURE”, “CURTAIN”, “DOOR”, “FAN”, dan “LAMP” sehingga ketika pengguna menekan salah satu menu dalam tampilan tersebut, pengguna dapat mengakses *Activity* yang mengatur *device* yang sesuai dengan menu tersebut. Keenam tombol ini diletakan pada *layout* berjenis GridView yang bentuknya berupa susunan 6 gambar dari atas ke bawah.
- Ketika pengguna menekan “ROOMS” pada *TabBar* di bagian atas *Activity*, pengguna dapat melihat menu yang ditampilkan berdasarkan lokasi setiap *device*. Ketika salah satu menu ruangan ditekan, akan ditampilkan *list* dari *device-device* yang berada di ruangan tersebut. Pengguna juga dapat menekan *ImageView* bergambar tempat sampah yang terletak diujung kanan bawah dari setiap *GridView room* untuk menampilkan *popup menu edit* dan *delete*. Ketika *popup menu edit* ditekan, pengguna diarahkan menuju *Activity* dengan tampilan serupa dengan *EditRoomActivity.java* dengan konfigurasi ruangan seperti sebelumnya. Ketika *popup menu delete* ditekan, *room* yang pengguna pilih akan segera dihapus. Daftar room juga ditampilkan dalam *GridView*.
- *Tab* terakhir adalah “SCENARIOS”, pengguna dapat mengaktifkan berbagai skenario yang sebelumnya sudah didaftarkan pada menu “Add Scenario” dengan cara menekan *ImageView* bergambar *power*, pengguna juga dapat melakukan *edit* ataupun *delete* terhadap suatu menu *scenario* dengan cara menekan *ImageView* bergambar tempat sampah yang terletak disebelah kanan *ImageView* bergambar *power*. Ketika *ImageView* tersebut ditekan, *popup menu edit* dan *delete* akan segera ditampilkan. Ketika pengguna memilih *edit*, maka pengguna akan diarahkan menuju tampilan pada *AddScenarioActivity.java* dengan konfigurasi *scenario* seperti sebelumnya dan ketika pengguna menekan *popup menu delete*, *scenario* yang pengguna pilih akan dihapus. Daftar *scenario* ditampilkan dalam *ListView*, yaitu suatu *layout* yang bentuknya daftar menurun dari atas ke bawah.

Bagian ketiga adalah sebagai berikut.

342

// FLOATING BUTTON

```

343
344     AddDevice = (Button) findViewById(R.id.badddevice);
345     AddRoom = (Button) findViewById(R.id.baddrroom);
346     AddScenario = (Button) findViewById(R.id.baddscenario);
347     //animasi floating button
348     AddDevice.setTypeface(font_title);
349     AddRoom.setTypeface(font_title);
350     AddScenario.setTypeface(font_title);
351     FabOpen =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fab_open);
352     FabOpen1 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fab_open1)
;
353     FabOpen2 =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fab_open2)
;
354     FabClose =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.fab_close)
;
355     FabRClockwise =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.rotate_clockwise);
356     FabRCClockwise =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.rotate_couterclock);
357
358     //ketika floating button diklik:
359     fab = (FloatingActionButton) findViewById(R.id.fab);
360     fab.setOnClickListener(new View.OnClickListener() {
361         @Override
362         public void onClick(View view) {
363             if (isOpen)
364             {
365                 AddDevice.startAnimation(FabClose);
366                 AddRoom.startAnimation(FabClose);
367                 AddScenario.startAnimation(FabClose);
368                 fab.startAnimation(FabRCClockwise);
369                 AddDevice.setClickable(false);
370                 AddRoom.setClickable(false);
371                 AddScenario.setClickable(false);
372                 isOpen = false;
373             } else {
374                 AddDevice.startAnimation(FabOpen);
375                 AddRoom.startAnimation(FabOpen1);
376                 AddScenario.startAnimation(FabOpen2);
377                 fab.startAnimation(FabRClockwise);
378                 AddDevice.setClickable(true);
379                 AddRoom.setClickable(true);
380                 AddScenario.setClickable(true);
381                 isOpen = true;
382             }
383         }
384     });
385     //Tombol-tombol yang muncul setelah floating button
ditekan
386     AddDevice.setOnClickListener(new View.OnClickListener() {
387
388         @Override

```

```

389         public void onClick(View v) {
390             if (!homeid.equals("")) {
391                 Intent afterloginIntent = new
Intent(MainActivity.this, AddDeviceActivity.class);
392             MainActivity.this.startActivity(afterloginIntent);
393             } else
394                 Toast.makeText(getApplicationContext(), "You
are not registered to a home yet", Toast.LENGTH_SHORT).show();
395             }
396
397         });
398
399         AddRoom.setOnClickListener(new View.OnClickListener() {
400
401             @Override
402             public void onClick(View v) {
403                 if (!homeid.equals("")) {
404                     Intent afterloginIntent = new
Intent(MainActivity.this, EditRoomActivity.class);
405                     afterloginIntent.putExtra("type", "add");
406
MainActivity.this.startActivity(afterloginIntent);
407                     } else
408                     Toast.makeText(getApplicationContext(), "You
are not registered to a home yet", Toast.LENGTH_SHORT).show();
409                 }
410
411         });
412         AddScenario.setOnClickListener(new View.OnClickListener()
{
413
414             @Override
415             public void onClick(View v) {
416                 if (!homeid.equals("")) {
417                     Intent afterloginIntent = new
Intent(MainActivity.this, AddScenarioActivity.class);
418                     afterloginIntent.putExtra("type", "add");
419
MainActivity.this.startActivity(afterloginIntent);
420                     } else
421                     Toast.makeText(getApplicationContext(), "You
are not registered to a home yet", Toast.LENGTH_SHORT).show();
422                 }
423
424         });

```

Bagian ketiga merupakan pengaturan *floating button* yang berada di ujung kanan bawah halaman. *FloatingButton* ini berguna untuk menampilkan menu tambahan untuk menambah *device*, *room* maupun *scenario*. Ketika ditekan, menu “Add Device” akan menampilkan tampilan dari *AddDeviceActivity.java*, ketika menu “Add Room” ditekan, pengguna akan menuju ke *EditRoomActiviy.java* dan ketika menu “Add Scenario” ditekan, tampilan akan berubah menjadi tampilan dari *AddScenarioActivity.java*.

Bagian keempat adalah sebagai berikut.

```

427         //DRAWER (NAVIGATION VIEW)
428         drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
429         ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
430                 this, drawer, toolbar,
431                 R.string.navigation_drawer_open, R.string.navigation_drawer_close);
432         drawer.setDrawerListener(toggle);
433         toggle.syncState();
434
435         NavigationView navView = (NavigationView)
436         findViewById(R.id.nav_view);
437         navView.setNavigationItemSelectedListener(this);
438         Menu m = navView.getMenu();
439         for (int i=0;i<m.size();i++) {
440             MenuItem mi = m.getItem(i);
441             //for applying a font to subMenu ...
442             SubMenu subMenu = mi.getSubMenu();
443             if (subMenu!=null && subMenu.size() >0 ) {
444                 for (int j=0; j <subMenu.size();j++) {
445                     MenuItem subMenuItem = subMenu.getItem(j);
446                     applyFontToMenuItem(subMenuItem);
447                 }
448             }
449         }
450
451         //mengatur foto profil, nama, dan username pada drawer
452         View headerview = navView.getHeaderView(0);
453         TextView usname_drawer = (TextView)
454         headerview.findViewById(R.id.userName);
455         usname_drawer.setText(usname);
456         TextView email_drawer = (TextView)
457         headerview.findViewById(R.id.userEmail);
458         email_drawer.setText(email);
459         ivprofpic = (ImageView)
460         headerview.findViewById(R.id.ivProfilePict);
461         if (profpic.equals("none"))
462             ivprofpic.setImageResource(R.drawable.profdefault);
463         else
464             ivprofpic.setImageBitmap(decodeBase64(profpic));

```

Bagian ini merupakan pengaturan drawer yang akan muncul ketika bagian kiri halaman di-*slide*. Bagian ini mengatur tampilan nama, foto, dan email pada *navigation bar*.

Bagian kelima adalah sebagai berikut.

```

463         //TOMBOL BURGER DAN SETTING PADA TITLE BAR
464         burgermenu = (ImageView) findViewById(R.id.burgermenu);
465         burgermenu.setOnClickListener(new View.OnClickListener() {
466
467             @Override
468             public void onClick (View view){
469                 //Creating the instance of PopupMenu
470                 drawer.openDrawer(GravityCompat.START);
471             }
472
473         });
474         settingmenu = (ImageView) findViewById(R.id.settingmenu);

```

```

475     settingmenu.setOnClickListener(new View.OnClickListener() {
476
477         @Override
478         public void onClick (View view){
479             Intent afterloginIntent = new
Intent(MainActivity.this,SettingActivity.class);
480                 MainActivity.this.startActivity(afterloginIntent);
481             }
482
483         });
484
485         //TOMBOL LOCK DAN UNLOCK PADA TITLE BAR
486         ivlock = (ImageView) findViewById(R.id.ivlocked);
487         ivunlock = (ImageView) findViewById(R.id.ivunlock);
488         llback = (LinearLayout) findViewById(R.id.lllockback);
489         lllock = (LinearLayout) findViewById(R.id.lllock);
490         TextView tvlocked = (TextView)
findViewById(R.id.tvlocked);
491         TextView tvlocked2 = (TextView)
findViewById(R.id.tvlocked2);
492         tvlocked.setTypeface(font_title);
493         tvlocked2.setTypeface(font_title);
494         Fabbacklocked =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.backlocked
);
495         Fabbackunlock =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.backunlock
ed);
496         Fablocked =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.locked);
497         Fabunlock =
AnimationUtils.loadAnimation(getApplicationContext(),R.anim.unlocked);
498         //ketika tombol lock ditekan
499         ivlock.setOnClickListener(new View.OnClickListener() {
500             @Override
501             public void onClick(View v) {
502                 ivlock.setVisibility(View.INVISIBLE);
503                 ivunlock.setVisibility(View.VISIBLE);
504                 llback.startAnimation(Fabbacklocked);
505                 lllock.startAnimation(Fablocked);
506                 burgermenu.setClickable(false);
507                 settingmenu.setClickable(false);
508                 tw.setEnabled(false);
509                 fab.setClickable(false);
510                 gridView.setEnabled(false);
511                 gridView2.setEnabled(false);
512                 listView3.setEnabled(false);
513                 host.setCurrentTab(0);
514
drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
515             try {
516                 JSONObject obj = new JSONObject();
517                 obj.put("head", "SC");
518                 obj.put("homeid", homeid);
519                 obj.put("name", "lock_scen");
520                 StringWriter toSent = new StringWriter();
521                 obj.writeJSONString(toSent);
522                 String message = toSent.toString();

```

```

523             AMQP.BasicProperties props = new
AMQP.BasicProperties
524                     .Builder()
525                     .correlationId(MainActivity.phoneid)
526                     .build();
527             MainActivity.channel.basicPublish("", 
MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
528         } catch (Exception e) {
529             }
530         }
531     }
532 }
533 //ketika tombol unlock ditekan
534 ivunlock.setOnClickListener(new View.OnClickListener() {
535     @Override
536     public void onClick(View v) {
537         ivlock.setVisibility(View.VISIBLE);
538         ivunlock.setVisibility(View.INVISIBLE);
539         llback.startAnimation(Fabbackunlocked);
540         lllock.startAnimation(Fabunlocked);
541         burgermenu.setClickable(true);
542         settingmenu.setClickable(true);
543         tw.setEnabled(true);
544         fab.setClickable(true);
545         gridView.setEnabled(true);
546         gridView2.setEnabled(true);
547         listView3.setEnabled(true);
548
549
550     drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED);
551     try {
552         JSONObject obj = new JSONObject();
553         obj.put("head", "SC");
554         obj.put("homeid", homeid);
555         obj.put("name", "unlock_scen");
556         StringWriter toSent = new StringWriter();
557         obj.writeJSONString(toSent);
558         String message = toSent.toString();
559         AMQP.BasicProperties props = new
AMQP.BasicProperties
560                     .Builder()
561                     .correlationId(MainActivity.phoneid)
562                     .build();
563             MainActivity.channel.basicPublish("", 
MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
564         } catch (Exception e) {
565             }
566         }
567     }
568 });

```

Bagian ini adalah pengaturan tombol-tombol yang ada di *title bar* halaman. Tombol-tombol tersebut berupa *ImageView*. *ImageView* di bagian kiri berfungsi sebagai *burger menu* untuk memunculkan *Navigation Drawer* dan salah satu *ImageView* di bagian kanan yang

bergambar *gear* berfungsi sebagai *shortcut* untuk menuju *SettingActivity.java* dan *ImageView* yang bergambar gembok berfungsi sebagai *shortcut* aktivasi *scenario lock*.

Scenario lock dapat digunakan untuk mengunci beberapa akses utama di dalam rumah. Dengan diaktifkannya *scenario lock*, seluruh aplikasi MINDS yang terdaftar pada rumah yang sama juga akan terkunci. Ketika dalam keadaan terkunci, pengguna tidak dapat menekan seluruh tombol yang ada kecuali *ImageView* bergambar gembok yang terbuka untuk melakukan *unlock* sehingga dapat membuka kembali aplikasi MINDS. Setiap member rumah akan diberikan notifikasi jika ada member yang membuka atau mengunci aplikasi MINDS. Fitur ini diciptakan untuk meningkatkan sistem keamanan rumah. Selain dengan menekan *burger menu*, pengguna juga dapat melakukan *swipe* dari tepian kiri layar ke kanan untuk memunculkan *Navigation Drawer*. Ketika *lock* atau *unlock button* ditekan, maka aplikasi akan mengirimkan data JSON untuk mengaktifkan skenario *lock* atau *unlock*.

Bagian berikutnya adalah sebagai berikut.

```
//SERVICE
571         mSensorService = new SensorService(this);
572         mServiceIntent = new Intent(this,
mSensorService.getClass());
573         if (CurrentStatNotif && !homeid.equals(""))
&& !isMyServiceRunning(mSensorService.getClass())) {
574             startService(mServiceIntent);
575         }
576
577         //LOAD ALL LIST FROM RASPI (DEVICES, ROOMS, SCENARIOS,
NOTIES)
578         if (!homeid.equals(""))
579             load_from_raspi();
580     }
```

Bagian ini merupakan bagian akhir dari fungsi *OnCreate()* activity ini. Bagian ini berisi pengaktifan service dan *load data* dari *host*. Service merupakan fitur dalam Android untuk menjalankan suatu operasi atau algoritma tertentu di belakang layar. Algoritma di dalam service akan selalu berjalan meskipun aplikasi ditutup. Algoritma di dalam service akan dijelaskan kemudian pada *SensorService.java*. Selain untuk mengaktifkan service, bagian ini juga untuk *load* data dari *host* melalui fungsi *load_from_raspi*. Fungsi ini akan dijelaskan pada bagian bawah.

Bagian selanjutnya adalah sebagai berikut.

```
boolean isMyServiceRunning(Class<?> serviceClass) {
583         ActivityManager manager = (ActivityManager)
getSystemService(Context.ACTIVITY_SERVICE);
584         for (ActivityManager.RunningServiceInfo service :
manager.getRunningServices(Integer.MAX_VALUE)) {
```

```

585         if
586             (serviceClass.getName().equals(service.service.getClassName())) {
587                 Log.i ("isMyServiceRunning?", true+"");
588                 return true;
589             }
590             Log.i ("isMyServiceRunning?", false+"");
591             return false;
592         }

```

Kita telah memasuki bagian luar dari fungsi OnCreate. Bagian ini merupakan definisi fungsi isMyServiceRunning. Fungsi ini bertujuan untuk memeriksa apakah ponsel sedang menjalankan *service* aplikasi ini. Jika iya, maka fungsi ini akan mengembalikan nilai *true*. Sebaliknya, nilai *false* akan dikembalikan jika aplikasi tidak sedang menjalankan *service*.

Bagian selanjutnya adalah sebagai berikut.

```

public Animation inFromRightAnimation()
595 {
596     Animation inFromRight = new TranslateAnimation(
597         Animation.RELATIVE_TO_PARENT, +1.0f,
598         Animation.RELATIVE_TO_PARENT, 0.0f,
599         Animation.RELATIVE_TO_PARENT, 0.0f,
600         Animation.RELATIVE_TO_PARENT, 0.0f);
601     inFromRight.setDuration(240);
602     inFromRight.setInterpolator(new AccelerateInterpolator());
603     return inFromRight;
604 }
605
606 public Animation outToRightAnimation()
607 {
608     Animation outtoLeft = new TranslateAnimation(
609         Animation.RELATIVE_TO_PARENT, -1.0f,
610         Animation.RELATIVE_TO_PARENT, 0.0f,
611         Animation.RELATIVE_TO_PARENT, 0.0f,
612         Animation.RELATIVE_TO_PARENT, 0.0f);
613     outtoLeft.setDuration(240);
614     outtoLeft.setInterpolator(new AccelerateInterpolator());
615     return outtoLeft;
616 }
617
618 private Menuu[] menus = {
619     new Menuu(R.string.switchstr, R.drawable.switchr),
620     new Menuu(R.string.chronostr, R.drawable.chronor),
621     new Menuu(R.string.curtainstr, R.drawable.curtainr),
622     new Menuu(R.string.doorstr, R.drawable.doorr),
623     new Menuu(R.string.fanstr, R.drawable.fanr),
624     new Menuu(R.string.lampstr, R.drawable.lampr),
625 };

```

Bagian ini merupakan pendefinisian animasi. Animasi ini digunakan pada *floating button* yang telah dijelaskan pada bagian-bagian sebelumnya. Pada saat menekan *FloatingButton*, digunakan animasi untuk menampilkan menu “Add Device”, “Add Room”, dan “Add Scenario”. Animasi dibuat satu persatu pada *folder R.anim*. Animasi yang dibuat

menggunakan tiga parameter yaitu *scale*, *rotate*, dan *alpha*. Pada bagian *scale*, menu ditampilkan dari ukuran 0% hingga 100% dengan *PivotX* dan *PivotY* yang bernilai 50% dan pada bagian *alpha*, nilai *alpha* diubah dari *alpha* dengan nilai 0.0 menjadi 1.0 dalam rentang waktu 300ms. Perbedaan animasi *scale* dan *alpha* pada ketiga menu adalah waktu animasi dimulai, pada menu “Add Device” *startOffSet* bernilai 0, sedangkan pada “Add Room”, *startOffSet* bernilai 50ms dan pada “Add Scenario” *startOffSet* bernilai 100ms. Perbedaan nilai *offset* ini menciptakan gerakan yang bergantian pada saat menu ditampilkan. Ketika *FloatingButton* ditekan sekali lagi menu disembunyikan dari ukuran 100% hingga 0% dengan *PivotX* dan *PivotY* yang sama dan *alpha* bernilai 1.0 hingga 0.0 dalam rentang waktu 300ms dan tanpa *offset*, sehingga ketiga menu menghilang pada waktu yang bersamaan. *FloatingButton* juga diberikan animasi *rotate*, sehingga ketika ditekan sekali akan berputar searah jarum jam sebesar 45° dan ketika ditekan sekali lagi akan berputar 45° berlawanan arah jarum jam dalam rentang waktu 300ms.

Selain itu, bagian ini juga merupakan bagian pendefinisian tipe data Menuu untuk Gridview pada *tab devices*. Tab ini juga telah dijelaskan pada bagian sebelumnya.

Bagian selanjutnya adalah sebagai berikut.

```
@Override
630     //FUNGSI YANG AKAN DIJALANKAN KETIKA TOMBOL BACK PADA PONSEL
DITEKAN
631     public void onBackPressed() {
632         DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
633         if (drawer.isDrawerOpen(GravityCompat.START)) {
634             drawer.closeDrawer(GravityCompat.START);
635         } else if (isOpen) {
636             AddDevice.startAnimation(FabClose);
637             AddRoom.startAnimation(FabClose);
638             AddScenario.startAnimation(FabClose);
639             fab.startAnimation(FabRCClockwise);
640             AddDevice.setClickable(false);
641             AddRoom.setClickable(false);
642             AddScenario.setClickable(false);
643             isOpen = false;
644         }
645     }
646     else {
647         //jika user tidak mencentang "Keep ME Logged In" pada
halaman sign in, maka ketika back ditekan, seluruh identitas user
dalam memori dihapus
648         if (!sharedPref.getBoolean("remember",false)) {
649             mainList.clear();
650             homeid = "";
651             homename = "";
652             email = "";
653             password = "";
```

```

654         usname = "";
655         profpic = "";
656         privilege = "";
657         AESkeyString = "";
658
659     try {
660         subscribeThread.interrupt();
661         connection.close();
662     } catch (Exception e) {
663
664     }
665     prefEditor.putString("homeid", homeid);
666     prefEditor.putString("homename", homename);
667     prefEditor.putString("email", email);
668     prefEditor.putString("password", password);
669     prefEditor.putString("usname", usname);
670     prefEditor.putString("picture", profpic);
671     prefEditor.putString("privilege", privilege);
672     prefEditor.putString("aeskey", AESkeyString);
673     prefEditor.putString("aeskeyexp", "");
674     prefEditor.apply();
675
676     stopService(new
Intent(this, mSensorService.getClass()));
677     Intent afterloginIntent = new
Intent(MainActivity.this, LoginActivity.class);
678     MainActivity.this.startActivity(afterloginIntent);
679 }
680     moveTaskToBack(true);
681 }
682
683 }
```

Bagian ini merupakan fungsi `OnBackPressed()`, yaitu fungsi yang akan dijalankan bila tombol back pada ponsel ditekan. Pada saat *back* ditekan, maka aplikasi akan menghapus profil pengguna di memori jika saat pengguna *sign in* tidak mencentang *keep me logged in*. Dengan demikian, saat pengguna membuka kembali aplikasi, maka pengguna akan kembali masuk ke halaman login.

Bagian berikutnya adalah sebagai berikut.

```

@Override
686     public boolean onCreateOptionsMenu(Menu menu) {
687         // Inflate the menu; this adds items to the action bar if
it is present.
688         getMenuInflater().inflate(R.menu.main, menu);
689         return true;
690     }
691
692     @Override
693     public boolean onOptionsItemSelected(MenuItem item) {
694         // Handle action bar item clicks here. The action bar will
695         // automatically handle clicks on the Home/Up button, so
long
696         // as you specify a parent activity in
AndroidManifest.xml.
```

```

697     int id = item.getItemId();
698
699     //noinspection SimplifiableIfStatement
700     if (id == R.id.action_settings) {
701         return true;
702     }
703
704     return super.onOptionsItemSelected(item);
705 }
706
707 @SuppressWarnings("StatementWithEmptyBody")
708 @Override
709 public boolean onNavigationItemSelected(MenuItem item) {
710     // Aksi yang dilakukan ketika salah satu tombol dalam
navigation view ditekan
711     int id = item.getItemId();
712
713     if (id == R.id.nav_dashboard) {
714         //NOTHING TO DO
715     } else if (id == R.id.nav_notification) {
716         if (!homeid.equals("")) {
717             Intent invitationIntent = new
Intent(MainActivity.this, NotificationActivity.class);
718             MainActivity.this.startActivity(invitationIntent);
719         } else
720             Toast.makeText(getApplicationContext(), "You are
not registered to a home yet", Toast.LENGTH_SHORT).show();
721
722     } else if (id == R.id.nav_data) {
723         if (!homeid.equals("")) {
724             Intent invitationIntent = new
Intent(MainActivity.this, DataActivity.class);
725             MainActivity.this.startActivity(invitationIntent);
726         } else
727             Toast.makeText(getApplicationContext(), "You are
not registered to a home yet", Toast.LENGTH_SHORT).show();
728
729     } else if (id == R.id.nav_members) {
730         if (!homeid.equals("")) {
731             Intent invitationIntent = new
Intent(MainActivity.this, CreateNewActivity2.class);
732             MainActivity.this.startActivity(invitationIntent);
733         } else
734             Toast.makeText(getApplicationContext(), "You are
not registered to a home yet", Toast.LENGTH_SHORT).show();
735
736     } else if (id == R.id.nav_invited) {
737         if (!homeid.equals("")) {
738             Toast.makeText(getApplicationContext(), "You are
already registered to a home", Toast.LENGTH_SHORT).show();
739
740     } else {
741         Intent invitationIntent = new
Intent(MainActivity.this, GettingStartedActivity.class);
742             MainActivity.this.startActivity(invitationIntent);
743     }
744
745     } else if (id == R.id.nav_setting) {

```

```

746             if (!homeid.equals("")) {
747                 Intent afterloginIntent = new
Intent(MainActivity.this, SettingActivity.class);
748                 MainActivity.this.startActivity(afterloginIntent);
749             } else
750                 Toast.makeText(getApplicationContext(), "You are
not registered to a home yet", Toast.LENGTH_SHORT).show();
751
752         } else if (id == R.id.nav_logout) {
753
754             //MENGIRIM PESAN KEPADA HOST DAN SERVER
755             try {
756                 AMQP.BasicProperties props = new
AMQP.BasicProperties
757                     .Builder()
758                     .correlationId(Settings.Secure.getString(g
etContentResolver(), Settings.Secure.ANDROID_ID))
759                     .build();
760                 JSONObject obj = new JSONObject();
761                 obj.put("head", "SO");
762                 obj.put("homeid", "123456");
763                 StringWriter toSent = new StringWriter();
764                 obj.writeJSONString(toSent);
765                 String message = toSent.toString();
766                 channel.basicPublish("", "AESqueue", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
767                 channel.basicPublish("", homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
768             } catch (Exception e) {
769                 Toast.makeText(getApplicationContext(), "Gagal Log
out",
770                         Toast.LENGTH_SHORT).show();
771             }
772
773             //MENGOSONGKAN SELURUH LIST
774             mainList.clear();
775             homeid = "";
776             email = "";
777             password = "";
778             usname = "";
779             profpic = "";
780             privilege = "";
781             AESkeyString = "";
782             homename = "";
783
784             //MENGHENTIKAN KONEKSI KE RABBITMQ
785             try {
786                 subscribeThread.interrupt();
787                 connection.close();
788             } catch (Exception e) {
789
790             }
791
792             //MENGHAPUS MEMORY
793             prefEditor.putString("homeid", homeid);
794             prefEditor.putString("homename", homename);
795             prefEditor.putString("email", email);
796             prefEditor.putString("password", password);

```

```

797         prefEditor.putString("uname",uname);
798         prefEditor.putString("picture",profpic);
799         prefEditor.putString("privilege",privilege);
800         prefEditor.putString ("aeskey",AESkeyString);
801         prefEditor.putString("aeskeyexp","");
802         prefEditor.apply();
803
804         //MENGHENTIKAN SERVICE
805         stopService(mServiceIntent);
806
807         //KEMBALI KE HALAMAN AWAL
808         Intent afterloginIntent = new
Intent(MainActivity.this,LoginActivity.class);
809         MainActivity.this.startActivity(afterloginIntent);
810     }
811
812     DrawerLayout drawer = (DrawerLayout)
findViewById(R.id.drawer_layout);
813     drawer.closeDrawer(GravityCompat.START);
814     return true;
815 }
816
817     private void applyFontToMenuItem(MenuItem mi) {
818         Typeface font = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Bold.otf");
819         SpannableString mNewTitle = new
SpannableString(mi.getTitle());
820         mNewTitle.setSpan(new CustomTypefaceSpan("") , font), 0 ,
mNewTitle.length(), Spannable.SPAN_INCLUSIVE_INCLUSIVE);
821         mi.setTitle(mNewTitle);
822     }

```

Bagian ini adalah untuk mengatur perilaku aplikasi ketika salah satu dari tombol pada *navigation drawer* ditekan. Terdapat beberapa menu yang terletak pada *Navigation Drawer*, yaitu “Dashboard”, “Notification”, “Data Analytics”, “House Members”, “Get Invited”, “Settings”, dan juga “Logout”.

- Menu “Dashboard” akan mengarahkan pengguna menuju menu utama (*MainActivity.java*)
- Menu “Notification” berguna untuk menampilkan *NotificationActivity.java* yang akan menampilkan peringatan-peringatan yang terjadi dalam sistem Rumah Cerdas, selain itu permintaan *invitation code* juga dikirimkan melalui menu ini.
- Menu “Data Analytics” mengarahkan pengguna menuju *DataActivity.java* yang menampilkan data-data penggunaan *device* yang terdaftar pada rumah tersebut, data tersebut diolah dalam bentuk grafik, sehingga pengguna dapat melihat data tersebut dengan lebih mudah.

- Menu “House Members” mengarahkan pengguna ke halaman *member* (*CreateNewActivity2.java*) untuk melihat setiap member rumah dan berbagai informasi member lainnya.
- Menu “Get Invited” dapat digunakan oleh pengguna yang baru berpindah rumah sehingga perlu untuk mendaftarkan kembali rumahnya.
- Untuk mengatur notifikasi, pengaturan *privilege*, pengaturan *profile* dan pengaturan lainnya, pengguna dapat menekan menu “Settings” sehingga pengguna akan segera diarahkan menuju *SettingActivity.java*.
- Menu terakhir adalah “Logout” yang dapat mengeluarkan akun pengguna dari aplikasi MINDS pada *smartphone* tersebut sehingga tampilan berubah menjadi *LoginActivity.java*. Ketika menu “Logout” ditekan, seluruh *SharedPreference* akan dikosongkan dengan cara melakukan input kosong terhadap seluruh *SharedPreference*.

Bagian selanjutnya adalah sebagai berikut.

```
//FUNGSI UNTUK UPDATE LAYOUT
825     public static void update_device (String address, String
command) {
826
827         for (ArrayList<String> element : mainList) {
828             if (element.get(1).equals(address)) {
829                 element.set(3, command);
830                 ((BaseAdapter)
SwitchActivity.listView.getAdapter()).notifyDataSetChanged();
831                     break;
832                 }
833             }
834         }
835
836     //FUNGSI UNTUK LOAD DATA DARI HOST (DIJALANKAN DI AWAL ON
CREATE)
837     void load_from_raspi () {
838         try {
839
840             ConnectionFactory factory = new ConnectionFactory();
841             factory.setUri(server_uri);
842             Connection connection = factory.newConnection();
843             Channel channel = connection.createChannel();
844
845             String replyQueueName =
channel.queueDeclare().getQueue();
846
847             AMQP.BasicProperties props = new AMQP.BasicProperties
.Builder()
848                 .correlationId(MainActivity.phoneid)
849                 .replyTo(replyQueueName)
850                 .build();
```

```

852
853         //PERIKSA APAKAH AES KEY SUDAH EXPIRED
854         //JIKA YA
855         if (new Date().after((new SimpleDateFormat("yyyy-MM-dd
HH:mm:ss")).parse(sharedPref.getString("aeskeyexp", "")))) {
856
857             //GENERASI KEY BARU DAN DISIMPAN
858             AESkeyString = generateAESkey();
859             SensorService.AESkeyString =
MainActivity.AESkeyString;
860             prefEditor.putString("aeskey", AESkeyString);
861             prefEditor.commit();
862
863             //KIRIM PESAN KE HOST
864             JSONObject obj = new JSONObject();
865             obj.put("head", "LO");
866             obj.put("type", "updatekey");
867             obj.put("aeskey", AESkeyString);
868             obj.put("homeid", homeid);
869             obj.put("privilege", privilege);
870             StringWriter toSent = new StringWriter();
871             obj.writeJSONString(toSent);
872             String message = toSent.toString();
873             channel.basicPublish("", homeid + "AES", props,
AESencrypt(message).getBytes());
874
875             //UPDATE KEY IN SERVER
876             obj = new JSONObject();
877             obj.put("head", "MU");
878             obj.put("type", "editaeskey");
879             obj.put("aeskey", AESkeyString);
880             toSent = new StringWriter();
881             obj.writeJSONString(toSent);
882             message = toSent.toString();
883             channel.basicPublish("", "AESqueue", props,
AESencrypt(message).getBytes());
884
885     }
886
887     //JIKA KEY BELUM EXPIRED
888     else {
889         //KIRIM PESAN KE HOST
890         JSONObject obj = new JSONObject();
891         obj.put("head", "LO");
892         obj.put("type", "nokey");
893         obj.put("homeid", homeid);
894         obj.put("privilege", privilege);
895         StringWriter toSent = new StringWriter();
896         obj.writeJSONString(toSent);
897         String message = toSent.toString();
898         channel.basicPublish("", homeid + "AES", props,
AESencrypt(message).getBytes());
899     }
900
901     final BlockingQueue<String> response = new
ArrayBlockingQueue<>(1);
902

```

```

903             channel.basicConsume(replyQueueName, true, new
DefaultConsumer(channel) {
904                 @Override
905                 public void handleDelivery(String consumerTag,
Envelope envelope, AMQP.BasicProperties properties, byte[] body)
throws IOException {
906                     if
(properties.getCorrelationId().equals(MainActivity.phoneid)) {
907                         response.offer(new String(body, "UTF-8"));
908
909                     }
910                 }
911             );
912             String res = AESdecrypt(response.take());
913
914             // IF USER HAS BEEN DELETED, SERVER WILL ANSWER
'deleted'
915             if (res.equals("deleted")) {
916                 homeid = "";
917                 privilege = "";
918                 stopService(mServiceIntent);
919                 prefEditor.putString("homeid", homeid);
920                 prefEditor.putString("email", email);
921                 prefEditor.commit();
922                 mainList.clear();
923             }
924
925             // IF NOT DELETED
926         else {
927
928             //PARSE THE LIST PACKAGE FROM HOST
929             JSONObject jsonObject = (JSONObject)
parser.parse(res);
930             String head = (String) jsonObject.get("head");
931             if (head.equals("LO")) {
932                 mainList = (ArrayList<ArrayList<String>>)
jsonObject.get("devices");
933                 ArrayList<ArrayList<String>> rooms =
(ArrayList<ArrayList<String>>) jsonObject.get("rooms");
934                 ArrayList<ArrayList<String>> scenarios =
(ArrayList<ArrayList<String>>) jsonObject.get("scenarios");
935                 NotificationActivity.notifs =
(ArrayList<ArrayList<String>>) jsonObject.get("notifications");
936                 homename = (String)
jsonObject.get("homename");
937
938             try {
939                 menus2.clear();
940                 for (ArrayList<String> element : rooms) {
941                     menus2.add(element.get(0));
942                 }
943                 menus3.clear();
944                 for (ArrayList<String> element :
scenarios) {
945                     menus3.add(element.get(0));
946                 }
947             }
948

```

```

949                     ((BaseAdapter)
gridView2.getAdapter()).notifyDataSetChanged();
950                     ((BaseAdapter)
listView3.getAdapter()).notifyDataSetChanged();
951             } catch (Exception e) {
952
953         }
954     }
955
956     //RESET AES KEY DI HANDPHONE
957     try {
958         Calendar calendar = Calendar.getInstance();
959         calendar.setTime(new Date());
960         calendar.add(Calendar.DATE,1);
961         prefEditor.putString("aeskeyexp", (new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss")).format(calendar.getTime()));
962         skeySpec = new
SecretKeySpec(AESkeyString.getBytes("UTF-8"), "AES");
963         SensorService.skeySpec = new
SecretKeySpec(AESkeyString.getBytes("UTF-8"), "AES");
964     } catch (Exception e) {}
965
966     //CEK STATUS KUNCI RUMAH
967     String lockstatus = (String)
jsonObject.get("lockstatus");
968     if (lockstatus.equals("lock")) {
969         ivlock.setVisibility(View.INVISIBLE);
970         ivunlock.setVisibility(View.VISIBLE);
971         llback.startAnimation(Fabbacklocked);
972         lllock.startAnimation(Fablocked);
973         burgermenu.setClickable(false);
974         settingmenu.setClickable(false);
975         tw.setEnabled(false);
976         fab.setClickable(false);
977         gridView.setEnabled(false);
978         gridView2.setEnabled(false);
979         listView3.setEnabled(false);
980         host.setCurrentTab(0);
981
982         drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
983     }
984     channel.close();
985     connection.close();
986
987
988 } catch (Exception e) {
989     Toast.makeText(getApplicationContext(),"Can't connect
to the server",
990             Toast.LENGTH_SHORT).show();
991 }
992 }
```

Bagian ini berisi fungsi update_device dan load_from_raspi. Update_device merupakan fungsi untuk melakukan update layout ketika ada perubahan device terjadi. Load_from_raspi merupakan fungsi untuk melakukan loading data dari Raspberry Pi.

Loading ini dilakukan pada fungsi OnCreate(). Loading data mengimplementasikan perintah dengan bentuk data JSON sebagai berikut.

Tabel 23 Data JSON untuk Loading Data dari Raspberry Pi

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
LO (Load)	-	<ul style="list-style-type: none"> • Home ID • Privilege • User • Kunci AES (jika kunci lama sudah kadaluarsa) 	Host	AES	<p>Bila kode benar:</p> <ul style="list-style-type: none"> • Daftar device • Daftar room • Daftar skenario

Fungsi load_from Raspi juga berisi perintah untuk memeriksa apakah kunci AES sudah kadaluarsa. Jika sudah, maka dilakukan pula *update* kunci AES, baik di ponsel, di server, maupun di Raspberry Pi.

Bagian berikutnya dari kode ini adalah sebagai berikut.

```
//FUNGSI UNTUK MENG-GENERATE AES KEY
995     public String generateAESkey () {
996         try {
997             KeyGenerator keyGenerator =
KeyGenerator.getInstance("AES");
998             keyGenerator.init(128);
999             SecretKey secretKey = keyGenerator.generateKey();
1000             return (Base64.encodeToString(secretKey.getEncoded(),
Base64.DEFAULT).substring(0,16));
1001         } catch (Exception e) {
1002             return null;
1003         }
1004     }
1005
1006
1007 //FUNGSI UNTUK AES ENCRYPTION
1008 public static String AESencrypt (String input) {
1009     String plain_text = "";
1010     try{
1011         aescipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv);
1012         byte[] a = iv_string.getBytes();
1013         byte[] b = aescipher.doFinal(input.getBytes());
1014         byte[] c = new byte[a.length + b.length];
1015         System.arraycopy(a, 0, c, 0, a.length);
1016         System.arraycopy(b, 0, c, a.length, b.length);
1017         plain_text = new String
(Base64.encodeToString(c,Base64.DEFAULT));
1018     } catch (Exception e) {
1019         System.err.println("Caught Exception: " +
e.getMessage());
1020     }
1021     return plain_text;
1022 }
1023
1024 //FUNGSI UNTUK AES DECRYPTION
```

```

1025     public static String AESdecrypt(String
1026     encrypted_encoded_string) {
1027         String plain_text = "";
1028         try{
1029             byte[] encrypted_decoded_bytes =
1030             Base64.decode(encrypted_encoded_string, Base64.DEFAULT);
1031             aescipher.init(Cipher.DECRYPT_MODE, skeySpec, iv);
1032             plain_text = new
1033             String(aescipher.doFinal(Arrays.copyOfRange(encrypted_decoded_bytes,
1034             16, encrypted_decoded_bytes.length)));//Returns garbage characters
1035             return plain_text;
1036         } catch (Exception e) {
1037             System.out.println("Caught Exception: " +
1038             e.getMessage());
1039         }
1040
1041         //FUNGSI UNTUK RSA ENCRYPTION
1042         public static String RSAencrypt (String input) {
1043             String plain_text = "";
1044             try {
1045                 Cipher cipher = Cipher.getInstance("RSA");
1046                 cipher.init(Cipher.ENCRYPT_MODE, pubkey);
1047                 plain_text = new String
1048                 (Base64.encodeToString(cipher.doFinal(input.getBytes("UTF-
1049                 8")),Base64.DEFAULT)));
1050             } catch (Exception e1) {
1051                 return null;
1052             }
1053             return plain_text;
1054         }
1055         //FUNGSI UNTUK RSA DECRYPTION
1056         public static String RSAdecrypt (String input) {
1057             try {
1058                 byte[] encrypted_decoded_bytes = Base64.decode(input,
1059                 Base64.DEFAULT);
1060                 Cipher cipher = Cipher.getInstance("RSA");
1061                 cipher.init(Cipher.DECRYPT_MODE, privkey);
1062                 return (new
1063                 String(cipher.doFinal(encrypted_decoded_bytes)));
1064             } catch (Exception e1) {
1065                 return null;
1066             }
1067             //FUNGSI UNTUK LOAD PRIVATE KEY RSA
1068             public static PrivateKey loadPrivateKey(String key64) {
1069                 byte[] clear =
1070                 Base64.decode(key64.getBytes(),Base64.DEFAULT);
1071                 PKCS8EncodedKeySpec keySpec = new
1072                 PKCS8EncodedKeySpec(clear);
1073                 try {

```

```

1072         KeyFactory fact = KeyFactory.getInstance("RSA");
1073         PrivateKey priv = fact.generatePrivate(keySpec);
1074         return priv;
1075     } catch (Exception e) {
1076         return null;
1077     }
1078 }
1079 }
1080
1081 //FUNGSI UNTUK LOAD PUBLIC KEY RSA
1082 public static PublicKey loadPublicKey(String stored) {
1083     byte[] data =
Base64.decode(stored.getBytes(), Base64.DEFAULT);
1084     X509EncodedKeySpec spec = new X509EncodedKeySpec(data);
1085     try {
1086         KeyFactory fact = KeyFactory.getInstance("RSA");
1087         return fact.generatePublic(spec);
1088     } catch (Exception e) {
1089         return null;
1090     }
1091 }

```

Bagian ini merupakan implementasi dari sistem keamanan jaringan. Bagian ini berisi fungsi-fungsi untuk enkripsi dan dekripsi AES dan RSA. Bagian ini juga berisi kode untuk melakukan generasi kunci AES yang baru.

Bagian berikutnya adalah sebagai berikut.

```

//FUNGSI UNTUK DECODE GAMBAR DARI STRING BASE64
1094     public static Bitmap decodeBase64(String input) {
1095         byte[] decodedByte = Base64.decode(input, 0);
1096         return BitmapFactory
1097             .decodeByteArray(decodedByte, 0,
decodedByte.length);
1098     }
1099
1100     //FUNGSI YANG DIJALANKAN KETIKA APLIKASI DISTOP
1101     @Override
1102     public void onStop()
1103     {
1104         super.onStop();
1105         SharedPreferences sharedPref =
getSharedPreferences("Authentication", MODE_PRIVATE);
1106         SharedPreferences.Editor prefEditor = sharedPref.edit();
1107         prefEditor.putString("homeid", homeid);
1108         prefEditor.putString("email", email);
1109         prefEditor.putString("password", password);
1110         prefEditor.apply();
1111     }
1112
1113     //FUNGSI YANG DIJALANKAN KETIKA ACTIVITY DIKILL
1114     @Override
1115     public void onDestroy() {
1116         super.onDestroy();
1117         if (!sharedPref.getBoolean("remember", false)) {
1118             //MENGHAPUS SELURUH DATA DI MEMORI KETIKA USER TIDAK
MENCENTANG "KEEP ME LOGIN"

```

```

1119         mainList.clear();
1120         homeid = "";
1121         email = "";
1122         password = "";
1123         usname = "";
1124         profpic = "";
1125         privilege = "";
1126         AESkeyString = "";
1127         homename = "";
1128
1129
1130     try {
1131         subscribeThread.interrupt();
1132         connection.close();
1133     } catch (Exception e) {
1134
1135     }
1136     prefEditor.putString("homeid",homeid);
1137     prefEditor.putString("homename",homename);
1138     prefEditor.putString("email",email);
1139     prefEditor.putString("password",password);
1140     prefEditor.putString("usname",usname);
1141     prefEditor.putString("picture",profpic);
1142     prefEditor.putString("privilege",privilege);
1143     prefEditor.putString("aeskey", AESkeyString);
1144     prefEditor.putString("aeskeyexp","");
1145
1146     prefEditor.apply();
1147
1148     stopService(new
Intent(this,mSensorService.getClass()));
1149     Intent afterloginIntent = new
Intent(MainActivity.this,LoginActivity.class);
1150     MainActivity.this.startActivity(afterloginIntent);
1151 }
1152 }
1153}

```

Bagian ini adalah bagian akhir dari `MainActivity.java`. Bagian ini berisi fungsi `OnStop` dan `OnDestroy`. Kedua fungsi ini akan dieksekusi ketika aplikasi dimatikan atau ditutup. Ketika aplikasi ditutup, aplikasi akan menghapus data-data dalam *shared preference* bila user tidak mencentang "keep me logged in" pada halaman *sign in*. Algoritma ini sama persis dengan isi algoritma `OnBackPressed`.

Selain penjelasan di atas, ada hal penting lain di dalam *activity* ini. Pada *Activity* ini, terdapat satu variabel penting yang menjadi patokan keberjalanannya hampir keseluruhan aplikasi. Variabel tersebut bernama `mainList`, yaitu sebuah variabel bertipe `ArrayList<ArrayList<String>>`. Seperti dilihat pada tipenya, variabel ini merupakan *ArrayList* 2 dimensi. Dimensi pertama (*row*) menyatakan banyaknya device yang terdaftar pada rumah. Dimensi kedua (*column*) terdiri dari 6 kolom, yaitu *name, address, type, status,*

room, dan *privilege*. Isi List ini sama persis dengan isi tabel *devices* pada database homeXXXXXX di Raspberry. *Name* adalah kolom nama *device*. *Address* adalah kolom alamat XBEE dari *device* yang bersangkutan. *Type* adalah jenis *device* (switch, temperature, curtain, door, fan, *lamp*). *Status* adalah keadaan terkini dari setiap *device*, misalnya ON atau OFF, 100% atau 50%, dan sebagainya (bergantung pada *type*). *Room* adalah kolom untuk menyimpan informasi di ruangan mana device tersebut berada. *Privilege* adalah kolom untuk menyimpan informasi *privilege* dari *device* yang bersangkutan. Variabel *mainList* selalu diperbarui oleh fungsi *load_from_raspi* yang dipanggil saat halaman utama dibuka.

Untuk membentuk GridView pada tab pertama, terdapat satu file java khusus untuk mengaturnya. Berikut adalah kode java MenuAdapter.java untuk mengatur perilaku tombol-tombol pada tab 1.

```
1  package com.example.billyausten.newfinal;
2
3  /**
4   * Created by Billy Austen on 2/3/2017.
5   */
6
7  import android.content.Context;
8  import android.content.res.AssetManager;
9  import android.graphics.Typeface;
10 import android.view.LayoutInflater;
11 import android.view.View;
12 import android.view.ViewGroup;
13 import android.widget.BaseAdapter;
14 import android.widget.ImageView;
15 import android.widget.TextView;
16
17 public class MenuAdapter extends BaseAdapter {
18
19     //DEKLARASI VARIABEL
20     private final Context mContext;
21     private final Menuuu[] menus;
22
23     //KONSTRUKTOR
24     public MenuAdapter(Context context, Menuuu[] menus) {
25         this.mContext = context;
26         this.menus = menus;
27     }
28
29     @Override
30     public int getCount() {
31         return menus.length;
32     }
33
34     @Override
35     public long getItemId(int position) {
36         return 0;
37     }
38 }
```

```

39     @Override
40     public Object getItem(int position) {
41         return null;
42     }
43
44     @Override
45     public View getView(int position, View convertView,
ViewGroup parent) {
46
47         final Menuu menuu = menus[position];
48
49         if (convertView == null) {
50             final LayoutInflater layoutInflater =
LayoutInflater.from(mContext);
51             convertView =
layoutInflater.inflate(R.layout.linearlayout_menu, null);
52         }
53
54         //IMAGE VIEW ICON TIAP TIPE DEVICE
55         final ImageView imageView =
(ImageView)convertView.findViewById(R.id.imageview_icon);
56         //TEXTVIEW NAMA DEVICE
57         final TextView nameTextView =
(TextView)convertView.findViewById(R.id.textview_name);
58         Typeface font_cb =
Typeface.createFromAsset(mContext.getAssets(), "fonts/Quicksand-
Bold.otf");
59         nameTextView.setTypeface(font_cb);
60
61         imageView.setImageResource(menuu.getImageResource());
62
nameTextView.setText(mContext.getString(menuu.getName()));
63
64         return convertView;
65     }
66
67
68 }

```

Pada kode di atas, tidak ada hal istimewa yang diatur. Pengaturan yang dilakukan hanyalah pengaturan tulisan dan tampilan.

Selanjutnya, untuk membentuk tab 2, diperlukan kode MenuAdapter2.java untuk mengatur perilaku isi GridView *rooms*. Berikut adalah MenuAdapter2.java.

```

1 package com.example.billyausten.newfinal;
2
3
4 /**
5  * Created by Billy Austen on 2/3/2017.
6  */
7
8 import android.content.Context;
9 import android.content.Intent;
10 import android.content.res.AssetManager;
11 import android.graphics.Typeface;
12 import android.support.v7.widget.PopupMenu;
13 import android.view.LayoutInflater;

```

```

14 import android.view.MenuItem;
15 import android.view.View;
16 import android.view.ViewGroup;
17 import android.widget.ArrayAdapter;
18 import android.widget.BaseAdapter;
19 import android.widget.ImageView;
20 import android.widget.TextView;
21 import android.widget.Toast;
22 import android.app.AlertDialog;
23 import android.content.DialogInterface;
24
25 import com.rabbitmq.client.AMQP;
26
27 import org.json.simple.JSONObject;
28
29 import java.lang.Runnable;
30
31 import java.io.StringWriter;
32 import java.util.ArrayList;
33
34 public class MenuAdapter2 extends BaseAdapter {
35
36     //DEKLARASI VARIABEL
37     private Context mContext;
38     private ArrayList<String> menus2;
39
40     //KONSTRUKTOR
41     public MenuAdapter2(Context context, ArrayList<String>
menus2) {
42         this.mContext = context;
43         this.menus2 = menus2;
44     }
45
46     @Override
47     public int getCount() {
48         return menus2.size();
49     }
50
51     @Override
52     public long getItemId(int position) {
53         return 0;
54     }
55
56     @Override
57     public Object getItem(int position) {
58         return null;
59     }
60
61     @Override
62     public View getView(int position, View convertView,
ViewGroup parent) {
63         //NAMA ROOM
64         final String room_name = menus2.get(position);
65
66         if (convertView == null) {
67             final LayoutInflator layoutInflater =
LayoutInflater.from(mContext);

```

```

68             convertView =
layoutInflater.inflate(R.layout.linearLayout_menu2, null);
69         }
70
71         //ICON ROOM
72         final ImageView imageView =
(ImageView)convertView.findViewById(R.id.imageview_icon);
73         imageView.setImageResource(R.drawable.house5r);
74         //TEXTVIEW NAMA ROOM
75         final TextView nameTextView =
(TextView)convertView.findViewById(R.id.textview_name);
76         Typeface font_cb =
Typeface.createFromAsset(mContext.getAssets(), "fonts/Quicksand-
Bold.otf");
77         nameTextView.setTypeface(font_cb);
78         nameTextView.setText(room_name.toUpperCase());
79
80         //ICON SAMPAH, UNTUK DELETE ATAU EDIT ROOM
81         ImageView trash = (ImageView)
convertView.findViewById(R.id.imageview_trash);
82         trash.setOnClickListener(new View.OnClickListener() {
83
84             @Override
85             public void onClick (View view) {
86                 //bila tombol ditekan, akan muncul pop
87                 up menu (Delete dan Edit)
88                 PopupMenu popup = new
PopupMenu(mContext, view);
89                 popup.getMenuInflater().inflate(R.menu.menu_trash,
popup.getMenu());
90                 popup.show();
91                 popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
92                     public boolean
onMenuItemClick(MenuItem item) {
93                         switch (item.getItemId()) {
94                             //bila delete:
95                             case R.id.action_delete_room:
96                                 AlertDialog.Builder(mContext)
97                                     .setTitle("Deleting Room")
98                                     .setMessage("Are you sure you want to delete this room?")
99                                     .setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
100                                         public void onClick(DialogInterface dialog, int which) {
101                                             // mengirim pesan delete room
102                                             try {
JSONObject obj = new JSONObject();

```

```

103         obj.put("head", "MD");
104         obj.put("homeid", MainActivity.homeid);
105         obj.put("name", room_name);
106         obj.put("type", "deleteroom");
107         obj.put("notifsender", MainActivity.usname);
108         obj.put("notifmessage", "I just deleted room '" +
room_name +"'");
109         StringWriter toSent = new StringWriter();
110         obj.writeJSONString(toSent);
111         String message = toSent.toString();
112         AMQP.BasicProperties props = new
AMQP.BasicProperties
113             .Builder()
114                 .correlationId(MainActivity.phoneid)
115                 .build();
116         MainActivity.channel.basicPublish("", "AESqueue",
props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
117         MainActivity.channel.basicPublish("", MainActivity.homeid + "AES",
props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
118
119     } catch (Exception e) {}
120 }
121
122     .setNegativeButton("No", new DialogInterface.OnClickListener() {
123
public void onClick(DialogInterface dialog, int which) {
124
// do nothing
125
}
126
127     .setIcon(android.R.drawable.ic_dialog_alert)
128
.show();
129
return true;
130
131
//bila tombol edit
yang ditekan

```

```

132                                         case
R.id.action_edit_room:
133                                         //masuk ke
halaman edit room
134                                         Intent
afterloginIntent = new Intent(mContext,EditRoomActivity.class);
135                                         afterloginIntent.putExtra("type","edit");
136                                         afterloginIntent.putExtra("name",room_name);
137                                         mContext.startActivity(afterloginIntent);
138                                         return true;
139                                         }
140                                         return true;
141                                         }
142                                         });
143                                         }
144                                         });
145                                         });
146                                         return convertView;
147                                         }
148                                         }
149                                         }
150                                         }
151                                         }

```

Pada kode ini, hal penting yang diatur adalah perilaku tombol *Trash* ketika diklik oleh user (baris 80). Bila tombol ini ditekan, maka akan muncul menu yang berisi 2 pilihan, yaitu *edit room* dan *delete room*. Bila tombol *delete* yang ditekan, maka aplikasi akan mengirimkan data JSON untuk menghapus room. Bila tombol *edit* yang ditekan, maka aplikasi akan mengirimkan data JSON untuk mengedit room.

Tabel 24 Data JSON untuk Delete dan Edit Room

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	deleteroom	<ul style="list-style-type: none"> • Home ID • Nama room yang hendak dihapus 	Host dan Server	AES	Tidak ada jawaban
	editroom	<ul style="list-style-type: none"> • Home ID • Nama room yang hendak diedit 	Host dan Server	AES	Tidak ada jawaban

		<ul style="list-style-type: none"> • Nama Baru • Daftar device baru di dalam room 			
--	--	---	--	--	--

Berikutnya, untuk membentuk tab 3, diperlukan satu kode java bernama ListScenario.java. Kode ini berisi pengaturan perilaku isi dari tab 3. Berikut adalah kodenya.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.app.AlertDialog;
5 import android.content.DialogInterface;
6 import android.content.Intent;
7 import android.graphics.Typeface;
8 import android.support.v7.widget.PopupMenu;
9 import android.view.LayoutInflater;
10 import android.view.MenuItem;
11 import android.view.View;
12 import android.view.ViewGroup;
13 import android.widget.ArrayAdapter;
14 import android.widget.BaseAdapter;
15 import android.widget.Button;
16 import android.widget.ImageView;
17 import android.widget.Switch;
18 import android.widget.TextView;
19 import android.widget.Toast;
20
21 import java.io.IOException;
22 import java.io.StringWriter;
23 import java.util.ArrayList;
24 import java.util.UUID;
25 import java.util.concurrent.ArrayBlockingQueue;
26 import java.util.concurrent.BlockingQueue;
27
28 import com.rabbitmq.client.AMQP;
29 import com.rabbitmq.client.Channel;
30 import com.rabbitmq.client.Connection;
31 import com.rabbitmq.client.ConnectionFactory;
32 import com.rabbitmq.client.DefaultConsumer;
33 import com.rabbitmq.client.Envelope;
34 import com.rabbitmq.client.QueueingConsumer;
35 import com.rabbitmq.client.Consumer;
36
37 import org.json.simple.parser.JSONParser;
38 import org.json.simple.JSONObject;
39
40
41 /**
42  * Created by Billy Austen on 2/25/2017.
43  */
44
45 public class ListScenario extends ArrayAdapter<String> {
46

```

```

47         //DEKLARASI VARIABEL
48         private ArrayList<String> names;
49         private Activity context;
50
51         //KONSTRUKTOR
52         public ListScenario(Activity context, ArrayList<String>
names) {
53             super(context, R.layout.list_scenario, names);
54             this.context = context;
55             this.names = names;
56         }
57
58         @Override
59         public View getView(final int position, View convertView,
ViewGroup parent) {
60             LayoutInflator inflater =
context.getLayoutInflater();
61             final View listViewItem =
inflater.inflate(R.layout.list_scenario, null, true);
62
63             //PENGATURAN FONT
64             Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
65
66             //TEXTVIEW NAMA SKENARIO
67             TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvscenname);
68             textViewName.setTypeface(font_title);
69             textViewName.setText(names.get(position));
70
71             //IMAGEVIEW TONG SAMPAH, UNTUK DELETE ATAU EDIT
72             ImageView trash = (ImageView)
listViewItem.findViewById(R.id.imageview_trash);
73
74             //IMAGEVIEW SWITCH, UNTUK MENGAKTIFKAN SKENARIO
75             ImageView switchscen = (ImageView)
listViewItem.findViewById(R.id.switchscen);
76
77             //IMAGEVIEW BERTULISKAN MODE, SEBAGAI LOGO SKENARIO
78             ImageView ivmode = (ImageView)
listViewItem.findViewById(R.id.ivmode);
79             if (names.get(position).equals("lock_scen"))
80                 ivmode.setImageResource(R.drawable.lockiconr);
81             else
82
83                 ivmode.setImageResource(R.drawable.unlockiconr);
84
85             //KETIKA LOGO TONG SAMPAH DIKLIK:
86             trash.setOnClickListener(new View.OnClickListener() {
87
88                 @Override
89                 public void onClick (View view) {
90                     //untuk skenario lock dan unlock, menu
yang akan keluar hanyalah edit karena kedua skenario ini tidak dapat
dihapus

```

```

90          if
91      (names.get(position).equals("lock_scen") ||  

92      names.get(position).equals("unlock_scen")) {  

93          PopupMenu popup = new  

94          PopupMenu(context, view);  

95          popup.getMenuInflater().inflate(R.menu.menu_trash_scen_lock,  

96          popup.getMenu());  

97          popup.show();  

98  

99          popup.setOnMenuItemClickListener(new  

100         PopupMenu.OnMenuItemClickListener() {  

101             public boolean  

102             onMenuItemClick(MenuItem item) {  

103                 switch  

104                 (item.getItemId()) {  

105                     case  

106                     R.id.action_edit_scenario:  

107                         try {  

108                             //melakukan request untuk mendaftar barang apa saja yang  

109                             terdaftar dalam skenario ini  

110                             String replyQueueName =  

111                             MainActivity.channel.queueDeclare().getQueue();  

112                             AMQP.BasicProperties props = new AMQP.BasicProperties  

113                             .Builder()  

114                             .correlationId(MainActivity.phoneid)  

115                             .replyTo(replyQueueName)  

116                             .build();  

117  

118                             //pesan JSON  

119                             JSONObject obj = new JSONObject();  

120                             obj.put("head", "MD");  

121                             obj.put("homeid", MainActivity.homeid);  

122                             obj.put("type", "editscenario_req");  

123                             obj.put("name", names.get(position));  

124  

125                             StringWriter toSent = new StringWriter();  

126                             obj.writeJSONString(toSent);  

127  

128                             String message = toSent.toString();  

129  

130                             //mengirim pesan

```

```

117     MainActivity.channel.basicPublish("", MainActivity.homeid +
118         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
119
120     //menunggu balasan
121     final BlockingQueue<String> response = new
122     ArrayBlockingQueue<>(1);
123
124     MainActivity.channel.basicConsume(replyQueueName, true, new
125     DefaultConsumer(MainActivity.channel) {
126
127         @Override
128
129         public void handleDelivery(String consumerTag, Envelope
130             envelope, AMQP.BasicProperties properties, byte[] body) throws
131             IOException {
132
133             if
134                 (properties.getCorrelationId().equals(MainActivity.phoneid)) {
135
136                 response.offer(new String(body, "UTF-8"));
137
138             }
139
140         }
141
142     });
143
144     //mengambil balasan
145     String res = MainActivity.AESdecrypt(response.take());
146
147     JSONParser parser = new JSONParser();
148
149     JSONObject jsonObject = (JSONObject) parser.parse(res);
150
151     ListAddDeviceRoom.checked = (ArrayList<ArrayList<String>>)
152     jsonObject.get("devices");
153
154     String time = (String) jsonObject.get("time");
155
156     Intent afterloginIntent = new Intent(context,
157     AddScenarioActivity.class);
158
159     afterloginIntent.putExtra("type", "edit");
160
161     afterloginIntent.putExtra("name", names.get(position));
162
163     afterloginIntent.putExtra("time", time);
164
165     context.startActivity(afterloginIntent);
166
167     } catch
168     (Exception e) {
169
170     }
171
172 }
```



```

174             String message = toSent.toString();
175
176             AMQP.BasicProperties props = new
177                 AMQP.BasicProperties()
178
179                 .correlationId(MainActivity.phoneid)
180
181                 .build();
182
183             MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
184
185             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
186             MainActivity.AESencrypt(message).getBytes("UTF-8"));
187
188         } catch (Exception e) {
189
190     }
191
192     }
193
194     .setNegativeButton("No", new DialogInterface.OnClickListener() {
195
196         public void onClick(DialogInterface dialog, int which) {
197
198             // do nothing
199
200         }
201
202     })
203
204     .setIcon(android.R.drawable.ic_dialog_alert)
205
206     .show();
207
208     return
209     true;
210
211     case
212     R.id.action_edit_scenario:
213
214         //untuk
215         edit scenario
216
217         try {
218
219             String replyQueueName =
220             MainActivity.channel.queueDeclare().getQueue();
221
222             AMQP.BasicProperties props = new AMQP.BasicProperties
223
224                 .Builder()

```

```

201         .correlationId(MainActivity.phoneid)
202         .replyTo(replyQueueName)
203         .build();
204     JSONObject obj = new JSONObject();
205     obj.put("head", "MD");
206     obj.put("homeid", MainActivity.homeid);
207     obj.put("type", "editscenario_req");
208     obj.put("name", names.get(position));
209     StringWriter toSent = new StringWriter();
210     obj.writeJSONString(toSent);
211     String message = toSent.toString();
212     MainActivity.channel.basicPublish("", MainActivity.homeid +
213     "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
214
215     final BlockingQueue<String> response = new
216     ArrayBlockingQueue<>(1);
217
218     MainActivity.channel.basicConsume(replyQueueName, true, new
219     DefaultConsumer(MainActivity.channel) {
220
221         @Override
222
223         public void handleDelivery(String consumerTag, Envelope
224         envelope, AMQP.BasicProperties properties, byte[] body) throws
225         IOException {
226
227             if
228             (properties.getCorrelationId().equals(MainActivity.phoneid)) {
229
230                 response.offer(new String(body, "UTF-8"));
231
232             }
233
234         }
235     });
236
237     String res = MainActivity.AESdecrypt(response.take());
238
239     JSONParser parser = new JSONParser();
240
241     JSONObject jsonObject = (JSONObject) parser.parse(res);

```

```

229     ListAddDeviceRoom.checked = (ArrayList<ArrayList<String>>)
jsonObject.get("devices");
230
231     String time = (String) jsonObject.get("time");
232
233     Intent afterloginIntent = new Intent(context,
AddScenarioActivity.class);
234
235     afterloginIntent.putExtra("type", "edit");
236
237     afterloginIntent.putExtra("name", names.get(position));
238
239     afterloginIntent.putExtra("time", time);
240
241     context.startActivity(afterloginIntent);
242
243     } catch
244     (Exception e) {
245
246     }
247
248
249     //BILA TOMBOL SWITCH SCENARIO DIKLIK, MAKA SKENARIO
DIAKTIFKAN:
250     switchscen.setOnClickListener(new
View.OnClickListener() {
251
252         @Override
253         public void onClick (View view) {
254             try {
255                 JSONObject obj = new JSONObject();
256                 obj.put("head", "SC");
257                 obj.put("homeid",
MainActivity.homeid);
258                 obj.put("name",
names.get(position));
259
260                 StringWriter toSent = new
StringWriter();
261
262                 obj.writeJSONString(toSent);
263                 String message =
264
AMQP.BasicProperties props = new
AMQP.BasicProperties
265
.Builder()
.correlationId(MainActivity.phoneid)
.build();

```

```

266     MainActivity.channel.basicPublish("", MainActivity.homeid
267     +"AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
268     } catch (Exception e) {
269         }
270     }
271 }
272 })
273
274     return listViewItem;
275 }
276
}

```

Kode ini mirip dengan kode MenuAdapter2.java. Kode ini berisi pengaturan perilaku tombol *Trash* ketika ditekan. Sama seperti MenuAdapter2, bila tombol *Trash* ditekan, maka akan muncul menu dengan dua pilihan, yaitu *delete scenario* dan *edit scenario*. Masing-masing pilihan akan membuat aplikasi mengirimkan data JSON *delete* atau *edit scenario*.

Tabel 25 Data JSON untuk Delete dan Edit Scenario

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	deletescenario	<ul style="list-style-type: none"> • Home ID • Nama skenario yang hendak dihapus 	Host dan Server	AES	Tidak ada jawaban
	editscenario	<ul style="list-style-type: none"> • Home ID • Nama skenario yang hendak diedit • Nama Baru • Daftar device baru di dalam skenario berikut statusnya 	Host dan Server	AES	Tidak ada jawaban

Selain itu, pada kode ListScenario.java juga diatur perilaku tombol *scenario switch* ketika ditekan. Ketika *scenario switch* ditekan, maka aplikasi akan mengirimkan perintah pengaktifan skenario dengan data JSON sebagai berikut.

Tabel 26 Data JSON Pengaktifan Skenario

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
SC (Scenario)	-	• Home ID	Host	AES	Tidak ada jawaban

Selain MainActivity.java, kode java lain yang berperan untuk halaman utama adalah SensorService.java. SensorService.java adalah kode yang berisi algoritma yang dijalankan pada fitur *Service* Android. *Service* merupakan suatu fitur di dalam perancangan aplikasi Android di mana aplikasi dapat menjalankan *thread* tertentu di dalam ponsel meskipun aplikasi telah ditutup. Isi kode SensorService.java adalah sebagai berikut.

```
1 package com.example.billyausten.newfinal;
2
3 import android.Manifest;
4 import android.app.AlarmManager;
5 import android.app.Notification;
6 import android.app.NotificationManager;
7 import android.app.PendingIntent;
8 import android.app.Service;
9 import android.content.Context;
10 import android.content.Intent;
11 import android.content.pm.PackageManager;
12 import android.location.Location;
13 import android.location.LocationListener;
14 import android.location.LocationManager;
15 import android.net.Uri;
16 import android.os.Bundle;
17 import android.os.Handler;
18 import android.os.IBinder;
19 import android.os.Message;
20 import android.support.annotation.Nullable;
21 import android.support.v4.app.ActivityCompat;
22 import android.support.v4.app.NotificationCompat;
23 import android.support.v4.widget.DrawerLayout;
24 import android.util.Base64;
25 import android.util.Log;
26 import android.view.View;
27 import android.widget.AdapterView;
28 import android.widget.Button;
29 import android.widget.EditText;
30 import android.widget.TextView;
31 import android.widget.Toast;
32 import android.content.SharedPreferences;
33 import android.os.PowerManager;
34
35 import com.rabbitmq.client.AMQP;
36 import com.rabbitmq.client.Channel;
37 import com.rabbitmq.client.Connection;
38 import com.rabbitmq.client.ConnectionFactory;
39 import com.rabbitmq.client.QueueingConsumer;
40
```

```

41 import org.json.simple.JSONObject;
42 import org.json.simple.parser.JSONParser;
43
44 import java.io.StringWriter;
45 import java.util.ArrayList;
46 import java.util.Arrays;
47 import java.util.Date;
48 import java.util.Timer;
49 import java.util.TimerTask;
50
51 import javax.crypto.Cipher;
52 import javax.crypto.spec.IvParameterSpec;
53 import javax.crypto.spec.SecretKeySpec;
54
55
56 public class SensorService extends Service {
57
58     //DEKLARASI VARIABEL
59     ConnectionFactory factory;
60     String server_uri =
61         "amqp://mindSTA:minds@167.205.24.10:5672/mindSTA132";
62     public static Connection connection;
63     public static Channel channel;
64     public static String phone_id;
65     public static String home_id;
66     JSONParser parser = new JSONParser();
67     Thread subscribeThread;
68     public static Double lngcur;
69     public static Double latcur;
70     Double lngstart;
71     Double latstart;
72     LocationManager locationManager;
73     LocationListener locationListenerNetwork;
74     PendingIntent pendingIntent;
75     AlarmManager manager;
76
77     //variabel enkripsi AES
78     public static String iv_string = "RandomInitVector";
79     public static String AESkeyString;
80     public static IvParameterSpec iv;
81     public static SecretKeySpec skeySpec;
82     public static Cipher aescipher;
83
84     public SensorService(Context applicationContext) {
85         super();
86     }
87
88     public SensorService() {
89     }
90     @Override
91     public int onStartCommand(Intent intent, int flags, int
92 startId) {
93         //INI ADALAH FUNGSI YANG DIJALANKAN KETIKA SERVICE
94         //PERTAMA KALI START
95         super.onStartCommand(intent, flags, startId);
96         factory = new ConnectionFactory();
97         //memperoleh informasi user dari memori

```

```

95         SharedPreferences sharedPref =
getSharedPreferences("Authentication", MODE_PRIVATE);
96         phone_id = sharedPref.getString("phoneid","");
97         home_id = sharedPref.getString("homeid","");
98         AESkeyString = sharedPref.getString("aeskey", "");
99         /*PowerManager pm = (PowerManager)
getSystemService(Context.POWER_SERVICE);
100        PowerManager.WakeLock wakelock =
pm.newWakeLock(PowerManager.PARTIAL_WAKE_LOCK,
getClass().getCanonicalName());
101        wakelock.acquire();*/
102
103        //menjalankan foreground service
104        activateForeground();
105
106        //SETUP AES KEY
107        try {
108            iv = new
IvParameterSpec(iv_string.getBytes());
109            skeySpec = new
SecretKeySpec(AESkeyString.getBytes("UTF-8"), "AES");
110            aescipher =
Cipher.getInstance("AES/CFB8/NoPadding");
111        } catch (Exception e) {
112
113        }
114
115        //INITIALIZATION (posisi)
116        lngstart = 0.0;
117        latstart = 0.0;
118        lngcur = 0.0;
119        latcur = 0.0;
120
121        //SETUP SUBSCRIBE FOR NOTIFICATION
122        setupSubscribe();
123
124        //SETUP LOCATION MANAGER
125        locationManager = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
126        locationListenerNetwork = new LocationListener() {
127            public void onLocationChanged(Location
location) {
128                lngcur = location.getLongitude();
129                latcur = location.getLatitude();
130            }
131            @Override
132            public void onStatusChanged(String s, int i,
Bundle bundle) {
133
134            }
135            @Override
136            public void onProviderEnabled(String s) {
137
138            }
139            @Override
140            public void onProviderDisabled(String s) {
141
}
141

```

```

142             if (ActivityCompat.checkSelfPermission(this,
143                     android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
144                     ActivityCompat.checkSelfPermission(this,
145                     Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
146                 //return;
147             }
148         }
149
150     @Override
151     public void onDestroy() {
152         super.onDestroy();
153         //KETIKA SERVICE DIMATIKAN, SEMUA THREAD DIMATIKAN
154         stopForeground(true);
155         if (ActivityCompat.checkSelfPermission(this,
156                     android.Manifest.permission.ACCESS_FINE_LOCATION) != PackageManager.PERMISSION_GRANTED &&
157                     ActivityCompat.checkSelfPermission(this,
158                     Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED) {
159             //return;
160         }
161
162         //FUNGSI UNTUK AES ENCRYPTION
163         public static String AESEncrypt (String input) {
164             String plain_text = "";
165             try{
166                 aescipher.init(Cipher.ENCRYPT_MODE, skeySpec,
167                             iv);
168                 byte[] a = iv_string.getBytes();
169                 byte[] b =
aescipher.doFinal(input.getBytes());
170                 byte[] c = new byte[a.length + b.length];
171                 System.arraycopy(a, 0, c, 0, a.length);
172                 System.arraycopy(b, 0, c, a.length, b.length);
173                 plain_text = new String
(Base64.encodeToString(c,Base64.DEFAULT));
174             } catch (Exception e) {
175                 System.err.println("Caught Exception: " +
e.getMessage());
176             }
177         }
178
179         //FUNGSI UNTUK AES DECRYPTION
180         public static String AESdecrypt (String
encrypted_encoded_string) {
181

```

```

182         String plain_text = "";
183         try{
184             byte[] encrypted_decoded_bytes =
Base64.decode(encrypted_encoded_string, Base64.DEFAULT);
185             aescipher.init(Cipher.DECRYPT_MODE, skeySpec,
iv);
186             plain_text = new
String(aescipher.doFinal(Arrays.copyOfRange(encrypted_decoded_bytes,
16, encrypted_decoded_bytes.length))); //Returns garbage characters
187             return plain_text;
188         } catch (Exception e) {
189             System.err.println("Caught Exception: " +
e.getMessage());
190         }
191         return plain_text;
192     }
193
194     //FUNGSI UNTUK SET UP SUBSCRIBE KE MAIN QUEUE
195     public void setupSubscribe() {
196         try {
197             factory.setAutomaticRecoveryEnabled(false);
198             factory.setUri(server_uri);
199
200         } catch (Exception e) {
201             Toast.makeText(getApplicationContext(),"Failed
to Connect",
202                     Toast.LENGTH_SHORT).show();
203         }
204         //ALGORITMA YANG AKAN DIJALANKAN KETIKA SUATU PESAN
205         //DATANG KE MAIN QUEUE
206         final Handler incomingMessageHandler = new Handler()
207         {
208             @Override
209             public void handleMessage(Message msg) {
210                 //dekripsi pesan
211                 String message =
AESdecrypt(msg.getData().getString("msg"));
212
213                 Toast.makeText(getApplicationContext(),message,Toast.LENGTH_SHOR
T).show();
214                 try {
215                     //menerjemahkan pesan
216                     JSONObject jsonObject =
(JSONObject) parser.parse(message);
217                     String head = (String)
jsonObject.get("head");
218                     if (head.equals("LC")) {
219                         if (SensorService.latcur !=
0.0 && SensorService.lngcur != 0.0) {
220                             try {
221                                 //mengirim
222                                 posisi
223
AMQP.BasicProperties props = new AMQP.BasicProperties

```

```
223     .Builder()
224     .correlationId(SensorService.phone_id)
225     .build();
226
227     JSONObject obj = new JSONObject();
228
229     obj.put("head",
230             "LC");
231     obj.put("homeid", SensorService.home_id);
232     obj.put("lat",
233             Double.toString(SensorService.latcur));
234     obj.put("lng",
235             Double.toString(SensorService.lngcur));
236
237     StringWriter toSent = new StringWriter();
238     toSent.writeJSONString(toSent);
239     String tosendstring = toSent.toString();
240
241     Toast.makeText(getApplicationContext(), Double.toString(SensorService.latcur) + " " +
242             Double.toString(SensorService.lngcur), Toast.LENGTH_SHORT).show();
243
244     SensorService.channel.basicPublish("", SensorService.home_id+"AES", props,
245             SensorService.AESencrypt(tosendstring).getBytes("UTF-8"));
246
247     SensorService.lngcur = 0.0;
248     SensorService.latcur = 0.0;
249
250     } catch (Exception e) {
251
252     Toast.makeText(getApplicationContext(), "Failed to Update
253             Location", Toast.LENGTH_SHORT).show();
254
255     }
256
257     }
258
259     else if (head.equals("CO")) {
260         //mengupdate posisi switch
261
262         String address = (String)
263             jsonObject.get("address");
264
265         String homeid_temp =
266             jsonObject.get("homeid");
267
268         String command = (String)
269             jsonObject.get("command");
270
271         if
272             (homeid_temp.equals(MainActivity.homeid)) {
```

```

254     MainActivity.update_device(address, command);
255
256             }
257         } if (head.equals("SC")) {
258             //mengupdate status locked
259             atau unlocked
260             String scen_name =
261             (String)jsonObject.get("address");
262             if
263             (scen_name.equals("lock_scen")) {
264                 MainActivity.ivlock.setVisibility(View.INVISIBLE);
265                 MainActivity.ivunlock.setVisibility(View.VISIBLE);
266                 MainActivity.llback.startAnimation(MainActivity.Fabbacklocked);
267                 MainActivity.lllock.startAnimation(MainActivity.Fablocked);
268                 MainActivity.burgermen.setClickable(false);
269                 MainActivity.settingmen.setClickable(false);
270                 MainActivity.tw.setEnabled(false);
271                 MainActivity.fab.setClickable(false);
272                 MainActivity.gridView.setEnabled(false);
273                 MainActivity.gridView2.setEnabled(false);
274                 MainActivity.listView3.setEnabled(false);
275                 MainActivity.host.setCurrentTab(0);
276                 MainActivity.drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_LOCKED_CLOSED);
277             } else if
278             (scen_name.equals("unlock_scen")) {
279                 MainActivity.ivlock.setVisibility(View.VISIBLE);
280                 MainActivity.ivunlock.setVisibility(View.INVISIBLE);
281                 MainActivity.llback.startAnimation(MainActivity.Fabbackunlocked);
282                 MainActivity.lllock.startAnimation(MainActivity.Fabunlocked);
283                 MainActivity.burgermen.setClickable(true);
284                 MainActivity.settingmen.setClickable(true);
285                 MainActivity.tw.setEnabled(true);
286                 MainActivity.fab.setClickable(true);

```

```

283     MainActivity.gridView.setEnabled(true);
284     MainActivity.gridView2.setEnabled(true);
285     MainActivity.listView3.setEnabled(true);
286     MainActivity.drawer.setDrawerLockMode(DrawerLayout.LOCK_MODE_UNLOCKED);
287             }
288         }
289         else if (head.equals("RE")) {
290             //mengupdate isi mainList,
291             list room, atau pun list skenario, dan memberikan notifikasi
292             String type = (String)
293             jsonObject.get("type");
294             if (type.equals("devices"))
295                 addNotif((String)
296             jsonObject.get("notifsender"), (String)
297             jsonObject.get("notifmessage"));
298             NotificationActivity.notifs = (ArrayList<ArrayList<String>>)
299             jsonObject.get("notifications");
300             MainActivity.mainList
301             = (ArrayList<ArrayList<String>>) jsonObject.get("devices");
302             Intent switchintent =
303             new Intent(SensorService.this, SwitchActivity.class);
304             SensorService.this.startActivity(switchintent);
305             //((BaseAdapter)
306             SwitchActivity.listView.setAdapter()).notifyDataSetChanged();
307             } else if
308             (type.equals("rooms")) {
309             ArrayList<ArrayList<String>> rooms =
310             (ArrayList<ArrayList<String>>) jsonObject.get("rooms");
311             addNotif((String)
312             jsonObject.get("notifsender"), (String)
313             jsonObject.get("notifmessage"));
314             NotificationActivity.notifs = (ArrayList<ArrayList<String>>)
315             jsonObject.get("notifications");
316             try {
317                 MainActivity.mainList = (ArrayList<ArrayList<String>>)
318                 jsonObject.get("devices");
319                 MainActivity.menus2.clear();
320                 for
321                 (ArrayList<String> element : rooms) {
322                     MainActivity.menus2.add(element.get(0));
323                 }
324                 ((BaseAdapter)
325                 MainActivity.gridView2.setAdapter()).notifyDataSetChanged();

```

```

312                                         } catch (Exception e)
313
314     Toast.makeText(getApplicationContext(), "failed to update room",
315     Toast.LENGTH_SHORT).show();
316                                         }
317 } else if
318     (type.equals("scenarios")) {
319                                         addNotif((String)
320     jsonObject.get("notifsender"), (String)
321     jsonObject.get("notifmessage"));
322
323     NotificationActivity.notifs = (ArrayList<ArrayList<String>>)
324     jsonObject.get("notifications");
325                                         try {
326
327     MainActivity.menus3.clear();
328                                         for
329     (ArrayList<String> element : scenarios) {
330
331         MainActivity.menus3.add(element.get(0));
332                                         }
333                                         ((BaseAdapter)
334     MainActivity.listView3.getAdapter()).notifyDataSetChanged();
335                                         } catch (Exception e)
336
337     Toast.makeText(getApplicationContext(), "failed to update
338     scenarios",
339                                         scenarios);
340                                         }
341                                         }
342                                         }
343                                         }
344                                         } catch (Exception e) {
345
346     Toast.makeText(getApplicationContext(), "Failed to
347     Update", Toast.LENGTH_SHORT).show();
348                                         }

```

```

347
348
349         }
350     );
351     subscribe(incomingMessageHandler);
352 }
353
354 void subscribe(final Handler handler)
355 {
356     subscribeThread = new Thread(new Runnable() {
357         @Override
358         public void run() {
359             while(true) {
360                 try {
361                     connection =
362                         factory.newConnection();
363                     channel =
364                         connection.createChannel();
365                     channel.basicQos(1);
366                     channel.queueDeclare(phone_id, false, false, false, null);
367                     //AMQP.Queue.DeclareOk q =
368                     channel.queueDeclare();
369                     //channel.queueBind(q.getQueue(), "", MainActivity.homeid+"RE");
370                     QueueingConsumer consumer =
371                     new QueueingConsumer(channel);
372                     channel.basicConsume(phone_id, true, consumer);
373                     // Process deliveries
374                     while (true) {
375                         QueueingConsumer.Delivery delivery =
376                             consumer.nextDelivery();
377                         String message = new
378                         String(delivery.getBody());
379                         Message msg =
380                         handler.obtainMessage();
381                         Bundle bundle = new
382                         Bundle();
383                         bundle.putString("msg", message);
384                         msg.setData(bundle);
385                         handler.sendMessage(msg);
386                     }
387                 } catch (InterruptedException e) {
388                     break;
389                 } catch (Exception e1) {
390                     Toast.makeText(getApplicationContext(),
391                         "Reconnecting...", Toast.LENGTH_SHORT).show();
392                     try {
393                         Thread.sleep(4000);
394                     } catch
395                     (InterruptedException e) {

```

```

388                                     //break;
389
390                                 }
391                         }
392                     }
393                 );
394             subscribeThread.start();
395         }
396
397     @Nullable
398     @Override
399     public IBinder onBind(Intent intent) {
400         return null;
401     }
402
403     //FUNGSI UNTUK SETUP FOREGROUND SERVICE
404     private void activateForeground() {
405         Intent notificationIntent = new Intent(this,
MainActivity.class);
406         PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent, 0);
407
408         Notification notif = new Notification.Builder(this)
409             .setContentTitle("MINDS")
410             .setContentText("Logged in as " +
MainActivity.usname)
411             .setSmallIcon(R.drawable.g6356)
412             .setContentIntent(contentIntent)
413             .build();
414         startForeground(1, notif);
415     }
416
417     //FUNGSI UNTUK MEMUNCULKAN WARNING NOTIFICATION
418     private void addWarningNotif() {
419         Intent notificationIntent = new Intent(this,
MainActivity.class);
420         PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent,
421             ///
PendingIntent.FLAG_UPDATE_CURRENT);
422             0);
423         Uri sound = Uri.parse("android.resource://" +
getPackageName() + "/" + R.raw.warning_ringtone);
424
425         NotificationCompat.Builder builder =
426             new NotificationCompat.Builder(this)
427
428             .setSmallIcon(R.drawable.g6356)
429             .setContentTitle("WARNING")
430             .setContentText("Someone is
trying to break through your home")
431             .setVibrate(new
long[]{NotificationCompat.DEFAULT_VIBRATE})
432             .setPriority(NotificationCompat.PRIORITY_MAX)
433             .setSound(sound)

```

```

433         .setDefaults(NotificationCompat.DEFAULT_LIGHTS |
NotificationCompat.DEFAULT_VIBRATE);
434
435
436         builder.setContentIntent(contentIntent);
437
438         // Add as notification
439         int m = (int) ((new Date().getTime() / 1000L) %
Integer.MAX_VALUE);
440         NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
441         manager.notify(m, builder.build());
442     }
443
444     //FUNGSI UNTUK MEMUNCULKAN NOTIFIKASI BIASA
445     private void addNotif(String sender, String message) {
446         Intent notificationIntent = new Intent(this,
MainActivity.class);
447         PendingIntent contentIntent =
PendingIntent.getActivity(this, 0, notificationIntent,
448             // 
PendingIntent.FLAG_UPDATE_CURRENT);
449             0);
450         Uri sound = Uri.parse("android.resource://" +
getPackageName() + "/" + R.raw.notif_ringtone);
451
452         NotificationCompat.Builder builder =
453             new NotificationCompat.Builder(this)
454
        .setSmallIcon(R.drawable.g6356)
455                     .setContentTitle(sender)
456                     .setContentText(message)
457                     .setVibrate(new
long[]{NotificationCompat.DEFAULT_VIBRATE})
458
        .setPriority(NotificationCompat.PRIORITY_MAX)
459                     .setSound(sound)
460
        .setDefaults(NotificationCompat.DEFAULT_LIGHTS |
NotificationCompat.DEFAULT_VIBRATE);
461
462
463         builder.setContentIntent(contentIntent);
464
465         // Add as notification
466         int m = (int) ((new Date().getTime() / 1000L) %
Integer.MAX_VALUE);
467         NotificationManager manager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
468         manager.notify(m, builder.build());
469     }
470
471 }

```

Hal-hal penting yang dijalankan di *service* adalah sebagai berikut.

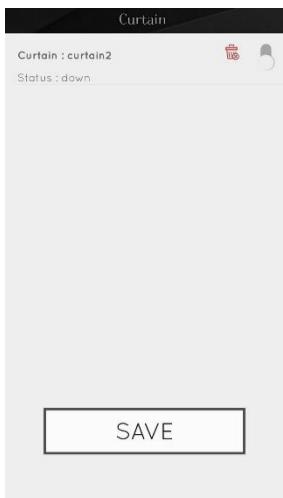
- Service melakukan *subscribe* terhadap antrian dengan nama yang sesuai dengan *phone ID*.
- Salah satu pesan yang mungkin masuk ke antrian ini adalah pesan *refresh*. Isi dari *refresh* adalah *update device*, *rooms*, atau *scenarios* baru di dalam rumah. Dengan demikian, elemen baru ini akan ditambahkan ke *ArrayList* dalam aplikasi dan ditampilkan pula ke layar.
- Selain membawa data *update*, pesan yang masuk ke antrian ini juga berisi notifikasi, misalnya notifikasi penambahan *device*, pengurangan *room*, dll.
- Kemungkinan pesan lain yang masuk ke antrian adalah pesan untuk mengirimkan lokasi terkini ponsel. Pesan ini dikirimkan oleh *host* setiap 5 menit supaya *host* dapat mengupdate lokasi setiap *user* dan melakukan kalkulasi apakah *user* sedang berada di dalam area rumah atau tidak. Bila aplikasi menerima pesan ini, maka aplikasi akan segera mengirimkan informasi lokasi (dari GPS) ke *host*.
- Jenis pesan lain yang mungkin masuk adalah pesan *warning*. Pesan ini dikirim oleh *host* bila ada pihak ilegal yang mencoba menerobos rumah ketika rumah sedang dalam keadaan *locked*. Pesan *warning* akan men-trigger aplikasi mengeluarkan notifikasi *warning*.

6.6 Halaman Device (*SwitchActivity.java*)

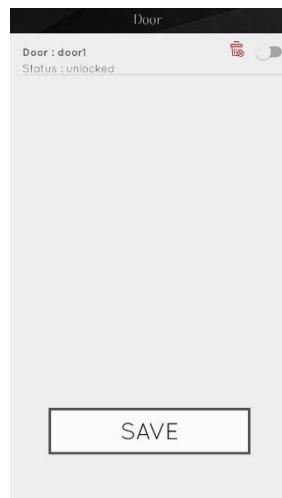
Pada bagian ini dijelaskan 6 buah *Device Activity* dalam *Tab Device* pada *MainActivity.java*. Pada saat pengguna menekan salah satu menu pada tampilan *GridView device*, pengguna akan langsung diarahkan menuju *Activity* dari *device* yang bersangkutan, menu “*Switch*” menampilkan nama, status dan *switch* dari seluruh *device* yang dikategorikan sebagai *switch*, menu “*Temperature*” menampilkan seluruh hasil pengukuran suhu maupun kelembaban yang diukur oleh *mesurement device*. Menu “*Curtain*” menampilkan seluruh nama, status dan *Switch* dari tirai yang sudah terpasang *device* untuk menurunkan ataupun menaikan tirai. Menu “*Door*” menampilkan seluruh pintu yang telah terpasang *device* untuk mendukung sistem penguncian dengan menggunakan aplikasi MINDS, sehingga tampilan pada *DoorActivity* serupa dengan tampilan *SwitchActivity* karena hanya mengandalkan *Switch toggle*. Menu “*Fan*” menggunakan *Seek bar* dibandingkan dengan *Switch* karena untuk mengatur kipas, pengguna dapat mengatur

kecepatan putar kipas agar keadaan udara di dalam rumah sesuai dengan keinginan pengguna. Menu terakhir adalah menu “Lamp” pada menu ini, pengguna dapat menyalakan dan mematikan lampu dengan cara menekan *Switch* dan mengatur warna dari lampu RGB yang sudah terpasang *device* dengan cara memilih warna dari lampu dengan cara menekan *ImageView* berbentuk 3 lingkaran dan menggeser *pointer* warna yang ditunjukan. Untuk menampilkan hal tersebut, dibutuhkan *library* “ambilwarna” yang di-*compile* pada *project* ini. Setelah mendaftarkan *library*, *dialog* untuk menampilkan warna dapat dimunculkan dengan cara memanggil fungsi “openDialog” yang juga memanggil fungsi lainnya agar *dialog* warna dapat terbuka. Setelah *dialog* warna terbuka, pengguna dapat memilih warna dan *output* dari pemilihan warna ini adalah 8 digit angka sebagai representasi warna. 2 digit pertama merupakan transparansi warna (FF merupakan warna solid dan 00 adalah warna transparan) dan 6 digit terakhir merupakan kode warna RGB. Pada seluruh *list* pada setiap menu memiliki *ImageView* bergambar tempat sampah yang dapat digunakan untuk menghapus *device* yang dipilih.

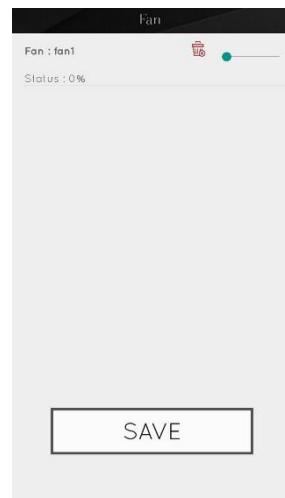
Seluruh halaman *device* disusun dengan tampilan *ListView* agar pengguna dapat melihat seluruh perangkat dengan lebih mudah. Pada setiap halaman juga terdapat *Button* “SAVE” agar pengguna dapat menyimpan perubahan yang mereka lakukan pada setiap *device*.



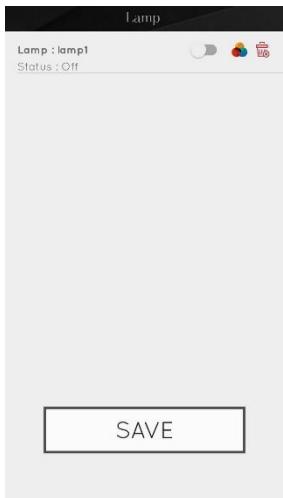
Gambar 31 Menu Curtain



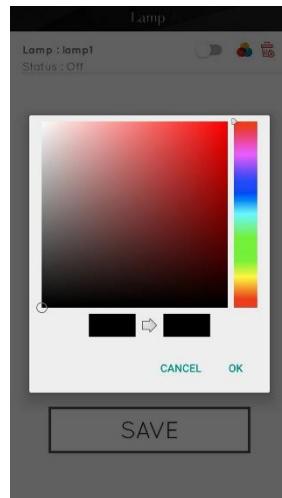
Gambar 32 Menu Door



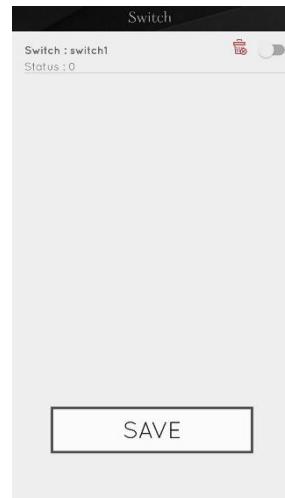
Gambar 33 Menu Fan



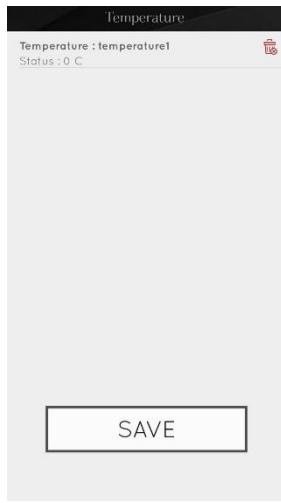
Gambar 34 Menu Lamp



Gambar 35 Dialog pada Menu
Lamp



Gambar 36 Menu Switch



**Gambar 37 Menu
Temperature**

Kode XML halaman ini adalah sebagai berikut (*activity_switch.xml*)

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:app="http://schemas.android.com/apk/res-auto"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        tools:context="com.example.billyausten.newfinal.SwitchActivity">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            android:background="?attr/colorPrimary"
            app:popupTheme="@style/PopupOverlay" />

    </android.support.design.widget.AppBarLayout>

    <include layout="@layout/content_switch" />

</android.support.design.widget.CoordinatorLayout>
```

Kode java dari halaman ini adalah sebagai berikut (*SwitchActivity.java*).

```
1 package com.example.billyausten.newfinal;
2
3 import android.graphics.Typeface;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.text.InputType;
```

```

7   import android.view.Menu;
8   import android.view.MenuItem;
9   import android.widget.Button;
10  import android.widget.TextView;
11  import android.view.View;
12  import android.widget.AdapterView;
13  import android.widget.BaseAdapter;
14  import android.widget.ListView;
15  import android.widget.Toast;
16
17  import java.util.ArrayList;
18
19  public class SwitchActivity extends AppCompatActivity {
20
21      // deklarasi variabel
22      public static ListView listView; //list view utk devices
23      ListSwitch customList; //adapter sebagai acuan listView
24      TextView activityTitle; //Judul activity
25      public static ArrayList<ArrayList<String>> devices = new
ArrayList<>(); //daftar devices yang akan ditampilkan pada listview
26
27      @Override
28      protected void onCreate(Bundle savedInstanceState) {
29          super.onCreate(savedInstanceState);
30          setContentView(R.layout.content_switch);
31
32          //PENGATURAN FONT
33          Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
34          Typeface font_title =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
35          Typeface font_or =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
BoldItalic.otf");
36          Typeface font_activity =
Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
37
38          //LISTVIEW YANG BERISI DAFTAR PERANGKAT
39          listView = (ListView) findViewById(R.id.listView);
40
41          //TULISAN JUDUL ACTIVITY
42          activityTitle = (TextView)
findViewById(R.id.activitytitle);
43          activityTitle.setTypeface(font_activity);
44
45          //TOMBOL SAVE
46          Button bsave = (Button) findViewById(R.id.bsave);
47          bsave.setTypeface(font_signin);
48          bsave.setOnClickListener(new View.OnClickListener()
{
49              @Override
50              public void onClick(View v) {
51                  finish();
52              }
53          });
54
55      }
56

```

```

57     public void onStart() {
58         super.onStart();
59
60         // onStart berisi algoritma utk seleksi devices yang
61         // termasuk ke dalam kategori yang dipilih user pada MainActivity
62         try {
63             //memperoleh informasi kategori apa yang
64             //dipilih
65             String base =
66             getIntent().getStringExtra("base");
67             String group =
68             getIntent().getStringExtra("group");
69             //activityTitle.setText(group);
70             String[] strArray = group.split(" ");
71             StringBuilder builder = new StringBuilder();
72             for (String s : strArray) {
73                 String cap = s.substring(0,
74                     1).toUpperCase() + s.substring(1);
75                 builder.append(cap + " ");
76             }
77             activityTitle.setText(builder.toString());
78
79             if (base.equals("room")) {
80                 for (ArrayList<String> element :
81                     MainActivity.mainList) {
82                     if (element.get(4).equals(group))
83                         devices.add(element);
84                 }
85             }
86         }
87     } catch (Exception e) {
88 }
89
90     //menetapkan adapter listView
91     customList = new ListSwitch(this, devices);
92     listView.setAdapter(customList);
93
94
95
96
97     //pengaturan listView
98     @Override
99     public boolean onCreateOptionsMenu(Menu menu) {
100         // Inflate the menu; this adds items to the action
101         // bar if it is present.
102         getMenuInflater().inflate(R.menu.main, menu);
103     }
104

```

```

105     @Override
106     public boolean onOptionsItemSelected(MenuItem item) {
107         // Handle action bar item clicks here. The action
bar will
108         // automatically handle clicks on the Home/Up
button, so long
109         // as you specify a parent activity in
AndroidManifest.xml.
110         int id = item.getItemId();
111
112         //noinspection SimplifiableIfStatement
113         if (id == R.id.action_settings) {
114             return true;
115         }
116
117         return super.onOptionsItemSelected(item);
118     }
119
120     @Override
121     public void onStop()
122     {
123         //onStop berisi penghapusan ArrayList devices
124         super.onStop();
125         devices.clear();
126     }
127 }
```

Bagian penting dari kode java di atas adalah bagian onStart. OnStart berisi algoritma untuk menyortir *device* apa saja yang akan ditampilkan ke layar. Ketika pada OnStart, aplikasi membaca *extra* dari *intent* untuk mengetahui kategori apakah yang hendak dibuka oleh *user*. Pembagian kategori dibagi dua, yaitu berdasarkan tipe atau berdasarkan *room*. Bila kategori dipilih berdasarkan tipe *device*, maka *device* yang akan ditampilkan disortir berdasarkan tipe. Bila kategori dipilih berdasarkan *room*, maka *device* disortir berdasarkan *room*. Daftar keseluruhan *device* diperoleh dari mainList pada MainActivity. Daftar *device* yang telah disortir dan akan ditampilkan ditampung di dalam arraylist bernama *devices*.

Setelah daftar *device* diperoleh, barulah setiap *device* tersebut ditampilkan ke layar. Tampilan di layar tentu bergantung pada jenis *device*. Tampilan untuk setiap *device* diatur dalam masing-masing kode xml *device*.

Kode list_switch.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
```

```

        android:textStyle="bold"
        android:id="@+id/textViewName"
        android:layout_row="0"
        android:layout_column="1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:id="@+id/textViewDesc"

        android:textStyle="italic"
        android:layout_row="1"
        android:layout_column="1" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_columnWeight="1"
        android:layout_marginRight="10dp"
        android:src="@drawable/trashr"
        android:layout_gravity="right"
        android:layout_row="0"
        android:id="@+id/ivtrash"/>

    <Switch
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/switch1"
        android:layout_row="0"

        android:textColorLink="?android:attr/colorActivatedHighlight"/>
    </GridLayout>

```

Kode list_curtain.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginBottom="8dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:id="@+id/textViewName"
        android:layout_row="0"
        android:layout_column="1" />

    <TextView

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:id="@+id/textViewDesc"
        android:layout_marginTop="4dp"
        android:textStyle="italic"
        android:layout_row="1"
        android:layout_column="1" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_columnWeight="1"
        android:layout_marginRight="10dp"
        android:layout_marginTop="4dp"
        android:src="@drawable/trashr"
        android:layout_gravity="right"
        android:layout_row="0"
        android:id="@+id/ivtrash"/>

    <Switch
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:layout_marginBottom="4dp"
        android:id="@+id/switch1"
        android:layout_row="0"
        android:rotation="-90"

        android:textColorLink="?android:attr/colorActivatedHighlight"/>

```

</GridLayout>

Kode list_fan.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:textStyle="bold"
        android:layout_marginTop="4dp"
        android:id="@+id/textViewName"
        android:layout_row="0"
        android:layout_column="1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:id="@+id/textViewDesc"

```

```

        android:layout_marginTop="4dp"
        android:textStyle="italic"
        android:layout_row="1"
        android:layout_column="1" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_columnWeight="1"
        android:layout_marginRight="10dp"
        android:src="@drawable/trashr"
        android:layout_gravity="right"
        android:layout_row="0"
        android:id="@+id/ivtrash"/>

    <SeekBar
        android:text=""
        android:layout_height="wrap_content"
        android:id="@+id/seek1"
        android:layout_marginTop="4dp"
        android:layout_row="0"
        android:textColorLink="?android:attr/colorActivatedHighlight"
        android:layout_width="100dp" />

</GridLayout>

```

Kode list_door.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:textStyle="bold"
        android:id="@+id/textViewName"
        android:layout_row="0"
        android:layout_column="1" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:id="@+id/textViewDesc"
        android:textStyle="italic"
        android:layout_row="1"
        android:layout_column="1" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_columnWeight="1"
        android:layout_marginRight="10dp"
        android:src="@drawable/trashr"

```

```

        android:layout_gravity="right"
        android:layout_row="0"
        android:id="@+id/ivtrash"/>

    <Switch
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/switch1"
        android:layout_row="0"

    android:textColorLink="?android:attr/colorActivatedHighlight"/>

</GridLayout>

```

Kode list_temperature.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:textStyle="bold"
        android:id="@+id/textViewName"
        android:layout_row="0"
        android:layout_column="0" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Text"
        android:layout_columnWeight="1"
        android:id="@+id/textViewDesc"
        android:textStyle="italic"
        android:layout_row="1"
        android:layout_column="0" />

    <ImageView
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_columnWeight="1"
        android:layout_marginRight="10dp"
        android:src="@drawable/trashr"
        android:layout_gravity="right"
        android:layout_row="0"
        android:id="@+id/ivtrash"/>

</GridLayout>

```

Kode list_lamp.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"

```

```

        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="220dp"
            android:layout_height="wrap_content"
            android:text="New Text"
            android:layout_columnWeight="1"
            android:layout_marginTop="5dp"
            android:textStyle="bold"
            android:id="@+id/textViewName"
            android:layout_row="0"
            android:layout_column="1" />

        <Switch
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="0dp"
            android:layout_gravity="center"
            android:id="@+id/switch1"
            />

        <ImageView
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:layout_marginLeft="10dp"
            android:src="@drawable/rgb"
            android:layout_gravity="center"
            android:id="@+id/ivrgb"/>

        <ImageView
            android:layout_width="20dp"
            android:layout_height="20dp"
            android:layout_marginLeft="10dp"
            android:src="@drawable/trashr"
            android:layout_gravity="center"
            android:id="@+id/ivtrash"/>

    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="New Text"
            android:id="@+id/textViewDesc"
            android:textStyle="italic" />

```

```
</LinearLayout>  
</LinearLayout>
```

Seluruh kode XML di atas memiliki kesamaan, yaitu mengandung nama *device*, status, dan ImageView berupa *Trash*. Perbedaan terjadi pada widget pengendali *device*. Untuk switch, curtain, dan door, *widget* pengendali berupa *switch*. Untuk fan, *widget* pengendali berupa *seekbar*. Untuk Lamp, *widget* pengendali berupa *color picker*. Untuk temperature, tidak ada *widget* pengendali karena *device* temperature tidak dapat dikendalikan.

Untuk mengatur perilaku isi tampilan tersebut, dibuat kode java bernama ListSwitch.java sebagai berikut. Kode java ini berlaku untuk semua jenis *device*.

```
1 package com.example.billyausten.newfinal;  
2  
3 import android.app.Activity;  
4 import android.app.AlertDialog;  
5 import android.content.DialogInterface;  
6 import android.content.Intent;  
7 import android.graphics.Typeface;  
8 import android.media.Image;  
9 import android.support.v7.widget.PopupMenu;  
10 import android.view.LayoutInflater;  
11 import android.view.MenuItem;  
12 import android.view.View;  
13 import android.view.ViewGroup;  
14 import android.widget.ArrayAdapter;  
15 import android.widget.CompoundButton;  
16 import android.widget.ImageView;  
17 import android.widget.SeekBar;  
18 import android.widget.Switch;  
19 import android.widget.TextView;  
20  
21 import com.rabbitmq.client.AMQP;  
22 import com.rabbitmq.client.Channel;  
23 import com.rabbitmq.client.Connection;  
24 import com.rabbitmq.client.ConnectionFactory;  
25 import com.rabbitmq.client.DefaultConsumer;  
26 import com.rabbitmq.client.Envelope;  
27  
28 import org.json.simple.JSONObject;  
29 import org.json.simple.parser.JSONParser;  
30  
31 import java.io.IOException;  
32 import java.io.StringWriter;  
33 import java.util.ArrayList;  
34 import java.util.UUID;  
35 import java.util.concurrent.ArrayBlockingQueue;  
36 import java.util.concurrent.BlockingQueue;  
37 import java.math.BigInteger;  
38  
39 import yuku.ambilwarna.AmbilWarnaDialog;  
40
```

```

41     public class ListSwitch extends ArrayAdapter<ArrayList<String>>
{
42
43         //kelas sebagai adapter dari listView pada SwitchActivity
44         private Activity context;
45         private ArrayList<ArrayList<String>> devices;
46
47         //variabel untuk color picker pada lamp
48         int color;
49
50         public ListSwitch(Activity context,
ArrayList<ArrayList<String>> devices) {
51             super(context, R.layout.list_switch, devices);
52             this.context = context;
53             this.devices = devices;
54         }
55
56         @Override
57         public View getView(final int position, View convertView,
ViewGroup parent) {
58             //fungsi utk mengatur tampilan listView
59             LayoutInflator inflater =
context.getLayoutInflater();
60
61             //terbagi menjadi beberapa kategori bergantung pada
jenis device
62             if (devices.get(position).get(2).equals("switch")) {
63                 View listViewItem =
inflater.inflate(R.layout.list_switch, null, true);
64                 TextView textViewName = (TextView)
listViewItem.findViewById(R.id.textViewName);
65                 final TextView textViewDesc = (TextView)
listViewItem.findViewById(R.id.textViewDesc);
66                 Switch switch1 = (Switch)
listViewItem.findViewById(R.id.switch1);
67
68                 Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
69                 textViewName.setTypeface(font_title);
70                 Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
71                 textViewDesc.setTypeface(font_desc);
72                 textViewName.setText("Switch : " +
devices.get(position).get(0));
73                 textViewDesc.setText("Status : " +
devices.get(position).get(3));
74                 if
(devices.get(position).get(3).equals("100"))
75                     switch1.setChecked(true);
76                 else
77                     switch1.setChecked(false);
78
79                 ImageView ivtrash = (ImageView)
listViewItem.findViewById(R.id.ivtrash);
80                 ivtrash.setOnClickListener(new
View.OnClickListener() {

```

```

81             @Override
82             public void onClick(View v) {
83                 PopupMenu popup = new
84                 PopupMenu(context, v);
85                 popup.getMenuInflater().inflate(R.menu.menu_trash_dev,
86                 popup.getMenu());
87                 popup.show();
88
89                 popup.setOnMenuItemClickListener(new
90                 PopupMenu.OnMenuItemClickListener() {
91                     public boolean
92                     onMenuItemClick(MenuItem item) {
93                         switch
94                         (item.getItemId()) {
95                             case
96                             R.id.action_delete_device:
97                             new
98                             AlertDialog.Builder(context)
99                             .setTitle("Deleting Device")
100                            .setMessage("Are you sure you want to delete this device?")
101                            .setPositiveButton("Yes", new DialogInterface.OnClickListener()
102 {
103
104                 public void onClick(DialogInterface dialog, int which) {
105
106                     // menghapus device
107
108                     try {
109
110                         JSONObject obj = new JSONObject();
111                         obj.put("head", "MD");
112                         obj.put("homeid", MainActivity.homeid);
113                         obj.put("address",
114                         devices.get(position).get(1));
115                         obj.put("type", "deletedevice");
116                         obj.put("notifsender", MainActivity.usname);
117                         obj.put("notifmessage", "I just deleted device
118                         " + devices.get(position).get(0) + "'");
119
120                         StringWriter toSent = new StringWriter();
121
122                         obj.writeJSONString(toSent);
123
124                         String message = toSent.toString();
125
126                         AMQP.BasicProperties props = new
127                         AMQP.BasicProperties

```

```

108                     .Builder()
109
110             .correlationId(MainActivity.phoneid)
111             .build();
112
113             MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
114
115             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
116             MainActivity.AESencrypt(message).getBytes("UTF-8"));
117
118             /*obj.clear();
119             obj.put("head", "NO");
120             obj.put("homeid", MainActivity.homeid);
121             obj.put("sender", MainActivity.usname);
122
123             obj.put("message", "I just deleted device '" +
124             devices.get(position).get(0) + "'");
125
126             toSent = new StringWriter();
127
128             obj.writeJSONString(toSent);
129
130             message = toSent.toString();
131
132             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
133             MainActivity.AESencrypt(message).getBytes("UTF-8"));
134
135         } catch (Exception e) {
136
137             }
138
139         }
140
141         .setNegativeButton("No", new DialogInterface.OnClickListener() {
142
143             public void onClick(DialogInterface dialog, int which) {
144
145                 // do nothing
146
147             }
148
149         })

```

```

135     .setIcon(android.R.drawable.ic_dialog_alert)
136     .show();
137
138         return
139         true;
140
141         case
142
143         Intent afterloginIntent = new
144         Intent(context,AddDeviceActivity.class);
145         afterloginIntent.putExtra("type","edit");
146         afterloginIntent.putExtra("name",devices.get(position).get(0));
147         afterloginIntent.putExtra("address",devices.get(position).get(1))
148         );
149         context.startActivity(afterloginIntent);
150
151     } catch
152     (Exception e) {
153
154     }
155
156     }
157
158     switch1.setOnCheckedChangeListener(new
159     CompoundButton.OnCheckedChangeListener() {
160
161             @Override
162             public void
163             onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
164
165                 try {
166                     JSONObject obj = new
167                     JSONObject();
168
169                     obj.put("head", "CO");
170                     obj.put("homeid",
171                     MainActivity.homeid);
172                     obj.put("address",
173                     devices.get(position).get(1));
174                     obj.put("type", "switch");
175                     if (isChecked) {
176                         obj.put("command",
177                         "100");
178
179                     Typeface font_desc =
180                     Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
181                     Regular.otf");
182
183                     textViewDesc.setTypeface(font_desc);

```

```

171     textViewDesc.setText("Status : on");
172     } else {
173         obj.put("command",
174             Typeface font_desc =
175             Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
176             Regular.otf");
177         textViewDesc.setTypeface(font_desc);
178     }
179     StringWriter toSent = new
180     obj.writeJSONString(toSent);
181     String message =
182     toSent.toString();
183     AMQP.BasicProperties props =
184     new AMQP.BasicProperties
185         .Builder()
186         .correlationId(MainActivity.phoneid)
187         .build();
188
189     MainActivity.channel.basicPublish("", MainActivity.homeid +
190     "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
191     } catch (Exception e) {
192     }
193     });
194     return listViewItem;
195 } else if
196 (devices.get(position).get(2).equals("curtain")) {
197     View listViewItem =
198     inflater.inflate(R.layout.list_curtain, null, true);
199     TextView textViewName = (TextView)
200     listViewItem.findViewById(R.id.textViewName);
201     final TextView textViewDesc = (TextView)
202     listViewItem.findViewById(R.id.textViewDesc);
203     Switch switch1 = (Switch)
204     listViewItem.findViewById(R.id.switch1);
205     Typeface font_title =
206     Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
207     Bold.otf");
208     textViewName.setTypeface(font_title);
209     Typeface font_desc =
210     Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
211     Regular.otf");
212     textViewDesc.setTypeface(font_desc);
213     textViewName.setText("Curtain : " +
214     devices.get(position).get(0));
215
216     if
217     (devices.get(position).get(3).equals("100")) {

```

```

206                     switch1.setChecked(true);
207                     textViewDesc.setText("Status : up");
208                 }
209             else {
210                 switch1.setChecked(false);
211                 textViewDesc.setText("Status : down" );
212             }
213
214             switch1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
215                 @Override
216                 public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
217                     try {
218                         JSONObject obj = new
JSONObject();
219                         obj.put("head", "CO");
220                         obj.put("homeid",
MainActivity.homeid);
221                         obj.put("address",
222                             obj.put("type", "curtain");
223                         if (isChecked) {
224                             obj.put("command",
"100");
225                         Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
226
textViewDesc.setTypeface(font_desc);
227
textViewDesc.setText("Status : up");
228                     } else {
229                         obj.put("command",
"0");
230                         Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
231
textViewDesc.setTypeface(font_desc);
232
textViewDesc.setText("Status : down");
233                 }
234
235             StringWriter toSent = new
StringWriter();
236             obj.writeJSONString(toSent);
237             String message =
AMQP.BasicProperties props =
new AMQP.BasicProperties
238
.Builder()
239
.correlationId(MainActivity.phoneid)
240
.build();
241
242     MainActivity.channel.basicPublish("", MainActivity.homeid +
"AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));

```

```
243             } catch (Exception e) {  
244                 }  
245             }  
246         } );  
247     }  
248  
249     ImageView ivtrash = (ImageView)  
listViewItem.findViewById(R.id.ivtrash);  
250     ivtrash.setOnClickListener(new  
View.OnClickListener() {  
251         @Override  
252         public void onClick(View v) {  
253             PopupMenu popup = new  
PopupMenu(context, v);  
254             popup.getMenuInflater().inflate(R.menu.menu_trash_dev,  
popup.getMenu());  
255             popup.show();  
256             popup.setOnMenuItemClickListener(new  
PopupMenu.OnMenuItemClickListener() {  
257                 public boolean  
onMenuItemClick(MenuItem item) {  
258                     switch  
(item.getItemId()) {  
259                         case  
R.id.action_delete_device:  
260                             AlertDialog.Builder(context)  
261                                 .setTitle("Deleting Device")  
262                                 .setMessage("Are you sure you want to delete this device?")  
263                                 .setPositiveButton("Yes", new DialogInterface.OnClickListener()  
{  
264                         public void onClick(DialogInterface dialog, int which) {  
265                             // menghapus device  
266                             try {  
267                                 JSONObject obj = new JSONObject();  
268                                 obj.put("head", "MD");  
269                                 obj.put("homeid", MainActivity.homeid);  
270                                 obj.put("address",  
devices.get(position).get(1));  
271                                 obj.put("type", "deletedevice");  
272                                 obj.put("notifsender", MainActivity.usname);  
273                                 obj.put("notifmessage", "I just deleted device  
" + devices.get(position).get(0) + "");  
274                             } catch (JSONException e) {  
e.printStackTrace();  
275                         }  
276                     }  
277                 }  
278             }  
279         }  
280     }  
281 }  
282 }  
283 }  
284 }  
285 }  
286 }  
287 }  
288 }  
289 }  
290 }  
291 }  
292 }  
293 }  
294 }  
295 }  
296 }  
297 }  
298 }  
299 }  
300 }  
301 }  
302 }  
303 }  
304 }  
305 }  
306 }  
307 }  
308 }  
309 }  
310 }  
311 }  
312 }  
313 }  
314 }  
315 }  
316 }  
317 }  
318 }  
319 }  
320 }  
321 }  
322 }  
323 }  
324 }  
325 }  
326 }  
327 }  
328 }  
329 }  
330 }  
331 }  
332 }  
333 }  
334 }  
335 }  
336 }  
337 }  
338 }  
339 }  
340 }  
341 }  
342 }  
343 }  
344 }  
345 }  
346 }  
347 }  
348 }  
349 }  
350 }  
351 }  
352 }  
353 }  
354 }  
355 }  
356 }  
357 }  
358 }  
359 }  
360 }  
361 }  
362 }  
363 }  
364 }  
365 }  
366 }  
367 }  
368 }  
369 }  
370 }  
371 }  
372 }  
373 }  
374 }  
375 }  
376 }  
377 }  
378 }  
379 }  
380 }  
381 }  
382 }  
383 }  
384 }  
385 }  
386 }  
387 }  
388 }  
389 }  
390 }  
391 }  
392 }  
393 }  
394 }  
395 }  
396 }  
397 }  
398 }  
399 }  
400 }  
401 }  
402 }  
403 }  
404 }  
405 }  
406 }  
407 }  
408 }  
409 }  
410 }  
411 }  
412 }  
413 }  
414 }  
415 }  
416 }  
417 }  
418 }  
419 }  
420 }  
421 }  
422 }  
423 }  
424 }  
425 }  
426 }  
427 }  
428 }  
429 }  
430 }  
431 }  
432 }  
433 }  
434 }  
435 }  
436 }  
437 }  
438 }  
439 }  
440 }  
441 }  
442 }  
443 }  
444 }  
445 }  
446 }  
447 }  
448 }  
449 }  
450 }  
451 }  
452 }  
453 }  
454 }  
455 }  
456 }  
457 }  
458 }  
459 }  
460 }  
461 }  
462 }  
463 }  
464 }  
465 }  
466 }  
467 }  
468 }  
469 }  
470 }  
471 }  
472 }  
473 }  
474 }  
475 }  
476 }  
477 }  
478 }  
479 }  
480 }  
481 }  
482 }  
483 }  
484 }  
485 }  
486 }  
487 }  
488 }  
489 }  
490 }  
491 }  
492 }  
493 }  
494 }  
495 }  
496 }  
497 }  
498 }  
499 }  
500 }  
501 }  
502 }  
503 }  
504 }  
505 }  
506 }  
507 }  
508 }  
509 }  
510 }  
511 }  
512 }  
513 }  
514 }  
515 }  
516 }  
517 }  
518 }  
519 }  
520 }  
521 }  
522 }  
523 }  
524 }  
525 }  
526 }  
527 }  
528 }  
529 }  
530 }  
531 }  
532 }  
533 }  
534 }  
535 }  
536 }  
537 }  
538 }  
539 }  
540 }  
541 }  
542 }  
543 }  
544 }  
545 }  
546 }  
547 }  
548 }  
549 }  
550 }  
551 }  
552 }  
553 }  
554 }  
555 }  
556 }  
557 }  
558 }  
559 }  
560 }  
561 }  
562 }  
563 }  
564 }  
565 }  
566 }  
567 }  
568 }  
569 }  
570 }  
571 }  
572 }  
573 }  
574 }  
575 }  
576 }  
577 }  
578 }  
579 }  
580 }  
581 }  
582 }  
583 }  
584 }  
585 }  
586 }  
587 }  
588 }  
589 }  
590 }  
591 }  
592 }  
593 }  
594 }  
595 }  
596 }  
597 }  
598 }  
599 }  
600 }  
601 }  
602 }  
603 }  
604 }  
605 }  
606 }  
607 }  
608 }  
609 }  
610 }  
611 }  
612 }  
613 }  
614 }  
615 }  
616 }  
617 }  
618 }  
619 }  
620 }  
621 }  
622 }  
623 }  
624 }  
625 }  
626 }  
627 }  
628 }  
629 }  
630 }  
631 }  
632 }  
633 }  
634 }  
635 }  
636 }  
637 }  
638 }  
639 }  
640 }  
641 }  
642 }  
643 }  
644 }  
645 }  
646 }  
647 }  
648 }  
649 }  
650 }  
651 }  
652 }  
653 }  
654 }  
655 }  
656 }  
657 }  
658 }  
659 }  
660 }  
661 }  
662 }  
663 }  
664 }  
665 }  
666 }  
667 }  
668 }  
669 }  
670 }  
671 }  
672 }  
673 }  
674 }  
675 }  
676 }  
677 }  
678 }  
679 }  
680 }  
681 }  
682 }  
683 }  
684 }  
685 }  
686 }  
687 }  
688 }  
689 }  
690 }  
691 }  
692 }  
693 }  
694 }  
695 }  
696 }  
697 }  
698 }  
699 }  
700 }  
701 }  
702 }  
703 }  
704 }  
705 }  
706 }  
707 }  
708 }  
709 }  
710 }  
711 }  
712 }  
713 }  
714 }  
715 }  
716 }  
717 }  
718 }  
719 }  
720 }  
721 }  
722 }  
723 }  
724 }  
725 }  
726 }  
727 }  
728 }  
729 }  
730 }  
731 }  
732 }  
733 }  
734 }  
735 }  
736 }  
737 }  
738 }  
739 }  
740 }  
741 }  
742 }  
743 }  
744 }  
745 }  
746 }  
747 }  
748 }  
749 }  
750 }  
751 }  
752 }  
753 }  
754 }  
755 }  
756 }  
757 }  
758 }  
759 }  
760 }  
761 }  
762 }  
763 }  
764 }  
765 }  
766 }  
767 }  
768 }  
769 }  
770 }  
771 }  
772 }  
773 }  
774 }  
775 }  
776 }  
777 }  
778 }  
779 }  
780 }  
781 }  
782 }  
783 }  
784 }  
785 }  
786 }  
787 }  
788 }  
789 }  
790 }  
791 }  
792 }  
793 }  
794 }  
795 }  
796 }  
797 }  
798 }  
799 }  
800 }  
801 }  
802 }  
803 }  
804 }  
805 }  
806 }  
807 }  
808 }  
809 }  
810 }  
811 }  
812 }  
813 }  
814 }  
815 }  
816 }  
817 }  
818 }  
819 }  
820 }  
821 }  
822 }  
823 }  
824 }  
825 }  
826 }  
827 }  
828 }  
829 }  
830 }  
831 }  
832 }  
833 }  
834 }  
835 }  
836 }  
837 }  
838 }  
839 }  
840 }  
841 }  
842 }  
843 }  
844 }  
845 }  
846 }  
847 }  
848 }  
849 }  
850 }  
851 }  
852 }  
853 }  
854 }  
855 }  
856 }  
857 }  
858 }  
859 }  
860 }  
861 }  
862 }  
863 }  
864 }  
865 }  
866 }  
867 }  
868 }  
869 }  
870 }  
871 }  
872 }  
873 }  
874 }  
875 }  
876 }  
877 }  
878 }  
879 }  
880 }  
881 }  
882 }  
883 }  
884 }  
885 }  
886 }  
887 }  
888 }  
889 }  
890 }  
891 }  
892 }  
893 }  
894 }  
895 }  
896 }  
897 }  
898 }  
899 }  
900 }  
901 }  
902 }  
903 }  
904 }  
905 }  
906 }  
907 }  
908 }  
909 }  
910 }  
911 }  
912 }  
913 }  
914 }  
915 }  
916 }  
917 }  
918 }  
919 }  
920 }  
921 }  
922 }  
923 }  
924 }  
925 }  
926 }  
927 }  
928 }  
929 }  
930 }  
931 }  
932 }  
933 }  
934 }  
935 }  
936 }  
937 }  
938 }  
939 }  
940 }  
941 }  
942 }  
943 }  
944 }  
945 }  
946 }  
947 }  
948 }  
949 }  
950 }  
951 }  
952 }  
953 }  
954 }  
955 }  
956 }  
957 }  
958 }  
959 }  
960 }  
961 }  
962 }  
963 }  
964 }  
965 }  
966 }  
967 }  
968 }  
969 }  
970 }  
971 }  
972 }  
973 }  
974 }  
975 }  
976 }  
977 }  
978 }  
979 }  
980 }  
981 }  
982 }  
983 }  
984 }  
985 }  
986 }  
987 }  
988 }  
989 }  
990 }  
991 }  
992 }  
993 }  
994 }  
995 }  
996 }  
997 }  
998 }  
999 }  
1000 }
```

```

274             StringWriter toSent = new StringWriter();
275
276             obj.writeJSONString(toSent);
277
278             String message = toSent.toString();
279
280             AMQP.BasicProperties props = new
AMQP.BasicProperties
281
282                     .Builder()
283
284                     .correlationId(MainActivity.phoneid)
285
286                     .build();
287
288             MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
289
290             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
291
292             /*obj.clear();
293
294             obj.put("head", "NO");
295
296             obj.put("homeid", MainActivity.homeid);
297
298             obj.put("sender", MainActivity.usname);
299
300             obj.put("message", "I just deleted device '" +
devices.get(position).get(0) + "'");
301
302             toSent = new StringWriter();
303
304             obj.writeJSONString(toSent);
305
306             message = toSent.toString();
307
308             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
309         */
310
311     }
312
313     }
314
315 }

```

```

300     .setNegativeButton("No", new DialogInterface.OnClickListener() {
301         public void onClick(DialogInterface dialog, int which) {
302             // do nothing
303         }
304     })
305     .setIcon(android.R.drawable.ic_dialog_alert)
306     .show();
307     return true;
308     case R.id.action_edit_device:
309     try {
310
311
312         Intent afterloginIntent = new
313         Intent(context,AddDeviceActivity.class);
314         afterloginIntent.putExtra("type","edit");
315         afterloginIntent.putExtra("name",devices.get(position).get(0));
316         afterloginIntent.putExtra("address",devices.get(position).get(1));
317         context.startActivity(afterloginIntent);
318     } catch
319     (Exception e) {
320
321
322         }
323         return true;
324     }
325     });
326 }
327
328     return listViewItem;
329     } else if
330     (devices.get(position).get(2).equals("fan")) {
331         View listViewItem =
332         inflater.inflate(R.layout.list_fan, null, true);
333         TextView textViewName = (TextView)
334         listViewItem.findViewById(R.id.textViewName);
335         final TextView textViewDesc = (TextView)
336         listViewItem.findViewById(R.id.textViewDesc);
337         SeekBar seek1 = (SeekBar)
338         listViewItem.findViewById(R.id.seek1);
339

```

```

335         Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
336         textViewName.setTypeface(font_title);
337         Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
338         textViewDesc.setTypeface(font_desc);
339         textViewName.setText("Fan : " +
devices.get(position).get(0));
340         textViewDesc.setText("Status : " +
devices.get(position).get(3) + "%");
341         seek1.setProgress(Integer.parseInt(devices.get(position).get(3))
);
342         seek1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
344             @Override
345             public void onProgressChanged(SeekBar
seekBar, int progress, boolean fromUser) {
346                 Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
347                 textViewDesc.setTypeface(font_desc);
348                 textViewDesc.setText("Status : " +
progress + "%");
349             }
350         }
351         @Override
352         public void onStartTrackingTouch(SeekBar
seekBar) {
353
354     }
355
356         @Override
357         public void onStopTrackingTouch(SeekBar
seekBar) {
358             try {
359                 JSONObject obj = new
JSONObject();
360                 obj.put("head", "CO");
361                 obj.put("homeid",
MainActivity.homeid);
362                 obj.put("address",
devices.get(position).get(1));
363                 obj.put("type", "fan");
364                 obj.put("command",
Integer.toString(seekBar.getProgress()));
365                 StringWriter toSent = new
StringWriter();
366                 obj.writeJSONString(toSent);
367                 String message =
toSent.toString();
368                 AMQP.BasicProperties props =
new AMQP.BasicProperties
369                 .Builder()

```

```

370     .correlationId(MainActivity.phoneid)
371                                         .build();
372
373     MainActivity.channel.basicPublish("", MainActivity.homeid +
374                                         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
375
376                                         } catch (Exception e) {
377                                         }
378                                         }
379
380             ImageView ivtrash = (ImageView)
listViewItem.findViewById(R.id.ivtrash);
381             ivtrash.setOnClickListener(new
View.OnClickListener() {
382
@Override
383             public void onClick(View v) {
384                 PopupMenu popup = new
PopupMenu(context, v);
385
popup.getMenuInflater().inflate(R.menu.menu_trash_dev,
popup.getMenu());
386
popup.show();
387
popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
388
public boolean
onMenuItemClick(MenuItem item) {
389
switch
(item.getItemId()) {
390
case
R.id.action_delete_device:
391
new
AlertDialog.Builder(context)
392
.setDialogTitle("Deleting Device")
393
.setsetMessage("Are you sure you want to delete this device?")
394
.setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
395
public void onClick(DialogInterface dialog, int which) {
396
// menghapus device
397
try {
398
JSONObject obj = new JSONObject();
399
obj.put("head", "MD");
400
obj.put("homeid", MainActivity.homeid);
401
obj.put("address",
devices.get(position).get(1));

```

```

402             obj.put("type", "deletedevice");
403             obj.put("notifsender", MainActivity.usname);
404             obj.put("notifmessage", "I just deleted device
'"+ devices.get(position).get(0) +"'");
405             StringWriter toSent = new StringWriter();
406             obj.writeJSONString(toSent);
407             String message = toSent.toString();
408             AMQP.BasicProperties props = new
AMQP.BasicProperties
409                     .Builder()
410
        .correlationId(MainActivity.phoneid)
411                     .build();
412
        MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-
8"));
413
        MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
414
        /*obj.clear();
415
        obj.put("head", "NO");
416
        obj.put("homeid", MainActivity.homeid);
417
        obj.put("sender", MainActivity.usname);
418
        obj.put("message", "I just deleted device '" +
devices.get(position).get(0) +"'");
419
        toSent = new StringWriter();
420
        obj.writeJSONString(toSent);
421
        message = toSent.toString();
422
        MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
423 */
424
425     } catch (Exception e) {
426

```

```

427
        }
428
    }
429
}
430     .setNegativeButton("No", new DialogInterface.OnClickListener() {
431         public void onClick(DialogInterface dialog, int which) {
432             // do nothing
433         }
434     })
435     .setIcon(android.R.drawable.ic_dialog_alert)
436     .show();
437
438
439 R.id.action_edit_device:
440
441
442     Intent afterloginIntent = new
443     Intent(context,AddDeviceActivity.class);
444     afterloginIntent.putExtra("type","edit");
445     afterloginIntent.putExtra("name",devices.get(position).get(0));
446     afterloginIntent.putExtra("address",devices.get(position).get(1));
447     context.startActivity(afterloginIntent);
448
449
450
451     } catch
452     (Exception e) {
453
454
455     }
456
457
458     return listViewItem;
459
460     } else if
461     (devices.get(position).get(2).equals("door")) {
462         View listViewItem =
463         inflater.inflate(R.layout.list_door, null, true);

```

```

461             TextView textViewName = (TextView)
listViewItem.findViewById(R.id.textViewName);
462             final TextView textViewDesc = (TextView)
listViewItem.findViewById(R.id.textViewDesc);
463             Switch switch1 = (Switch)
listViewItem.findViewById(R.id.switch1);
464
465
466             Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
467             textViewName.setTypeface(font_title);
468             Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
469             textViewDesc.setTypeface(font_desc);
470             textViewName.setText("Door : " +
devices.get(position).get(0));
471             textViewDesc.setText("Status : " +
devices.get(position).get(3));
472             if
(devices.get(position).get(3).equals("100")) {
473                 switch1.setChecked(true);
474                 textViewDesc.setText("Status : locked");
475             } else {
476                 switch1.setChecked(false);
477                 textViewDesc.setText("Status :
unlocked");
478             }
479
480             switch1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
481                 @Override
482                 public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
483                     try {
484                         JSONObject obj = new
JSONObject();
485                         obj.put("head", "CO");
486                         obj.put("homeid",
MainActivity.homeid);
487                         obj.put("type", "door");
488                         obj.put("address",
devices.get(position).get(1));
489                         if (isChecked) {
490                             obj.put("command",
"100");
491                         Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
492                         textViewDesc.setTypeface(font_desc);
493                         textViewDesc.setText("Status : locked");
494                     } else {
495                         obj.put("command",
"0");

```

```

496         Typeface font_desc =
497         Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
498         textViewDesc.setTypeface(font_desc);
499         textViewDesc.setText("Status : unlocked");
500     }
501     StringWriter toSent = new
502     obj.writeJSONObject(toSent);
503     String message =
504     toSent.toString();
505     AMQP.BasicProperties props =
506     new AMQP.BasicProperties
507         .Builder()
508         .correlationId(MainActivity.phoneid)
509         .build();
510     MainActivity.channel.basicPublish("", MainActivity.homeid +
511     "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
512     } catch (Exception e) {
513     }
514   });
515   ImageView ivtrash = (ImageView)
516   listViewItem.findViewById(R.id.ivtrash);
517   ivtrash.setOnClickListener(new
518   View.OnClickListener() {
519     @Override
520     public void onClick(View v) {
521       PopupMenu popup = new
522       PopupMenu(context, v);
523       popup.getMenuInflater().inflate(R.menu.menu_trash_dev,
524       popup.getMenu());
525       popup.show();
526       popup.setOnMenuItemClickListener(new
527       PopupMenu.OnMenuItemClickListener() {
528         public boolean
529         onMenuItemClick(MenuItem item) {
530           switch
531             (item.getItemId()) {
532             case
533               R.id.action_delete_device:
534               AlertDialog.Builder(context)
535                 .setTitle("Deleting Device")
536                 .setMessage("Are you sure you want to delete this device?")

```

```

530     .setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
531         public void onClick(DialogInterface dialog, int which) {
532             // menghapus device
533             try {
534                 JSONObject obj = new JSONObject();
535                 obj.put("head", "MD");
536                 obj.put("homeid", MainActivity.homeid);
537                 obj.put("address",
devices.get(position).get(1));
538                 obj.put("type", "deletedevice");
539                 obj.put("notifsender", MainActivity.usname);
540                 obj.put("notifmessage", "I just deleted device
'" + devices.get(position).get(0) + "'");
541                 StringWriter toSent = new StringWriter();
542                 obj.writeJSONString(toSent);
543                 String message = toSent.toString();
544                 AMQP.BasicProperties props = new
AMQP.BasicProperties
545                     .Builder()
546
                .correlationId(MainActivity.phoneid)
547
                    .build();
548
                    MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-
8"));
549
                    MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
550
                    /*obj.clear();
551
                    obj.put("head", "NO");
552
                    obj.put("homeid", MainActivity.homeid);
553
                    obj.put("sender", MainActivity.usname);

```

```

554                 obj.put("message", "I just deleted device '" +
devices.get(position).get(0) + "'");
555
556                 toSent = new StringWriter();
557
558                 obj.writeJSONString(toSent);
559
560                 message = toSent.toString();
561
562             MainActivity.channel.basicPublish("", 
MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
563         */
564
565         } catch (Exception e) {
566
567     }
568
569 }
570
571 .setNegativeButton("No", new DialogInterface.OnClickListener() {
572
573     public void onClick(DialogInterface dialog, int which) {
574
575         // do nothing
576
577     }
578
579     .setIcon(android.R.drawable.ic_dialog_alert)
580
581     .show();
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2298
2299
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2398
2399
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2498
2499
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2598
2599
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2698
2699
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
27
```

```

583                                         } catch
584             (Exception e) {
585
586             return true;
587         }
588     }
589
590     });
591 }
592
593     return listViewItem;
594 } else if
595 (devices.get(position).get(2).equals("lamp")) {
596
597     View listViewItem =
598     inflater.inflate(R.layout.list_lamp, null, true);
599     listViewItem.findViewById(R.id.textViewName);
600     final TextView textViewDesc = (TextView)
601     listViewItem.findViewById(R.id.textViewDesc);
602     final Switch switch1 = (Switch)
603     listViewItem.findViewById(R.id.switch1);
604     ImageView ivrgb = (ImageView)
605     listViewItem.findViewById(R.id.ivrgb);
606
607     Typeface font_title =
608     Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
609     Bold.otf");
610     textViewName.setTypeface(font_title);
611     Typeface font_desc =
612     Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
613     Regular.otf");
614     textViewDesc.setTypeface(font_desc);
615     textViewName.setText("Lamp : " +
616     devices.get(position).get(0));
617     color = (new
618     BigInteger(devices.get(position).get(3), 16)).intValue();
619     if
620     (devices.get(position).get(3).equals("ff000000")) {
621         switch1.setChecked(false);
622         textViewDesc.setText("Status : Off");
623     } else {
624         switch1.setChecked(true);
625         textViewDesc.setText("Status : On");
626     }
627
628     switch1.setOnCheckedChangeListener(new
629     CompoundButton.OnCheckedChangeListener() {
630         @Override
631         public void
632         onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
633             if(isChecked) {
634                 try {
635                     textViewDesc.setText("Status : On");
636                 }
637             }
638         }
639     });
640 }

```

```

622                 JSONObject obj = new
623                 JSONObject();
624
625                 obj.put("head", "CO");
626                 obj.put("homeid",
627                 MainActivity.homeid);
628                 obj.put("address",
629                 devices.get(position).get(1));
630                 obj.put("type",
631                 "lamp");
632                 obj.put("command",
633                 String.format("%08x", color));
634                 StringWriter toSent =
635                 new StringWriter();
636                 obj.writeJSONObject(toSent);
637                 String message =
638                 MainActivity.channel.basicPublish("", MainActivity.homeid +
639                 "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
640
641             } catch (Exception e) {
642             }
643         }
644         textViewDesc.setText("Status : Off");
645         JSONObject obj = new
646         JSONObject();
647
648         obj.put("head", "CO");
649         obj.put("homeid",
650         MainActivity.homeid);
651         obj.put("address",
652         devices.get(position).get(1));
653         obj.put("type",
654         "lamp");
655         obj.put("command",
656         "ff000000");
657         StringWriter toSent =
658         new StringWriter();
659         obj.writeJSONObject(toSent);
660         String message =
661         MainActivity.channel.basicPublish("", MainActivity.homeid +
662         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
663
664     } catch (Exception e) {
665     }

```

```

657     MainActivity.channel.basicPublish("", MainActivity.homeid +
658     "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
659             } catch (Exception e) {
660         }
661     }
662     }
663     }
664     });
665
666     ivrgb.setOnClickListener(new
View.OnClickListener() {
667         @Override
668         public void onClick(View view) {
669             openDialog(false,
devices.get(position).get(1), switch1.isChecked());
670         }
671     });
672 });
673
674     ImageView ivtrash = (ImageView)
listViewItem.findViewById(R.id.ivtrash);
675     ivtrash.setOnClickListener(new
View.OnClickListener() {
676         @Override
677         public void onClick(View v) {
678             PopupMenu popup = new
PopupMenu(context, v);
679             popup.getMenuInflater().inflate(R.menu.menu_trash_dev,
popup.getMenu());
680             popup.show();
681
682             popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
683                 public boolean
onMenuItemClick(MenuItem item) {
684                     switch
(item.getItemId()) {
685                         case
R.id.action_delete_device:
686                         AlertDialog.Builder(context)
687                             .setTitle("Deleting Device")
688                             .setMessage("Are you sure you want to delete this device?")
689                             .setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
690
691                     public void onClick(DialogInterface dialog, int which) {
692                         // menghapus device
693
try {

```

```

692                 JSONObject obj = new JSONObject();
693
694                 obj.put("head", "MD");
695
696                 obj.put("homeid", MainActivity.homeid);
697
698                 obj.put("address",
devices.get(position).get(1));
699
700                 obj.put("type", "deletedevice");
701
702                 obj.put("notifsender", MainActivity.usname);
703
704                 obj.put("notifmessage", "I just deleted device
'" + devices.get(position).get(0) + "'");
705
706                 StringWriter toSent = new StringWriter();
707
708                 obj.writeJSONString(toSent);
709
710                 String message = toSent.toString();
711
712                 AMQP.BasicProperties props = new
AMQP.BasicProperties
713
714                 .Builder()
715
716                 .correlationId(MainActivity.phoneid)
717
718                 .build();
719
720                 MainActivity.channel.basicPublish("", "AESqueue", props,
MainActivity.AESencrypt(message).getBytes("UTF-
8"));
721
722                 MainActivity.channel.basicPublish("", MainActivity.homeid +
"AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
723
724                 /*obj.clear();
725
726                 obj.put("head", "NO");
727
728                 obj.put("homeid", MainActivity.homeid);
729
730                 obj.put("sender", MainActivity.usname);
731
732                 obj.put("message", "I just deleted device '" +
devices.get(position).get(0) + "'");
733
734                 toSent = new StringWriter();
735
736                 obj.writeJSONString(toSent);
737
738                 message = toSent.toString();

```

```

716                         MainActivity.channel.basicPublish("",  

MainActivity.homeid + "AES", props,  

MainActivity.AESencrypt(message).getBytes("UTF-8"));  

717     */  

718  

719         } catch (Exception e) {  

720  

721             }  

722         }  

723     })  

724     .setNegativeButton("No", new DialogInterface.OnClickListener() {  

725         public void onClick(DialogInterface dialog, int which) {  

726             // do nothing  

727         }  

728     })  

729     .setIcon(android.R.drawable.ic_dialog_alert)  

730     .show();  

731                                         return  

true;  

732                                         case  

R.id.action_edit_device:  

733                                         try {  

734  

735  

736             Intent afterloginIntent = new  

Intent(context,AddDeviceActivity.class);  

737             afterloginIntent.putExtra("type","edit");  

738             afterloginIntent.putExtra("name",devices.get(position).get(0));  

739             afterloginIntent.putExtra("address",devices.get(position).get(1))  

);  

740             context.startActivity(afterloginIntent);  

741         } catch  

(Exception e) {  

742  

743         }  

744  

true;  

745  

746  

747         }  

748     });
}

```

```

749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
    }
    );
    return listViewItem;
} else if
(devices.get(position).get(2).equals("temperature")) {
    View listViewItem =
inflater.inflate(R.layout.list_temperature, null, true);
    TextView textViewName = (TextView)
listViewItem.findViewById(R.id.textViewName);
    TextView textViewDesc = (TextView)
listViewItem.findViewById(R.id.textViewDesc);
    Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
    textViewName.setTypeface(font_title);
    Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
    textViewDesc.setTypeface(font_desc);
    textViewName.setText("Temperature : " +
devices.get(position).get(0));
    textViewDesc.setText("Status : " +
devices.get(position).get(3) + " C");
    ImageView ivtrash = (ImageView)
listViewItem.findViewById(R.id.ivtrash);
    ivtrash.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        PopupMenu popup = new
PopupMenu(context, v);
        popup.getMenuInflater().inflate(R.menu.menu_trash_dev,
popup.getMenu());
        popup.show();
        popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            public boolean
onMenuItemClick(MenuItem item) {
                switch
(item.getItemId()) {
                    case
R.id.action_delete_device:
                        new
AlertDialog.Builder(context)
                            .setTitle("Deleting Device")
                            .setMessage("Are you sure you want to delete this device?")
                            .setPositiveButton("Yes", new DialogInterface.OnClickListener()
{
                            public void onClick(DialogInterface dialog, int which) {
                                // menghapus device

```

```

780
781         try {
782             JSONObject obj = new JSONObject();
783             obj.put("head", "MD");
784             obj.put("homeid", MainActivity.homeid);
785             obj.put("address",
786 devices.get(position).get(1));
787             obj.put("type", "deletedevice");
788             obj.put("notifsender", MainActivity.usname);
789             obj.put("notifmessage", "I just deleted device
790             '" + devices.get(position).get(0) + "'");
791             StringWriter toSent = new StringWriter();
792             obj.writeJSONString(toSent);
793             String message = toSent.toString();
794             AMQP.BasicProperties props = new
795             AMQP.BasicProperties
796             .Builder()
797             .correlationId(MainActivity.phoneid)
798             .build();
799
800             MainActivity.channel.basicPublish("", "AESqueue", props,
801             MainActivity.AESencrypt(message).getBytes("UTF-
8"));
802             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
803             MainActivity.AESencrypt(message).getBytes("UTF-8"));
804             /*obj.clear();
805             obj.put("head", "NO");
806             obj.put("homeid", MainActivity.homeid);
807             obj.put("sender", MainActivity.usname);
808             obj.put("message", "I just deleted device '" +
809             devices.get(position).get(0) + "'");
810             toSent = new StringWriter();
811             obj.writeJSONString(toSent);

```

```

804                     message = toSent.toString();
805                     MainActivity.channel.basicPublish("", 
806                     MainActivity.homeid + "AES", props,
807                     MainActivity.AESencrypt(message).getBytes("UTF-8"));
808                     */
809             } catch (Exception e) {
810
811         }
812     }
813 }
814 .setNegativeButton("No", new DialogInterface.OnClickListener() {
815
816     public void onClick(DialogInterface dialog, int which) {
817
818         // do nothing
819     }
820
821     .setIcon(android.R.drawable.ic_dialog_alert)
822
823     .show();
824
825
826
827     Intent afterloginIntent = new
828     Intent(context,AddDeviceActivity.class);
829
830     afterloginIntent.putExtra("type","edit");
831
832     afterloginIntent.putExtra("name",devices.get(position).get(0));
833
834     afterloginIntent.putExtra("address",devices.get(position).get(1));
835
836     context.startActivity(afterloginIntent);
837
838     } catch
839     (Exception e) {
840
841
842     }
843
844     true;
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885

```

```

836                                     return true;
837                                 }
838                             });
839                         }
840                     );
841                     return listViewItem;
842                 }
843             }
844             return null;
845         }
846     }
847     void openDialog(boolean supportsAlpha, final String address, final boolean switchstat) {
848         AmbilWarnaDialog dialog = new
849         AmbilWarnaDialog(context, color, supportsAlpha, new
850         AmbilWarnaDialog.OnAmbilWarnaListener() {
851             @Override
852             public void onOk(AmbilWarnaDialog dialog, int
853             color) {
854                 //Toast.makeText(getApplicationContext(), "ok",
855                 Toast.LENGTH_SHORT).show();
856                 ListSwitch.this.color = color;
857                 if (switchstat) {
858                     try {
859                         JSONObject obj = new
860                         JSONObject();
861                         obj.put("head", "CO");
862                         obj.put("homeid",
863                         MainActivity.homeid);
864                         obj.put("address", address);
865                         obj.put("type", "lamp");
866                         obj.put("command",
867                         String.format("%08x", color));
868                         StringWriter toSent = new
869                         StringWriter();
870                         obj.writeJSONString(toSent);
871                         String message =
872                         toSent.toString();
873                         AMQP.BasicProperties props =
874                         new AMQP.BasicProperties
875                         .Builder()
876                         .correlationId(MainActivity.phoneid)
877                         .build();
878
879                         MainActivity.channel.basicPublish("", MainActivity.homeid +
880                         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
881                     } catch (Exception e) {
882                         }
883                     }
884                 }
885             }
886             @Override
887             public void onCancel(AmbilWarnaDialog dialog)
888             {

```

```

878         //Toast.makeText(getApplicationContext(), "cancel",
879         Toast.LENGTH_SHORT).show();
880     }
881 }
882 }
883 }

```

Kode di atas terbagi-bagi menjadi beberapa segmen *if* bergantung pada jenis *device*. Untuk *device* dengan pengendali *switch*, hanya ada 2 kemungkinan status, yaitu off dan on. Off direpresentasikan dengan *command* 0, dan on direpresentasikan dengan *command* 100. Untuk *seekbar*, terdapat 100 kemungkinan nilai *command* karena nilai *seekbar* fleksibel dari 0 sampai 100. Untuk color picker, bentuk *command* adalah *command* RGB. Dengan demikian, *command* terdiri dari 3 bilangan 8 bit (0 sampai 255).

Ketika ada perintah pengendalian, maka aplikasi akan mengirimkan data JSON sebagai berikut.

Tabel 27 Data JSON Perintah Pengendalian Devices

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
CO (Command)	-	<ul style="list-style-type: none"> • Home ID • Address XBee Perangkat • Isi perintah (disesuaikan dengan jenis perangkat, dijelaskan pada 3.1.4. 	<i>Host</i>	AES	Tidak ada jawaban

Selain itu, tombol *trash* berguna untuk melakukan *delete* atau *edit*. Ketika tombol ini ditekan, maka akan muncul menu *delete device* dan *edit device*. Ketika tombol *delete device* ditekan, aplikasi akan mengirimkan pesan JSON berikut.

Tabel 28 Data JSON Delete Devices

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	deletedevice	<ul style="list-style-type: none"> • Home ID • Address perangkat yang 	<i>Host</i> dan Server	AES	Tidak ada jawaban

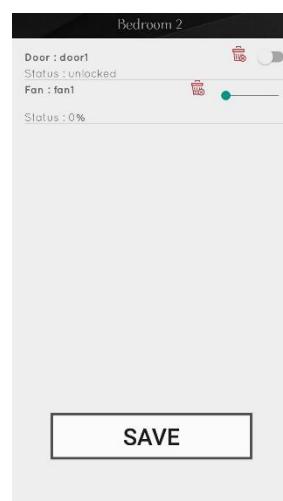
		ingin dihapus			
--	--	------------------	--	--	--

Ketika tombol *edit device* ditekan, maka halaman akan berpindah ke AddDeviceActivity dengan operasi "edit" (akan dijelaskan kemudian).

6.7 Halaman Room (*RoomActivity.java*)

Pada halaman ini, pengguna dapat memilih *device* yang akan diaktifkan berdasarkan lokasi *device* tersebut. *Tab room* digunakan untuk mempermudah pengguna memilih *device* dengan cepat dan tepat sehingga ketika pengguna berada di suatu ruangan dan ingin mengubah status dari sebuah *device*, pengguna tidak perlu mencari *device* tersebut diantara seluruh *device* dalam sistem Rumah Cerdas. Pada saat pengguna menekan salah satu ruangan pada *MainActivity.java*, pengguna akan menuju *RoomActivity.java*. *RoomActivity.java* akan menyesuaikan tampilan dan konten dari *Activity* sesuai dengan ruangan yang dipilih. Pada *RoomActivity* terdapat *list* dari *device* yang ada pada ruangan tersebut. Pengguna dapat segera mengganti status dari *device* pada *ListView* tersebut seperti pada tampilan *device* pada *Device Activity* hanya saja seluruh jenis *device* dijadikan satu.

Pada *Activity* ini, hanya digunakan *title bar*, *ListView* dan *Button* sehingga membutuhkan sebuah *file xml* dan *java* tambahan untuk mengatur tampilan pada *ListView*. *ListView* pada *RoomActivity.java* bersifat dinamis, karena berisi *device-device* dari berbagai kategori yang dapat berubah. *Button* "SAVE" berfungsi untuk menyimpan seluruh konfigurasi dan mengubah tampilan menjadi *MainActivity.java*.



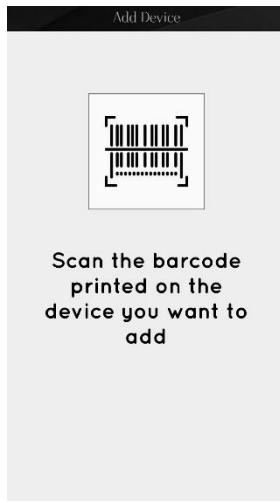
Gambar 38 Halaman Room (*RoomActivity.java*)

Cara menampilkan *ListView* pada *Activity* ini sangat mirip dengan yang ada pada *SwitchActivity.java*. Perbedaan hanya terjadi pada proses pemilihan *device*. Proses pemilihan *device* pada *Activity* ini adalah berdasarkan kolom *room*, bukan *type*. Selain daripada itu, seluruh proses keberjalanannya *Activity* ini sama dengan *SwitchActivity.java*.

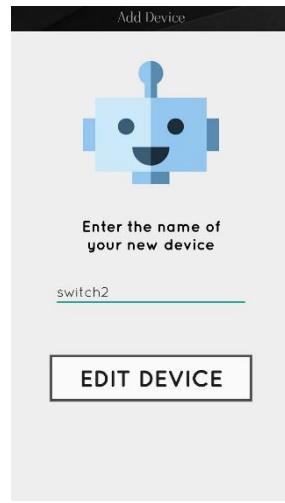
6.8 Halaman Add Device (*AddDevice1Activity.java*, *AddDeviceActivity.java*)

Halaman ini dapat diakses dari menu pada *FloatingButton* yang terdapat pada *MainActivity.java*. Pada halaman ini, pengguna dapat mendaftarkan *device* baru yang sudah terpasang di dalam rumah dengan cara melakukan *scan QR code* pada *device* yang dipasang. Karena *device* yang didaftarkan harus pada keadaan sudah aktif terpasang, *QR code* dari *device* tersebut dicetak dalam sebuah kertas terpisah agar mudah untuk dipindai. Untuk meningkatkan keamanan, *QR code* berupa data ID *device* yang sudah terenkripsi. *Scanning device QR code* dilakukan dengan menggunakan kamera *smartphone*, sehingga ketika *scan* berhasil, pengguna langsung dialihkan menuju halaman *AddDeviceActivity.java* untuk memberi nama *device* tersebut. Ketika *device* sudah diberi nama, *device* secara otomatis sudah terdaftar pada server. Pada kode setiap *device* mengandung informasi jenis *device* dan protokol *device* tersebut sehingga pengguna tidak perlu menuliskan jenis dari *device* yang didaftarkan.

Pada *AddDevice1Activity.java* pengguna dapat menekan *ImageView* bergambar *barcode* untuk melakukan *scanning barcode* dengan menggunakan kamera. Untuk melakukan *scanning barcode*, digunakan *library* diluar Android Studio, yaitu *library* dari zxing dengan kode "info.vividcode.android.zxing.CaptureActivity". Dengan menggunakan *library* ini, MINDS dapat digunakan untuk melakukan *scan* terhadap QR Code dengan cara memanggil *Intent* agar dapat beralih ke *CaptureActivity.class* yang merupakan kelas bawaan dari *library* zxing. Setelah membaca QR Code dan mendapatkan hasilnya, hasil akan disimpan pada sebuah *TextView* kosong yang berada dibawah *ImageView* bergambar barcode dan juga dikirim ke *server* untuk diberikan *feedback* berupa tambahan *device* pada MINDS.



**Gambar 39 Halaman Add Device
(AddDevice1Activity.java)**



**Gambar 40 Halaman Nama Device
(AddDeviceActivity.java)**

Kode XML halaman ini adalah sebagai berikut.

Kode activity_add_device1.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_add_device1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    android:orientation="vertical"

    tools:context="com.example.billyausten.newfinal.AddDevice1Activity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <TextView
                android:layout_width="290dp"
                android:layout_height="wrap_content"
                android:text="Add Device"
                android:id="@+id/activitytitle"
```

```

        android:textAlignment="center"
        android:layout_gravity="center_vertical"
        android:textSize="18dp"
        android:textColor="#AAAAAA"
        android:layout_marginTop="3dp"/>

    </LinearLayout>
</LinearLayout>

<ImageView
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_gravity="center"
    android:layout_marginTop="80dp"
    android:src="@drawable/adddevicecam2"
    android:id="@+id/ivbarcode"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="device number"
    android:id="@+id/tvdevicenum"
    android:layout_gravity="center"
    android:visibility="invisible"/>

<TextView
    android:layout_width="265dp"
    android:layout_marginTop="30dp"
    android:layout_height="155dp"
    android:layout_gravity="center"
    android:text="Scan the barcode printed on the device you want
to add"
    android:textSize="28dp"
    android:id="@+id/tvEnterCode"
    android:textAlignment="center"
    android:textColor="#000000"
    />
</LinearLayout>

```

Kode activity_add_device.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_add_device"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.example.billyausten.newfinal.AddDeviceActivity"
    android:descendantFocusability="beforeDescendants"
    android:focusableInTouchMode="true">

    <LinearLayout
        android:layout_width="match_parent"

```

```

        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_gravity="center">

        <TextView
            android:layout_width="290dp"
            android:layout_height="wrap_content"
            android:text="Add Device"
            android:id="@+id/activitytitle"
            android:textAlignment="center"
            android:layout_gravity="center_vertical"
            android:textSize="18dp"
            android:textColor="#AAAAAA"
            android:layout_marginTop="3dp"/>

    </LinearLayout>
</LinearLayout>

<ImageView
    android:layout_width="175dp"
    android:layout_height="175dp"
    android:layout_gravity="center"
    android:id="@+id/ivdevicename"
    android:src="@drawable/robot"
    android:layout_marginTop="30dp"/>
<TextView
    android:layout_width="220dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Enter the name of your new device"
    android:textSize="20dp"
    android:layout_marginTop="35dp"
    android:id="@+id/tvEnterCode"
    android:textAlignment="center"
    android:textColor="#000000"
    />

<EditText
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:hint="Device Name"
    android:id="@+id/etDevName"
    android:layout_marginTop="30dp"
    android:layout_gravity="center"/>

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Device"
    android:textSize="30dp"
    android:layout_marginTop="60dp"

```

```

        android:layout_gravity="center"
        android:id="@+id/badddev"
        android:background="@drawable/signin" />

    </LinearLayout>

```

Kode AddDeviceActivity1.java adalah sebagai berikut.

```

1  package com.example.billyausten.newfinal;
2
3  import android.app.Activity;
4  import android.content.Intent;
5  import android.graphics.Typeface;
6  import android.support.v4.view.GravityCompat;
7  import android.support.v4.widget.DrawerLayout;
8  import android.support.v7.app.AppCompatActivity;
9  import android.os.Bundle;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.ImageView;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 import info.vividcode.android.zxing.CaptureActivity;
18 import info.vividcode.android.zxing.CaptureActivityIntents;
19
20 public class AddDevice1Activity extends AppCompatActivity {
21
22     //DEKLARASI VARIABEL
23     TextView tvdevicenum;
24     char status;
25
26     @Override
27     protected void onCreate(Bundle savedInstanceState) {
28         super.onCreate(savedInstanceState);
29         setContentView(R.layout.activity_add_device1);
30
31         //PENGATURAN FONT
32         Typeface font_title =
33             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
34         Typeface font_activity =
35             Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
36
37         //TEXTVIEW KETERANGAN
38         TextView tvEnterCode = (TextView)
39             findViewById(R.id.tvEnterCode);
40         tvEnterCode.setTypeface(font_title);
41
42         //JUDUL ACTIVITY
43         TextView activitytitle = (TextView)
44             findViewById(R.id.activitytitle);
45         activitytitle.setTypeface(font_activity);
46
47         //TEXTVIEW PEMBACAAN QR CODE
48         tvdevicenum = (TextView)
49             findViewById(R.id.tvdevicenum);
50
51         //INISIALISASI

```

```

47             //status adalah variabel indikator
48             //jika status = '0', maka QR code belum atau tidak
berhasil terbaca
49             //jika status = '1', maka QR code terbaca
50             status = '0';
51
52             //IMAGE UNTUK MASUK KE HALAMAN QR SCANNING
53             ImageView ivScan = (ImageView)
findViewById(R.id.ivbarcode);
54             ivScan.setOnClickListener(new View.OnClickListener()
{
55                 @Override
56                 public void onClick(View view) {
57                     // Membuat intent baru untuk memanggil
CaptureActivity bawaan ZXing
58                     Intent captureIntent = new
Intent(AddDevice1Activity.this, CaptureActivity.class);
59                     // Kemudian kita mengeset pesan yang
akan ditampilkan ke user saat menjalankan QRCode scanning
60
CaptureActivityIntents.setPromptMessage(captureIntent, "Barcode
scanning...");
61                     // Melakukan startActivityForResult,
untuk menangkap balikan hasil dari QR Code scanning
62                     startActivityForResult(captureIntent,
63                     0);
64                 }
65             });
66
67             @Override
68             public void onBackPressed() {
69                 Intent afterloginIntent = new
Intent(AddDevice1Activity.this, MainActivity.class);
70
AddDevice1Activity.this.startActivity(afterloginIntent);
71             }
72
73             @Override
74             public void onStart() {
75                 super.onStart();
76                 //QRCODE BERHASIL TERBACA, statu = '1'
77                 if (status == '1') {
78                     status = '0';
79                     //Masuk ke halaman berikutnya untuk memasukan
nama device
80                     Intent afterloginIntent = new
Intent(AddDevice1Activity.this, AddDeviceActivity.class);
81                     //address dan tipe device pada QR code
disimpan dalam Intent Extra
82
afterloginIntent.putExtra("devicenum", tvdevicenum.getText().toSt
ring());
83                     afterloginIntent.putExtra("type", "add");
84
AddDevice1Activity.this.startActivity(afterloginIntent);
85             }
86

```

```

87
88
89     //FUNGSI YANG AKAN DIJALANKAN UNTUK MEMBACA QR CODE
90     @Override
91     public void onActivityResult(int requestCode, int
resultCode, Intent data) {
92
93         if (requestCode == 0) {
94             if (resultCode == Activity.RESULT_OK &&
data != null) {
95                 //QR code berhasil terbaca
96                 String value =
MainActivity.RSAdecrypt(data.getStringExtra("SCAN_RESULT"));
97                 Toast.makeText
(getApplicationContext(), value, Toast.LENGTH_SHORT).show();
98                 tvdevicenum.setText(value);
99                 status = '1';
100            } else if (resultCode ==
Activity.RESULT_CANCELED) {
101                //QR code tidak berhasil terbaca
102                tvdevicenum.setText("Scanning Failed,
Please try again");
103            }
104        } else {
105
106    }
107    super.onActivityResult(requestCode, resultCode,
data);
108}
109}

```

Bagian penting dari kode java di atas adalah:

- Pada kode ini terdapat variabel bernama status. Variabel ini diinisialisasi 0. Status bernilai 0 menandakan bahwa QR code belum berhasil terbaca. Status bernilai 1 menandakan bahwa QR code berhasil terbaca.
- Baris 53 sampai 64, yaitu bagian pengaturan perilaku tombol *scan QR Code*. Ketika tombol ini ditekan, maka tampilan akan berpindah ke gambar tangkapan kamera. Aplikasi akan mulai melakukan *QR Code Scanning*.
- Baris 91 sampai akhir, yaitu bagian untuk menentukan langkah apa yang dilakukan setelah QR Code berhasil terbaca. Bila QR code tidak berhasil terbaca, maka variabel status tetap bernilai 0. Bila QR code berhasil terbaca, maka status diubah menjadi 1. Bila status bernilai 1, maka aplikasi akan membentuk data JSON yang isinya sama persis dengan isi QR Code, yaitu alamat Xbee *Device* dan tipe *device*. Setelah itu, halaman akan berpindah ke AddDeviceActivity dengan ekstra data JSON di atas.

Kode java AddDeviceActivity.java adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.WindowManager;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import com.rabbitmq.client.AMQP;
15
16 import org.json.simple.parser.JSONParser;
17 import org.json.simple.JSONObject;
18 import org.json.simple.parser.ParseException;
19
20 import java.io.StringWriter;
21
22 public class AddDeviceActivity extends AppCompatActivity {
23
24     //DEKLARASI VARIABEL
25     JSONParser parser = new JSONParser();
26     String oldname;
27     String address;
28
29     @Override
30     protected void onCreate(Bundle savedInstanceState) {
31         super.onCreate(savedInstanceState);
32         setContentView(R.layout.activity_add_device);
33
34         //PENGATURAN FONT
35         Typeface font_title =
36             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
37         Typeface font_edit=
38             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
39         Typeface font_activity =
40             Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
41
42         //TEXTVIEW KETERANGAN
43         TextView tvEnterCode = (TextView)
44             findViewById(R.id.tvEnterCode);
45         tvEnterCode.setTypeface(font_title);
46
47         //EDITTEXT NAMA DEVICE
48         final EditText etDevName = (EditText)
49             findViewById(R.id.etDevName);
50         etDevName.setTypeface(font_edit);
51
52         //BUTTON UNTUK MENAMBAH DEVICE
53         Button badddev = (Button)
54             findViewById(R.id.badddev);
55         badddev.setTypeface(font_title);
56
57         //JUDUL ACTIVITY

```

```

52         TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
53             activitytitle.setTypeface(font_activity);
54
55             //MENGETAHUI TIPE OPERASI, APAKAH ADD ATAU EDIT
56             final String operation_type =
getIntent().getStringExtra("type");
57
58             if (operation_type.equals("edit")) {
59                 //Jika edit device
60                 activitytitle.setText("Edit Device");
61                 oldname = getIntent().getStringExtra("name");
//nama lama device
62                 address =
getIntent().getStringExtra("address");      //device address
63                 etDevName.setText(oldname);
//EditText diisi dengan nama lama
64                 badddev.setText("Edit Device");
65             }
66
67             badddev.setOnClickListener(new
View.OnClickListener() {
68
69                 @Override
70                 public void onClick(View v) {
71                     //jika add device
72                     if (operation_type.equals("add")) {
73                         //diperiksa dulu apakah nama
device terdiri dari 1 sampai 20 karakter
74                         if
(! (etDevName.getText().toString().equals("")) &&
etDevName.getText().toString().length() <= 20) {
75
76                             try {
77                                 //memperoleh address
dan tipe dari Intent Extra
78                                 String devicenum =
getIntent().getStringExtra("devicenum");
79
80                                 //menyusun kembali
pesan yang akan dikirim
81                                 JSONObject obj =
(JSONObject) parser.parse(devicenum);
82                                 obj.put("name",
etDevName.getText().toString());
83                                 obj.put("homeid",
MainActivity.homeid);
84                                 obj.put("head", "MD");
85                                 obj.put("type",
"adddevice");
86                                 obj.put("notifsender",
MainActivity.usname);
87                                 obj.put("notifmessage", "I just added device '" +
etDevName.getText().toString() + "'");
88                                 StringWriter toSent =
new StringWriter();

```

```

89         obj.writeJSONString(toSent);
90                                         String message =
91         toSent.toString();
92                                         //mengisi properti
93         pesan (correlation ID diisi phone ID)
94                                         AMQP.BasicProperties
95         props = new AMQP.BasicProperties
96                                         .Builder()
97                                         .correlationId(MainActivity.phoneid)
98                                         .build();
99                                         //mengirim pesan
100
101         MainActivity.channel.basicPublish("", "AESqueue", props,
102         MainActivity.AESencrypt(message).getBytes("UTF-8"));
103
104         MainActivity.channel.basicPublish("", MainActivity.homeid +
105         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
106                                         //kembali ke halaman
107                                         utama
108                                         finish();
109         } catch (Exception e2) {
110
111         }
112
113
114         //jika edit device
115         else if (operation_type.equals("edit"))
116         {
117             if
118             (!(etDevName.getText().toString().equals("")) &&
119             etDevName.getText().toString().length() <= 20) {
120                 try {
121
122                     //menyusun pesan yang
123                     akan dikirim
124                     JSONObject obj = new
125                     JSONobject();
126                     obj.put("oldname",
127                     oldname);
128                     obj.put("homeid",
129                     MainActivity.homeid);
130                     obj.put("head", "MD");
131                     obj.put("type",
132                     "editdevice");

```

```

125                                     obj.put("name",
etDevName.getText().toString());
126                                     obj.put("address",
address);
127                                     obj.put("notifsender",
MainActivity.usname);
128
129                                     obj.put("notifmessage", "I just edited device '" + oldname +
to '' + etDevName.getText().toString() + "'");
130                                     StringWriter toSent =
new StringWriter();
131                                     obj.writeJSONString(toSent);
132                                     String message =
toSent.toString();
133                                     //mengisi properti
pesan
134                                     AMQP.BasicProperties
props = new AMQP.BasicProperties
135                                     .Builder()
136                                     .correlationId(MainActivity.phoneid)
137                                     .build();
138                                     //mengirim pesan
139
MainActivity.channel.basicPublish("", "AESqueue", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
140
141                                     //kembali ke halaman
utama
142                                     finish();
143 } catch (Exception e2) {
144
145 }
146 } else {
147
Toast.makeText(getApplicationContext(), "Name should be between
0 to 20 character",
148
Toast.LENGTH_SHORT).show();
149
}
150
}
151
}
152
}
153
})
154
}
155
}

```

Hal penting dari kode ini adalah sebagai berikut.

- Halaman ini sebenarnya digunakan untuk 2 hal sekaligus, yaitu untuk *add device* dan *edit device*. Untuk membedakan kedua jenis operasi ini, setiap kali *activity*

terbentuk, aplikasi mengambil *intent extra* dari *activity* sebelumnya, apakah *add* atau *edit* (baris 56).

- Baris 67 merupakan bagian yang penting, yaitu pengaturan perilaku button *add device* atau *edit device*. Ketika *button* ini diklik, maka aplikasi akan mengirimkan data JSON *add device* ataupun *edit device*.

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	adddevice	<ul style="list-style-type: none"> • Home ID • Address XBee Perangkat • Tipe perangkat • Nama perangkat 	Host dan Server	AES	Tidak ada jawaban
MD (Manajemen Device)	editdevice	<ul style="list-style-type: none"> • Home ID • Address perangkat yang ingin diedit • Nama baru perangkat 	Host dan Server	AES	Tidak ada jawaban

6.9 Halaman Add Room (*EditRoomActivity.java*, *AddDeviceRoom.java*)

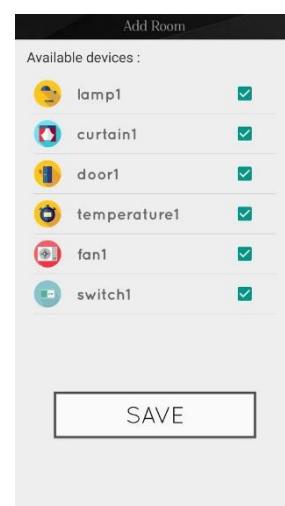
Halaman ini dapat diakses dengan menekan *Button* “Add Room” dari *FloatingButton* yang terdapat pada *MainActivity.java* atau ketika tombol *edit room* pada salah satu *room* ditekan. Pada halaman ini, pengguna dapat menambah kategori *room* yang akan ditampilkan pada *tab room*. Untuk menambah sebuah *room*, dibutuhkan nama *room* dan *device* apa saja yang berada di dalam *room* tersebut. Pada *EditRoomActivity.java*, pengguna dapat memasukan nama *room* dalam sebuah *EditText* yang berada dibawah *ImageView*. *EditText* untuk nama *room* memiliki *InputType* “textPersonName” sehingga tombol *enter* pada *keyboard* pengguna dapat berfungsi sebagai tombol *next* dan mencegah pengguna untuk memasukan nama ruangan sebanyak dua baris atau lebih. Setelah itu terdapat *TextView* “Add Device” yang berguna untuk mendaftarkan *device* ke dalam *room* tersebut. *TextView* “Add Device” akan mengantarkan pengguna ke dalam halaman baru

yaitu *AddDeviceRoom.java*, pada halaman tersebut pengguna dapat memilih *device* dari *list device* yang sudah tersedia dengan cara menekan *Checkbox* yang terletak disebelah kanan pada *list device*. Ketika sudah selesai memilih *device*, pengguna dapat menekan *Button* “ADD” untuk menyimpan konfigurasi penambahan *device* dan pengguna segera kembali pada tampilan *EditRoomActivity.java*. Jika pengguna ingin menghapus *device* yang sudah dipilih, pengguna dapat menekan tombol “x” yang terletak di sebelah nama *device* pada *EditDeviceActivity.java*.

Pada *EditDeviceActivity.java*, terdapat *ListView* yang berisi seluruh *device* yang berada pada ruangan tersebut. Dalam *list* tersebut terdapat sebuah *ImageView* yang berupa logo dari kategori jenis *device*, *TextView* yang berupa nama *device* dan juga *ImageView* yang berbentuk huruf “x” yang dapat digunakan untuk menghapus *device* dari ruangan tersebut. Pada *AddDeviceRoom.java* juga terdapat *ListView* yang berisi *ImageView* yang berupa logo dari kategori jenis *device*, *TextView* nama *device* dan juga *Checkbox* yang dapat digunakan oleh pengguna untuk memilih *device* apa saja yang ingin didaftarkan pada ruangan tersebut. Pada *EditRoomActivity.java* diberikan kode “OnBackPressed” dengan *Intent* menuju *MainActivity.java* agar pengguna tidak kembali menuju *AddDeviceRoom.java* ketika pengguna menekan tombol *back* setelah selesai menyimpan *device*.



**Gambar 41 Halaman Add Room
(EditRoomActivity.java)**



**Gambar 42 Halaman Add Device pada Room
(AddDeviceRoom.java)**

Kode XML halaman ini adalah sebagai berikut (activity_edit_room.xml).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_edit_room"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    android:orientation="vertical"
    tools:context="com.example.billyausten.newfinal.EditRoomActivity"
    android:descendantFocusability="beforeDescendants"
    android:focusableInTouchMode="true">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <TextView
                android:layout_width="290dp"
                android:layout_height="wrap_content"
                android:text="Add Room"
                android:id="@+id/activitytitle"
                android:textAlignment="center"
                android:layout_gravity="center_vertical"
                android:textSize="18dp"
                android:textColor="#AAAAAA"
                android:layout_marginTop="3dp"/>

        </LinearLayout>
    </LinearLayout>

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="80dp"
        android:layout_gravity="center"
        android:src="@drawable/house5r"
        android:layout_marginTop="15dp"/>

    <EditText
        android:layout_width="250dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:hint="Room Name"
        android:inputType="textPersonName"
```

```

        android:textAlignment="center"
        android:textSize="16dp"
        android:layout_marginTop="0dp"
        android:id="@+id/etroomname"/>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal">

        <TextView
            android:layout_width="225dp"
            android:layout_height="50dp"
            android:layout_gravity="center"
            android:text="Device in this room:"
            android:layout_marginTop="10dp"
            android:id="@+id/tvdeviceinthisroom"
            android:textSize="18dp"
            android:textColor="#000000"/>

        <TextView
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:textAlignment="center"
            android:text="Add Device"
            android:layout_marginTop="0dp"
            android:textSize="16dp"
            android:id="@+id/tvadddevice"
            android:background="@drawable/signin3"/>
    </LinearLayout>

    <ListView
        android:layout_width="310dp"
        android:layout_height="254dp"
        android:id="@+id/listView"
        android:layout_gravity="center">

    </ListView>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Add Room"
        android:layout_marginTop="20dp"
        android:textSize="30dp"
        android:id="@+id/baddrrom"
        android:background="@drawable/signin"/>

```

Kode java dari halaman ini adalah sebagai berikut.

```

1  package com.example.billyausten.newfinal;
2
3  import android.content.Intent;
4  import android.graphics.Typeface;

```

```

5   import android.os.Bundle;
6   import android.support.v7.app.AppCompatActivity;
7   import android.view.Menu;
8   import android.view.MenuItem;
9   import android.view.View;
10  import android.widget.Button;
11  import android.widget.EditText;
12  import android.widget.ListView;
13  import android.widget.TextView;
14  import android.os.Handler;
15  import android.widget.Toast;
16
17  import com.rabbitmq.client.AMQP;
18
19  import org.json.simple.JSONObject;
20
21  import java.io.StringWriter;
22  import java.util.ArrayList;
23
24  public class EditRoomActivity extends AppCompatActivity {
25
26      //DEKLARASI VARIABEL
27      private ListView listView;
28      String operation_type;
29      String roomname;
30
31      @Override
32      protected void onCreate(Bundle savedInstanceState) {
33          super.onCreate(savedInstanceState);
34          setContentView(R.layout.activity_edit_room);
35
36          //PENGATURAN FONT
37          Typeface font_signin =
38              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
39          Typeface font_title =
40              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
41          Typeface font_or =
42              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
43 BoldItalic.otf");
43          Typeface font_activity =
44              Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
45
46          //JUDUL ACTIVITY
47          TextView activitytitle = (TextView)
48              findViewById(R.id.activitytitle);
49          activitytitle.setTypeface(font_activity);
50
51          //EDITTEXT NAMA ROOM
52          final EditText etroomname = (EditText)
53              findViewById(R.id.etroomname);
54          etroomname.setTypeface(font_title);
55
56          //TEXTVIEW KETERANGAN
57          TextView tvdeviceinthis = (TextView)
58              findViewById(R.id.tvdeviceinthisroom);
59          tvdeviceinthis.setTypeface(font_title);

```

```
54 //TOMBOL UNTUK MENAMBAH ATAU MENGURANGI DEVICE DALAM
ROOM
55         TextView tvadddevice = (TextView)
findViewById(R.id.tvadddevice);
56             tvadddevice.setTypeface(font_or);
57             tvadddevice.setOnClickListener(new
View.OnClickListener() {
58
59                 @Override
60                     public void onClick(View v) {
61
62                         Intent afterloginIntent = new
Intent(EditRoomActivity.this,AddDeviceRoom.class);
63
64                         afterloginIntent.putExtra("type","room");
65
66                         EditRoomActivity.this.startActivity(afterloginIntent);
67                     }
68
69             });
69 //BUTTON UNTUK MENAMBAH ROOM
70         Button baddrrom = (Button)
findViewById(R.id.baddrrom);
71             baddrrom.setTypeface(font_signin);
72
73 //LISTVIEW UNTUK MENDAFTAR SELURUH DEVICE DALAM ROOM
74         listView = (ListView) findViewById(R.id.listView);
75
76 //MEMBACA TIPE OPERASI (EDIT ATAU ADD)
77         operation_type = getIntent().getStringExtra("type");
78         try {
79             if (operation_type.equals("edit")) {
80                 roomname =
getIntent().getStringExtra("name");
81                 etroomname.setText(roomname);
82                 for (ArrayList<String> element :
MainActivity.mainList) {
83                     if
(element.get(4).equals(roomname)) {
84
ListAddDeviceRoom.checked.add(element);
85                     }
86                 }
87                 baddrrom.setText("Edit Room");
88                 activitytitle.setText("Edit Room");
89             }
90         } catch (Exception e) {}
91
92
93 //PERLAKUAN ADDROOM KETIKA DIKLIK
94         baddrrom.setOnClickListener(new
View.OnClickListener() {
95
96                 @Override
97                     public void onClick(View v) {
98                         if (operation_type.equals("add")) {
99                             try {
```

```

99                                         //membentuk pesan JSON untuk
menambah room
100
JSONObject obj = new
101
obj.put("head", "MD");
102
obj.put("homeid",
103
etroomname.getText().toString());
104
obj.put("type", "addrroom");
105
obj.put("name",
106
ListAddDeviceRoom.checked);
107
obj.put("devices",
108
MainActivity.usname);
109
obj.put("notifsender",
110
just added room '" + etroomname.getText().toString() + "'");
111
StringWriter toSent = new
112
obj.writeJSONString(toSent);
113
String message =
114
toSent.toString();
115
//properti
116
AMQP.BasicProperties props =
117
new AMQP.BasicProperties
118
.Builder()
119
.correlationId(MainActivity.phoneid)
120
.build();
121
//mengirim pesan
122
MainActivity.channel.basicPublish("", "AESqueue", props,
123
MainActivity.AESencrypt(message).getBytes("UTF-8"));
124
125
MainActivity.channel.basicPublish("", MainActivity.homeid +
126
"AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
127
ListAddDeviceRoom.checked.clear();
128
} catch (Exception e) {}
129
new Handler().postDelayed(new
Runnable() {
130
@Override
131
public void run() {
132
finish();
133
}
134
}, 200);
} else if
(operation_type.equals("edit")) {
135
try {
136
//membentuk pesan JSON untuk
edit room
137
JSONObject obj = new
138
obj.put("head", "MD");
139
obj.put("homeid",
140
MainActivity.homeid);
141
obj.put("type", "editroom");
142
obj.put("name",
143
etroomname.getText().toString());

```

```

135                     obj.put("oldname", roomname);
136                     obj.put("devices",
137                         MainActivity.usname);
138                     obj.put("notifsender",
139                         just edited room '" + roomname + "' to '" +
140                         etroomname.getText().toString() + "'");
140                     StringWriter toSent = new
141                     obj.writeJSONObject(toSent);
142                     String message =
143                     toSent.toString();
144                     //properti
145                     AMQP.BasicProperties props =
146                         .Builder()
147                         .correlationId(MainActivity.phoneid)
148                         .build();
149                     //mengirim pesan
150                     MainActivity.channel.basicPublish("", "AESqueue", props,
151                         MainActivity.AESencrypt(message).getBytes("UTF-8"));
152                     MainActivity.channel.basicPublish("", MainActivity.homeid +
153                         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
154                     ListAddDeviceRoom.checked.clear();
155                     } catch (Exception e) {
156                     }
157                     new Handler().postDelayed(new
158                         Runnable() {
159                             @Override
160                             public void run() {
161                                 finish();
162                             }
163                         }, 200);
164                     }
165                 }
166             }
167             @Override
168             public boolean onCreateOptionsMenu(Menu menu) {
169                 // Inflate the menu; this adds items to the action
170                 bar if it is present.
171                 getMenuInflater().inflate(R.menu.main, menu);
172                 return true;
173             }
174             @Override
175             public boolean onOptionsItemSelected(MenuItem item) {
176                 // Handle action bar item clicks here. The action
177                 bar will

```

```

177         // automatically handle clicks on the Home/Up
button, so long
178         // as you specify a parent activity in
AndroidManifest.xml.
179         int id = item.getItemId();
180
181         //noinspection SimplifiableIfStatement
182         if (id == R.id.action_settings) {
183             return true;
184         }
185
186         return super.onOptionsItemSelected(item);
187     }
188
189     @Override
190     public void onBackPressed() {
191         ListAddDeviceRoom.checked.clear();
192         finish();
193     }
194
195     @Override
196     public void onStart() {
197         super.onStart();
198         final ListDeviceRoom customList = new
ListDeviceRoom(this, ListAddDeviceRoom.checked);
199         listView.setAdapter(customList);
200     }
201 }

```

Bagian penting dari kode java di atas adalah sebagai berikut.

- Baris 57 sampai 67, yaitu bagian yang mengatur algoritma yang akan dijalankan ketika tombol *add device* diklik. Ketika tombol *add device* diklik, maka aplikasi akan berpindah ke halaman AddDeviceRoom. Halaman ini berguna untuk mengedit barang apa saja yang ada di dalam ruangan tersebut.
- Sama seperti halaman *Add Device*, halaman ini juga tidak hanya berguna untuk *add room*, namun juga *edit room*. Dengan demikian, perlu ada langkah pembacaan tipe operasi, apakah *add* atau *edit*. Langkah ini dilakukan pada baris 77.
- Baris 94 sampai 164, yaitu bagian yang berisi algoritma ketika tombol *add room* atau *edit room* diklik. Ketika tombol ini diklik, maka aplikasi akan mengirimkan pesan JSON *add* atau *edit room* sebagai berikut.

Tabel 29 Data JSON Add atau Edit Room

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	addrroom	<ul style="list-style-type: none"> • Home ID • Nama room 	Host dan Server	AES	Tidak ada jawaban

		<ul style="list-style-type: none"> • Daftar device yang ada dalam room 			
	editroom	<ul style="list-style-type: none"> • Home ID • Nama room yang hendak diedit • Nama Baru Daftar device baru di dalam room 	Host dan Server	AES	Tidak ada jawaban

Selain hal yang telah dijelaskan di atas, pada halaman ini terdapat *listview* yang berisi daftar *device* yang berada di dalam *room*. Tampilan *listview* ini diatur melalui kode XML *list_room.xml*.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="70dp">

    <ImageView
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:src="@drawable/fanr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <ImageView
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_gravity="center"
        android:background="@drawable/ic_clear_black_24dp"
        android:backgroundTint="#FF0000"/>

</LinearLayout>

```

ListView tersebut juga diatur dengan kode java *ListDeviceRoom.java*.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;

```

```

4   import android.graphics.Typeface;
5   import android.view.LayoutInflater;
6   import android.view.View;
7   import android.view.ViewGroup;
8   import android.widget.ArrayAdapter;
9   import android.widget.ImageView;
10  import android.widget.TextView;
11
12  import java.util.ArrayList;
13
14  /**
15   * Created by Billy Austen on 2/22/2017.
16   */
17
18  public class ListDeviceRoom extends
ArrayAdapter<ArrayList<String>> {
19
20      //DEKLARASI VARIABEL
21      private Activity context;
22      private ArrayList<ArrayList<String>> devices = new
ArrayList<>();
23
24      //konstruktor
25      public ListDeviceRoom(Activity context,
ArrayList<ArrayList<String>> devices) {
26          super(context, R.layout.list_room, devices);
27          this.context = context;
28          this.devices = devices;
29      }
30
31      @Override
32      public View getView(int position, View convertView,
ViewGroup parent) {
33          //PENGATURAN TAMPILAN, BERGANTUNG PADA JENIS DEVICE
34          LayoutInflater inflater =
context.getLayoutInflater();
35          View listViewItem =
inflater.inflate(R.layout.list_room, null, true);
36          TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
37          ImageView imageView = (ImageView)
listViewItem.findViewById(R.id.ivdeviceroom);
38          Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
39          textViewName.setTypeface(font_title);
40          textViewName.setText(devices.get(position).get(0));
41          if (devices.get(position).get(2).equals("switch"))
42
imageView.setImageResource(R.drawable.switchr);
43          else if
(devices.get(position).get(2).equals("curtain"))
44
imageView.setImageResource(R.drawable.curtainr);
45          else if (devices.get(position).get(2).equals("fan"))
46
imageView.setImageResource(R.drawable.fanr);
47          else if
(devices.get(position).get(2).equals("temperature"))

```

```

48         imageView.setImageResource(R.drawable.chronor);
49             else if
50             (devices.get(position).get(2).equals("lamp"))
51                 imageView.setImageResource(R.drawable.lampr);
52             else if
53             (devices.get(position).get(2).equals("door"))
54                 imageView.setImageResource(R.drawable.doorr);
55         }
}

```

Hal penting dari kode di atas ada pada baris 41 sampai akhir. Pada bagian ini, dilakukan pengaturan logo *device*. Logo ini bergantung pada tipe *device*.

Ketika tombol *add device* ditekan, maka aplikasi akan memasuki halaman khusus untuk memilih *device* mana saja yang terdaftar di ruangan tersebut. Halaman tersebut adalah AddDeviceRoom. Berikut adalah kode XML halaman tersebut.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_add_device_room"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:orientation="vertical"
    tools:context="com.example.billyausten.newfinal.AddDeviceRoom">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:layout_marginLeft="16dp"
        android:textSize="20dp"
        android:textColor="#000000"
        android:text="Available Devices:"
        android:id="@+id/tvavailabledevice"/>

    <ListView
        android:layout_width="311dp"
        android:layout_height="382dp"
        android:layout_marginTop="20dp"
        android:id="@+id/listView"
        android:layout_gravity="center">

    </ListView>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Add"/>

```

```

        android:layout_marginTop="30dp"
        android:textSize="30dp"
        android:id="@+id/badddeviceroom"
        android:background="@drawable/signin"/>
    </LinearLayout>

```

Kode java AddDeviceRoom.java adalah sebagai berikut.

```

1  package com.example.billyausten.newfinal;
2
3  import android.content.Intent;
4  import android.graphics.Typeface;
5  import android.os.Bundle;
6  import android.support.v7.app.AppCompatActivity;
7  import android.view.View;
8  import android.widget.BaseAdapter;
9  import android.widget.Button;
10 import android.widget.ListView;
11 import android.widget.TextView;
12 import android.widget.Toast;
13
14 import java.util.ArrayList;
15
16 public class AddDeviceRoom extends AppCompatActivity {
17
18     //DEKLARASI VARIABEL
19     private ListView listView;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_add_device_room);
25
26         //PENGATURAN FONT
27         Typeface font_signin =
28             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
29         Typeface font_title =
30             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
31         Typeface font_or =
32             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
33 BoldItalic.otf");
34
35         //TEXT VIEW KETERANGAN
36         TextView tvAvailable = (TextView)
37             findViewById(R.id.tvavailabledevice);
38         tvAvailable.setTypeface(font_title);
39
40         //BUTTON ADD DEVICE TO ROOM
41         Button badddeviceroom = (Button)
42             findViewById(R.id.badddeviceroom);
43         badddeviceroom.setTypeface(font_signin);
44         ListAddDeviceRoom customList;
45
46         //MEMERIKSA TIPE OPERASI
47         //Jika ruangan
48         if
49             (getIntent().getStringExtra("type").equals("room"))
50                 customList = new ListAddDeviceRoom(this,
51             MainActivity.mainList);

```

```

44          //Jika skenario
45      else {
46          ArrayList<ArrayList<String>> temp = new
ArrayList<>();
47          //skenario tidak mengikutkan device
termperature karena temperature tidak bisa diberikan aksi tertentu,
hanya bisa dibaca
48          for (ArrayList<String> element :
MainActivity.mainList) {
49              if
(!element.get(2).equals("temperature"))
50                  temp.add(element);
51          }
52          customList = new ListAddDeviceRoom(this,
temp);
53      }
54
55      //set adapter dari listview
56      listView = (ListView) findViewById(R.id.listView);
57      listView.setAdapter(customList);
58
59      //jika button ditekan, kembali ke halaman sebelumnya
60      badddeviceroom.setOnClickListener(new
View.OnClickListener() {
61
62          @Override
63          public void onClick(View v) {
64              finish();
65          }
66
67      });
68  }
69 }
```

Hal penting dalam kode di atas adalah sebagai berikut.

- Baris 38, yaitu deklarasi list *device*. Karena bentuk *widget* yang digunakan adalah *listview*, maka diperlukan deklarasi adapter. Adapter ini bernama ListAddDeviceRoom.java. Isi dari adapter ini adalah seluruh *device* yang terdaftar dalam rumah.
- Baris 40 sampai 52, yaitu bagian untuk memperoleh jenis operasi, apakah operasi *room* atau *scenario*. Perlu diketahui bahwa halaman ini terbuka baik untuk *add/edit room* maupun *add/edit scenario* (akan dijelaskan kemudian). Dengan demikian, bagian ini menjadi sangat penting. Untuk *add* atau *edit scenario*, perangkat berjenis temperature tidak perlu ditampilkan karena perangkat temperatur tidak mungkin dimasukan ke dalam skenario (tidak bisa diberi perintah kendali apapun).
- Baris 63 sampai akhir, yaitu pengaturan algoritma ketika *button* selesai ditekan. Ketika *button* ini ditekan, maka aplikasi akan kembali ke halaman sebelumnya.

Untuk mengisi *ListView* pada halaman ini, diperlukan kode XML dan kode java. Kode XML untuk mengatur isi *listview* ini adalah *list_add_device.xml*.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:weightSum="1"
        android:paddingTop="8dp"
        android:paddingBottom="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_gravity="center"
        android:src="@drawable/fanr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <CheckBox
        android:layout_width="26dp"
        android:layout_height="wrap_content"
        android:id="@+id/checkBox"
        android:layout_gravity="center"
        android:layout_weight="0.04" />

</LinearLayout>
```

Kode java untuk mengisi *listview* pada halaman ini adalah *ListAddDeviceRoom.java*.

```
1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.graphics.Typeface;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ArrayAdapter;
9 import android.widget.ImageView;
10 import android.widget.TextView;
11 import android.widget.CheckBox;
12 import android.widget.CompoundButton;
13 import android.widget.Toast;
14
15 import java.util.ArrayList;
16
17 /**
18 * Created by Billy Austen on 2/22/2017.
```

```

19     */
20
21     public class ListAddDeviceRoom extends
ArrayAdapter<ArrayList<String>> {
22
23         //DEKLARASI VARIABEL
24         private Activity context;
25         private ArrayList<ArrayList<String>> devices = new
ArrayList<>();
26         public static ArrayList<ArrayList<String>> checked = new
ArrayList<>();
27         boolean found;
28         int i;
29
30         //KONSTRUKTOR
31         public ListAddDeviceRoom(Activity context,
ArrayList<ArrayList<String>> devices) {
32             super(context, R.layout.list_add_device, devices);
33             this.context = context;
34             this.devices = devices;
35         }
36
37         @Override
38         public View getView(final int position, View convertView,
ViewGroup parent) {
39             //FUNGSI UNTUK MENGATUR TAMPILAN
40
41             LayoutInflator inflater =
context.getLayoutInflator();
42             View listViewItem =
inflater.inflate(R.layout.list_add_device, null, true);
43
44             //PENGATURAN FONT
45             Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
46             //TEXTVIEW NAMA DEVICE
47             TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
48             textViewName.setTypeface(font_title);
49             //IMAGEVIEW LOGO TIPE DEVICE
50             ImageView imageView = (ImageView)
listViewItem.findViewById(R.id.ivdeviceroom);
51
52             //PENGATURAN TEXT PADA TEXTVIEW NAMA DEVICE
53             textViewName.setText(devices.get(position).get(0));
54             if (devices.get(position).get(2).equals("switch"))
55
56                 imageView.setImageResource(R.drawable.switchr);
57             else if
(devices.get(position).get(2).equals("curtain"))
58
59                 imageView.setImageResource(R.drawable.curtainr);
60             else if
(devices.get(position).get(2).equals("fan"))
61                 imageView.setImageResource(R.drawable.fanr);
62             else if
(devices.get(position).get(2).equals("temperature"))

```

```

61         imageView.setImageResource(R.drawable.chronor);
62     else if
63     (devices.get(position).get(2).equals("lamp"))
64             imageView.setImageResource(R.drawable.lampr);
65     else if
66     (devices.get(position).get(2).equals("door"))
67             imageView.setImageResource(R.drawable.doorr);
68
69         //CHECKBOX DI SEBELAH NAMA, UNTUK MENGATUR APAKAH
70         DEVICE DIMASUKAN KE DALAM ROOM ATAU TIDAK
71         CheckBox checkBox = (CheckBox)
72         listViewItem.findViewById(R.id.checkBox);
73
74         //CHECK THE CHECKBOX OR NOT
75         found = false;
76         i=0;
77         while (!checked.isEmpty() && !found && i <
78         checked.size()) {
79             if
80             (devices.get(position).get(1).equals(checked.get(i).get(1))) {
81                 found = true;
82             } else
83                 i++;
84         }
85         if (found)
86             checkBox.setChecked(true);
87         else
88             checkBox.setChecked(false);
89
90         //PENGATURAN PERILAKU CHECKBOX KETIKA DIUBAH STATUS
91         //CHECK NYA
92         checkBox.setOnCheckedChangeListener(new
93         CompoundButton.OnCheckedChangeListener() {
94             @Override
95             public void onCheckedChanged(CompoundButton
96             buttonView, boolean isChecked) {
97                 if (isChecked) {
98                     checked.add(devices.get(position));
99                 } else {
100                     //hapus dari list
101                     found = false;
102                     i=0;
103                     while (!checked.isEmpty()
104                     && !found && i < checked.size()) {
105                         if
106                         (devices.get(position).get(1).equals(checked.get(i).get(1))) {
107                             found = true;
108                         } else
109                             i++;
110                     }
111                     if (found)
112                         checked.remove(i);
113                 }
114             }
115         );

```

```
106         return listViewItem;
107     }
108 }
```

Bagian penting dari kode di atas adalah sebagai berikut.

- Baris 52 sampai 66, yaitu bagian yang mengatur logo *device* pada tiap-tiap baris *listview* bergantung pada jenis *device*.
- Pada kode di atas, terdapat variabel ArrayList bernama checked. Checked merupakan variabel yang menampung daftar *device* yang dicentang oleh *user*. Dengan demikian, bagian penting berikutnya adalah baris 84 sampai akhir. Ketika *checkbox* mengalami perubahan status centang (misal di-*check* atau di-*uncheck*), maka hal ini akan mengubah isi ArrayList checked.
- Bagian penting lainnya adalah baris 70 sampai 82, yaitu baris untuk memberikan kondisi awal (inisialisasi) status *checkbox* masing-masing baris dalam *listview*. Inisialisasi ini dilakukan dengan cara mengecek dulu apakah *device* pada baris itu terdaftar atau tidak pada ArrayList checked.

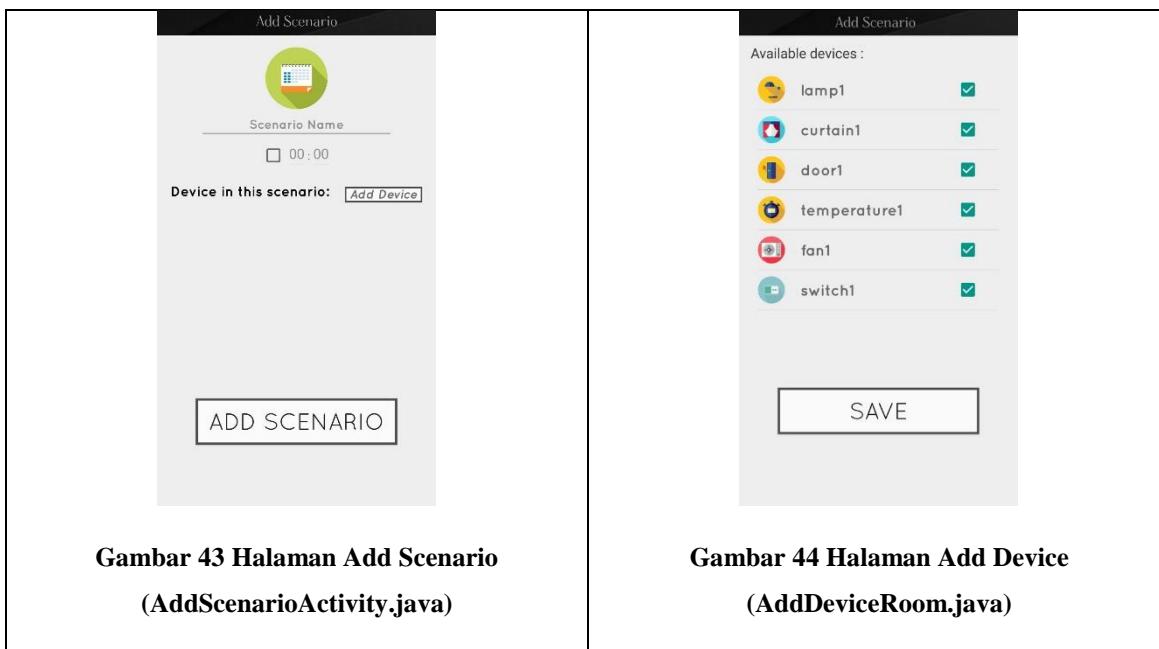
Perlu diketahui bahwa ArrayList checked juga menjadi patokan isi dari *listview* pada halaman EditRoom.java. Dengan demikian, isi *listview* pada halaman EditRoom.java selalu sinkron dengan *device* yang dicentang oleh *user* pada halaman ini.

6.10 Halaman Add Scenario (*AddScenarioActivity.java*)

Halaman ini dapat diakses dengan menekan *Button* “Add Scenario” pada *FloatingButton* dalam *MainActivity.java*. Pada halaman ini pengguna dapat menambahkan *scenario* penggunaan *device* pada sistem Rumah Cerdas. *Scenario* yang dimaksudkan adalah mode penggunaan *device*, contohnya adalah ketika pengguna memilih *scenario* “Sunday Morning” maka lampu taman, lampu balkon dan lampu teras dimatikan, seluruh tirai dibuka 100% dan pintu depan terbuka. Contoh lainnya adalah pada saat pengguna mengaktifkan *scenario* “Family Gathering”, seluruh lampu kamar dan kipas di kamar dimatikan dan kipas di ruang keluarga dan lampu di ruang keluarga dinyalakan. Dengan adanya fitur ini, pengguna tidak perlu menghabiskan banyak waktu untuk mengaktifkan *device* satu per satu, cukup dengan mengaktifkan *scenario* yang didaftarkan saja. Untuk mendaftarkan *scenario*, pengguna perlu memasukan nama *scenario* dan *device* apa saja yang perlu diaktifkan. Jika *device* berupa *fan*, *RGB lamp* ataupun *curtain*, pengguna dapat melakukan konfigurasi dengan mengatur *SeekBar* pada *device* tersebut. Tampilan pada

Activity ini serupa dengan tampilan pada *EditRoomActivity.java*, hanya saja pada halaman ini terdapat kolom *input* berupa waktu yang dapat digunakan pengguna untuk otomatisasi pengubahan status *device*. *ListView* pada halaman ini juga berbeda dengan halaman *EditRoomActivity.java* karena pada halaman ini, *list* menampilkan status *device* yang sudah diatur sebelumnya. Dengan melakukan pengaturan *scenario* berdasarkan waktu, pengguna dapat mengganti status sebuah maupun beberapa *device* secara bersamaan dengan otomatis.

Pada *AddScenarioActivity.java* terdapat *EditText* untuk menerima *input* berupa nama *scenario*. Seperti nama *room*, nama *scenario* juga memiliki *InputType* “textPersonName” agar pengguna tidak memberikan nama *scenario* sebanyak dua baris atau lebih. Setelah *EditText*, terdapat *ListView* yang berisi daftar *device* yang dapat diatur oleh *AddScenarioActivity.java*. Pada *list* ini, ditampilkan logo, nama dan status dari setiap *device* yang terdaftar dalam *scenario* ini. Pengguna dapat menambahkan *device* seperti pada *EditRoomActivity.java*. Seperti pada *EditRoomActivity.java*, *AddScenarioActivity.java* juga diberikan kode “OnBackPressed” agar ketika tombol *back* ditekan pengguna dapat menuju ke *MainActivity.java* dengan bantuan *Intent*.



Kode XML dari halaman ini adalah sebagai berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_add_scenario"
    android:layout_width="match_parent"
```

```

        android:layout_height="match_parent"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:paddingLeft="0dp"
        android:paddingRight="0dp"
        android:paddingTop="0dp"
        android:orientation="vertical"

    tools:context="com.example.billyausten.newfinal.AddScenarioActivity"
    android:weightSum="1"
    android:descendantFocusability="beforeDescendants"
    android:focusableInTouchMode="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="33dp"
            android:layout_gravity="center"
            android:orientation="vertical"
            android:background="@drawable/backblack2"
            android:id="@+id/backtitle">
            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_gravity="center">

                    <TextView
                        android:layout_width="290dp"
                        android:layout_height="wrap_content"
                        android:text="Add Scenario"
                        android:id="@+id/activitytitle"
                        android:textAlignment="center"
                        android:layout_gravity="center_vertical"
                        android:textSize="18dp"
                        android:textColor="#AAAAAA"
                        android:layout_marginTop="3dp"/>

                </LinearLayout>
            </LinearLayout>
        </LinearLayout>

        <ImageView
            android:layout_width="match_parent"
            android:layout_height="80dp"
            android:layout_gravity="center"
            android:src="@drawable/scenario1"
            android:layout_marginTop="15dp"/>

        <EditText
            android:layout_width="250dp"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:hint="Scenario Name"
            android:inputType="textPersonName"
            android:textAlignment="center"
            android:textSize="16dp"
            android:layout_marginTop="0dp"
            android:id="@+id/etscenariename"/>

        <LinearLayout

```

```

        android:layout_width="wrap_content"
        android:layout_height="40dp"
        android:orientation="horizontal"
        android:layout_gravity="center"
        android:layout_weight="0.01">>

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_gravity="center"
        android:id="@+id/checkbox"/>

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:hint="00"
        android:maxLength="2"
        android:inputType="datetime"
        android:id="@+id/ettime1"
        android:enabled="false"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text":""
        android:layout_gravity="center"
        android:id="@+id/textView" />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:hint="00"
        android:maxLength="2"
        android:id="@+id/ettime2"
        android:inputType="datetime"
        android:enabled="false"/>

</LinearLayout>

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:orientation="horizontal">

    <TextView
        android:layout_width="225dp"
        android:layout_height="50dp"
        android:layout_gravity="center"
        android:text="Device in this scenario:"
        android:layout_marginTop="10dp"
        android:id="@+id/tvdeviceinthisscenario"
        android:textSize="18dp"
        android:textColor="#000000"/>

    <TextView
        android:layout_width="100dp"
        android:layout_height="wrap_content"

```

```

        android:layout_gravity="center"
        android:textAlignment="center"
        android:text="Add Device"
        android:layout_marginTop="0dp"
        android:textSize="16dp"
        android:id="@+id/tvadddevice"
        android:background="@drawable/signin3"/>

    </LinearLayout>

    <ListView
        android:layout_width="320dp"
        android:layout_height="213dp"
        android:id="@+id/listView"
        android:layout_gravity="center"
        android:layout_marginRight="15dp"
        android:layout_marginLeft="15dp">

    </ListView>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Add Scenario"
        android:layout_marginTop="20dp"
        android:textSize="30dp"
        android:id="@+id/baddscenario"
        android:background="@drawable/signin"/>

</LinearLayout>

```

Kode java halaman ini adalah sebagai berikut.

```

1  package com.example.billyausten.newfinal;
2
3  import android.content.Intent;
4  import android.graphics.Typeface;
5  import android.os.Bundle;
6  import android.os.Handler;
7  import android.support.v7.app.AppCompatActivity;
8  import android.view.Menu;
9  import android.view.MenuItem;
10 import android.view.View;
11 import android.widget.AdapterView;
12 import android.widget.Button;
13 import android.widget.CheckBox;
14 import android.widget.CompoundButton;
15 import android.widget.EditText;
16 import android.widget.ListView;
17 import android.widget.TextView;
18 import android.widget.Toast;
19
20 import com.rabbitmq.client.AMQP;
21
22 import org.json.simple.JSONObject;
23
24 import java.io.StringWriter;
25 import java.util.ArrayList;

```

```

26
27     public class AddScenarioActivity extends AppCompatActivity {
28
29         private ListView listView;
30         String operation_type;
31
32         @Override
33         protected void onCreate(Bundle savedInstanceState) {
34             super.onCreate(savedInstanceState);
35             setContentView(R.layout.activity_add_scenario);
36
37             //PENGATURAN FONT
38             Typeface font_signin =
39                 Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
40             Typeface font_title =
41                 Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
42             Typeface font_or =
43                 Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
44                 BoldItalic.otf");
45             Typeface font_activity =
46                 Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
47
48             //JUDUL ACTIVITY
49             TextView activitytitle = (TextView)
50             findViewById(R.id.activitytitle);
51             activitytitle.setTypeface(font_activity);
52
53             //EDITTEXT UTK NAMA SCENARIO
54             final EditText etscenarioname = (EditText)
55             findViewById(R.id.etscenarioname);
56             etscenarioname.setTypeface(font_title);
57
58             //TEXTVIEW KETERANGAN
59             TextView tvdeviceinthis = (TextView)
60             findViewById(R.id.tvdeviceinthisscenario);
61             tvdeviceinthis.setTypeface(font_title);
62
63             //BUTTON UNTUK MELAKUKAN ADD/EDIT SCENARIO
64             Button baddscenario = (Button)
65             findViewById(R.id.baddscenario);
66             baddscenario.setTypeface(font_signin);
67
68             //CHECKBOX DAN EDITTEXT UTK SCENARIO BERJADWAL
69             // (SCHEDULING)
70             final CheckBox Checkbox = (CheckBox)
71             findViewById(R.id.Checkbox);
72             final EditText ettime1 = (EditText)
73             findViewById(R.id.ettime1);
74             final EditText ettime2 = (EditText)
75             findViewById(R.id.ettime2);
76
77             //MENDAPAT INFORMASI JENIS OPERASI
78             operation_type = getIntent().getStringExtra("type");
79             //jika edit
80             if (operation_type.equals("edit")) {
81
82                 etscenarioname.setText(getIntent().getStringExtra("name"));
83                 //edittext nama diisi dengan nama skenario

```

```

69             String time =
getIntent().getStringExtra("time");           //waktu skenario
70             if (time.equals("none"))
71                 Checkbox.setChecked(false);
72             else {
73                 Checkbox.setChecked(true);
74                 //edittext diisi dengan waktu
75                 ettime1.setText(time.substring(0,2));
76                 ettime2.setText(time.substring(2,4));
77                 ettime1.setEnabled(true);
78                 ettime2.setEnabled(true);
79             }
80             //button dan judul diisi dengan edit scenario
81             baddscenario.setText("Edit Scenario");
82             activitytitle.setText("Edit Scenario");
83         }
84
85         //PENGATURAN CHECKBOX
86         //jika checkbox dicentang, maka edittext waktu
enabled
87         Checkbox.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
88             @Override
89             public void
onCheckedChanged(CompoundButton buttonView,boolean isChecked) {
90                 if (isChecked) {
91                     ettime1.setEnabled(true);
92                     ettime2.setEnabled(true);
93                 } else {
94                     ettime1.setEnabled(false);
95                     ettime2.setEnabled(false);
96                 }
97             }
98         });
99     }
100
101     //LISTVIEW DEVICE PADA SKENARIO
102     listView = (ListView) findViewById(R.id.listView);
103
104
105     //TOMBOL UNTUK MENAMBAH ATAU MENGURANGI BARANG DALAM
SKENARIO
106     TextView tvadddevice = (TextView)
findViewById(R.id.tvadddevice);
107     tvadddevice.setTypeface(font_or);
108     tvadddevice.setOnClickListener(new
View.OnClickListener() {
109
110             @Override
111             public void onClick(View v) {
112
113                 Intent afterloginIntent = new
Intent(AddScenarioActivity.this,AddDeviceRoom.class);
114
115                 afterloginIntent.putExtra("type","scenario");
116                 AddScenarioActivity.this.startActivity(afterloginIntent);
117             }
118         });

```

```

117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

        });

        //PENGATURAN BUTTON JIKA DICLICK
        baddscenario.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //list barang kosong, skenario tidak
        dapat dibuat
        if (ListAddDeviceRoom.checked.isEmpty())
            Toast.makeText(getApplicationContext(), "The device list is
empty",
            Toast.LENGTH_SHORT).show();
        //nama skenario kosong, skenario tidak
        dapat dibuat
        else if
(etscenarioname.getText().toString().equals(""))
            Toast.makeText(getApplicationContext(), "Scenario name is
empty",
            Toast.LENGTH_SHORT).show();
        //jika nama skenario dan list device
        diisi, skenario berhasil dibuat
        else {
            //jika skenario tidak dijadwal
            (checkbox tidak dicentang)
            if (!(Checkbox.isChecked())) {
                try {
                    //membentuk pesan JSON
                    JSONObject obj = new
                    obj.put("head", "MD");
                    obj.put("homeid",
MainActivity.homeid);
                    if
(operation_type.equals("edit")) {
                        obj.put("type",
"editscenario");
                        obj.put("oldname", getIntent().getStringExtra("name"));
                    } else {
                        obj.put("type",
"addscenario");
                    }
                    obj.put("name",
etscenarioname.getText().toString());
                    obj.put("devices",
ListAddDeviceRoom.checked);
                    obj.put("time",
"none");
                }
            }
        }
    }
}
);

```

```

152                                         obj.put("notifsender",
MainActivity.usname);
153                                         if
154 (operation_type.equals("edit")) {
155                                         obj.put("notifmessage", "I just edited scenario '" +
getIntent().getStringExtra("name") +"' to '" +
etscenarioname.getText().toString() +"'");
156                                         } else
157                                         obj.put("notifmessage", "I just added scenario '" +
etscenarioname.getText().toString() +"'");
158                                         StringWriter toSent =
159 new StringWriter();
160                                         obj.writeJSONString(toSent);
161                                         String message =
162                                         toSent.toString();
163                                         //membentuk properti
pesan
164                                         AMQP.BasicProperties
props = new AMQP.BasicProperties
165                                         .Builder()
166                                         .correlationId(MainActivity.phoneid)
167                                         .build();
168                                         //mengirim pesan ke
server dan host
169                                         MainActivity.channel.basicPublish("", "AESqueue", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
170                                         MainActivity.channel.basicPublish("", MainActivity.homeid +
"AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
171                                         //arraylist daftar
device dikosongkan
173                                         ListAddDeviceRoom.checked.clear();
174                                         } catch (Exception e) {}
175                                         new
Handler().postDelayed(new Runnable() {
177                                         @Override
178                                         public void run() {
179                                         finish();
180                                         }
181                                         }, 200);
182                                         }
183                                         //jika skenario dijadwal (checkbox
dicentang, edittext waktu diisi dengan waktu yang valid)
184                                         else if
(Integer.parseInt(ettimel.getText().toString()) >= 0 &&
Integer.parseInt(ettimel.getText().toString()) < 24 &&

```

```

Integer.parseInt(ettime2.getText().toString()) >= 0 &&
Integer.parseInt(ettime2.getText().toString()) < 60 {
186                                     try {           //membentuk pesan JSON
187                                         JSONObject obj = new
untuk dikirim
188                                         JSONObject () ;
189                                         obj.put("head", "MD");
190                                         obj.put("homeid",
MainActivity.homeid);
191                                         if
192                                         (operation_type.equals("edit")) {
193                                         obj.put("oldname", getIntent().getStringExtra("name"));
194                                         } else {
195                                         obj.put("type",
"addscenario");
196                                         }
197                                         obj.put("name",
etscenario.name.getText().toString());
198                                         obj.put("devices",
ListAddDeviceRoom.checked);
199                                         obj.put("notifsender",
MainActivity.email);
200                                         if
201                                         (operation_type.equals("edit")) {
201                                         obj.put("notifmessage", "I just editted scenario '" +
getIntent().getStringExtra("name") +"' to '" +
etscenario.name.getText().toString() +"'");
202                                         } else
203                                         obj.put("notifmessage", "I just added scenario '" +
etscenario.name.getText().toString() +"'");
204                                         String hr =
ettime1.getText().toString();
205                                         String mn =
ettime2.getText().toString();
206                                         if (hr.length() == 1)
//1 digit hour
207                                         hr = "0" + hr;
208                                         if (mn.length() == 1)
//1 digit minute
209                                         mn = "0" + mn;
210                                         obj.put("time",
hr+mn);
211                                         StringWriter toSent =
212                                         new StringWriter();
213                                         obj.writeJSONString(toSent);
214                                         String message =
toSent.toString();
215                                         //membentuk properti
216                                         pesan

```

```

217                                     AMQP.BasicProperties
218     props = new AMQP.BasicProperties
219                                         .Builder()
220                                         .correlationId(MainActivity.phoneid)
221                                         .build();
222                                         //mengirim pesan ke
223                                         server dan host
224
225                                         MainActivity.channel.basicPublish("", "AESqueue", props,
226                                         MainActivity.AESencrypt(message).getBytes("UTF-8"));
227                                         MainActivity.channel.basicPublish("", MainActivity.homeid +
228                                         "AES", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
229                                         //arraylist device
230                                         dikosongkan
231                                         ListAddDeviceRoom.checked.clear();
232                                         } catch (Exception e) {} new
233                                         Handler().postDelayed(new Runnable() {
234                                         @Override
235                                         public void run() {
236                                         finish();
237                                         }
238                                         Toast.makeText(getApplicationContext(), "The time is not valid",
239                                         Toast.LENGTH_SHORT).show();
240                                         }
241                                         }
242                                         });
243                                         }
244                                         @Override
245                                         public boolean onCreateOptionsMenu(Menu menu) {
246                                         // Inflate the menu; this adds items to the action
247                                         // bar if it is present.
248                                         getMenuInflater().inflate(R.menu.main, menu);
249                                         return true;
250                                         }
251                                         @Override
252                                         public boolean onOptionsItemSelected(MenuItem item) {
253                                         // Handle action bar item clicks here. The action
254                                         // bar will
255                                         // automatically handle clicks on the Home/Up
256                                         // button, so long
257                                         // as you specify a parent activity in
258                                         // AndroidManifest.xml.
259                                         int id = item.getItemId();
260

```

```

259         //noinspection SimplifiableIfStatement
260         if (id == R.id.action_settings) {
261             return true;
262         }
263
264         return super.onOptionsItemSelected(item);
265     }
266
267     @Override
268     public void onBackPressed() {
269         finish();
270         ListAddDeviceRoom.checked.clear();
271     }
272
273     @Override
274     public void onStart() {
275         super.onStart();
276         final ListDeviceCat customList = new
ListDeviceCat(this, ListAddDeviceRoom.checked);
277         listView.setAdapter(customList);
278     }
279 }

```

Halaman ini sangat mirip dengan halaman EditRoom.java. Perbedaannya ada pada bagian pengisian waktu. Hal-hal penting dalam halaman ini adalah sebagai berikut.

- Sama seperti halaman *Add Device* dan *Add Room*, halaman ini juga tidak hanya berguna untuk *add scenario*, namun juga *edit scenario*. Dengan demikian, perlu ada langkah pembacaan tipe operasi, apakah *add* atau *edit*. Langkah ini dilakukan pada baris 65.
- Baris 85 sampai 99, yaitu bagian untuk mengatur perilaku *checkbox*. Ketika status *checkbox* dalam keadaan dicentang, maka *EditText* waktu akan aktif (bisa diedit).
- Baris 108 sampai 118, yaitu bagian yang mengatur algoritma yang akan dijalankan ketika tombol *add device* diklik. Ketika tombol *add device* diklik, maka aplikasi akan berpindah ke halaman AddDeviceRoom (sama seperti halaman *add room*). Halaman ini berguna untuk mengedit barang apa saja yang ada di dalam skenario tersebut.
- Baris 120 sampai 243, yaitu bagian yang berisi algoritma ketika tombol *add room* atau *edit room* diklik. Ketika tombol ini diklik, maka aplikasi akan mengirimkan pesan JSON *add* atau *edit room* sebagai berikut.

Tabel 30 Data JSON untuk *Add* dan *Edit Scenario*

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
--------	------	----------------	--------	----------	---------------------

MD (Manajemen Device)	addscenario	<ul style="list-style-type: none"> • Home ID • Nama skenario • Daftar device yang ada dalam skenario dan aksinya 	Host dan Server	AES	Tidak ada jawaban
	editscenario	<ul style="list-style-type: none"> • Home ID • Nama skenario yang hendak diedit • Nama Baru • Daftar device baru di dalam skenario berikut statusnya 	Host dan Server	AES	Tidak ada jawaban

Sama seperti halaman *add room*, pada halaman ini juga terdapat *listview* perangkat apa saja yang dikendalikan dalam skenario ini. Isi *listview* itu tergantung pada jenis *device*. Dengan demikian, ada kemungkinan 5 jenis tampilan dalam *listview*.

Kode *list_scenario_switch.xml*:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:weightSum="1"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:paddingBottom="8dp"
    android:paddingTop="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_gravity="center"
        android:src="@drawable/switchr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
```

```

        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <Switch
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/switch1"
        android:layout_marginLeft="40dp"
        android:layout_gravity="center"
        android:textColorLink="?android:attr/colorActivatedHighlight"
    />

</GridLayout>

```

Kode list_scenario_curtain.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:weightSum="1"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:paddingBottom="8dp"
    android:paddingTop="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_gravity="center"
        android:src="@drawable/curtainr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <SeekBar
        android:text=""
        android:layout_height="wrap_content"
        android:id="@+id/seek1"
        android:layout_marginTop="4dp"
        android:layout_gravity="center"
        android:textColorLink="?android:attr/colorActivatedHighlight"
        android:layout_width="100dp" />

```

```
</GridLayout>
```

Kode list_scenario_fan:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:weightSum="1"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:paddingBottom="8dp"
    android:paddingTop="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_gravity="center"
        android:src="@drawable/fanr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <SeekBar
        android:text=""
        android:layout_height="wrap_content"
        android:id="@+id/seek1"
        android:layout_marginTop="4dp"
        android:layout_gravity="center"
        android:textColorLink="?android:attr/colorActivatedHighlight"
        android:layout_width="100dp" />

</GridLayout>
```

Kode list_scenario_door:

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:weightSum="1"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    android:paddingBottom="8dp"
    android:paddingTop="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:layout_gravity="center"
```

```

        android:src="@drawable/doorr"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <Switch
        android:text=""
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/switch1"
        android:layout_marginLeft="40dp"
        android:layout_gravity="center"
        android:textColorLink="?android:attr/colorActivatedHighlight"
    />

</GridLayout>

```

Kode list_scenario_lamp:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:layout_marginTop="8dp"
        android:layout_marginBottom="8dp"
        android:paddingBottom="8dp"
        android:paddingTop="8dp">

    <ImageView
        android:layout_width="35dp"
        android:layout_height="35dp"
        android:src="@drawable/lampr"
        android:layout_gravity="center"
        android:id="@+id/ivdeviceroom"/>

    <TextView
        android:layout_width="170dp"
        android:layout_height="wrap_content"
        android:textSize="16dp"
        android:text="Central Fan"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        android:id="@+id/tvdeviceroom"/>

    <Switch
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="0dp"
        android:layout_gravity="center"
        android:id="@+id/switch1" />

```

```

<ImageView
    android:layout_width="20dp"
    android:layout_height="20dp"
    android:layout_marginLeft="10dp"
    android:src="@drawable/rgb"
    android:layout_gravity="center"
    android:id="@+id/ivrgb"/>

</LinearLayout>

```

Kode java yang mengatur isi *listview* adalah ListDeviceCat.Java.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.graphics.Typeface;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ArrayAdapter;
9 import android.widget.CompoundButton;
10 import android.widget.ImageView;
11 import android.widget.SeekBar;
12 import android.widget.TextView;
13 import android.widget.Switch;
14
15 import org.json.simple.JSONObject;
16
17 import java.io.StringWriter;
18 import java.math.BigInteger;
19 import java.util.ArrayList;
20
21 import yuku.ambilwarna.AmbilWarnaDialog;
22
23 /**
24 * Created by Billy Austen on 2/24/2017.
25 */
26
27 public class ListDeviceCat extends
ArrayAdapter<ArrayList<String>> {
28     private ArrayList<ArrayList<String>> devices;
29     private Activity context;
30     int color;
31
32     public ListDeviceCat(Activity context,
ArrayList<ArrayList<String>> devices) {
33         super(context, R.layout.list_scenario_fan, devices);
34         this.context = context;
35         this.devices = devices;
36     }
37
38     @Override
39     public View getView(final int position, View convertView,
ViewGroup parent) {
40         LayoutInflater inflater =
context.getLayoutInflater();
41

```

```

42         //Isi list ini bergantung pada ArrayList Checked.
Bila suatu device terdaftar pada Checked, maka checkbox akan
tercentang
43
44         //PENGATURAN TAMPILAN UNTUK DEVICE BERJENIS SWITCH
45         if
(ListAddDeviceRoom.checked.get(position).get(2).equals("switch")) {
46             View listViewItem =
inflater.inflate(R.layout.list_scenario_switch, null, true);
47             TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
48             Switch switch1 = (Switch)
listViewItem.findViewById(R.id.switch1);
49             Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
50             textViewName.setTypeface(font_title);
51
textViewName.setText(devices.get(position).get(0));
52         if
(ListAddDeviceRoom.checked.get(position).get(3).equals("0"))
53             switch1.setChecked(false);
54         else
55             switch1.setChecked(true);
56             switch1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
57                 @Override
58                 public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
59                     if (isChecked)
60
ListAddDeviceRoom.checked.get(position).set(3,"100");
61                     else
62
ListAddDeviceRoom.checked.get(position).set(3,"0");
63                 }
64             });
65             return listViewItem;
66         }
67         //PENGATURAN TAMPILAN UNTUK DEVICE BERJENIS CURTAIN
68         else if
(ListAddDeviceRoom.checked.get(position).get(2).equals("curtain")) {
69             View listViewItem =
inflater.inflate(R.layout.list_scenario_curtain, null, true);
70             TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
71             SeekBar seek1 = (SeekBar)
listViewItem.findViewById(R.id.seek1);
72             Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
73             textViewName.setTypeface(font_title);
74
textViewName.setText(devices.get(position).get(0));
75             seek1.setMax(255);
76
seek1.setProgress(Integer.parseInt(devices.get(position).get(3)))
);

```

```

77             seek1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
78                 @Override
79                 public void onProgressChanged(SeekBar
seekBar, int progress, boolean fromUser) {
80
81                     }
82
83                     @Override
84                     public void onStartTrackingTouch(SeekBar
seekBar) {
85
86                     }
87
88                     @Override
89                     public void onStopTrackingTouch(SeekBar
seekBar) {
90
ListAddDeviceRoom.checked.get(position).set(3, Integer.toString(s
eekBar.getProgress())));
91
}
92
})
93
return listViewItem;
94
}
95
//PENGATURAN TAMPILAN UNTUK DEVICE BERJENIS FAN
96
else if
(ListAddDeviceRoom.checked.get(position).get(2).equals("fan")) {
98
View listViewItem =
inflater.inflate(R.layout.list_scenario_fan, null, true);
99
TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
100
SeekBar seek1 = (SeekBar)
listViewItem.findViewById(R.id.seek1);
101
Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
102
textViewName.setTypeface(font_title);
103
textViewName.setText(devices.get(position).get(0));
104
seek1.setMax(255);
105
seek1.setProgress(Integer.parseInt(devices.get(position).get(3)));
);
106
seek1.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
107
@Override
108
public void onProgressChanged(SeekBar
seekBar, int progress, boolean fromUser) {
109
}
110
}
111
@Override
112
public void onStartTrackingTouch(SeekBar
seekBar) {
113
}
114
}
115
}
116
}

```

```

117                     @Override
118                     public void onStopTrackingTouch(SeekBar
seekBar) {
119
120                         ListAddDeviceRoom.checked.get(position).set(3,
Integer.toString(seekBar.getProgress()));
121                         }
122
123                     return listViewItem;
124                 }
125             //PENGATURAN TAMPILAN UNTUK DEVICE BERJENIS DOOR
126             else if
(ListAddDeviceRoom.checked.get(position).get(2).equals("door")) {
127                 View listViewItem =
inflater.inflate(R.layout.list_scenario_door, null, true);
128                 TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
129                 Switch switch1 = (Switch)
listViewItem.findViewById(R.id.switch1);
130                 Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
131                 textViewName.setTypeface(font_title);
132
133                 textViewName.setText(devices.get(position).get(0));
134                 if
(ListAddDeviceRoom.checked.get(position).get(3).equals("0"))
135                     switch1.setChecked(false);
136                 else
switch1.setChecked(true);
137                 switch1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
138                     @Override
139                     public void
onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
140                         if (isChecked)
141
ListAddDeviceRoom.checked.get(position).set(3,"100");
142                         else
143
ListAddDeviceRoom.checked.get(position).set(3,"0");
144                         }
145                     });
146                     return listViewItem;
147                 }
148             //PENGATURAN TAMPILAN UNTUK DEVICE BERJENIS LAMP
149             else if
(ListAddDeviceRoom.checked.get(position).get(2).equals("lamp")) {
150                 View listViewItem =
inflater.inflate(R.layout.list_scenario_lamp, null, true);
151                 TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvdeviceroom);
152                 final Switch switch1 = (Switch)
listViewItem.findViewById(R.id.switch1);
153                 ImageView ivrgb = (ImageView)
listViewItem.findViewById(R.id.ivrgb);

```

```

154         Typeface font_title =
155         Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
156         Bold.otf");
157         textViewName.setTypeface(font_title);
158         textViewName.setText(devices.get(position).get(0));
159         color = (new
160         BigInteger(devices.get(position).get(3), 16)).intValue();
161         if
162             (devices.get(position).get(3).equals("ff000000")) {
163                 switch1.setChecked(false);
164             } else {
165                 switch1.setChecked(true);
166             }
167             switch1.setOnCheckedChangeListener(new
168             CompoundButton.OnCheckedChangeListener() {
169                 @Override
170                 public void
171                 onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
172                     if(isChecked) {
173                         devices.get(position).set(3, String.format("%08x", color));
174                     }
175                     else {
176                         devices.get(position).set(3, "ff000000");
177                     }
178                 });
179             ivrgb.setOnClickListener(new
180             View.OnClickListener() {
181                 @Override
182                 public void onClick(View view) {
183                     openDialog(false, position,
184                     switch1.isChecked());
185                 }
186             });
187             return listViewItem;
188         }
189     }
190
191     void openDialog(boolean supportsAlpha, final int position,
192     final boolean switchstat) {
193         AmbilWarnaDialog dialog = new
194         AmbilWarnaDialog(context, color, supportsAlpha, new
195         AmbilWarnaDialog.OnAmbilWarnaListener() {
196             @Override
197             public void onOk(AmbilWarnaDialog dialog, int
198             color) {
199                 ListDeviceCat.this.color = color;
200                 if (switchstat) {

```

```

197     devices.get(position).set(3, String.format("%08x", color));
198         }
199     }
200
201     @Override
202     public void onCancel(AambilWarnaDialog dialog)
203     {
204         //Toast.makeText(getApplicationContext(), "cancel",
205         //Toast.LENGTH_SHORT).show();
206         }
207     });
208 }

```

Hal penting dari kode di atas adalah sebagai berikut.

- Kode terbagi menjadi beberapa subbagian *if*. Pembagian dilakukan berdasarkan tipe *device*. Hal yang membedakan setiap *device* adalah *widget* pengendali yang ditampilkan. Widget ini sama dengan yang ada pada halaman *device*.
- *Device* yang ditampilkan pada *listview* ini adalah yang terdaftar dalam *ArrayList checked* pada *ListAddDeviceRoom.java*. Ini sama dengan yang terjadi pada halaman *add room*.

Sama seperti pada halaman *EditRoom.java*, ketika tombol *add device* ditekan, maka aplikasi akan memasuki halaman *AddDeviceRoom*, yaitu halaman untuk penyentangan *device* yang hendak dikendalikan melalui skenario yang bersangkutan. *Activity* tersebut merupakan *activity* yang sama persis dan tidak mengalami perubahan sama sekali.

6.11 Halaman Notifikasi dan Pesan (*NotificationActivity.java*, *MessageActivity.java*)

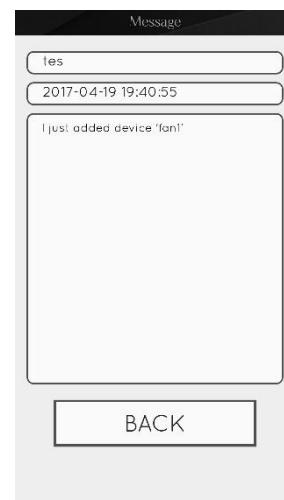
Pengguna dapat menerima berbagai informasi mengenai MINDS pada *Notification Activity*. Ketika ada orang baru yang mendaftarkan diri pada MINDS dalam suatu rumah, Admin rumah segera mendapatkan pesan permintaan penambahan member seperti yang sebelumnya sudah dibahas. Admin akan menerima 6 buah angka yang dapat dimasukan pengguna baru ke dalam *Invite1Activity.java* untuk menyelesaikan tahap pendaftaran. Selain urusan pendaftaran, *NotificationActivity.java* juga berguna untuk menampilkan informasi mengenai rumah tersebut, baik adanya *device* baru yang terdaftar, member baru yang masuk maupun ruangan atau *scenario* baru yang ditambahkan.

Notification Activity akan menampilkan sebuah *ListView*. *ListView* berisikan tanggal maupun jam pesan tersebut dikirimkan, pengirim pesan dan tampilan singkat pesan tersebut. Tampilan singkat dibuat dengan menambahkan kode “singleLine = true” pada XML program. Ketika suatu pesan ditekan, maka tampilan akan beralih ke tampilan pesan tersebut secara keseluruhan pada *MessageActivity.java*. Untuk melakukan *passing* data pengirim, isi dari notifikasi dan juga waktu pengiriman pada saat perpindahan dari *NotificationActivity.java* ke *MessageActivity.java*, dilakukan pengiriman *extra* dengan cara menambahkan kode “*putExtra*” pada *Intent* ketika suatu notifikasi ditekan dan akan berpindah *Activity*. Sehingga, untuk mendapatkan isi *extra* yang dikirim oleh *Activity* sebelumnya, perlu dilakukan “*getExtra*” pada *MessageActivity.java* sesuai dengan variabel mana yang dibutuhkan.

Pada tampilan notifikasi secara keseluruhan, digunakan tiga buah *TextView* yang memuat nama pengirim notifikasi, tanggal dan waktu notifikasi dikirim dan isi notifikasi tersebut. Pada bagian bawah dari *MessageActivity.java* terdapat *Button* “BACK” untuk kembali pada halaman *NotificationActivity.java*.



**Gambar 45 Halaman Notifikasi
(NotificationActivity.java)**



**Gambar 46 Halaman Pesan
(MessageActivity.java)**

Kode XML untuk *activity* pertama adalah sebagai berikut (*activity_notification.xml*).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_notification"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
```

```

        android:paddingLeft="0dp"
        android:paddingRight="0dp"
        android:paddingTop="0dp"
        android:orientation="vertical"

    tools:context="com.example.billyausten.newfinal.NotificationActivity">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="33dp"
            android:layout_gravity="center"
            android:orientation="vertical"
            android:background="@drawable/backblack2"
            android:id="@+id/backtitle">
            <LinearLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:layout_gravity="center">
                <TextView
                    android:layout_width="290dp"
                    android:layout_height="wrap_content"
                    android:text="Notifications"
                    android:id="@+id/activitytitle"
                    android:textAlignment="center"
                    android:layout_gravity="center_vertical"
                    android:textSize="18dp"
                    android:textColor="#AAAAAA"
                    android:layout_marginTop="3dp"/>

            </LinearLayout>
        </LinearLayout>

        <ListView
            android:layout_width="match_parent"
            android:layout_height="420dp"
            android:id="@+id/listView"
            android:layout_marginTop="10dp">

        </ListView>
        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Back"
            android:layout_marginTop="30dp"
            android:textSize="30dp"
            android:id="@+id/bback"
            android:background="@drawable/signin"/>
    </LinearLayout>

```

Kode java NotificationActivity.java adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.view.View;

```

```

8   import android.widget.AdapterView;
9   import android.widget.Button;
10  import android.widget.ListView;
11  import android.widget.TextView;
12
13  import java.util.ArrayList;
14
15  public class NotificationActivity extends AppCompatActivity {
16
17      //DEKLARASI VARIABEL
18      private ListView listView;
19      String SelectedSender;
20      String SelectedMessage;
21      String SelectedDate;
22      public static ArrayList<ArrayList<String>> notifs;
23
24      @Override
25      protected void onCreate(Bundle savedInstanceState) {
26          super.onCreate(savedInstanceState);
27          setContentView(R.layout.activity_notification);
28
29          //PENGATURAN FONT
30          Typeface font_signin =
31              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
32          Typeface font_title =
33              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Bold.otf");
34          Typeface font_or =
35              Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
36              BoldItalic.otf");
36          Typeface font_activity =
37              Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
38
39          //TITLE BAR
40          TextView activitytitle = (TextView)
41              findViewById(R.id.activitytitle);
42          activitytitle.setTypeface(font_activity);
43
44          //LIST UNTUK MENAMPUNG SEMUA NOTIFIKASI
45          final ListNotif customList = new ListNotif(this,
46              notifs);
47          listView = (ListView) findViewById(R.id.listView);
48          listView.setAdapter(customList);
49          //KETIKA ELEMEN LISTVIEW DIKLIK:
50          listView.setOnItemClickListener(new
51              AdapterView.OnItemClickListener() {
52                  @Override
53                  public void onItemClick(AdapterView<?>
54                      adapterView, View view, int i, long l) {
55
56                      SelectedSender = notifs.get(i).get(1);
57                      SelectedMessage = notifs.get(i).get(2);
58                      SelectedDate = notifs.get(i).get(3);
59
60                      //masuk ke MessageActivity
61                      Intent intent = new
62                          Intent(NotificationActivity.this, MessageActivity.class);
63                      intent.putExtra("SelectedSender",
64                          SelectedSender);

```

```

55                     intent.putExtra("SelectedMessage",
56 SelectedMessage);
57                     intent.putExtra("SelectedDate",
58 SelectedDate);
59                     startActivity(intent);
60     });
61 //BUTTON UNTUK BACK
62 Button bback = (Button) findViewById(R.id.bback);
63 bback.setTypeface(font_signin);
64 bback.setOnClickListener(new View.OnClickListener() {
65
66     @Override
67     public void onClick(View v) {
68
69         finish();
70     }
71
72 });
73
74 }

```

Hal penting yang ada pada kode di atas adalah sebagai berikut.

- Halaman ini berisi sebuah *ListView* yang isinya berdasarkan isi dari adapter bertipe *ListNotif* (sebuah kelas yang akan dijelaskan kemudian). Deklarasi adapter ini ada pada baris 39 sampai 42.
- Baris 43 sampai 59 berisi pengaturan perilaku ketika terjadi klik pada salah satu elemen *ListView*. Bila salah satu isi *ListView* notifikasi ditekan, maka aplikasi akan berpindah ke halaman *MessageActivity.java* (akan dijelaskan kemudian). Perpindahan halaman ini dilakukan dengan *Intent* yang diberi ekstra. Isi dari ekstra tersebut adalah identitas pengirim notifikasi, isi notifikasi, dan waktu notifikasi. Ketiga ekstra ini nantinya akan dibaca oleh *MessageActivity.java*.
- Baris 61 sampai akhir berisi pengaturan perilaku ketika tombol *back* diklik. Ketika tombol ini diklik, maka aplikasi akan kembali ke halaman sebelumnya (*MainActivity.java*).

Seperti yang telah dijelaskan di atas, pada halaman ini terdapat suatu kelas bernama *ListNotif.java* yang merupakan adapter dari *ListView* notifikasi. Kode XML isi *ListView* ini adalah sebagai berikut.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="horizontal"
        android:layout_width="match_parent"

```

```

    android:layout_height="match_parent">
<LinearLayout
    android:layout_width="270dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="5dp"
    android:layout_marginBottom="5dp">
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="MINDS"
    android:textColor="#000000"
    android:id="@+id/tvsender"/>
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:text="Invitation Code: Carrel,
caroluscarrel@gmail.com"
    android:singleLine="true"
    android:textColor="#606060"
    android:id="@+id/tvsubject"/>
</LinearLayout>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:layout_gravity="center"
    android:layout_marginRight="10dp">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:layout_marginLeft="8dp"
    android:text="01/03/17"
    android:textColor="#000000"
    android:id="@+id/tvdate"/>
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="right"
    android:layout_marginLeft="8dp"
    android:text="00:00"
    android:textColor="#000000"
    android:id="@+id/tvtime"/>
</LinearLayout>
</LinearLayout>

```

Kode java ListNotif.java adalah sebagai berikut.

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.graphics.Typeface;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ArrayAdapter;

```

```

9   import android.widget.TextView;
10
11  import java.util.ArrayList;
12
13  /**
14   * Created by Billy Austen on 3/1/2017.
15   */
16
17  public class ListNotif extends ArrayAdapter<ArrayList<String>> {
18
19      //DEKLARASI VARIABEL
20      private ArrayList<ArrayList<String>> notifs;
21      private Activity context;
22
23      //KONSTRUKTOR
24      public ListNotif(Activity context,
ArrayList<ArrayList<String>> notifs) {
25          super(context, R.layout.list_notif, notifs);
26          this.context = context;
27          this.notifs = notifs;
28      }
29
30      @Override
31      public View getView(int position, View convertView,
ViewGroup parent) {
32          LayoutInflator inflater =
context.getLayoutInflater();
33
34          //PENGATURAN FONT
35          Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
36          Typeface font_signin =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
37
38          View listViewItem =
inflater.inflate(R.layout.list_notif, null, true);
39
40          //TEXTVIEW NAMA PENGIRIM
41          TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvsender);
42          textViewName.setTypeface(font_title);
43          textViewName.setText(notifs.get(position).get(1));
44
45          //TEXTVIEW DESKRIPSI NOTIFIKASI
46          TextView textViewDesc = (TextView)
listViewItem.findViewById(R.id.tvssubject);
47          textViewDesc.setTypeface(font_signin);
48          textViewDesc.setText(notifs.get(position).get(2));
49
50          //TEXTVIEW TANGGAL DAN WAKTU
51          TextView textViewDate = (TextView)
listViewItem.findViewById(R.id.tvdate);
52          TextView textViewTime = (TextView)
listViewItem.findViewById(R.id.tvttime);
53          textViewDate.setTypeface(font_signin);

```

```

54         textViewDate.setText(notifs.get(position).get(3).substring(0,10))
55         textViewTime.setTypeface(font_signin);
56     textViewTime.setText(notifs.get(position).get(3).substring(11,19)
57 );
58     return listViewItem;
59 }
60 }
```

Bila dilihat dari kode di atas, isi dari kode tersebut hanyalah pengaturan tulisan dan tampilan. Tampilan isi ListView ini terdiri dari nama pengirim, isi notifikasi, dan waktu penerimaan notifikasi. Baris 40 sampai 56 adalah pengesetan isi TextView ketiga informasi di atas.

Berikutnya, ketika salah satu isi ListView ditekan, maka aplikasi akan berpindah ke MessageActivity.java. Berikut adalah kode xml activity_message.xml.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_message"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="0dp"
    android:paddingRight="0dp"
    android:paddingTop="0dp"
    tools:context="com.example.billyausten.newfinal.MessageActivity"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">
            <TextView
                android:layout_width="290dp"
                android:layout_height="wrap_content"
                android:text="Message"
                android:id="@+id/activitytitle"
                android:textAlignment="center"
                android:layout_gravity="center_vertical"
                android:textSize="18dp"
                android:textColor="#AAAAAA"
```

```

        android:layout_marginTop="3dp"/>

    </LinearLayout>
</LinearLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:background="@drawable/senderbox"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="20dp">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="26dp"
        android:textSize="18dp"
        android:textColor="#000000"
        android:id="@+id/tvsender"
        android:layout_marginLeft="10dp"
        android:layout_centerVertical="true"
        android:layout_marginRight="10dp"/>
</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="30dp"
    android:background="@drawable/senderbox"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="10dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="26dp"
        android:textSize="18dp"
        android:textColor="#000000"
        android:id="@+id/tvtime"
        android:layout_marginLeft="10dp"
        android:layout_centerVertical="true"
        android:layout_marginRight="10dp"/>
</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="350dp"
    android:background="@drawable/senderbox"
    android:layout_marginLeft="15dp"
    android:layout_marginRight="15dp"
    android:layout_marginTop="10dp">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="350dp"
        android:textSize="15dp"
        android:textColor="#000000"
        android:id="@+id/tvmessage"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="10dp"
        android:layout_marginRight="10dp"/>
</RelativeLayout>
<Button
    android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Back"
        android:layout_marginTop="20dp"
        android:textSize="30dp"
        android:id="@+id/bback"
        android:background="@drawable/signin"/>
    
```

Kode MessageActivity.java adalah sebagai berikut.

```

1  package com.example.billyausten.newfinal;
2
3  import android.content.Intent;
4  import android.graphics.Typeface;
5  import android.os.Bundle;
6  import android.support.v7.app.AppCompatActivity;
7  import android.view.View;
8  import android.widget.Button;
9  import android.widget.TextView;
10
11 public class MessageActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_message);
17
18         //PENGATURAN FONT
19         Typeface font_activity =
Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
20         Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
21
22         //JUDUL ACTIVITY (TITLE BAR)
23         TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
24         activitytitle.setTypeface(font_activity);
25
26         //TEXTVIEW PENGIRIM NOTIFIKASI
27         TextView tvsender = (TextView)
findViewById(R.id.tvsender);
28         tvsender.setTypeface(font_signin);
29
30         //TEXTVIEW WAKTU NOTIFIKASI
31         TextView tvttime = (TextView)
findViewById(R.id.tvttime);
32         tvttime.setTypeface(font_signin);
33
34         //TEXTVIEW ISI NOTIFIKASI
35         TextView tvmessage = (TextView)
findViewById(R.id.tvmassage);
36         tvmassage.setTypeface(font_signin);
37
38         //MEMPEROLEH PENGIRIM, ISI, DAN WAKTU DARI INTENT
EXTRA
39         String SelectedSender =
getIntent().getStringExtra("SelectedSender");

```

```

40         String SelectedMessage =
getIntent().getStringExtra("SelectedMessage");
41         String SelectedDate =
getIntent().getStringExtra("SelectedDate");
42         String sender = (" " + SelectedSender);
43         String datetime = (" " + SelectedDate);
44         String message = (" " + SelectedMessage);
45
46         //SET TULISAN PADA TEXTVIEW SESUAI DENGAN NILAI YANG
TELAH DIPEROLEH
47         tvsender.setText(sender);
48         tvtime.setText(datetime);
49         tvmessage.setText(message);
50
51         //TOMBOL BACK
52         Button bback = (Button) findViewById(R.id.bback);
53         bback.setTypeface(font_signin);
54         bback.setOnClickListener(new View.OnClickListener()
{
55
56             @Override
57             public void onClick(View v) {
58                 finish();
59             }
60
61         });
62     }
63 }
64

```

Hal penting dari kode di atas adalah sebagai berikut.

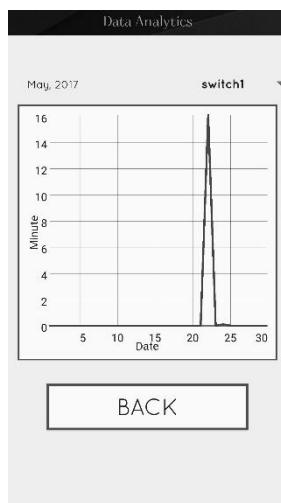
- Baris 38 sampai 44, yaitu bagian untuk mengambil nilai-nilai ekstra dari *intent* yang diberikan oleh halaman sebelumnya (*NotificationActivity.java*). Setelah nilai-nilai ekstra diperoleh, ketiga nilai tersebut dimasukan sebagai tulisan pada TextView *tvsender*, *tvtime*, dan *tvmessage*.
- Baris 52 sampai akhir, yaitu pengaturan klik tombol *back*. Ketika tombol *back* diklik, maka aplikasi akan berpindah ke halaman sebelumnya (dengan perintah *finish*).

6.12 Halaman Analisis Data (*DataActivity.java*)

Untuk memantau penggunaan keseluruhan *device* dalam rumah, pengguna dapat melihat grafik penggunaan *device* yang ditampilkan pada *DataActivity.java*. *Activity* ini tergolong penting untuk membantu pengguna melakukan efisiensi terhadap penggunaan energi di dalam rumah. Ketika terjadi peningkatan penggunaan energi, pengguna dapat segera mematikan *device-device* yang dirasa tidak perlu untuk digunakan. Grafik yang ditunjukkan merupakan grafik waktu penggunaan *device* dalam jam terhadap tanggal pada

bulan tersebut. Diatas grafik terdapat *TextView* yang menunjukan bulan dan tahun dari grafik yang sedang ditampilkan.

Digunakan *TextView* untuk menampilkan bulan dan tahun grafik yang sedang ditampilkan. Terdapat sebuah *Spinner* berisi *device-device* di dalam rumah tersebut, pengguna dapat melihat grafik penggunaan berdasarkan *device* maupun secara keseluruhan. Grafik dibuat dengan menggunakan *library GraphView*. Untuk menggunakan *GraphView*, perlu ditambahkan kode *compile 'com.jjoe64:graphview:4.2.1'* pada build.gradle (Module:app) sebagai pendaftaran *library GraphView*. Sebelum dapat melakukan berbagai *setting* terhadap grafik, perlu dilakukan inisiasi *GridLabelRender* pada *DataActivity.java*. Untuk memunculkan grafik yang ditampilkan oleh gambar, diperlukan penambahan judul pada sumbu X dan Y dengan menambahkan kode *setVerticalAxisTitle* dan *setHorizontalAxisTitle*. Sedangkan untuk mengatur warna dari grafik, digunakan kode *setColor* pada *series* dari grafik. Pada bagian luar dari grafik, ditambahkan sebuah *ImageView* yang tidak memiliki gambar tetapi memiliki *background berupa segi empat dengan tebal 3dp*. *ImageView* ini berfungsi sebagai kotak yang membatasi antara grafik dan objek lain pada *DataActivity.java*. Terdapat juga *Button* “Back” yang berfungsi untuk mengalihkan *Activity* menjadi *MainActivity.java*.



Gambar 47 Halaman Analisis Data (*DataActivity.java*)

Kode XML *activity_data.xml* adalah sebagai berikut.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
```

```

        android:layout_height="match_parent">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="33dp"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:background="@drawable/backblack2"
        android:id="@+id/backtitle">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:layout_gravity="center">

            <TextView
                android:layout_width="290dp"
                android:layout_height="wrap_content"
                android:text="Data Analytics"
                android:id="@+id/activitytitle"
                android:textAlignment="center"
                android:layout_gravity="center_vertical"
                android:textSize="18dp"
                android:textColor="#AAAAAA"
                android:layout_marginTop="3dp"/>
        </LinearLayout>
    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="50dp">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginBottom="5dp"
            android:layout_marginTop="5dp"
            android:id="@+id/tvmonth"
            android:text="March, 2017"
            android:layout_marginLeft="25dp"
            android:textColor="#000000"/>
        <Spinner
            android:layout_width="120dp"
            android:layout_height="wrap_content"
            android:layout_gravity="right"
            android:layout_marginTop="15dp"
            android:layout_marginLeft="150dp"
            android:id="@+id/spinner" />
    </LinearLayout>

    <RelativeLayout
        android:layout_width="335dp"
        android:layout_height="330dp"
        android:layout_gravity="center">
        <ImageView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@drawable/kotak"

```

```

        android:layout_alignParentBottom="true"
        android:layout_alignParentTop="true" />

    <com.jjoe64.graphview.GraphView
        android:layout_width="310dp"
        android:layout_height="310dp"
        android:id="@+id/graph"
        app:seriesType="line"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />
    </RelativeLayout>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="Back"
        android:layout_marginTop="30dp"
        android:textSize="30dp"
        android:id="@+id/bback"
        android:background="@drawable/signin"/>
</LinearLayout>

```

Kode java activity ini adalah sebagai berikut (DataActivity.java).

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.graphics.Color;
6 import android.graphics.Typeface;
7 import android.os.Bundle;
8 import android.view.View;
9 import android.view.ViewGroup;
10 import android.widget.AdapterView;
11 import android.widget.ArrayAdapter;
12 import android.widget.BaseAdapter;
13 import android.widget.Button;
14 import android.widget.Spinner;
15 import android.widget.TextView;
16 import android.widget.Toast;
17
18 import com.jjoe64.graphview.GraphView;
19 import com.jjoe64.graphview.GridLabelRenderer;
20 import com.jjoe64.graphview.series.DataPoint;
21 import com.jjoe64.graphview.series.LineGraphSeries;
22 import com.rabbitmq.client.AMQP;
23 import com.rabbitmq.client.Channel;
24 import com.rabbitmq.client.Connection;
25 import com.rabbitmq.client.ConnectionFactory;
26 import com.rabbitmq.client.DefaultConsumer;
27 import com.rabbitmq.client.Envelope;
28
29 import org.json.simple.JSONObject;
30 import org.json.simple.parser.JSONParser;
31
32 import java.io.IOException;
33 import java.io.StringWriter;
34 import java.util.ArrayList;

```

```

35 import java.util.UUID;
36 import java.util.concurrent.ArrayBlockingQueue;
37 import java.util.concurrent.BlockingQueue;
38
39 import java.util.Calendar;
40 import java.util.Date;
41 import java.text.DateFormat;
42 import java.text.SimpleDateFormat;
43
44 /**
45 * Created by Billy Austen on 1/22/2017.
46 */
47
48 public class DataActivity extends Activity {
49
50     ArrayList<String> data = new ArrayList<>();
51     Spinner spinner;
52     GraphView graph;
53     LineGraphSeries<DataPoint> series;
54     DataPoint[] listDataPoint = new DataPoint[31];
55     int i;
56
57     @Override
58     protected void onCreate(Bundle savedInstanceState) {
59         super.onCreate(savedInstanceState);
60         setContentView(R.layout.activity_data);
61
62         //PENGATURAN FONT
63         Typeface font_activity =
Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
64         Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
65
66         //JUDUL
67         TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
68         activitytitle.setTypeface(font_activity);
69
70         //REQUEST DATA ELEMEN PERTAMA PADA SPINNER
71         requestData(0);
72
73         //MEMBENTUK KURVA
74         for (i=0; i < 31; i++)
75             listDataPoint[i] = new DataPoint(i+1,
Float.parseFloat(data.get(i)));
76
77         graph = (GraphView) findViewById(R.id.graph);
78         series = new LineGraphSeries<>(listDataPoint);
79         graph.addSeries(series);
80         series.setColor(Color.DKGRAY);
81         series.setThickness(8);
82
83         graph.getViewport().setMinY(0);
84         graph.getViewport().setMinX(1);
85         graph.getViewport().setMaxX(31);
86         graph.getViewport().setXAxisBoundsManual(true);
87         graph.getViewport().setYAxisBoundsManual(true);
88

```

```

89         GridLabelRenderer gridLabel =
graph.getGridLabelRenderer();
90         gridLabel.setHorizontalAxisTitle("Date");
91         gridLabel.setVerticalAxisTitle("Minute");
92         gridLabel.setLabelsSpace(10);
93         gridLabel.setPadding(10);
94
95         graph.getViewport().setScrollable(true); // enables
horizontal scrolling
96         graph.getViewport().setScrollableY(true); // enables
vertical scrolling
97         graph.getViewport().setScalable(true); // enables
horizontal zooming and scrolling
98         graph.getViewport().setScalableY(true); // enables
vertical zooming and scrolling
99
100        //BUTTON UNTUK KEMBALI
101        Button bback = (Button) findViewById(R.id.bback);
102        bback.setTypeface(font_signin);
103        bback.setOnClickListener(new View.OnClickListener()
{
104
105            @Override
106            public void onClick(View v) {
107
108                finish();
109            }
110
111        });
112
113        //TEXTVIEW BULAN DAN TAHUN
114        TextView tvmonth = (TextView)
findViewById(R.id.tvmonth);
115        tvmonth.setTypeface(font_signin);
116
117        String month = "May";
118        String year = "2017";
119        String monthyear = String.format("%s, %s",
120                                         month,
121                                         year);
122        tvmonth.setText(monthyear);
123
124        //PENGATURAN SPINNER
125        String[] devices = new
String[MainActivity.mainList.size()];
126        for (i=0; i < MainActivity.mainList.size(); i++) {
127            devices[i] =
MainActivity.mainList.get(i).get(0);
128        }
129
130        ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
devices){
131            public View getView(int position, View
convertView, ViewGroup parent) {
132                View v = super.getView(position,
convertView, parent);
133

```

```

134                                     Typeface
externalFont=Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
Bold.otf");
135                                         ((TextView)
v).setTypeface(externalFont);
136
137                                         return v;
138 }
139
140
141         public View getDropDownView(int position,
View convertView, ViewGroup parent) {
142             View v =super.getDropDownView(position,
convertView, parent);
143
144             Typeface
externalFont=Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
Regular.otf");
145             ((TextView)
v).setTypeface(externalFont);
146
147             return v;
148 }
149 }
150
151         // Drop down layout style - list view with radio
button
152
dataAdapter.setDropDownViewResource(android.R.layout.simple_spin
ner_dropdown_item);
153
154     spinner = (Spinner) findViewById(R.id.spinner);
155     // attaching data adapter to spinner
156     spinner.setAdapter(dataAdapter);
157     // Spinner click listener
158     spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {
159         @Override
160         public void onItemSelected(AdapterView<?>
parent, View view, int position, long id) {
161             requestData(position);
162         }
163
164         @Override
165         public void onNothingSelected(AdapterView<?>
parent) {
166
167         }
168     });
169 }
170
171     void requestData (int position) {
172         //FUNGSI UNTUK MELAKUKAN REQUEST DATA PENGGUNAAN
DEVICE PADA SUATU INDEX TERTENTU (position) DALAM SPINNER
173         try {
174             DateFormat df = new
SimpleDateFormat("MMyyyy");
175             Date today = Calendar.getInstance().getTime();

```

```

176             String stringMonthYear = df.format(today);
177
178             String replyQueueName =
MainActivity.channel.queueDeclare().getQueue();
179
180             AMQP.BasicProperties props = new
AMQP.BasicProperties
181                     .Builder()
182
183                     .correlationId(MainActivity.phoneid)
184                     .replyTo(replyQueueName)
185                     .build();
186
187             JSONObject obj = new JSONObject();
188
189                     obj.put("head", "DA");
190                     obj.put("homeid", MainActivity.homeid);
191                     obj.put("monthyear", stringMonthYear);
192                     obj.put("address",
MainActivity.mainList.get(position).get(1));
193
194             StringWriter toSent = new StringWriter();
195             obj.writeJSONString(toSent);
196             String message = toSent.toString();
197             MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes());
198
199             final BlockingQueue<String> response = new
ArrayBlockingQueue<>(1);
200
201             MainActivity.channel.basicConsume(replyQueueName, true, new
DefaultConsumer(MainActivity.channel) {
202                     @Override
203                     public void handleDelivery(String consumerTag, Envelope envelope, AMQP.BasicProperties properties,
byte[] body) throws IOException {
204                     if
(properties.getCorrelationId().equals(MainActivity.phoneid)) {
205                     response.offer(new
String(body, "UTF-8"));
206
207                     }
208                 }
209             String res =
MainActivity.AESdecrypt(response.take());
210
211             JSONParser parser = new JSONParser();
212             data = (ArrayList<String>) parser.parse(res);
213             for (i=0; i < 31; i++)
214                     listDataPoint[i] = new DataPoint(i+1,
Float.parseFloat(data.get(i)));
215
216             graph.removeAllSeries();
217             series = new LineGraphSeries<>(listDataPoint);
218             graph.addSeries(series);

```

```

219         series.setColor(Color.DKGRAY);
220         series.setThickness(8);
221
222         graph.getViewport().setMinY(0);
223         graph.getViewport().setMinX(1);
224         graph.getViewport().setMaxX(31);
225
226         graph.getViewport().setXAxisBoundsManual(true);
227         graph.getViewport().setYAxisBoundsManual(true);
228     } catch (Exception e) {
229
230     }
231 }
232 }
```

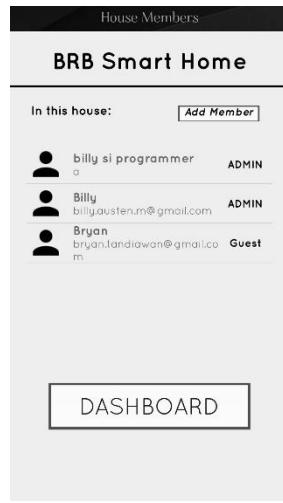
Bagian penting dari kode di atas adalah sebagai berikut.

- Baris 73 sampai 98, yaitu bagian pembentukan kurva. Bagian ini menetapkan label kurva, nilai maksimum sumbu, dan pengaturan *scrollable* kurva.
- Baris 124 sampai akhir, yaitu bagian untuk mengatur *spinner*. Ketika salah satu elemen *spinner* dipilih, maka bentuk kurva akan berubah menjadi data penggunaan *device* yang terpilih dari *spinner* tersebut. Dengan demikian, pada kode ini terdapat fungsi requestData. Fungsi ini bertujuan untuk melakukan permintaan data penggunaan *device* ke *host* menggunakan data JSON sebagai berikut.

Tabel 31 Data JSON Permintaan Data Penggunaan Device

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
DA (Data)	-	<ul style="list-style-type: none"> • Home ID • Address device yang ingin dilihat datanya 	<i>Host</i>	AES	Data penggunaan device selama 1 bulan

6.13 Halaman Member (*CreateNewActivity2.java*)



Gambar 48 Halaman Member (CreateNewActivity2.java)

Kode XML pada halaman ini sebagai berikut (activity_create_new2.xml):

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_create_new2"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="0dp"
10    android:paddingRight="0dp"
11    android:paddingTop="0dp"
12    android:orientation="vertical"
13    tools:context="com.example.billyausten.newfinal.CreateNewActivity2"
14    android:weightSum="1">
15
16    <LinearLayout
17        android:layout_width="match_parent"
18        android:layout_height="33dp"
19        android:layout_gravity="center"
20        android:orientation="vertical"
21        android:background="@drawable/backblack2"
22        android:id="@+id/backtitle">
23        <LinearLayout
24            android:layout_width="wrap_content"
25            android:layout_height="wrap_content"
26            android:orientation="horizontal"
27            android:layout_gravity="center">
28            <TextView
29                android:layout_width="290dp"
30                android:layout_height="wrap_content"
31                android:text="House Members"
32                android:id="@+id/activitytitle"
33                android:textAlignment="center"
34                android:layout_gravity="center_vertical"
35                android:textSize="18dp"
```

```

36             android:textColor="#AAAAAA"
37             android:layout_marginTop="3dp"/> 
38
39         
40     
41
42     <TextView
43         android:layout_width="match_parent"
44         android:layout_height="wrap_content"
45         android:text="MY HOME"
46         android:layout_gravity="center"
47         android:textAlignment="center"
48         android:layout_marginTop="20dp"
49         android:textSize="30dp"
50         android:id="@+id/tvHouseName"
51         android:textColor="#000000"/>
52
53     <ImageView
54         android:layout_width="match_parent"
55         android:layout_height="3dp"
56         android:layout_gravity="center"
57         android:layout_marginTop="15dp"
58         android:src="@drawable/pembatasdrawer2"/>
59
60     <LinearLayout
61         android:layout_width="match_parent"
62         android:layout_height="wrap_content"
63         android:orientation="horizontal"
64         android:layout_gravity="center_horizontal"
65         android:layout_marginLeft="18dp"
66         android:layout_marginTop="20dp">
67
68         <TextView
69             android:layout_width="120dp"
70             android:layout_height="match_parent"
71             android:layout_gravity="center"
72             android:text="In this house:"
73             android:textSize="16dp"
74             android:id="@+id/tvinthishouse"
75             android:textColor="#000000"/>
76
77         <TextView
78             android:layout_width="105dp"
79             android:textAlignment="center"
80             android:layout_height="match_parent"
81             android:text="Add Member"
82             android:layout_marginLeft="70dp"
83             android:layout_marginTop="2dp"
84             android:textSize="14dp"
85             android:id="@+id/tvaddmember"
86             android:layout_gravity="center"
87             android:textColor="#000000"
88             android:background="@drawable/signin3"/>
89     
90
91     <ListView
92         android:layout_width="317dp"
93         android:layout_height="250dp"

```

```

94        android:id="@+id/listView"
95        android:layout_gravity="center"
96        android:layout_marginTop="30dp"
97        android:layout_weight="0.27" />
98
99    <Button
100        android:layout_width="wrap_content"
101        android:layout_height="wrap_content"
102        android:layout_gravity="center"
103        android:text="Dashboard"
104        android:layout_marginTop="30dp"
105        android:textSize="30dp"
106        android:id="@+id/bback"
107        android:background="@drawable/signin"/>
108
109</LinearLayout>

```

Kode XML untuk *list* pada halaman ini sebagai berikut (list_member.xml):

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:orientation="horizontal"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:weightSum="1">
8
9     <ImageView
10        android:layout_width="50dp"
11        android:layout_height="50dp"
12        android:layout_gravity="center"
13        android:src="@drawable/ic_person_black_24dp"
14        android:layout_marginLeft="0dp"/>
15
16     <LinearLayout
17         android:layout_width="170dp"
18         android:layout_height="wrap_content"
19         android:layout_gravity="center"
20         android:orientation="vertical"
21         android:layout_weight="0.62"
22         android:layout_marginLeft="10dp"
23         android:layout_marginRight="10dp">
24
25         <TextView
26             android:layout_width="wrap_content"
27             android:layout_height="wrap_content"
28             android:id="@+id/tvmembername"
29             android:text="test"
30             android:textSize="16dp"/>
31         <TextView
32             android:layout_width="wrap_content"
33             android:layout_height="wrap_content"
34             android:id="@+id/tvmemberemail"
35             android:text="testadd"
36             android:textSize="14dp"/>
37
38     </LinearLayout>

```

```

39 <TextView
40     android:layout_width="wrap_content"
41     android:layout_height="wrap_content"
42     android:layout_gravity="center"
43     android:gravity="end"
44     android:text="ADMIN"
45     android:textColor="#000000"
46     android:id="@+id/tvprivilege"/>
47
48</LinearLayout>

```

Halaman ini dapat dibuka dengan menekan menu “House Members” pada *NavigationDrawer*. Pada halaman ini pengguna dapat melihat nama rumah yang telah didaftarkan sebelumnya dan ditampilkan dalam bentuk *TextView*. Setelah itu, pengguna dapat mendaftarkan member baru dengan cara menekan *TextView* “Add Member” yang diberikan *background* persegi panjang yang terletak di bawah pembatas. Pada halaman ini, pengguna dapat melihat seluruh pengguna lainnya yang terdaftar pada rumah yang sama pada MINDS. Dalam *list member* tersebut, pengguna dapat melihat nama, *email* dan *privilege* dari setiap *member*. *Privilege* bertujuan untuk mengatur batas wewenang setiap *member* dalam mengatur peralatan di dalam rumah tersebut. Setiap kategori *privilege* dapat di-setting oleh admin rumah sesuai dengan keinginan admin. Sebagai contoh, admin hanya memperbolehkan kategori *privilege* “Guest” untuk dapat mengakses pintu pagar dan pintu keluar rumah sehingga member dengan *privilege* “Guest” tidak dapat mengubah status *device* lainnya selain pintu pagar dan pintu keluar.

Kode java pada halaman ini sebagai berikut (CreateNewActivity2.java):

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.AlertDialog;
4 import android.content.DialogInterface;
5 import android.content.Intent;
6 import android.graphics.Typeface;
7 import android.os.StrictMode;
8 import android.support.v7.app.AppCompatActivity;
9 import android.os.Bundle;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.ListView;
13 import android.widget.TextView;
14 import android.widget.Toast;
15 import android.widget.AdapterView;
16 import android.support.v7.widget.PopupMenu;
17 import android.view.MenuItem;
18 import com.rabbitmq.client.AMQP;
19 import com.rabbitmq.client.Channel;

```

```

20 import com.rabbitmq.client.Connection;
21 import com.rabbitmq.client.ConnectionFactory;
22 import com.rabbitmq.client.DefaultConsumer;
23 import com.rabbitmq.client.Envelope;
24 import org.json.simple.JSONObject;
25 import org.json.simple.parser.JSONParser;
26 import java.io.IOException;
27 import java.io.StringWriter;
28 import java.util.ArrayList;
29 import java.util.Random;
30 import java.util.UUID;
31 import java.util.concurrent.ArrayBlockingQueue;
32 import java.util.concurrent.BlockingQueue;
33
34 public class CreateNewActivity2 extends AppCompatActivity {
35
36     //DEKLARASI VARIABEL
37     private ListView listView;
38     ListMember customList;
39     ArrayList<ArrayList<String>> users = new ArrayList<>();
40     JSONParser parser = new JSONParser();
41
42     @Override
43     protected void onCreate(Bundle savedInstanceState) {
44         super.onCreate(savedInstanceState);
45         setContentView(R.layout.activity_create_new2);
46
47         StrictMode.ThreadPolicy policy = new
48         StrictMode.ThreadPolicy.Builder().permitAll().build();
49         StrictMode.setThreadPolicy(policy);
50
51         //PENGATURAN FONT
52         Typeface font_signin =
53             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
54         Typeface font_title = Typeface.createFromAsset(getAssets(),
55             "fonts/Quicksand-Bold.otf");
56         Typeface font_or = Typeface.createFromAsset(getAssets(),
57             "fonts/Quicksand-BoldItalic.otf");
58         Typeface font_activity =
59             Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
60
61         //JUDUL ACTIVITY
62         TextView activitytitle = (TextView)
63         findViewById(R.id.activitytitle);
64         activitytitle.setTypeface(font_activity);
65
66         //NAMA RUMAH
67         TextView tvHouseName = (TextView)
68         findViewById(R.id.tvHouseName);
69         tvHouseName.setTypeface(font_title);
70         if (!MainActivity.homename.equals(""))
71             tvHouseName.setText(MainActivity.homename);
72
73         //TEXTVIEW KETERANGAN
74         TextView tvinthishouse = (TextView)
75         findViewById(R.id.tvinthishouse);
76         tvinthishouse.setTypeface(font_title);
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
279
280
281
282
283
284
285
286
287
288
289
289
290
291
292
293
294
295
296
297
298
299
299
300
301
302
303
304
305
306
307
308
309
309
310
311
312
313
314
315
316
317
318
319
319
320
321
322
323
324
325
326
327
328
329
329
330
331
332
333
334
335
336
337
338
339
339
340
341
342
343
344
345
346
347
348
349
349
350
351
352
353
354
355
356
357
358
359
359
360
361
362
363
364
365
366
367
368
369
369
370
371
372
373
374
375
376
377
378
379
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1088
1089
1090
1091
1092
1093
1094
1095
1095
1096
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1187
1188
1189
1190
1191
1192
1193
1194
1194
1195
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1287
1288
1289
1290
1291
1292
1293
1294
1295
1295
1296
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1387
1388
1389
1390
1391
1392
1393
1394
1394
1395
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1487
1488
1489
1490
1491
1492
1493
1494
1494
1495
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1587
1588
1589
1590
1591
1592
1593
1594
1594
1595
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1687
1688
1689
1690
1691
1692
1693
1694
1694
1695
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1787
1788
1789
1790
1791
1792
1793
1794
1794
1795
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1887
1888
1889
1890
1891
1892
1893
1894
1894
1895
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1988
1989
1990
1991
1992
1993
1994
1994
1995
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2088
2089
2090
2091
2092
2093
2094
2095
2095
2096
2097
2098
2099
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2188
2189
2190
2191
2192
2193
2194

```

```

70         //BUTTON UNTUK KEMBALI KE DASHBOARD
71         Button bback = (Button) findViewById(R.id.bback);
72         bback.setTypeface(font_signin);
73         bback.setOnClickListener(new View.OnClickListener() {
74
75             @Override
76             public void onClick(View v) {
77                 finish();
78             }
79         });
80     });
81
82     //TOMBOL UNTUK MASUK KE HALAMAN MEMBER INVITING
83     TextView tvaddmember = (TextView)
findViewById(R.id.tvaddmember);
84     tvaddmember.setTypeface(font_or);
85     tvaddmember.setOnClickListener(new View.OnClickListener() {
86
87         @Override
88         public void onClick(View v) {
89
90             Intent afterloginIntent = new
Intent(CreateNewActivity2.this,AddMemberActivity.class);
91             CreateNewActivity2.this.startActivity(afterloginIntent);
92         }
93     });
94 );
95
96     //LISTVIEW UNTUK MEMBER DALAM RUMAH
97     final ListMember customList = new ListMember(this, users);
98     listView = (ListView) findViewById(R.id.listView);
99     listView.setAdapter(customList);
100
101    listView.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
102        @Override
103        public void onItemClick(AdapterView<?> adapterView,
View view, final int i, long l) {
104            //Ketika diklik, akan muncul menu untuk delete
member
105            PopupMenu popup = new
PopupMenu(CreateNewActivity2.this, view);
106            popup.getMenuInflater().inflate(R.menu.menu_member,
popup.getMenu());
107            popup.show();
108            popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
109                public boolean onMenuItemClick(MenuItem item) {
110                    switch (item.getItemId()) {
111                        //jika tombol delete ditekan
112                        case R.id.action_delete_member:
113                            //user diperiksa dulu apakah admin,
jika user admin, maka user diizinkan menambah anggota
114                            if
(MainActivity.privilege.equals("ADMIN")) {
115                                try {
116                                    //membentuk data JSON

```

```

117         JSONObject obj = new
118         JSONObject();
119
120         obj.put("head", "MU");
121         obj.put("homeid",
122             obj.put("email",
123                 obj.put("type",
124                     obj.put("notifsender",
125                     obj.put("notifmessage", "I
126 just deleted member '" + users.get(i).get(1) + "'");
127
128                     StringWriter toSent = new
129                     StringWriter();
130
131                     obj.writeJSONString(toSent);
132
133                     toSent.toString();
134
135                     String message =
136                     //properti pesan
137                     AMQP.BasicProperties props
138                     = new AMQP.BasicProperties
139                         .Builder()
140                         .correlationId(MainActivity.phoneid)
141                         .build();
142
143                     //mengirim pesan
144
145                     MainActivity.channel.basicPublish("", "AESqueue", props,
146                     MainActivity.AESencrypt(message).getBytes("UTF-8"));
147
148                     MainActivity.channel.basicPublish("", MainActivity.homeid + "AES",
149                     props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
150
151                     Intent afterloginIntent =
152                     new Intent(CreateNewActivity2.this, MainActivity.class);
153
154                     CreateNewActivity2.this.startActivity(afterloginIntent);
155
156                     } catch (Exception e) {
157
158                     }
159
160                     Toast.makeText(CreateNewActivity2.this, "You Deleted " +
161                     users.get(i).get(1), Toast.LENGTH_SHORT).show();
162
163                     } else {
164
165                     Toast.makeText(CreateNewActivity2.this, "You are not ADMIN.\nOnly
166                     ADMIN can delete users", Toast.LENGTH_SHORT).show();
167
168                     }
169
170                     return true;
171
172                     }
173
174                     return true;
175
176                     }
177
178                     );
179
180                     }
181
182                     );
183
184                     }

```

```

154
155     @Override
156     protected void onResume() {
157         super.onResume();
158
159         String response = rpc();
160
161         try {
162             users = (ArrayList)parser.parse(response);
163             customList = new ListMember(this, users);
164             listView.setAdapter(customList);
165
166         } catch (Exception e) {
167
168         }
169     }
170
171     public String rpc () {
172         try {
173             String replyQueueName;
174
175             replyQueueName =
>MainActivity.channel.queueDeclare().getQueue();
176
177             AMQP.BasicProperties props = new AMQP.BasicProperties
178                 .Builder()
179                 .correlationId(MainActivity.phoneid)
180                 .replyTo(replyQueueName)
181                 .build();
182
183             JSONObject obj = new JSONObject();
184
185             obj.put("homeid", MainActivity.homeid);
186             obj.put("head", "MU");
187             obj.put("type", "list");
188             StringWriter toSent = new StringWriter();
189             obj.writeJSONString(toSent);
190             String message = toSent.toString();
191             MainActivity.channel.basicPublish("", "AESqueue",
props, MainActivity.AESencrypt(message).getBytes());
192             final BlockingQueue<String> response = new
ArrayBlockingQueue<>(2);
193
194             MainActivity.channel.basicConsume(replyQueueName, true,
new DefaultConsumer(MainActivity.channel) {
195                 @Override
196                 public void handleDelivery(String consumerTag,
Envelope envelope, AMQP.BasicProperties properties, byte[] body)
throws IOException {
197                     if
(properties.getCorrelationId().equals(MainActivity.phoneid)) {
198                         response.offer(new String(body, "UTF-8"));
199
200                     }
201                 }
202             });
203             String res = MainActivity.AESdecrypt(response.take());
204             return res;

```

```

205
206     } catch (Exception e) {
207         Toast.makeText(getApplicationContext(),"Can't connect
208             to the server\nCheck your connection",Toast.LENGTH_SHORT).show();
209     }
210 }
211}

```

Kode java untuk *list* pada halaman ini sebagai berikut (ListMember.java):

```

1 package com.example.billyausten.newfinal;
2
3 import android.app.Activity;
4 import android.graphics.Typeface;
5 import android.view.LayoutInflater;
6 import android.view.View;
7 import android.view.ViewGroup;
8 import android.widget.ArrayAdapter;
9 import android.widget.TextView;
10import java.util.ArrayList;
11
12public class ListMember extends ArrayAdapter<ArrayList<String>> {
13
14     //DEKLARASI VARIABEL
15     private ArrayList<ArrayList<String>> users;
16     private Activity context;
17
18     //KONSTRUKTOR
19     public ListMember(Activity context, ArrayList<ArrayList<String>>
users) {
20         super(context, R.layout.list_member, users);
21         this.context = context;
22         this.users = users;
23     }
24
25     @Override
26     public View getView(int position, View convertView, ViewGroup
parent) {
27         //PENGATURAN TAMPILAN
28         LayoutInflater inflater = context.getLayoutInflater();
29
30         //PENGATURAN FONT
31         Typeface font_title =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Bold.otf");
32         Typeface font_desc =
Typeface.createFromAsset(context.getAssets(), "fonts/Quicksand-
Regular.otf");
33
34         View listViewItem = inflater.inflate(R.layout.list_member,
null, true);
35
36         //TEXTVIEW NAMA USER
37         TextView textViewName = (TextView)
listViewItem.findViewById(R.id.tvmembername);
38         textViewName.setTypeface(font_title);
39         textViewName.setText(users.get(position).get(0));

```

```

40         //TEXTVIEW EMAIL USER
41         TextView textViewEmail = (TextView)
listViewItem.findViewById(R.id.tvmemberemail);
43         textViewEmail.setTypeface(font_desc);
44
45         //TEXTVIEW PRIVILEGE USER
46         TextView textViewPrivilege = (TextView)
listViewItem.findViewById(R.id.tvprivilege);
47         textViewPrivilege.setTypeface(font_title);
48         textViewEmail.setText(users.get(position).get(1));
49
50         if (users.get(position).get(2).length() > 6)
51
textViewPrivilege.setText(users.get(position).get(2).substring(0, 6));
52         else
53             textViewPrivilege.setText(users.get(position).get(2));
54         return listViewItem;
55     }
56}

```

Kode yang cukup penting pada *CreateNewActivity2.java* adalah:

- Baris 51 sampe 94 merupakan inisiasi berbagai komponen penyusun halaman ini termasuk dengan pengaturan *font*-nya. Pada baris 85 sampai 94, dilakukan *setOnClickListener* pada *TextView* “Add Member” agar ketika pengguna menekan *TextView* ini, pengguna dapat menuju halaman berikutnya untuk melakukan invitasi kepada member baru.
- Baris 97 sampai 152 merupakan pengaturan untuk *ListView* yang akan menampilkan seluruh member. Pada baris 97 sampai 99, tampilan maupun isi dari *ListView* dikonfigurasi menggunakan *ListMember.java*. pada baris 101 sampai 152, dilakukan *setOnClickListener* terhadap *ListView*, sehingga ketika *ListView* ditekan, instruksi-instruksi untuk mengirimkan perintah berupa JSON lakukan. Instruksi tersebut dapat dilihat pada bagian bawah penjelasan halaman ini.
- Baris 171 sampai 210 merupakan fungsi dari RPC yang dilakukan pada baris sebelumnya.

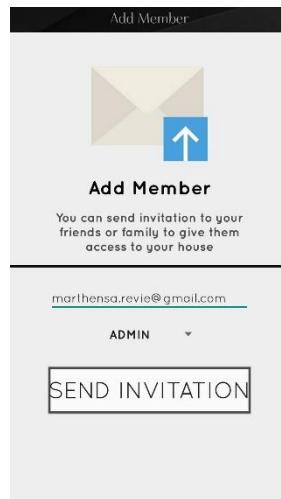
List member dibuat dengan menggunakan *ListView* yang didesain pada *file java* yang berbeda yaitu *ListMember.java*. Untuk menjadikan *ListMember.java* sebagai *list adapter* dari *CreateNewActivity2.java*, dilakukan *setAdapter* terhadap *ListMember*. *ListMember.java* sendiri berisikan objek apa saja yang akan ditampilkan dalam *ListView* pada *CreateNewActivity2.java*. Oleh karena itu, pada *ListMember.java*, dilakukan inisiasi nama, email dan privilege dalam sebuah *context* serta *font* yang digunakan oleh setiap objek

tersebut. Karena menggunakan *ListView*, ketika jumlah *member* melebihi kapasitas yang bisa ditampilkan pada layar, pengguna dapat melakukan *scroll* untuk melihat *member* yang tidak terlihat. Admin juga dapat melakukan menghapus *member* dalam *list* rumah dengan cara memilih menu pada *popup* menu yang muncul ketika member yang tersebut ditekan. Ketika pengguna menekan *TextView* “Add Member”, pengguna diarahkan menuju *AddMemberActivity.java* sedangkan ketika pengguna menekan *Button* “BACK”, Pengguna diarahkan menuju *MainActivity.java*.

Tabel 32 Tabel JSON House Member

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen Users)	list	• Home ID	Server	AES	Daftar seluruh <i>users</i> yang terdaftar di rumah yang bersangkutan

6.14 Halaman Add Member (*AddMemberAcitivty.java*)



Gambar 49 Halaman Add Member (*AddMemberActivity.java*)

Kode XML pada halaman ini sebagai berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_add_member"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9 
```

```

8      android:paddingLeft="0dp"
9      android:paddingRight="0dp"
10     android:paddingTop="0dp"
11     android:orientation="vertical"
12
13     tools:context="com.example.billyausten.newfinal.AddMemberActivity">
14
15     <LinearLayout
16         android:layout_width="match_parent"
17         android:layout_height="33dp"
18         android:layout_gravity="center"
19         android:orientation="vertical"
20         android:background="@drawable/backblack2"
21         android:id="@+id/backtitle">
22         <LinearLayout
23             android:layout_width="wrap_content"
24             android:layout_height="wrap_content"
25             android:orientation="horizontal"
26             android:layout_gravity="center">
27             <TextView
28                 android:layout_width="290dp"
29                 android:layout_height="wrap_content"
30                 android:text="Add Member"
31                 android:id="@+id/activitytitle"
32                 android:textAlignment="center"
33                 android:layout_gravity="center_vertical"
34                 android:textSize="18dp"
35                 android:textColor="#AAAAAA"
36                 android:layout_marginTop="3dp"/>
37
38         </LinearLayout>
39     </LinearLayout>
40
41     <ImageView
42         android:layout_width="150dp"
43         android:layout_height="150dp"
44         android:layout_gravity="center"
45         android:src="@drawable/upmail"
46         android:layout_marginTop="30dp"/>
47
48     <TextView
49         android:layout_width="match_parent"
50         android:layout_height="wrap_content"
51         android:text="Add Member"
52         android:layout_gravity="center"
53         android:textAlignment="center"
54         android:layout_marginTop="5dp"
55         android:textSize="25dp"
56         android:id="@+id/tvaddmember"
57         android:textColor="#000000"/>
58
59     <TextView
60         android:layout_width="250dp"
61         android:layout_height="wrap_content"
62         android:textAlignment="center"
63         android:layout_gravity="center"
64         android:textSize="16dp"

```

```

65        android:layout_marginTop="15dp"
66        android:text="You can send invitation to your friends or
family to give them access to your house"
67        android:id="@+id/tvnote"
68        android:textColor="#505050"/>
69
70    <ImageView
71        android:layout_width="match_parent"
72        android:layout_height="3dp"
73        android:layout_gravity="center"
74        android:layout_marginTop="15dp"
75        android:src="@drawable/pembatasdrawer2"/>
76
77    <EditText
78        android:layout_width="260dp"
79        android:layout_height="wrap_content"
80        android:layout_gravity="center"
81        android:hint="New member's email"
82        android:textSize="16dp"
83        android:id="@+id/etEmail"
84        android:inputType="textEmailAddress"
85        android:layout_marginTop="20dp" />
86
87    <Spinner
88        android:layout_width="120dp"
89        android:layout_height="wrap_content"
90        android:layout_gravity="center"
91        android:layout_marginTop="15dp"
92        android:id="@+id/spinner" />
93
94    <Button
95        android:layout_width="wrap_content"
96        android:layout_height="wrap_content"
97        android:layout_gravity="center"
98        android:text="Send Invitation"
99        android:layout_marginTop="30dp"
100       android:textSize="30dp"
101       android:id="@+id/bsend"
102       android:background="@drawable/signin"/>
103
104</LinearLayout>

```

Halaman ini diakses dengan menekan *TextView* “Add Member” pada *CreateNewActivity2.java*. Halaman ini memiliki fungsi sebagai sarana bagi tuan rumah untuk mengirimkan invitasi kepada pengguna lain yang ingin ikut mengakses rumah tersebut. Pada *Activity* ini, hanya terdapat sebuah *EditText*, yaitu *email* yang dapat digunakan untuk memasukan *email* dari *user* yang hendak diundang. Selain itu, terdapat juga sebuah *Spinner* yang berisi jenis *privilege* apa yang hendak diberikan kepada pengguna tersebut. Ada 2 jenis *privilege*, yaitu admin dan guest. Masing-masing memiliki perbedaan pada kedalaman akses rumah.

Kode java pada halaman ini sebagai berikut:

```
1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.view.View;
8 import android.view.ViewGroup;
9 import android.widget.AdapterView;
10 import android.widget.ArrayAdapter;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.Spinner;
14 import android.widget.TextView;
15 import android.widget.Toast;
16 import com.rabbitmq.client.AMQP;
17 import com.rabbitmq.client.DefaultConsumer;
18 import com.rabbitmq.client.Envelope;
19 import org.json.simple.JSONObject;
20 import java.io.IOException;
21 import java.io.StringWriter;
22 import java.util.ArrayList;
23 import java.util.List;
24 import java.util.UUID;
25 import java.util.concurrent.ArrayBlockingQueue;
26 import java.util.concurrent.BlockingQueue;
27
28 public class AddMemberActivity extends AppCompatActivity {
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_add_member);
34
35         //PENGATURAN FONT
36         Typeface font_signin =
37             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
38         Typeface font_title = Typeface.createFromAsset(getAssets(),
39             "fonts/Quicksand-Bold.otf");
40         Typeface font_or = Typeface.createFromAsset(getAssets(),
41             "fonts/Quicksand-BoldItalic.otf");
42         Typeface font_activity =
43             Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
44
45         //TEXTVIEW KETERANGAN
46         TextView tvAddMember = (TextView)
47             findViewById(R.id.tvaddmember);
48         tvAddMember.setTypeface(font_title);
49
50         //TEXTVIEW KETERANGAN
51         TextView tvNote = (TextView) findViewById(R.id.tvnote);
52         tvNote.setTypeface(font_title);
53
54         //EDITTEXT UTK EMAIL YANG HENDAK DIUNDANG
55         final EditText etEmail = (EditText)
56             findViewById(R.id.etEmail);
```

```

51         etEmail.setTypeface(font_signin);
52
53     //JUDUL ACTIVITY
54     TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
55         activitytitle.setTypeface(font_activity);
56
57     //SPINNER UTK PRIVILEGE
58     final Spinner spinner = (Spinner)
findViewById(R.id.spinner);
59     String[] categories = new String[2];
60     categories[0] = "ADMIN";
61     categories[1] = "Guest";
62
63     //MEMBUAT ADAPTER UTK SPINNER
64     ArrayAdapter<String> dataAdapter = new
ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
categories){
65         public View getView(int position, View convertView,
ViewGroup parent) {
66
67             //SET FONT
68             View v = super.getView(position, convertView,
parent);
69             Typeface
externalFont=Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
Bold.otf");
70             ((TextView) v).setTypeface(externalFont);
71             return v;
72         }
73
74
75         public View getDropDownView(int position, View
convertView, ViewGroup parent) {
76
77             //SET FONT
78             View v =super.getDropDownView(position,
convertView, parent);
79             Typeface
externalFont=Typeface.createFromAsset(getAssets(), "fonts/Quicksand-
Regular.otf");
80             ((TextView) v).setTypeface(externalFont);
81             return v;
82         }
83     };
84
85     // MENGATUR DROPODOWN SYTLE DARI SPINNER
86
dataAdapter.setDropDownViewResource(android.R.layout.simple_spinner_dr
opdown_item);
87
88     // SET ADAPTER UTK SPINNER
89     spinner.setAdapter(dataAdapter);
90
91     //BUTTON UTK MENGIRIM PERMINTAAN INVITE
92     Button bsend = (Button) findViewById(R.id.bsend);
93     bsend.setTypeface(font_signin);
94     bsend.setOnClickListener(new View.OnClickListener() {

```

```

95         @Override
96         public void onClick(View v) {
97             //mengirim pesan ke server berdasarkan data yang
telah dimasukan user
98             String response =
99             rpc(etEmail.getText().toString(), spinner.getSelectedItem().toString());
100            if (response.equals("1")) //Email belum didaftarkan
oleh invited user
101                Toast.makeText(AddMemberActivity.this, "The
email has not been signed up",
102                    Toast.LENGTH_LONG).show();
103            else if (response.equals("2")) //Pengirim bukan
ADMIN
104                Toast.makeText(AddMemberActivity.this, "You are
not ADMIN\nOnly ADMIN can manage users",
105                    Toast.LENGTH_LONG).show();
106            else { //Email sudah didaftarkan oleh invited user,
inviter menerima respon berupa kode undangan
107                Toast.makeText(AddMemberActivity.this,
"Invitation code: " + response,
108                    Toast.LENGTH_LONG).show();
109                finish();
110            }
111        });
112    }
113
114    public String rpc (String email, String privilege) {
115        //fungsi untuk mengirim pesan kepada server sebagai
permintaan untuk invite
116        //pengiriman menggunakan skema RPC
117        try {
118            String replyQueueName =
MainActivity.channel.queueDeclare().getQueue();
119
120            AMQP.BasicProperties props = new AMQP.BasicProperties
121                .Builder()
122                .correlationId(MainActivity.phoneid)
123                .replyTo(replyQueueName)
124                .build();
125            JSONObject obj = new JSONObject();
126
127            //data yang dikirimkan
128            obj.put("homeid", MainActivity.homeid);
129            obj.put("sender", MainActivity.email);
130            obj.put("head", "MU");
131            obj.put("type", "register");
132            obj.put("email", email);
133            obj.put("privilege", privilege);
134            StringWriter toSent = new StringWriter();
135            obj.writeJSONString(toSent);
136            String message = toSent.toString();
137            //mengirim pesan
138            MainActivity.channel.basicPublish("", "AESqueue",
props, MainActivity.AESencrypt(message).getBytes("UTF-8"));

```

```

139         MainActivity.channel.basicPublish("",  

MainActivity.homeid+"AES", props,  

MainActivity.AESencrypt(message).getBytes("UTF-8"));  

140  

141         //menunggu dalam antrian  

142         final BlockingQueue<String> response = new  

ArrayBlockingQueue<>(2);  

143         MainActivity.channel.basicConsume(replyQueueName, true,  

new DefaultConsumer(MainActivity.channel) {  

144             @Override  

145             public void handleDelivery(String consumerTag,  

Envelope envelope, AMQP.BasicProperties properties, byte[] body)  

throws IOException {  

146                 if  

(properties.getCorrelationId().equals(MainActivity.phoneid)) {  

147                     response.offer(new String(body, "UTF-8"));  

148                 }  

149             }  

150         };  

151     };  

152     //jawaban  

153     String res = MainActivity.AESdecrypt(response.take());  

154     return res;  

155  

156 } catch (Exception e) {  

157     //gagal terkoneksi ke server  

158     Toast.makeText(getApplicationContext(),"Can't connect  

to the server\nCheck your connection",Toast.LENGTH_SHORT).show();  

159     return null;  

160 }  

161 }  

162 }

```

Kode yang cukup penting pada java diatas adalah:

- Baris 36 sampai 55 merupakan inisiasi komponen-komponen penyusun halaman ini termasuk pengaturan *font*.
- Baris 58 sampai 89 merupakan pengaturan *Spinner*. Pada baris 58 sampai 61, dilakukan inisiasi *Spinner* untuk pemilihan *privilege*. *Spinner* berisi dua buah kategori *privilege* yaitu admin dan guest dengan tipe *string*. Pada baris 64 sampai 83 kode digunakan mengganti *font* dari *Spinner*. Pengubahan *font* pada *Spinner* membutuhkan *ArrayAdapter* yang berisikan *getView* untuk pilihan yang dipilih dan *getDropDownView* untuk seluruh pilihan *Spinner*.
- Baris 94 sampai 111 berisi instruksi-instruksi yang dikirimkan aplikasi ketika *Button* “SEND” ditekan. Ketika *Button* ditekan, respon aplikasi bergantung dari kondisi yang sedang terjadi. Ketika pengguna sudah memenuhi syarat, kode invitasi akan

ditampilkan dengan menggunakan *Toast*. Jika tidak maka akan ada *Toast* lain yang memperingatkan pengguna.

- Baris 114 sampai 161 merupakan fungsi yang dapat digunakan aplikasi untuk mengirimkan perintah kepada server berupa data JSON dengan skema pengiriman RPC.

Tabel 33 Tabel JSON Add Member

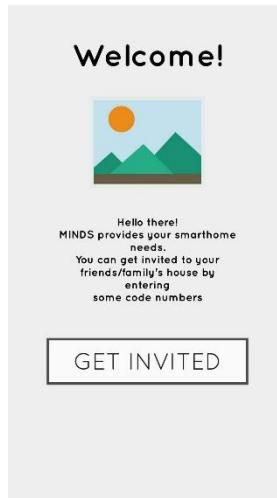
Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	register	<ul style="list-style-type: none"> • Home ID • Email yang ingin diundang • Privilege 	Server	AES	<p>Bila email terundang valid (belum terdaftar di rumah lain):</p> <ul style="list-style-type: none"> • 6 digit kode registrasi <p>Bila email terundang sudah ada di rumah lain:</p> <ul style="list-style-type: none"> • Karakter ‘1’

Perlu diketahui bahwa *email* yang dimasukan haruslah *email* yang pernah melakukan *sign up* (seperti yang dijelaskan pada *SignUpActivity.java*). Dengan demikian, ketika server menerima pesan di atas, hal pertama yang dilakukan adalah pengecekan apakah *email* tersebut sudah pernah *sign up*. Bila sudah, maka *email* tersebut akan ditambahkan sebagai *user* yang dapat mengakses rumah tersebut. Namun, supaya *user* benar-benar terkonfirmasi, maka server akan melakukan generasi 6 digit angka, lalu dikirimkan sebagai balasan kepada pengundang. Enam digit angka ini adalah kode yang harus dimasukan oleh *user* yang diundang pada *AddMemberActivity.java* (telah dijelaskan pada bagian sebelumnya). Dengan demikian, selama kode ini belum dimasukan oleh *user* terundang, *user* terundang tersebut belum bisa melakukan akses ke rumah. Setelah kode dimasukan, barulah *user* bisa mengakses rumah.

Sebaliknya, bila *user* yang hendak diundang ternyata belum terdaftar atau belum melakukan *sign up*, maka server akan membalas dengan pesan ‘0’, dan pada MINDS akan muncul *Toast* “Email has not been registered”.

Ketika tombol “SEND INVITATION” ditekan, maka MINDS akan kembali berpindah ke *CreateNewActivity2.java*.

6.15 Halaman GettingStarted (GettingStartedActivity.java)



Gambar 50 Halaman Getting Started (GettingStartedActivity.java)

Kode XML pada halaman ini sebagai berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_login"
6     android:layout_width="match_parent"
7     android:orientation="vertical"
8     android:layout_height="match_parent"
9     android:paddingBottom="@dimen/activity_vertical_margin"
10    android:paddingLeft="@dimen/activity_horizontal_margin"
11    android:paddingRight="@dimen/activity_horizontal_margin"
12    android:paddingTop="@dimen/activity_vertical_margin"
13    tools:context="com.example.billyausten.newfinal.LoginActivity"
14    android:weightSum="1">
15
16    <TextView
17        android:layout_width="wrap_content"
18        android:layout_height="wrap_content"
19        android:layout_gravity="center"
20        android:textSize="40dp"
21        android:textColor="#000000"
22        android:id="@+id/tvWelcome"
23        android:text="Welcome!"
24        android:layout_marginTop="25dp"/>
25
26    <ImageView
27        android:layout_width="150dp"
28        android:layout_height="150dp"
29        android:layout_gravity="center"
30        android:src="@drawable/welcomeimg"
31        android:layout_marginTop="15dp"
32        android:layout_marginBottom="20dp"/>
```

```

32
33     <TextView
34         android:layout_width="wrap_content"
35         android:layout_height="wrap_content"
36         android:text="Hello there!"
37         android:layout_gravity="center"
38         android:textSize="14dp"
39         android:textColor="#000000"
40         android:id="@+id/tvwelcome1"/>
41
42     <TextView
43         android:layout_width="250dp"
44         android:layout_height="wrap_content"
45         android:text="MINDS provides your smarthome needs."
46         android:layout_gravity="center"
47         android:textAlignment="center"
48         android:textSize="14dp"
49         android:textColor="#000000"
50         android:id="@+id/tvwelcome2"/>
51
52     <TextView
53         android:layout_width="230dp"
54         android:layout_height="wrap_content"
55         android:text="You can get invited to your friends/family's
house by entering"
56         android:layout_gravity="center"
57         android:textAlignment="center"
58         android:textSize="14dp"
59         android:textColor="#000000"
60         android:id="@+id/tvwelcome3"/>
61
62     <TextView
63         android:layout_width="230dp"
64         android:layout_height="wrap_content"
65         android:text="some code numbers"
66         android:layout_gravity="center"
67         android:textAlignment="center"
68         android:textSize="14dp"
69         android:textColor="#000000"
70         android:id="@+id/tvwelcome4"/>
71
72     <Button
73         android:layout_width="wrap_content"
74         android:layout_height="wrap_content"
75         android:layout_gravity="center"
76         android:text="Get Invited"
77         android:layout_marginTop="45dp"
78         android:textSize="30dp"
79         android:id="@+id/bnext"
80         android:background="@drawable/signin"/>
81
82</LinearLayout>

```

Halaman *GettingStartedActivity.java* hanya akan ditampilkan kepada pengguna baru karena pengguna baru belum memiliki rumah yang terdaftar pada aplikasi MINDS sehingga pengguna baru perlu didaftarkan oleh admin rumah untuk dapat mengakses aplikasi

MINDS. Untuk memasukan kode izin dari admin rumah, pengguna dapat menekan *Button* “GET INVITED” agar dapat menuju ke *InviteActivity1.java*. Seperti pada halaman pertama, perpindahan antar *Activity* dilakukan dengan menggunakan *Intent* pada *setOnClickListener* kedua *Button*. Pada *GettingStartedActivity.java* ini, ditambahkan juga kode “OnBackPressed” agar ketika pengguna menekan *Button back* pengguna akan keluar dari aplikasi dan pengguna tidak akan kembali ke *SignInActivity.java*.

Kode java pada halaman ini sebagai berikut:

```
1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;
5 import android.os.Bundle;
6 import android.support.v7.app.AppCompatActivity;
7 import android.view.View;
8 import android.widget.Button;
9 import android.widget.TextView;
10
11public class GettingStartedActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_getting_started);
17
18         //PENGATURAN FONT
19         Typeface font_title = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Bold.otf");
20         Typeface font_signin = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Regular.otf");
21
22         //TEXTVIEW KETERANGAN
23         TextView tvwelcome = (TextView)
findViewById(R.id.tvWelcome);
24         tvwelcome.setTypeface(font_title);
25         TextView tvwelcome1 = (TextView)
findViewById(R.id.tvwelcome1);
26         tvwelcome1.setTypeface(font_title);
27         TextView tvwelcome2 = (TextView)
findViewById(R.id.tvwelcome2);
28         tvwelcome2.setTypeface(font_title);
29         TextView tvwelcome3 = (TextView)
findViewById(R.id.tvwelcome3);
30         tvwelcome3.setTypeface(font_title);
31         TextView tvwelcome4 = (TextView)
findViewById(R.id.tvwelcome4);
32         tvwelcome4.setTypeface(font_title);
33
34         //TOMBOL UNTUK MELANJUTKAN KE HALAMAN BERIKUTNYA, YAITU
HALAMAN UNTUK MEMASUKAN KODE UNIK PENGUNDANGAN
35         Button bnnext = (Button) findViewById(R.id.bnnext);
36         bnnext.setTypeface(font_signin);
```

```

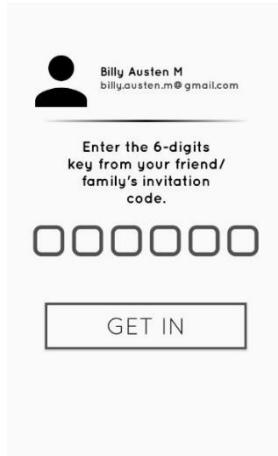
37         bnext.setOnClickListener(new View.OnClickListener() {
38
39             @Override
40             public void onClick(View v) {
41                 Intent afterloginIntent = new
Intent(GettingStartedActivity.this, InviteActivity1.class);
42
GettingStartedActivity.this.startActivity(afterloginIntent);
43             }
44         });
45     }
46 }

```

Kode yang cukup penting pada java diatas adalah:

- Baris 19 sampai 36 merupakan inisiasi komponen-komponen penyusun halaman ini. Pengaturan *font* juga dilakukan pada baris ini
- Baris 37 sampai 44 merupakan instruksi ketika *Button* “GET INVITED” ditekan.

6.16 Halaman Get Invited (*InviteActivity1.java*)



Gambar 51 Halaman Get Invited (*InviteActivity1.java*)

Kode XML pada halaman ini sebagai berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_invite1"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="@dimen/activity_horizontal_margin"
10    android:paddingRight="@dimen/activity_horizontal_margin"
11    android:paddingTop="@dimen/activity_vertical_margin"

```

```

11      android:orientation="vertical"
12
13      <LinearLayout
14          android:layout_width="match_parent"
15          android:layout_marginTop="35dp"
16          android:layout_height="wrap_content"
17          android:orientation="horizontal"
18          android:layout_gravity="center">
19
20          <ImageView
21              android:layout_width="100dp"
22              android:layout_height="100dp"
23              android:src="@drawable/ic_person_black_24dp"
24              android:layout_gravity="center"
25              android:layout_marginLeft="5dp"/>
26
27          <LinearLayout
28              android:layout_width="215dp"
29              android:layout_height="match_parent"
30              android:orientation="vertical">
31
32          <TextView
33              android:layout_width="match_parent"
34              android:layout_height="wrap_content"
35              android:text="Billy Austen M"
36              android:id="@+id/tvname"
37              android:textSize="16dp"
38              android:paddingTop="28dp"
39              android:textColor="#000000"/>
40
41          <TextView
42              android:layout_width="match_parent"
43              android:layout_height="wrap_content"
44              android:text="billy.austen.m@gmail.com"
45              android:id="@+id/tvemail"
46              android:textSize="14dp"
47              android:textColor="#404040"/>
48
49      </LinearLayout>
50  </LinearLayout>
51
52  <ImageView
53      android:layout_width="match_parent"
54      android:layout_height="3dp"
55      android:src="@drawable/pembatasdrawer2"/>
56
57  <TextView
58      android:layout_width="220dp"
59      android:layout_gravity="center"
60      android:layout_height="wrap_content"
61      android:layout_marginTop="20dp"
62      android:textSize="19dp"
63      android:textColor="#000000"
64      android:textAlignment="center"
65      android:id="@+id/tvdigit"
66      android:text="Enter the 6-digits key from your
friend/family's invitation code."/>

```

```

67
68    <LinearLayout
69        android:layout_width="wrap_content"
70        android:layout_height="wrap_content"
71        android:orientation="horizontal"
72        android:layout_gravity="center"
73        android:layout_marginTop="25dp">
74
75        <EditText
76            android:layout_width="40dp"
77            android:layout_height="40dp"
78            android:background="@drawable/digitcode"
79            android:id="@+id/etdigit1"
80            android:textSize="20dp"
81            android:maxLength="1"
82            android:inputType="number"
83            android:textAlignment="center"/>
84
85        <EditText
86            android:layout_width="40dp"
87            android:layout_height="40dp"
88            android:background="@drawable/digitcode"
89            android:layout_marginLeft="10dp"
90            android:id="@+id/etdigit2"
91            android:textSize="20dp"
92            android:maxLength="1"
93            android:inputType="number"
94            android:textAlignment="center"/>
95
96        <EditText
97            android:layout_width="40dp"
98            android:layout_height="40dp"
99            android:background="@drawable/digitcode"
100           android:layout_marginLeft="10dp"
101           android:id="@+id/etdigit3"
102           android:textSize="20dp"
103           android:maxLength="1"
104           android:inputType="number"
105           android:textAlignment="center"/>
106
107        <EditText
108            android:layout_width="40dp"
109            android:layout_height="40dp"
110            android:background="@drawable/digitcode"
111            android:layout_marginLeft="10dp"
112            android:id="@+id/etdigit4"
113            android:textSize="20dp"
114            android:maxLength="1"
115            android:inputType="number"
116            android:textAlignment="center"/>
117
118        <EditText
119            android:layout_width="40dp"
120            android:layout_height="40dp"
121            android:background="@drawable/digitcode"
122            android:layout_marginLeft="10dp"
123            android:id="@+id/etdigit5"
124            android:textSize="20dp"

```

```

125        android:maxLength="1"
126        android:inputType="number"
127        android:textAlignment="center"/>
128
129    <EditText
130        android:layout_width="40dp"
131        android:layout_height="40dp"
132        android:background="@drawable/digitcode"
133        android:layout_marginLeft="10dp"
134        android:textSize="20dp"
135        android:textAlignment="center"
136        android:maxLength="1"
137        android:inputType="number"
138        android:id="@+id/etdigit6"/>
139
140
141    <Button
142        android:layout_width="wrap_content"
143        android:layout_height="wrap_content"
144        android:layout_gravity="center"
145        android:text="Get In"
146        android:layout_marginTop="60dp"
147        android:textSize="30dp"
148        android:id="@+id/bgetin"
149        android:background="@drawable/signin"/>
150
151</LinearLayout>

```

Halaman ini dapat diakses melalui *GettingStartedActivity.java* dengan menekan *Button* “GET INVITED” ataupun dari *MainActivity.java* dengan menekan menu “Get Invited” pada *Navigation Drawer*. Pengguna baru dapat menggunakan halaman ini untuk memasukan *invitation code* dari admin rumah agar dapat bisa menggunakan MINDS untuk mengatur *device* pada rumah yang terdaftar. Untuk mendapatkan *invitation code*, admin dapat melakukan *request* kepada server dengan cara mengisi *email* dan nama serta *privilege* dari pengguna baru pada *AddMemberActivity.java* dan menekan *Button* “SEND INVITATION” untuk segera mengakses *invitation code*. *Privilege* dapat dipilih dari *Spinner* yang terletak dibawah *EditText* “Email”. Setelah mengajukan penambahan member baru, admin akan mendapatkan notifikasi berupa enam digit angka *invitation code* yang dapat diisi oleh pengguna baru pada *InviteActivity1.java*. Pengguna baru yang berhasil mendaftarkan diri menggunakan *invitation code* yang dikirimkan oleh admin langsung dapat menggunakan MINDS dengan konfigurasi *email*, nama dan *privilege* yang sudah diatur oleh admin.

Untuk melakukan *input invitation code*, pengguna dapat mengisi setiap digit angka satu per satu pada setiap *EditText* dengan *background* persegi. *EditText* menggunakan

InputType “Number” sehingga *keyboard* yang akan tampil pada saat mengisi *EditText* ini adalah *keyboard* numerik. Pada *EditText* ini juga diberikan kode “maxlength=1” untuk membatasi jumlah karakter yang dapat pengguna isi pada setiap *EditText* sebanyak satu karakter. Setelah mengisi *invitation code*, pengguna dapat langsung menekan *Button* “Get In” agar server dapat segera melakukan verifikasi kode. Ketika telah disetujui, pengguna dapat langsung menuju ke *MainActivity.java* dan menggunakan seluruh fitur aplikasi MINDS. Jika *invitation code* yang di salah, maka akan muncul *Toast* yang memberitahu bahwa *invitation code* yang dimasukan salah.

Kode java pada halaman ini sebagai berikut:

```
1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.content.SharedPreferences;
5 import android.graphics.Typeface;
6 import android.os.Bundle;
7 import android.os.StrictMode;
8 import android.provider.Settings;
9 import android.support.v7.app.AppCompatActivity;
10 import android.view.View;
11 import android.widget.Button;
12 import android.widget.EditText;
13 import android.widget.TextView;
14 import android.widget.Toast;
15 import com.rabbitmq.client.AMQP;
16 import com.rabbitmq.client.Channel;
17 import com.rabbitmq.client.Connection;
18 import com.rabbitmq.clientConnectionFactory;
19 import com.rabbitmq.client.DefaultConsumer;
20 import com.rabbitmq.client.Envelope;
21 import org.json.simple.JSONObject;
22 import org.json.simple.parser.JSONParser;
23 import java.io.IOException;
24 import java.io.StringWriter;
25 import java.util.Random;
26 import java.util.UUID;
27 import java.util.concurrent.ArrayBlockingQueue;
28 import java.util.concurrent.BlockingQueue;
29
30 public class InviteActivity1 extends AppCompatActivity {
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState) {
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.activity_invite1);
36
37         StrictMode.ThreadPolicy policy = new
38         StrictMode.ThreadPolicy.Builder().permitAll().build();
39         StrictMode.setThreadPolicy(policy);
```

```

40         //PENGATURAN FONT
41         Typeface font_title = Typeface.createFromAsset(getAssets() ,
"fonts/Quicksand-Bold.otf");
42         Typeface font_signin =
Typeface.createFromAsset(getAssets() , "fonts/Quicksand-Regular.otf");
43
44         //TEXTVIEW NAMA USER
45         TextView tvname = (TextView) findViewById(R.id.tvname);
46         tvname.setTypeface(font_title);
47         tvname.setText(MainActivity.usname);
48
49         //TEXTVIEW EMAIL USER
50         TextView tvemail = (TextView) findViewById(R.id.tvemail);
51         tvemail.setTypeface(font_title);
52         tvemail.setText(MainActivity.email);
53
54         //TEXTVIEW KETERANGAN
55         TextView tvdigit = (TextView) findViewById(R.id.tvdigit);
56         tvdigit.setTypeface(font_title);
57
58         //EDITTEXT UNTUK MEMASUKAN 6 DIGIT KODE INVITASI
59         final EditText etdigit1 = (EditText)
findViewById(R.id.etdigit1);
60         etdigit1.setTypeface(font_signin);
61         final EditText etdigit2 = (EditText)
findViewById(R.id.etdigit2);
62         etdigit2.setTypeface(font_signin);
63         final EditText etdigit3 = (EditText)
findViewById(R.id.etdigit3);
64         etdigit3.setTypeface(font_signin);
65         final EditText etdigit4 = (EditText)
findViewById(R.id.etdigit4);
66         etdigit4.setTypeface(font_signin);
67         final EditText etdigit5 = (EditText)
findViewById(R.id.etdigit5);
68         etdigit5.setTypeface(font_signin);
69         final EditText etdigit6 = (EditText)
findViewById(R.id.etdigit6);
70         etdigit6.setTypeface(font_signin);
71
72         //BUTTON UNTUK MENGIRIM KODE UNIK
73         Button bgetin = (Button) findViewById(R.id.bgetin);
74         bgetin.setTypeface(font_signin);
75         bgetin.setOnClickListener(new View.OnClickListener() {
76
77             @Override
78             public void onClick(View v) {
79                 String code = etdigit1.getText().toString() +
etdigit2.getText().toString() + etdigit3.getText().toString() +
etdigit4.getText().toString() + etdigit5.getText().toString() +
etdigit6.getText().toString();
80                 String response = rpc(code);
81                 JSONParser parser = new JSONParser();
82                 try {
83                     JSONObject jsonObject = (JSONObject)
parser.parse(response);
84                     String id = (String) jsonObject.get("homeid");

```

```

85             String priv = (String)
jsonObject.get("privilege");
86
87             if (id.equals("0"))
Toast.makeText(InviteActivity1.this, "Wrong
invitation code",
88                                         Toast.LENGTH_SHORT).show();
88
89             else {
90                 //MENGIRIM PESAN SIGN IN KEPADA HOST
91                 try {
92                     AMQP.BasicProperties props = new
AMQP.BasicProperties
93                                         .Builder()
94                                         .correlationId(MainActivity.phoneid)
95                                         .build();
96
97                     JSONObject obj = new JSONObject();
98                     obj.put("head", "SI");
99                     obj.put("aeskey",
100 MainActivity.AESkeyString);
101
102                     obj.put("email",MainActivity.email);
103                     obj.put("privilege",priv);
103
StringWriter toSent = new
StringWriter();
104
obj.writeJSONObject(toSent);
105
String message = toSent.toString();
106
MainActivity.channel.basicPublish("", id+"RSA", props, MainActivity.RSAencrypt(message).getBytes("UTF-8"));
107
} catch (Exception e) {
109
Toast.makeText(getApplicationContext(),"gagal
kirim",Toast.LENGTH_SHORT).show();
110
}
111
SharedPreferences sharedPref =
getSharedPreferences("Authentication", MODE_PRIVATE);
112
SharedPreferences.Editor prefEditor =
sharedPref.edit();
113
prefEditor.putString("homeid", id);
114
String usname = (String)
jsonObject.get("usname");
115
prefEditor.putString("usname", usname);
116
prefEditor.putString("profpic", "none");
117
prefEditor.putString("privilege", priv);
118
prefEditor.apply();
119
Intent afterloginIntent = new
Intent(InviteActivity1.this, MainActivity.class);
120
InviteActivity1.this.startActivity(afterloginIntent);
121
}
122
} catch (Exception e) {
123
}
124
}
125
}
126
);
127
}
128
}

```

```

129     public String rpc (String code) {
130         //FUNGSI UNTUK MENGIRIM PESAN KODE INVITASI DENGAN SKEMA
131         RPC
132         try {
133             String replyQueueName =
134             MainActivity.channel.queueDeclare().getQueue ();
135             //properti
136             AMQP.BasicProperties props = new AMQP.BasicProperties
137                 .Builder ()
138                 .correlationId (MainActivity.phoneid)
139                 .replyTo (replyQueueName)
140                 .build ();
141
142             //pesan JSON
143             JSONObject obj = new JSONObject ();
144             obj.put ("head", "MU");
145             obj.put ("type", "confirm");
146             obj.put ("email", MainActivity.email);
147             obj.put ("code", code);
148             StringWriter toSent = new StringWriter ();
149             obj.writeJSONObject (toSent);
150             String message = toSent.toString ();
151             MainActivity.channel.basicPublish ("", "AESqueue",
152             props, MainActivity.AESencrypt (message).getBytes ("UTF-8"));
153
154             //menunggu balasan
155             final BlockingQueue<String> response = new
156             ArrayBlockingQueue<>(2);
157             MainActivity.channel.basicConsume (replyQueueName, true,
158             new DefaultConsumer (MainActivity.channel) {
159                 @Override
160                 public void handleDelivery (String consumerTag,
161                 Envelope envelope, AMQP.BasicProperties properties, byte[] body)
162                 throws IOException {
163                     if
164                     (properties.getCorrelationId ().equals (MainActivity.phoneid)) {
165                         response.offer (new String (body, "UTF-8"));
166                     }
167                 }
168             });
169             //mengambil balasan
170             String res = MainActivity.AESdecrypt (response.take ());
171             return res;
172         } catch (Exception e) {
173             Toast.makeText (getApplicationContext (), "Can't connect
174             to the server\nCheck your connection", Toast.LENGTH_SHORT).show ();
175             return null;
176         }
177     }
178 }
```

Kode yang cukup penting pada java diatas adalah:

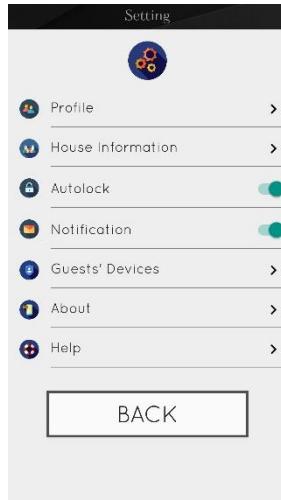
- Baris 41 sampai 74 merupakan inisiasi dari komponen-komponen penyusun halaman ini. Pengaturan *font* juga terdapat dalam baris ini.
- Baris 75 sampai 127 berisi seluruh instruksi ketika *Button* “GET IN” ditekan. Instruksi yang dikirimkan berupa JSON dengan format seperti dibawah.
- Baris 129 sampai 169 merupakan fungsi untuk mengirimkan kode invitasional menggunakan skema RPC.

Tabel 34 Tabel JSON pada get invited

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	Confirm	<ul style="list-style-type: none"> • 6 digit kode konfirmasi 	Server	AES	<p>Bila kode benar:</p> <ul style="list-style-type: none"> • Home ID <p>Bila kode salah: Jawaban berupa karakter '0'</p>

Bila MINDS menerima balasan berupa ID Rumah, maka MINDS akan berpindah *Activity* ke *MainActivity.java*. Bila balasan yang diterima berupa ‘0’, maka MINDS akan menampilkan pesan toast “Wrong invitation code”.

6.17 Halaman Setting (*SettingActivity.java*)



Gambar 52 Halaman Setting (*SettingActivity.java*)

Kode XML pada halaman ini sebagai berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
```

```

2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_setting"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="0dp"
10    android:paddingRight="0dp"
11    android:paddingTop="0dp"
12
13    tools:context="com.example.billyausten.newfinal.SettingActivity"
14    android:orientation="vertical">
15
16    <LinearLayout
17        android:layout_width="match_parent"
18        android:layout_height="33dp"
19        android:layout_gravity="center"
20        android:orientation="vertical"
21        android:background="@drawable/backblack2"
22        android:id="@+id/backtitle">
23            <LinearLayout
24                android:layout_width="wrap_content"
25                android:layout_height="wrap_content"
26                android:orientation="horizontal"
27                android:layout_gravity="center">
28
29                <TextView
30                    android:layout_width="290dp"
31                    android:layout_height="wrap_content"
32                    android:text="Setting"
33                    android:id="@+id/activitytitle"
34                    android:textAlignment="center"
35                    android:layout_gravity="center_vertical"
36                    android:textSize="18dp"
37                    android:textColor="#AAAAAA"
38                    android:layout_marginTop="3dp"/>
39
40
41            <ImageView
42                android:layout_width="50dp"
43                android:layout_height="50dp"
44                android:src="@drawable/settingicon"
45                android:layout_marginTop="15dp"
46                android:layout_gravity="center"/>
47
48            <LinearLayout
49                android:layout_width="match_parent"
50                android:layout_height="wrap_content"
51                android:layout_marginTop="25dp">
52
53                <ImageView
54                    android:layout_width="25dp"
55                    android:layout_height="25dp"
56                    android:src="@drawable/setprof"
57                    android:backgroundTint="#808080"

```

```

58         android:layout_marginLeft="15dp"/>
59
60     <TextView
61         android:layout_width="275dp"
62         android:layout_height="wrap_content"
63         android:textSize="18dp"
64         android:text="Profile"
65         android:textColor="#202020"
66         android:layout_marginLeft="15dp"
67         android:id="@+id/tvprofile"/>
68
69     <ImageView
70         android:layout_width="25dp"
71         android:layout_height="25dp"
72         android:src="@drawable/ic_chevron_right_black_24dp"
73         android:backgroundTint="#808080"
74         android:id="@+id/ivprof"/>
75 </LinearLayout>
76
77     <ImageView
78         android:layout_width="match_parent"
79         android:layout_height="1dp"
80         android:layout_marginLeft="55dp"
81         android:backgroundTint="#808080"
82         android:background="@drawable/side_nav_bar"
83         android:layout_marginTop="10dp"/>
84
85     <LinearLayout
86         android:layout_width="match_parent"
87         android:layout_height="wrap_content"
88         android:layout_marginTop="15dp">
89
90         <ImageView
91             android:layout_width="25dp"
92             android:layout_height="25dp"
93             android:src="@drawable/sethouse"
94             android:backgroundTint="#808080"
95             android:layout_marginLeft="15dp"/>
96
97         <TextView
98             android:layout_width="275dp"
99             android:layout_height="wrap_content"
100            android:textSize="18dp"
101            android:text="House Information"
102            android:textColor="#202020"
103            android:layout_marginLeft="15dp"
104            android:id="@+id/tvhouse"/>
105
106         <ImageView
107             android:layout_width="25dp"
108             android:layout_height="25dp"
109             android:src="@drawable/ic_chevron_right_black_24dp"
110             android:backgroundTint="#808080"
111             android:id="@+id/ivhouse"/>
112     </LinearLayout>
113
114     <ImageView
115         android:layout_width="match_parent"

```

```

116        android:layout_height="1dp"
117        android:layout_marginLeft="55dp"
118        android:backgroundTint="#808080"
119        android:background="@drawable/side_nav_bar"
120        android:layout_marginTop="10dp"/>
121
122    <LinearLayout
123        android:layout_width="match_parent"
124        android:layout_height="wrap_content"
125        android:layout_marginTop="15dp">
126
127        <ImageView
128            android:layout_width="25dp"
129            android:layout_height="25dp"
130            android:src="@drawable/setlock"
131            android:backgroundTint="#808080"
132            android:layout_marginLeft="15dp"/>
133
134        <TextView
135            android:layout_width="260dp"
136            android:layout_height="wrap_content"
137            android:textSize="18dp"
138            android:text="Autolock"
139            android:textColor="#202020"
140            android:layout_marginLeft="15dp"
141            android:id="@+id/tvlock"/>
142
143        <Switch
144            android:layout_width="45dp"
145            android:layout_height="25dp"
146            android:id="@+id/swlock"/>
147    </LinearLayout>
148
149    <ImageView
150        android:layout_width="match_parent"
151        android:layout_height="1dp"
152        android:layout_marginLeft="55dp"
153        android:backgroundTint="#808080"
154        android:background="@drawable/side_nav_bar"
155        android:layout_marginTop="10dp"/>
156
157    <LinearLayout
158        android:layout_width="match_parent"
159        android:layout_height="wrap_content"
160        android:layout_marginTop="15dp">
161
162        <ImageView
163            android:layout_width="25dp"
164            android:layout_height="25dp"
165            android:src="@drawable/setnotification"
166            android:backgroundTint="#808080"
167            android:layout_marginLeft="15dp"/>
168
169        <TextView
170            android:layout_width="260dp"
171            android:layout_height="wrap_content"
172            android:textSize="18dp"
173            android:text="Notification"

```

```

174         android:textColor="#202020"
175         android:layout_marginLeft="15dp"
176         android:id="@+id/tvnotif"/>
177
178     <Switch
179         android:layout_width="45dp"
180         android:layout_height="25dp"
181         android:id="@+id/swnotif"/>
182 </LinearLayout>
183
184 <ImageView
185     android:layout_width="match_parent"
186     android:layout_height="1dp"
187     android:layout_marginLeft="55dp"
188     android:backgroundTint="#808080"
189     android:background="@drawable/side_nav_bar"
190     android:layout_marginTop="10dp"/>
191
192 <LinearLayout
193     android:layout_width="match_parent"
194     android:layout_height="wrap_content"
195     android:layout_marginTop="15dp">
196
197     <ImageView
198         android:layout_width="25dp"
199         android:layout_height="25dp"
200         android:src="@drawable/devprivr"
201         android:backgroundTint="#808080"
202         android:layout_marginLeft="15dp"/>
203
204     <TextView
205         android:layout_width="275dp"
206         android:layout_height="wrap_content"
207         android:textSize="18dp"
208         android:text="Guests' Devices"
209         android:textColor="#202020"
210         android:layout_marginLeft="15dp"
211         android:id="@+id/tvprivilege"/>
212
213     <ImageView
214         android:layout_width="25dp"
215         android:layout_height="25dp"
216         android:src="@drawable/ic_chevron_right_black_24dp"
217         android:backgroundTint="#808080"
218         android:id="@+id/ivprivilege"/>
219 </LinearLayout>
220
221 <ImageView
222     android:layout_width="match_parent"
223     android:layout_height="1dp"
224     android:layout_marginLeft="55dp"
225     android:backgroundTint="#808080"
226     android:background="@drawable/side_nav_bar"
227     android:layout_marginTop="10dp"/>
228
229 <LinearLayout
230     android:layout_width="match_parent"
231     android:layout_height="wrap_content"

```

```

232         android:layout_marginTop="15dp">
233
234     <ImageView
235         android:layout_width="25dp"
236         android:layout_height="25dp"
237         android:src="@drawable/setabout"
238         android:backgroundTint="#808080"
239         android:layout_marginLeft="15dp"/>
240
241     <TextView
242         android:layout_width="275dp"
243         android:layout_height="wrap_content"
244         android:textSize="18dp"
245         android:text="About"
246         android:textColor="#202020"
247         android:layout_marginLeft="15dp"
248         android:id="@+id/tvabout"/>
249
250     <ImageView
251         android:layout_width="25dp"
252         android:layout_height="25dp"
253         android:src="@drawable/ic_chevron_right_black_24dp"
254         android:backgroundTint="#808080"
255         android:id="@+id/ivabout"/>
256 </LinearLayout>
257
258     <ImageView
259         android:layout_width="match_parent"
260         android:layout_height="1dp"
261         android:layout_marginLeft="55dp"
262         android:backgroundTint="#808080"
263         android:background="@drawable/side_nav_bar"
264         android:layout_marginTop="10dp"/>
265
266     <LinearLayout
267         android:layout_width="match_parent"
268         android:layout_height="wrap_content"
269         android:layout_marginTop="15dp">
270
271         <ImageView
272             android:layout_width="25dp"
273             android:layout_height="25dp"
274             android:src="@drawable/sethelp"
275             android:backgroundTint="#808080"
276             android:layout_marginLeft="15dp"/>
277
278         <TextView
279             android:layout_width="275dp"
280             android:layout_height="wrap_content"
281             android:textSize="18dp"
282             android:text="Help"
283             android:textColor="#202020"
284             android:layout_marginLeft="15dp"
285             android:id="@+id/tvhelp"/>
286
287         <ImageView
288             android:layout_width="25dp"
289             android:layout_height="25dp"

```

```

290         android:src="@drawable/ic_chevron_right_black_24dp"
291         android:backgroundTint="#808080"
292         android:id="@+id/ivhelp"/>
293     </LinearLayout>
294
295     <ImageView
296         android:layout_width="match_parent"
297         android:layout_height="1dp"
298         android:layout_marginLeft="55dp"
299         android:backgroundTint="#808080"
300         android:background="@drawable/side_nav_bar"
301         android:layout_marginTop="10dp"/>
302
303     <Button
304         android:layout_width="wrap_content"
305         android:layout_height="wrap_content"
306         android:layout_gravity="center"
307         android:text="Back"
308         android:layout_marginTop="30dp"
309         android:textSize="30dp"
310         android:id="@+id/bsignin"
311         android:background="@drawable/signin"/>
312
313</LinearLayout>

```

Pengguna dapat mengatur beberapa hal penting pada MINDS melalui *Activity* ini. Hal yang dapat diubah melalui *Activity* ini adalah profil, informasi rumah, *autolock*, notifikasi dan pengaturan *privilege*. Selain itu, terdapat informasi tambahan mengenai MINDS yang dapat dilihat dengan menekan menu “About” dan “Help”. Setiap menu mengarahkan pengaturan yang lebih lanjut melalui *Activity*-nya masing-masing ketika menu tersebut ditekan, kecuali untuk *autolock* dan *notification* yang statusnya dapat diubah secara langsung pada *SettingActivity.java*.

Menu pada *SettingActivity.java* tidak menggunakan *ListView* karena menu ini merupakan menu yang statis sehingga perlu menggunakan *ListView*. Menu merupakan gabungan dari *ImageView* dan *TextView* kecuali *autolock* dan *notification* yang juga menggunakan *Switch Bar* untuk mengganti statusnya. Seluruh *ImageView* maupun *TextView* diberikan *setOnClickListener* sehingga pengguna dapat menekan seluruh bagian menu untuk mengubah konfigurasi menu tersebut. Pada bagian bawah, terdapat *Button* “Back” yang dapat mengarahkan pengguna menuju *MainActivity.java* ketika ditekan.

Kode java pada halaman ini sebagai berikut:

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Context;

```

```

4 import android.content.Intent;
5 import android.content.SharedPreferences;
6 import android.graphics.Typeface;
7 import android.os.Bundle;
8 import android.support.v7.app.AppCompatActivity;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.CompoundButton;
12 import android.widget.ImageView;
13 import android.widget.Switch;
14 import android.widget.TextView;
15 import android.widget.Toast;
16
17 public class SettingActivity extends AppCompatActivity {
18
19     //DEKLARASI VARIABEL
20     SharedPreferences sharedpreferences;
21     SharedPreferences.Editor editor;
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         super.onCreate(savedInstanceState);
26         setContentView(R.layout.activity_setting);
27
28         //PENGATURAN FONT
29         Typeface font_signin =
30             Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
31         Typeface font_title = Typeface.createFromAsset(getAssets(),
32             "fonts/Quicksand-Bold.otf");
33         Typeface font_or = Typeface.createFromAsset(getAssets(),
34             "fonts/Quicksand-BoldItalic.otf");
35         Typeface font_activity =
36             Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
37
38         //TITLE BAR
39         TextView activitytitle = (TextView)
40             findViewById(R.id.activitytitle);
41         activitytitle.setTypeface(font_activity);
42
43         //TEXTVIEW TOMBOL 1
44         TextView tvprof = (TextView) findViewById(R.id.tvprofile);
45         tvprof.setTypeface(font_signin);
46
47         //TEXTVIEW TOMBOL 2
48         TextView tvhouse = (TextView) findViewById(R.id.tvhouse);
49         tvhouse.setTypeface(font_signin);
50
51         //TEXTVIEW TOMBOL 3
52         TextView tvlock = (TextView) findViewById(R.id.tvlock);
53         tvlock.setTypeface(font_signin);
54
55         //TEXTVIEW TOMBOL 4
56         TextView tvnotif = (TextView) findViewById(R.id.tvnotif);
57         tvnotif.setTypeface(font_signin);
58
59         //TEXTVIEW TOMBOL 5
60         TextView tvabout = (TextView) findViewById(R.id.tvabout);
61         tvabout.setTypeface(font_signin);

```

```

57
58     //TEXTVIEW TOMBOL 6
59     TextView tvhelp = (TextView) findViewById(R.id.tvhelp);
60     tvhelp.setTypeface(font_signin);
61
62     //TEXTVIEW TOMBOL 7
63     TextView tvprivilege = (TextView)
findViewById(R.id.tvprivilege);
64     tvprivilege.setTypeface(font_signin);
65
66     //BUTTON UNTUK BACK
67     Button bback = (Button) findViewById(R.id.bsignin);
68     bback.setTypeface(font_signin);
69     bback.setOnClickListener(new View.OnClickListener() {
70
71         @Override
72         public void onClick(View v) {
73             finish();
74         }
75     });
76
77     //SHARED PREFERENCE
78     sharedpreferences =
getSharedPreferences("Authentication", MODE_PRIVATE);
79     editor = sharedpreferences.edit();
80
81     //SWITCH ACTIVATE NOTIFICATION
82     Switch swnotif = (Switch) findViewById(R.id.swnotif);
83     if (MainActivity.CurrentStatNotif) {
84         swnotif.setChecked(true);
85     } else {
86         swnotif.setChecked(false);
87     }
88     swnotif.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
89         @Override
90         public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
91             if(isChecked) {
92                 editor.putBoolean("NotifStat", true);
93                 editor.commit();
94                 MainActivity.CurrentStatNotif = true;
95                 startService(MainActivity.mServiceIntent);
96             }
97             else {
98                 editor.putBoolean("NotifStat", false);
99                 editor.commit();
100                MainActivity.CurrentStatNotif = false;
101                stopService(MainActivity.mServiceIntent);
102            }
103        }
104    });
105
106     //SWITCH UNTUK AUTOLOCK
107     Switch swlock = (Switch) findViewById(R.id.swlock);
108     if (MainActivity.CurrentStatLock) {
109         swlock.setChecked(true);
110     } else {

```

```

111         swlock.setChecked(false);
112     }
113     swlock.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {
114         @Override
115         public void onCheckedChanged(CompoundButton buttonView,
boolean isChecked) {
116             if(isChecked) {
117                 editor.putBoolean("LockStat", true);
118                 editor.commit();
119                 MainActivity.CurrentStatLock = true;
120             }
121             else {
122                 editor.putBoolean("LockStat", false);
123                 editor.commit();
124                 MainActivity.CurrentStatLock = false;
125             }
126         }
127     });
128
129     tvhouse.setOnClickListener(new View.OnClickListener() {
130
131         @Override
132         public void onClick(View v) {
133             Intent afterloginIntent = new
Intent(SettingActivity.this,CreateNewActivity1.class);
134             SettingActivity.this.startActivity(afterloginIntent);
135         }
136     );
137
138     ImageView ivhouse = (ImageView) findViewById(R.id.ivhouse);
139     ivhouse.setOnClickListener(new View.OnClickListener() {
140
141         @Override
142         public void onClick(View v) {
143             Intent afterloginIntent = new
Intent(SettingActivity.this,CreateNewActivity1.class);
144             SettingActivity.this.startActivity(afterloginIntent);
145         }
146     );
147
148     tvprof.setOnClickListener(new View.OnClickListener() {
149
150         @Override
151         public void onClick(View v) {
152             Intent afterloginIntent = new
Intent(SettingActivity.this,UserActivity.class);
153             SettingActivity.this.startActivity(afterloginIntent);
154         }
155     );
156
157     ImageView ivprof = (ImageView) findViewById(R.id.ivprof);
158     ivprof.setOnClickListener(new View.OnClickListener() {
159
160         @Override

```

```

161         public void onClick(View v) {
162             Intent afterloginIntent = new
Intent(SettingActivity.this,UserActivity.class);
163             SettingActivity.this.startActivity(afterloginIntent);
164         }
165     });
166
167     tvhelp.setOnClickListener(new View.OnClickListener() {
168
169         @Override
170         public void onClick(View v) {
171             Intent afterloginIntent = new
Intent(SettingActivity.this,PagerActivity.class);
172             SettingActivity.this.startActivity(afterloginIntent);
173         }
174     );
175
176     ImageView ivhelp = (ImageView) findViewById(R.id.ivhelp);
177     ivhelp.setOnClickListener(new View.OnClickListener() {
178
179         @Override
180         public void onClick(View v) {
181             Intent afterloginIntent = new
Intent(SettingActivity.this,PagerActivity.class);
182             SettingActivity.this.startActivity(afterloginIntent);
183         }
184     );
185
186     tvabout.setOnClickListener(new View.OnClickListener() {
187
188         @Override
189         public void onClick(View v) {
190             Intent afterloginIntent = new
Intent(SettingActivity.this,AboutActivity.class);
191             SettingActivity.this.startActivity(afterloginIntent);
192         }
193     );
194
195     ImageView ivabout = (ImageView) findViewById(R.id.ivabout);
196     ivabout.setOnClickListener(new View.OnClickListener() {
197
198         @Override
199         public void onClick(View v) {
200             Intent afterloginIntent = new
Intent(SettingActivity.this,AboutActivity.class);
201             SettingActivity.this.startActivity(afterloginIntent);
202         }
203     );
204
205     tvprivilege.setOnClickListener(new View.OnClickListener() {
206
207         @Override
208         public void onClick(View v) {

```

```

209             if (MainActivity.privilege.equals("ADMIN")) {
210                 Intent afterloginIntent = new
Intent(SettingActivity.this, PrivilegeActivity.class);
211
SettingActivity.this.startActivity(afterloginIntent);
212         } else {
213             Toast.makeText(getApplicationContext(),"You are
not ADMIN.\nOnly ADMIN can edit this.",
214                                         Toast.LENGTH_SHORT).show();
215         }
216     }
217 );
218
219     ImageView ivprivilege = (ImageView)
findViewById(R.id.ivabout);
220     ivprivilege.setOnClickListener(new View.OnClickListener() {
221
222         @Override
223         public void onClick(View v) {
224             if (MainActivity.privilege.equals("ADMIN")) {
225                 Intent afterloginIntent = new
Intent(SettingActivity.this, PrivilegeActivity.class);
226
SettingActivity.this.startActivity(afterloginIntent);
227             } else {
228                 Toast.makeText(getApplicationContext(),"You are
not ADMIN.\nOnly ADMIN can edit this.",
229                                         Toast.LENGTH_SHORT).show();
230             }
231         }
232     );
233 }
234 }
```

Kode yang cukup penting pada *file java* halaman ini adalah:

- Baris 28 sampai 75 merupakan inisiasi dari setiap komponen halaman Setting dan pengaturan *font* yang digunakan pada halaman ini.
- Baris 81 sampai 104 merupakan pengaturan status dari fitur notifikasi. Pada baris 83 sampai 87, dilakukan pengaturan status awal dari *Switch* berdasarkan pengaturan terakhir yang dilakukan oleh pengguna. Jika pengguna mengaktifkan fitur notifikasi pada *Switch*, bagian notifikasi “NotifStat” pada *shared preference* diubah menjadi bernilai *true* dan “CurrentStatNotif” juga bernilai *true*. Setelah mengubah kedua hal tersebut, *Service* aplikasi MINDS dinyalakan. Dengan aktifnya *service*, aplikasi dapat berjalan walaupun aplikasi sudah ditutup sehingga notifikasi dapat terus diterima oleh pengguna. Ketika pengguna mematikan fitur notifikasi, “NotifStat” dan “CurrentStatNotif” akan bernilai *false* dan *Service* akan dimatikan. Hal yang serupa dilakukan untuk Auto-Lock pada baris 107 sampai 127.

- Baris 129 sampai 232 merupakan pengaturan perpindahan halaman ketika ada menu yang ditekan. Perpindahan halaman dilakukan dengan menggunakan *Intent*.

Fitur *auto-lock* digunakan untuk mengaktifkan *scenario lock* secara otomatis berdasarkan lokasi *mobile device*. Seperti yang sebelumnya sudah dijelaskan, *scenario lock* dapat digunakan untuk mengunci seluruh akses utama rumah. Sehingga pada saat tidak ada member yang berada pada rentang jarak tertentu dari pusat rumah, sebagai contoh, ketika radius yang didaftarkan adalah 100 meter, maka ketika tidak ada member pada jarak 100 meter dari pusat rumah, MINDS akan mengunci seluruh akses utama rumah tersebut. Sistem *auto-lock* mengantisipasi keadaan ketika pengguna lupa mengunci rumah ketika berpergian. *Switch Bar* pada *notification* menandakan status notifikasi pengguna. Ketika *Switch Bar* berada pada posisi ON, maka pengguna memperbolehkan MINDS untuk mengirimkan notifikasi kepada *smartphone* pengguna, begitu juga sebaliknya. *Notification* selalu dikirimkan kepada *smartphone* setiap saat walaupun MINDS dalam keadaan tertutup sekalipun karena *notification* bekerja pada *service*. *Service* merupakan suatu proses yang tetap berjalan walaupun aplikasi tersebut sedang tidak dijalankan maupun sudah dihapus dari *task manager*. *Service* hanya akan berhenti beroperasi ketika *service* tersebut dimatikan atau aplikasi di-uninstall. Pengubahan status *notification* akan mengubah status aktif atau tidaknya *service* pada MINDS sehingga notifikasi tidak akan diterima ketika *service* dimatikan pada saat status *notification* OFF.

6.18 Halaman User (UserActivity.java)



Gambar 53 Halaman User (UserActivity.java)

Kode XML pada halaman ini sebagai berikut:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:id="@+id/activity_user"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:paddingBottom="@dimen/activity_vertical_margin"
9      android:paddingLeft="0dp"
10     android:paddingRight="0dp"
11     android:paddingTop="0dp"
12     tools:context="com.example.billyausten.newfinal.UserActivity"
13     android:orientation="vertical"
14     android:descendantFocusability="beforeDescendants"
15     android:focusableInTouchMode="true">
16
17     <LinearLayout
18         android:layout_width="match_parent"
19         android:layout_height="33dp"
20         android:layout_gravity="center"
21         android:orientation="vertical"
22         android:background="@drawable/backblack2"
23         android:id="@+id/backtitle">
24
25         <LinearLayout
26             android:layout_width="wrap_content"
27             android:layout_height="wrap_content"
28             android:orientation="horizontal"
29             android:layout_gravity="center">
30
31             <TextView
32                 android:layout_width="290dp"
33                 android:layout_height="wrap_content"
34                 android:text="Profile"
35                 android:id="@+id/activitytitle"
36                 android:textAlignment="center"
37                 android:layout_gravity="center_vertical"
38                 android:textSize="18dp"
39                 android:textColor="#AAAAAA"
40                 android:layout_marginTop="3dp"/>
41
42         </LinearLayout>
43
44     </LinearLayout>
45
46     <ImageView
47         android:layout_width="150dp"
48         android:layout_height="150dp"
49         android:layout_gravity="center"
50         android:layout_marginTop="30dp"
51         android:src="@drawable/billy"
52         android:id="@+id/ivProfilePict"/>
53
54     <TextView
55         android:layout_width="wrap_content"
56         android:layout_height="wrap_content"
```

```

56        android:layout_gravity="center"
57        android:id="@+id/tvemail"
58        android:layout_marginTop="30dp"
59        android:textSize="18dp"
60        android:text="billy.austen.m@gmail.com"
61        android:textColor="#404040"/>
62
63    <RelativeLayout
64        android:layout_width="match_parent"
65        android:layout_height="wrap_content"
66        android:layout_marginTop="0dp">
67
68        <TextView
69            android:layout_width="150dp"
70            android:layout_height="wrap_content"
71            android:gravity="center_horizontal"
72            android:text="Username"
73            android:textAlignment="center"
74            android:layout_centerHorizontal="true"
75            android:textSize="18dp"
76            android:layout_marginTop="10dp"
77            android:id="@+id/tvusername"
78            android:textColor="#404040"
79            android:maxLines="1"/>
80
81        <EditText
82            android:layout_width="150dp"
83            android:layout_height="wrap_content"
84            android:textAlignment="center"
85            android:layout_centerHorizontal="true"
86            android:hint="Username"
87            android:textSize="18dp"
88            android:visibility="invisible"
89            android:id="@+id/etusername"
90            android:inputType="textPersonName"/>
91
92        <Button
93            android:layout_width="40dp"
94            android:layout_height="18dp"
95            android:text="Edit"
96            android:background="@drawable/kotak"
97            android:layout_toRightOf="@+id/etusername"
98            android:layout_marginTop="13dp"
99            android:layout_marginLeft="10dp"
100           android:id="@+id/bedituser"
101           android:clickable="true"/>
102
103        <Button
104            android:layout_width="40dp"
105            android:layout_height="18dp"
106            android:text="Save"
107            android:background="@drawable/kotak"
108            android:layout_toRightOf="@+id/etusername"
109            android:layout_marginTop="13dp"
110            android:layout_marginLeft="10dp"
111            android:id="@+id/bsaveuser"
112            android:visibility="invisible"
113            android:clickable="false"/>

```

```

114
115      <TextView
116          android:layout_width="150dp"
117          android:layout_height="wrap_content"
118          android:gravity="center_horizontal"
119          android:text="Password"
120          android:textAlignment="center"
121          android:layout_centerHorizontal="true"
122          android:textSize="18dp"
123          android:layout_marginTop="40dp"
124          android:id="@+id/tvpassword"
125          android:password="true"
126          android:textColor="#404040"/>
127
128      <EditText
129          android:layout_width="150dp"
130          android:layout_height="wrap_content"
131          android:textAlignment="center"
132          android:layout_centerHorizontal="true"
133          android:hint="Password"
134          android:textSize="18dp"
135          android:visibility="invisible"
136          android:layout_marginTop="30dp"
137          android:id="@+id/etPassword"
138          android:inputType="textPassword"/>
139
140      <EditText
141          android:layout_width="150dp"
142          android:layout_height="wrap_content"
143          android:textAlignment="center"
144          android:layout_centerHorizontal="true"
145          android:hint="Password"
146          android:textSize="18dp"
147          android:visibility="invisible"
148          android:layout_marginTop="60dp"
149          android:id="@+id/etPasswordc"
150          android:inputType="textPassword"/>
151
152      <Button
153          android:layout_width="40dp"
154          android:layout_height="18dp"
155          android:text="Edit"
156          android:background="@drawable/kotak"
157          android:layout_toRightOf="@+id/etusername"
158          android:layout_marginTop="43dp"
159          android:layout_marginLeft="10dp"
160          android:id="@+id/beditpass"/>
161
162      <Button
163          android:layout_width="40dp"
164          android:layout_height="18dp"
165          android:text="Save"
166          android:background="@drawable/kotak"
167          android:layout_toRightOf="@+id/etusername"
168          android:layout_marginTop="43dp"
169          android:layout_marginLeft="10dp"
170          android:id="@+id/bsavepass"
171          android:visibility="invisible"

```

```

172        android:clickable="false"/>
173    
```

`</RelativeLayout>`

```

175
176    <Button
177        android:layout_width="wrap_content"
178        android:layout_height="wrap_content"
179        android:layout_gravity="center"
180        android:text="Save"
181        android:layout_marginTop="50dp"
182        android:textSize="30dp"
183        android:id="@+id/bsave"
184        android:background="@drawable/signin"/>
185
186</LinearLayout>
```

Pada *Activity* ini, pengguna dapat mengganti beberapa informasi pada profil seperti *nama* dan *password* akun MINDS. *Nama* dan *password* yang sukses diganti akan langsung ditampilkan pada *Navigation Drawer* sebagai *preview* dari profil pengguna. Pengguna dapat mengganti *profile picture* menggunakan gambar pada galeri ataupun mengambil gambar menggunakan kamera dengan cara menekan *profile picture*, dan segera tampil *popup menu* untuk memilih metode pengambilan gambar. Pengguna juga dapat mengubah *nama* dan *password* pengguna pada aplikasi MINDS dengan menekan *Button* “EDIT” pada sebelah kanan *TextView* *password* maupun *nama*.

Pada tampilan *UserActivity.java* digunakan *LinearLayout* dan *RelativeLayout*. *LinearLayout* digunakan untuk mempermudah penyusunan setiap komponen *Activity*, sedangkan *RelativeLayout* digunakan untuk menyusun *Button*, *EditText* maupun *TextView* yang saling tumpang tindih seperti *Button* “SAVE” dan *Button* “EDIT”. Kedua *Button* tersebut saling tumpang tindih dan berukuran sama, tetapi *visibility* *Button* “SAVE” bernilai *invisible*, sehingga *Button* tersebut tidak akan terlihat bersamaan dengan *Button* “EDIT”. Ketika *Button* “EDIT” ditekan, dilakukan pengubahan nilai *visibility* dari *Button* “SAVE” dan *Button* “EDIT”. *Button* “SAVE” diubah menjadi *visible* dan *Button* “EDIT” diubah menjadi *invisible*, sehingga yang terlihat hanyalah *Button* “SAVE”. Hal yang sama juga dilakukan untuk *TextView* dan *EditText* pada *user* maupun *password*.

Kode java pada halaman ini sebagai berikut:

```

1 package com.example.billyausten.newfinal;
2
3
4 import android.content.ActivityNotFoundException;
5 import android.content.Context;
```

```

6 import android.content.Intent;
7 import android.content.SharedPreferences;
8 import android.content.pm.PackageManager;
9 import android.graphics.Bitmap;
10 import android.graphics.Typeface;
11 import android.graphics.drawable.BitmapDrawable;
12 import android.net.Uri;
13 import android.os.Environment;
14 import android.os.StrictMode;
15 import android.support.v4.app.ActivityCompat;
16 import android.support.v7.app.AppCompatActivity;
17 import android.os.Bundle;
18 import android.support.v7.widget.PopupMenu;
19 import android.util.Base64;
20 import android.util.DisplayMetrics;
21 import android.view.MenuItem;
22 import android.view.View;
23 import android.view.inputmethod.InputMethodManager;
24 import android.widget.Button;
25 import android.widget.EditText;
26 import android.widget.ImageView;
27 import android.widget.TextView;
28 import android.widget.Toast;
29 import android.graphics.BitmapFactory;
30 import com.rabbitmq.client.AMQP;
31 import org.json.simple.JSONObject;
32 import java.io.ByteArrayOutputStream;
33 import java.io.File;
34 import java.io.StringWriter;
35
36 public class UserActivity extends AppCompatActivity {
37
38     //DEKLARASI VARIABEL
39     public static final String Usernamez = "usname";
40     public static final String Passwordz = "password";
41     public static final String ProfPicz = "profpic";
42     SharedPreferences sharedpreferences;
43     String susername;
44     String spassword;
45     String sprofpic;
46     SharedPreferences.Editor editor;
47     ImageView ivprofpic;
48     File file;
49     Uri uri;
50     Intent CamIntent, GalIntent, CropIntent ;
51     public static final int RequestPermissionCode = 1 ;
52     DisplayMetrics displayMetrics ;
53     int width, height;
54
55     @Override
56     protected void onCreate(Bundle savedInstanceState) {
57         super.onCreate(savedInstanceState);
58         setContentView(R.layout.activity_user);
59
60         //PENGATURAN FONT
61         Typeface font_activity =
Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");

```

```

62         Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");
63         Typeface font_title = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Bold.otf");
64         Typeface font_or = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-BoldItalic.otf");
65
66         //TITLE BAR
67         TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
68         activitytitle.setTypeface(font_activity);
69
70         //TEXTVIEW EMAIL USER
71         TextView tvemail = (TextView) findViewById(R.id.tvemail);
72         tvemail.setTypeface(font_signin);
73         tvemail.setText(MainActivity.email);
74
75         //TEXTVIEW USERNAME
76         final TextView tvusername = (TextView)
findViewById(R.id.tvusername);
77         tvusername.setTypeface(font_signin);
78
79         //TEXTVIEW PASSWORD
80         final TextView tvpassword = (TextView)
findViewById(R.id.tvpassword);
81         tvpassword.setTypeface(font_signin);
82
83         //EDITTEXT UNTUK EDIT USERNAME
84         final EditText etusername = (EditText)
findViewById(R.id.etusername);
85         etusername.setTypeface(font_signin);
86
87         //EDITTEXT UNTUK EDIT PASSWORD
88         final EditText etpassword = (EditText)
findViewById(R.id.etPassword);
89         etpassword.setTypeface(font_signin);
90
91         //EDITTEXT UNTUK MEMASUKAN KONFIRMASI PASSWORD BARU
92         final EditText etpasswordc = (EditText)
findViewById(R.id.etPasswordc);
93         etpasswordc.setTypeface(font_signin);
94
95         //BUTTON UNTUK EDIT USERNAME
96         final Button bedituser = (Button)
findViewById(R.id.bedituser);
97         bedituser.setTypeface(font_signin);
98
99         //BUTTON UNTUK SAVE USERNAME BARU
100        final Button bsaveuser = (Button)
findViewById(R.id.bsaveuser);
101        bsaveuser.setTypeface(font_signin);
102
103        //BUTTON UNTUK EDIT PASSWORD
104        final Button beditpass = (Button)
findViewById(R.id.beditpass);
105        beditpass.setTypeface(font_signin);
106
107        //BUTTON UNTUK SAVE PASSWORD BARU

```

```

108     final Button bsavepass = (Button)
findViewById(R.id.bsavepass);
109     bsavepass.setTypeface(font_signin);
110
111     //BUTTON UNTUK SAVE KESELURUHAN PENGUBAHAN PROFIL
112     Button bsave = (Button) findViewById(R.id.bsave);
113     bsave.setTypeface(font_signin);
114
115     //PENGISIAN MASING-MASING WIDGET
116     ivprofpic = (ImageView) findViewById(R.id.ivProfilePict);
117
118     sharedpreferences =
getSharedPreferences("Authentication", MODE_PRIVATE);
119     editor = sharedpreferences.edit();
120
121     susername = sharedpreferences.getString(Usernamez,
"Username");
122     spassword = sharedpreferences.getString>Passwordz,
>Password");
123     tvusername.setText(susername);
124     tvpassword.setText(spassword);
125
126     etusername.setClickable(false);
127     etpassword.setClickable(false);
128     etpassworddc.setClickable(false);
129
130     bedituser.setOnClickListener(new View.OnClickListener() {
131
132         @Override
133         public void onClick(View v) {
134
135             bedituser.setVisibility(View.INVISIBLE);
136             bedituser.setClickable(false);
137             bsaveuser.setVisibility(View.VISIBLE);
138             bsaveuser.setClickable(true);
139             tvusername.setVisibility(View.INVISIBLE);
140             etusername.setVisibility(View.VISIBLE);
141             etusername.setClickable(true);
142             susername = sharedpreferences.getString(Usernamez,
"Username");
143             etusername.setText(susername);
144         }
145     );
146
147     bsaveuser.setOnClickListener(new View.OnClickListener() {
148
149         @Override
150         public void onClick(View v) {
151
152             bedituser.setVisibility(View.VISIBLE);
153             bedituser.setClickable(true);
154             bsaveuser.setVisibility(View.INVISIBLE);
155             bsaveuser.setClickable(false);
156             tvusername.setVisibility(View.VISIBLE);
157             etusername.setVisibility(View.INVISIBLE);
158             etusername.setClickable(false);
159             InputMethodManager imm =
(InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);

```

```

160         imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
161         susername = etusername.getText().toString();
162         tvusername.setText(susername);
163     }
164 }
165
166 beditpass.setOnClickListener(new View.OnClickListener() {
167
168     @Override
169     public void onClick(View v) {
170
171         beditpass.setVisibility(View.INVISIBLE);
172         beditpass.setClickable(false);
173         bsavepass.setVisibility(View.VISIBLE);
174         bsavepass.setClickable(true);
175         tvpassword.setVisibility(View.INVISIBLE);
176         etpassword.setVisibility(View.VISIBLE);
177         etpassword.setClickable(true);
178         etpassworddc.setVisibility(View.VISIBLE);
179         etpassworddc.setClickable(true);
180         etpassword.setText("");
181         etpassworddc.setText("");
182     }
183 });
184
185 bsavepass.setOnClickListener(new View.OnClickListener() {
186
187     @Override
188     public void onClick(View v) {
189
190         if
191             (etpassword.getText().toString().equals(etpassworddc.getText().toString()
192             )) {
193                 beditpass.setVisibility(View.VISIBLE);
194                 beditpass.setClickable(true);
195                 bsavepass.setVisibility(View.INVISIBLE);
196                 bsavepass.setClickable(false);
197                 tvpassword.setVisibility(View.VISIBLE);
198                 etpassword.setVisibility(View.INVISIBLE);
199                 etpassword.setClickable(false);
200                 etpassworddc.setVisibility(View.INVISIBLE);
201                 etpassworddc.setClickable(false);
202                 InputMethodManager imm =
203                     (InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE);
204                 imm.hideSoftInputFromWindow(v.getWindowToken(), 0);
205                 spassword = etpassword.getText().toString();
206                 tvpassword.setText(spassword);
207             } else {
208                 Toast.makeText(UserActivity.this,"Your Password
209                 and Password Confirmation is different",Toast.LENGTH_SHORT).show();
210             }
211
212         }
213     }
214
215 //KETIKA FOTO PROFIL DIKLICK, MAKAN AKAN MUNCUL POP UP MENU,
216 //PILIH FOTO YANG SUDAH ADA DI GALLERY, ATAU AMBIL FOTO BARU DARI KAMERA

```

```
122     ivprofpic.setOnClickListener(new View.OnClickListener() {
123
124         @Override
125         public void onClick(View v) {
126             PopupMenu popup = new PopupMenu(UserActivity.this,
127                                         v);
128             //PopupMenu popup = new
129             //PopupMenu(getApplicationContext(), view);
130             //Inflating the Popup using xml file
131             popup.getMenuInflater().inflate(R.menu.menu_prof,
132                                         popup.getMenu());
133             popup.show();
134             //registering popup with OnMenuItemClickListener
135             popup.setOnMenuItemClickListener(new
136             PopupMenu.OnMenuItemClickListener() {
137                 public boolean onMenuItemClick(MenuItem item) {
138                     switch (item.getItemId()) {
139                         case R.id.action_camera:
140                             ClickImageFromCamera();
141                             return true;
142                         case R.id.action_gallery:
143                             GetImageFromGallery();
144                             return true;
145                         }
146                     return true;
147                 }
148             });
149         });
150     });
151
152     sprofpic = sharedpreferences.getString(ProfPicz, null);
153     if (sprofpic.equals("none"))
154         ivprofpic.setImageResource(R.drawable.profdefault);
155     else
156         ivprofpic.setImageBitmap(decodeBase64(sprofpic));
157
158     //KETIKA BUTTON SAVE DITEKAN:
159     bsave.setOnClickListener(new View.OnClickListener() {
160
161         @Override
162         public void onClick(View v) {
163             //mengirim pesan editprofile
164             try {
165                 Bitmap bmprofpic = ((BitmapDrawable)
166 ivprofpic.getDrawable()).getBitmap();
167                 editor.putString(ProfPicz,
168 encodeTobase64(bmprofpic));
169                 editor.putString(Usernamez, susername);
170                 editor.putString>Passwordz, spassword);
171                 editor.commit();
172                 JSONObject obj = new JSONObject();
173                 obj.put("head", "MU");
174                 obj.put("type", "editprof");
175                 obj.put("homeid", MainActivity.homeid);
176                 obj.put("email", MainActivity.email);
177                 obj.put("name", susername);
178                 obj.put("password", spassword);
179                 obj.put("picture", encodeTobase64(bmprofpic));
180             }
181         }
182     });
183 }
```

```

264             StringWriter toSent = new StringWriter();
265             obj.writeJSONString(toSent);
266             String message = toSent.toString();
267             AMQP.BasicProperties props = new
AMQP.BasicProperties
268                     .Builder()
269                     .correlationId(MainActivity.phoneid)
270                     .build();
271             MainActivity.channel.basicPublish("", "AESqueue", props, MainActivity.AESencrypt(message).getBytes("UTF-8"));
272
273         } catch (Exception e) {
274
275     }
276
277     //PINDAH ACTIVITY KE MAINACTIVITY
278     Intent mainIntent = new
Intent(UserActivity.this,MainActivity.class);
279     UserActivity.this.startActivity(mainIntent);
280 }
281 })
282
283     StrictMode.VmPolicy.Builder builder = new
StrictMode.VmPolicy.Builder();
284     StrictMode.setVmPolicy(builder.build());
285 }
286
287 //FUNGSI UNTUK MENGAMBIL FOTO DARI KAMERA
288 public void ClickImageFromCamera() {
289     CamIntent = new
Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE);
290     file = new File(Environment.getExternalStorageDirectory(),
291                     "file" + String.valueOf(System.currentTimeMillis())
+ ".jpg");
292     uri = Uri.fromFile(file);
293
CamIntent.putExtra(android.provider.MediaStore.EXTRA_OUTPUT, uri);
294     CamIntent.putExtra("return-data", true);
295     startActivityForResult(CamIntent, 0);
296 }
297
298 //FUNGSI UNTUK MENGAMBIL FOTO DARI GALERI
299 public void GetImageFromGallery(){
300     GalIntent = new Intent(Intent.ACTION_PICK,
301 android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
302     startActivityForResult(Intent.createChooser(GalIntent,
"Select Image From Gallery"), 2);
303 }
304
305 @Override
306 protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
307     if (requestCode == 0 && resultCode == RESULT_OK) {
308         ImageCropFunction();
309     }
310     else if (requestCode == 2) {

```

```

311         if (data != null) {
312             uri = data.getData();
313             ImageCropFunction();
314         }
315     }
316     else if (requestCode == 1) {
317         if (data != null) {
318             Bundle bundle = data.getExtras();
319             Bitmap bitmap = bundle.getParcelable("data");
320             ivprofpic.setImageBitmap(bitmap);
321         }
322     }
323 }
324
325 //FUNGSI UNTUK IMAGE CROP
326 public void ImageCropFunction() {
327
328     // Image Crop Code
329     try {
330         CropIntent = new
Intent("com.android.camera.action.CROP");
331         CropIntent.setDataAndType(uri, "image/*");
332         CropIntent.putExtra("crop", "true");
333         CropIntent.putExtra("outputX", 180);
334         CropIntent.putExtra("outputY", 180);
335         CropIntent.putExtra("aspectX", 4);
336         CropIntent.putExtra("aspectY", 4);
337         CropIntent.putExtra("scaleUpIfNeeded", true);
338         CropIntent.putExtra("return-data", true);
339         startActivityForResult(CropIntent, 1);
340     } catch (ActivityNotFoundException e) {
341
342     }
343 }
344
345
346     public void EnableRuntimePermission() {
347
348         if
(ActivityCompat.shouldShowRequestPermissionRationale(UserActivity.this
,
349
            android.Manifest.permission.CAMERA)) {
350
351         } else {
352             ActivityCompat.requestPermissions(UserActivity.this,new
String[] {
353
                android.Manifest.permission.CAMERA},
RequestPermissionCode);
354         }
355     }
356
357     @Override
358     public void onRequestPermissionsResult(int RC, String per[], int[] PResult) {
359
360         switch (RC) {
361             case RequestPermissionCode:

```

```

362             if (PResult.length > 0 && PResult[0] ==
PackageManager.PERMISSION_GRANTED) {
363                 } else {
364                     }
365                     }
366                     break;
367                 }
368             }
369         }
370     }
371     //FUNGSI UNTUK ENCODE GAMBAR KE BASE64 STRING
372     public static String encodeToBase64(Bitmap image) {
373         Bitmap immage = image;
374         ByteArrayOutputStream baos = new ByteArrayOutputStream();
375         immage.compress(Bitmap.CompressFormat.PNG, 100, baos);
376         byte[] b = baos.toByteArray();
377         String imageEncoded = Base64.encodeToString(b,
Base64.DEFAULT);
378         return imageEncoded;
379     }
380
381     //FUNGSI UNTUK DECODE BASE64 STRING KE GAMBAR
382     public static Bitmap decodeBase64(String input) {
383         byte[] decodedByte = Base64.decode(input, 0);
384         return BitmapFactory
385             .decodeByteArray(decodedByte, 0,
decodedByte.length);
386     }
387 }

```

Kode yang cukup penting pada java diatas adalah:

- Baris 60 sampai 116 merupakan inisiasi dari berbagai komponen yang digunakan pada halaman ini. Selain inisiasi komponen, pada baris ini juga terdapat pengaturan *font* yang digunakan.
- Pada baris 118 sampai 122 dilakukan pembacaan *shared preference* untuk mendapatkan informasi terakhir mengenai pengguna.
- Baris 130 sampai baris 209 merupakan pengaturan tampilan dan penerimaan *input* informasi *username* dan *password* yang dilakukan oleh pengguna. *Input* yang dimasukan oleh pengguna akan ditampilkan pada *TextView* yang bersangkutan.
- Baris 212 sampai 236 merupakan pemunculan *popup Menu* ketika *profile picture* ditekan. Mengenai proses penggantian *profile picture* akan dijelaskan dibagian bawah pada bagian ini.
- Baris 238 sampai 242 menampilkan *profile picture* sesuai dengan *profile picture* yang terakhir disimpan oleh pengguna.

- Baris 245 sampai 281 merupakan pengiriman perintah JSON oleh MINDS kepada server untuk proses *edit profile*. Tabel perintah JSON dapat dilihat pada bagian bawah mengenai halaman ini.
- Baris 287 sampai 303 merupakan fungsi untuk mendapatkan *profile picture* baik dengan mengambil gambar menggunakan kamera ataupun mengambil gambar pada galeri *smartphone* pengguna.
- Baris 305 sampai 323 merupakan sebuah fungsi untuk memanggil apakah *profile picture* harus di-*crop* ataupun ditampilkan pada halaman *profile*.
- Baris 326 sampai 344 merupakan fungsi untuk melakukan *cropping* terhadap *profile picture*.
- Baris 371 sampai 386 merupakan fungsi untuk melakukan *encode* dan *decode* terhadap *profile picture* dalam bentuk *bitmap* maupun *base64 string*.

Untuk mengganti *profile picture*, diperlukan *setOnItemClickListener* pada *profile picture* sehingga ketika *profile picture* ditekan, muncul *popup menu* yang sebelumnya sudah dijelaskan. Untuk menampilkan *popup menu*, digunakan kode *getMenuInflater* yang akan memanggil menu yang sudah didesain sebelumnya pada *menu_prof*. Untuk dapat memilih setiap menu pada *popup menu*, digunakan kembali *setOnItemClickListener* pada *popup menu*. Ketika pengguna memilih untuk mengambil gambar menggunakan kamera, MINDS segera memanggil *Intent(android.provider.MediaStore.ACTION_IMAGE_CAPTURE)* sehingga aplikasi dapat digunakan untuk mengambil gambar menggunakan kamera pada *smartphone*. Setelah mengambil gambar menggunakan kamera, MINDS memanggil *Intent("com.android.camera.action.CROP")* yang berfungsi untuk menampilkan tampilan untuk melakukan *crop* terhadap gambar tersebut. Karena memiliki nilai *aspectX* dan *aspectY* yang sama, gambar hanya dapat di-*crop* dengan ukuran persegi sehingga sesuai dengan *ImageView* yang tersedia yang juga berukuran persegi. Karena kode *scaleUpIsNeeded* bernilai *true*, pengguna dapat mengubah ukuran daerah *crop* sesuai dengan keinginan pengguna dan hasil *output*-nya akan disesuaikan dengan ukuran *ImageView* yang tersedia. Setelah melakukan *cropping*, hasil *crop* akan langsung tampil pada *profile picture* pengguna. Hal serupa juga dijalankan ketika pengguna memilih untuk mengambil gambar pada galeri. Ketika menu mengambil gambar pada galeri ditekan, MINDS segera memanggil *Intent(Intent.ACTION_PICK, android.provider.MediaStore.Images.Media.EXTERNAL_CONTENT_URI)*

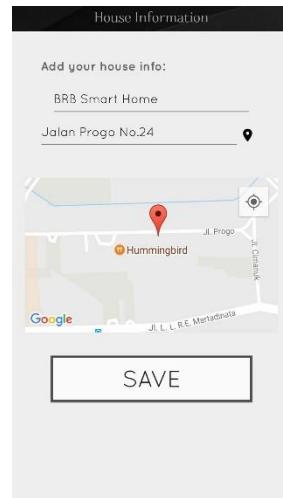
NT_URI) sehingga tampilan segera berubah menjadi tampilan galeri. Ketika pengguna sudah memilih gambar untuk *profile picture* pada galeri, tampilan berubah menjadi tampilan untuk melakukan *crop* terhadap gambar seperti pada menu sebelumnya dan gambar hasil *crop* ditampilkan pada *ImageView profile picture*. Untuk melakukan penggantian terhadap *profile picture*, dibutuhkan berbagai *permission* untuk mengakses kamera ataupun galeri, seperti `<uses-feature android:name="android.hardware.camera" />`, `<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />`, `<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />`, dan `<uses-permission android:name="android.permission.CAMERA" />`.

Pada halaman ini, digunakan proses *encoding* dengan menggunakan fungsi `encodeToBase64` terhadap *profile picture* agar data mengenai *profile picture* dapat dipindah-pindahkan kepada server maupun *Activity* lain. *Encoding* dilakukan dengan mengubah gambar berupa *bitmap* menjadi *base64 string*. Ketika gambar sudah berbentuk *string*, gambar dengan mudah disimpan pada berbagai memori seperti *shared preference* dan juga server. Untuk mengembalikan gambar, dilakukan *decode* dari *base64 string* menjadi *bitmap* dengan menggunakan fungsi `decodeBase64`.

Tabel 35 JSON untuk edit profile

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	editprof	<ul style="list-style-type: none"> • Email user • Nama baru • Foto baru • Password baru 	Server	AES	Tidak ada jawaban

6.19 Halaman Informasi Rumah (*CreateNewActivity1.java*)



Gambar 54 Halaman Create New (CreateNewActivity1.java)

Kode halaman XML sebagai berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_create_new1"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="0dp"
10    android:paddingRight="0dp"
11    android:paddingTop="0dp"
12    android:orientation="vertical"
13
14    tools:context="com.example.billyausten.newfinal.CreateNewActivity1"
15    android:weightSum="1"
16    android:descendantFocusability="beforeDescendants"
17    android:focusableInTouchMode="true">
18
19    <LinearLayout
20        android:layout_width="match_parent"
21        android:layout_height="33dp"
22        android:layout_gravity="center"
23        android:orientation="vertical"
24        android:background="@drawable/backblack2"
25        android:id="@+id/backtitle">
26
27        <LinearLayout
28            android:layout_width="wrap_content"
29            android:layout_height="wrap_content"
30            android:orientation="horizontal"
31            android:layout_gravity="center">
32
33            <TextView
34                android:layout_width="290dp"
35                android:layout_height="wrap_content"
36                android:text="House Information"
37                android:id="@+id/activitytitle"
```

```

36         android:textAlignment="center"
37         android:layout_gravity="center_vertical"
38         android:textSize="18dp"
39         android:textColor="#AAAAAA"
40         android:layout_marginTop="3dp"/>/>
41
42     
43
44 
45
46 <TextView
47     android:layout_width="wrap_content"
48     android:layout_height="wrap_content"
49     android:text="Add your house info:"
50     android:layout_marginTop="35dp"
51     android:layout_marginLeft="37dp"
52     android:id="@+id/tvAdd"
53     android:textSize="16dp"/>
54
55 <EditText
56     android:layout_width="260dp"
57     android:layout_height="wrap_content"
58     android:layout_gravity="center"
59     android:hint="House Name"
60     android:textSize="16dp"
61     android:inputType="textPersonName"
62     android:id="@+id/etHouseName"
63     android:layout_marginTop="20dp"/>
64
65 <LinearLayout
66     android:layout_width="368dp"
67     android:layout_height="50dp">
68
69     <EditText
70         android:layout_width="260dp"
71         android:layout_height="wrap_content"
72         android:layout_gravity="center"
73         android:hint="Address"
74         android:textSize="16dp"
75         android:layout_marginLeft="34dp"
76         android:id="@+id/etAddress"
77         android:inputType="textEmailAddress"/>
78
79     <ImageView
80         android:layout_width="20dp"
81         android:layout_height="20dp"
82         android:layout_gravity="center"
83         android:src="@drawable/ic_location_on_black_24dp"
84         android:id="@+id/btn_show"/>
85
86 
87
88 <fragment
89     android:id="@+id/map"
90
91     android:name="com.google.android.gms.maps.SupportMapFragment"
92     android:layout_width="326dp"
93     android:layout_height="200dp"/>

```

```

93     android:layout_marginTop="30dp"
94     android:layout_gravity="center"/>
95
96     <Button
97         android:layout_width="wrap_content"
98         android:layout_height="wrap_content"
99         android:layout_gravity="center"
100        android:text="Save"
101        android:layout_marginTop="30dp"
102        android:textSize="30dp"
103        android:id="@+id/bsave"
104        android:background="@drawable/signin"/>
105
106</LinearLayout>

```

Halaman ini digunakan untuk membantu pengguna dalam mendaftarkan rumah mereka. Pengguna dapat memasukan nama dan alamat rumah mereka pada *EditText* yang tersedia. Nama rumah dapat diisi sesuai dengan keinginan pengguna untuk keperluan identifikasi alamat. Pada halaman ini, terdapat fitur sinkronisasi antara *EditText* untuk alamat dan fitur Google Maps sehingga ketika *EditText* diisi dengan alamat, Google Maps dapat mencari posisi alamat tersebut dan pengguna hanya perlu melakukan verifikasi terhadap alamat yang ditunjuk oleh *pointer* dengan cara menggeser letak *selected point* pada Google Maps. Setelah dilakukan verifikasi, *EditText* akan berisi alamat baru yang telah diverifikasi.

Pada halaman ini digunakan *fragment* untuk memuat Google Maps pada bagian tengah dari *Activity*. Untuk melakukan akses pada Google Maps, perlu dilakukan beberapa perubahan pada bagian “AndroidManifest” agar aplikasi mendapatkan izin untuk menggunakan GPS pada *smartphone*. Selain itu, aplikasi MINDS perlu didaftarkan pada Google API agar MINDS dapat mengakses Google Maps. Paket Google Maps yang digunakan adalah paket *Basic* dengan pertimbangan bahwa MINDS masih dalam tahap pengembangan sehingga pengguna yang mengakses Google Maps belum terlalu banyak, selain itu MINDS juga tidak memerlukan fitur yang lebih canggih karena hanya perlu mengakses sebuah lokasi yang ditunjuk oleh *marker*. Pada bagian bawah dari *CreateNewActivity1*, terdapat satu tombol *Button* untuk melanjutkan proses *Create New* pada sistem Rumah Cerdas.

Kode halaman java sebagai berikut:

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;

```

```

5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.view.WindowManager;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ImageView;
12 import android.widget.TextView;
13 import android.Manifest;
14 import android.content.Context;
15 import android.content.pm.PackageManager;
16 import android.location.Address;
17 import android.location.Geocoder;
18 import android.location.Location;
19 import android.os.AsyncTask;
20 import android.os.Build;
21 import android.support.v4.app.ActivityCompat;
22 import android.support.v4.app.FragmentActivity;
23 import android.os.Bundle;
24 import android.support.v4.content.ContextCompat;
25 import android.view.Menu;
26 import android.widget.TextView;
27 import android.widget.Toast;
28 import com.google.android.gms.common.ConnectionResult;
29 import com.google.android.gms.common.api.GoogleApiClient;
30 import com.google.android.gms.location.LocationListener;
31 import com.google.android.gms.location.LocationRequest;
32 import com.google.android.gms.location.LocationServices;
33 import com.google.android.gms.maps.CameraUpdateFactory;
34 import com.google.android.gms.maps.GoogleMap;
35 import com.google.android.gms.maps.OnMapReadyCallback;
36 import com.google.android.gms.maps.SupportMapFragment;
37 import com.google.android.gms.maps.model.BitmapDescriptorFactory;
38 import com.google.android.gms.maps.model.LatLng;
39 import com.google.android.gms.maps.model.LatLngBounds;
40 import com.google.android.gms.maps.model.Marker;
41 import com.google.android.gms.maps.model.MarkerOptions;
42 import java.io.IOException;
43 import java.io.StringWriter;
44 import java.util.List;
45 import java.io.BufferedReader;
46 import java.io.IOException;
47 import java.io.InputStream;
48 import java.io.InputStreamReader;
49 import java.io.UnsupportedEncodingException;
50 import java.net.HttpURLConnection;
51 import java.net.URL;
52 import java.net.URLEncoder;
53 import java.util.HashMap;
54 import java.util.List;
55 import java.util.UUID;
56 import java.util.concurrent.ArrayBlockingQueue;
57 import java.util.concurrent.BlockingQueue;
58 import org.json.JSONObject;
59 import org.json.simple.parser.JSONParser;
60 import android.os.AsyncTask;
61 import android.os.Bundle;
62 import android.support.v4.app.FragmentActivity;

```

```

63 import android.util.Log;
64 import android.view.Menu;
65 import android.view.View;
66 import android.view.View.OnClickListener;
67 import android.widget.Button;
68 import android.widget.EditText;
69 import android.widget.Toast;
70 import com.google.android.gms.maps.CameraUpdateFactory;
71 import com.google.android.gms.maps.GoogleMap;
72 import com.google.android.gms.maps.SupportMapFragment;
73 import com.google.android.gms.maps.model.LatLng;
74 import com.google.android.gms.maps.model.MarkerOptions;
75 import com.rabbitmq.client.AMQP;
76 import com.rabbitmq.client.Channel;
77 import com.rabbitmq.client.Connection;
78 import com.rabbitmq.client.ConnectionFactory;
79 import com.rabbitmq.client.DefaultConsumer;
80 import com.rabbitmq.client.Envelope;
81
82 public class CreateNewActivity1 extends FragmentActivity implements
OnMapReadyCallback,
83         GoogleApiClient.ConnectionCallbacks,
84         GoogleApiClient.OnConnectionFailedListener,
85         LocationListener {
86
87     //DEKLARASI VARIABEL
88     private GoogleMap mMap;
89     GoogleApiClient mGoogleApiClient;
90     Location mLastLocation;
91     Marker mCurrLocationMarker;
92     LocationRequest mLocationRequest;
93     MarkerOptions markerOptions;
94     LatLng latLng;
95     ImageView mBtnFind;
96     EditText etPlace;
97     double latget;
98     double lngget;
99     Geocoder geocoder;
100
101    @Override
102    protected void onCreate(Bundle savedInstanceState) {
103        super.onCreate(savedInstanceState);
104        setContentView(R.layout.activity_create_new1);
105
106        if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
107            checkLocationPermission();
108        }
109        // MEMINTA FRAGMENT UNTUK MAPS
110        SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
111            .findFragmentById(R.id.map);
112        mapFragment.getMapAsync(this);
113
114        //PENGATURAN FONT
115        Typeface font_signin =
Typeface.createFromAsset(getAssets(), "fonts/Quicksand-Regular.otf");

```

```

116     Typeface font_activity =
Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
117     Typeface font_title = Typeface.createFromAsset(getAssets(),
"fonts/Quicksand-Bold.otf");
118
119     //JUDUL ACTIVITY
120     TextView activitytitle = (TextView)
findViewById(R.id.activitytitle);
121     activitytitle.setTypeface(font_activity);
122
123     //EDITTEXT UNTUK NAMA RUMAH
124     final EditText etHouseName = (EditText)
findViewById(R.id.etHouseName);
125     etHouseName.setTypeface(font_signin);
126
127     //TEXTVIEW KETERANGAN
128     TextView tvAdd = (TextView) findViewById(R.id.tvAdd);
129     tvAdd.setTypeface(font_title);
130
131     //EDIT TEXT UNTUK ALAMAT RUMAH
132     etPlace = (EditText) findViewById(R.id.etAddress);
133     etPlace.setTypeface(font_signin);
134
135     //TOMBOL UNTUK MENCARI LOKASI/ALAMAT PADA MAPS
136     mBtnFind = (ImageView) findViewById(R.id.btn_show);
137     mBtnFind.setOnClickListener(new OnClickListener() {
138
139         @Override
140         public void onClick(View v) {
141             // Memperoleh lokasi alamat pada maps
142             String location = etPlace.getText().toString();
143             if(location==null || location.equals("")){
144                 Toast.makeText(getApplicationContext(), "No Place is
entered", Toast.LENGTH_SHORT).show();
145                 return;
146             }
147
148             String url =
"https://maps.googleapis.com/maps/api/geocode/json?";
149             try {
150                 // encoding special characters like space in
the user input place
151                 location = URLEncoder.encode(location, "utf-
8");
152             } catch (UnsupportedEncodingException e) {
153                 e.printStackTrace();
154             }
155
156             //membentuk URL untuk mencari
157             String address = "address=" + location;
158             String sensor = "sensor=false";
159             url = url + address + "&" + sensor;
160
161             //mengunduh lokasi
162             DownloadTask downloadTask = new DownloadTask();
163             downloadTask.execute(url);
164         }
165     });

```

```

166     geocoder = new Geocoder(this);
167
168     //BUTTON UNTUK SAVE
169     Button bsave = (Button) findViewById(R.id.bsave);
170     bsave.setTypeface(font_signin);
171     bsave.setOnClickListener(new View.OnClickListener() {
172
173         @Override
174         public void onClick(View v) {
175             try {
176                 //PENGIRIMAN PESAN KE SERVER
177
178                 //membentuk properti pesan
179                 AMQP.BasicProperties props = new
AMQP.BasicProperties
180                     .Builder()
181                     .correlationId(MainActivity.phoneid)
182                     .build();
183
184                 //membentuk format JSON
185                 org.json.simple.JSONObject obj = new
org.json.simple.JSONObject();
186                     obj.put("head", "MH");
187                     obj.put("homeid", MainActivity.homeid);
188                     obj.put("type", "edit");
189                     obj.put("name",
etHouseName.getText().toString());
190                     obj.put("lat",
Double.toString(latLng.latitude));
191                     obj.put("lng",
Double.toString(latLng.longitude));
192                     StringWriter toSent = new StringWriter();
193                     obj.writeJSONString(toSent);
194                     String message = toSent.toString();
195
196                     //mengirim pesan
197                     MainActivity.channel.basicPublish("",,
MainActivity.homeid + "AES", props,
MainActivity.AESEncrypt(message).getBytes());
198                     MainActivity.channel.basicPublish("",,
"AESqueue", props, MainActivity.AESEncrypt(message).getBytes());
199                     //kembali ke halaman sebelumnya
(settingactivity)
200                     finish();
201             } catch (Exception e) {
202                 Toast.makeText(getApplicationContext(), "Failed
to edit house information", Toast.LENGTH_SHORT).show();
203             }
204         }
205
206     });
207
208     //KETIKA ACTIVITYINI TERBUKA, APLIKASI AKAN MEMINTA DATA
NAMA RUMAH DAN LOKASI YANG TERSIMPAN DI SERVER
209     try {
210         JSONParser parser = new JSONParser();
211         //memperoleh informasi menggunakan skema RPC
212         String message = rpc();

```

```

213             org.json.simple.JSONObject jsonObject =
214                 (org.json.simple.JSONObject) parser.parse(message);
215                 //memperoleh koordinat latitude dan longitude dari
server
216                 latget = Double.parseDouble((String)
jsonObject.get("lat"));
217                 lngget = Double.parseDouble((String)
jsonObject.get("lng"));
218                 LatLng = new LatLng(latget,lngget);
219                 etHouseName.setText((String) jsonObject.get("name"));
220             } catch (Exception e) {
221                 Toast.makeText(getApplicationContext(), "out",
Toast.LENGTH_SHORT).show();
222             }
223         }
224
225     @Override
226     public void onMapReady(GoogleMap googleMap) {
227         mMap = googleMap;
228         mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
229
230         //Inisiasi Google Play Service
231         if (android.os.Build.VERSION.SDK_INT >=
Build.VERSION_CODES.M) {
232             if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {
233                 buildGoogleApiClient();
234                 mMap.setMyLocationEnabled(true);
235             }
236         }
237     }
238
239     else {
240         buildGoogleApiClient();
241         mMap.setMyLocationEnabled(true);
242     }
243
244     //Meletakkan marker pada lokasi yang ditekan
245     mMap.setOnMapClickListener(new
GoogleMap.OnMapClickListener() {
246
247         @Override
248         public void onMapClick(LatLng arg0) {
249
250             // Getting the Latitude and Longitude of the
touched location
251             LatLng = arg0;
252
253             // Clears the previously touched position
254             mMap.clear();
255
256             // Animating to the touched position
257
mMap.animateCamera(CameraUpdateFactory.newLatLng(latLng));
258
259             // Creating a marker
260             markerOptions = new MarkerOptions();
261

```

```

262         // Setting the position for the marker
263         markerOptions.position(latLng);
264
265         // Placing a marker on the touched position
266         mMap.addMarker(markerOptions);
267
268         // Adding Marker on the touched location with
269         address
270         new
271     });
272
273     mMap.addMarker(new MarkerOptions()
274         .position(new LatLng(latget, lngget))
275         .title("Your Home"));
276     mMap.moveCamera(CameraUpdateFactory.newLatLng(new
277     LatLng(latget, lngget)));
278
279     try {
280         List<Address> address =
281     geocoder.getFromLocation(latget, lngget, 1);
282         etPlace.setText(address.get(0).getAddressLine(0));
283     } catch (Exception e) {
284         Toast.makeText(getApplicationContext(), "Failed to get
285         address", Toast.LENGTH_SHORT).show();
286     }
287
288     public String rpc () {
289         //FUNGSI UNTUK MENGIRIM REQUEST NAMA DAN LOKASI DARI SERVER
290         try {
291             String replyQueueName =
292             MainActivity.channel.queueDeclare().getQueue();
293
294             //properti pesan
295             AMQP.BasicProperties props = new AMQP.BasicProperties
296                 .Builder()
297                 .correlationId(MainActivity.phoneid)
298                 .replyTo(replyQueueName)
299                 .build();
300
301             //pesan JSON
302             org.json.simple.JSONObject obj = new
303             org.json.simple.JSONObject();
304             obj.put("head", "MH");
305             obj.put("homeid", MainActivity.homeid);
306             obj.put("type", "list");
307             StringWriter toSent = new StringWriter();
308             obj.writeJSONString(toSent);
309             String message = toSent.toString();
310             //mengirim pesan
311             MainActivity.channel.basicPublish("", MainActivity.homeid+"AES", props,
312             MainActivity.AESencrypt(message).getBytes("UTF-8"));
313
314             //menunggu balasan

```

```

311             final BlockingQueue<String> response = new
ArrayBlockingQueue<>(2);
312             MainActivity.channel.basicConsume(replyQueueName, true,
new DefaultConsumer(MainActivity.channel) {
313                 @Override
314                 public void handleDelivery(String consumerTag,
Envelope envelope, AMQP.BasicProperties properties, byte[] body)
throws IOException {
315                     if
(properties.getCorrelationId().equals(MainActivity.phoneid)) {
316                         response.offer(new String(body, "UTF-8"));
317                     }
318                 }
319             );
320             //mendapat balasan
321             String res = MainActivity.AESdecrypt(response.take());
322             return res;
323
324         } catch (Exception e) {
325             Toast.makeText(getApplicationContext(),"Can't connect
to the server\nCheck your connection",Toast.LENGTH_SHORT).show();
326             return null;
327         }
328     }
329
330     protected synchronized void buildGoogleApiClient() {
331         mGoogleApiClient = new GoogleApiClient.Builder(this)
332             .addConnectionCallbacks(this)
333             .addOnConnectionFailedListener(this)
334             .addApi(LocationServices.API)
335             .build();
336         mGoogleApiClient.connect();
337     }
338
339     @Override
340     public void onConnected(Bundle bundle) {
341         mLocationRequest = new LocationRequest();
342         mLocationRequest.setInterval(1000);
343         mLocationRequest.setFastestInterval(1000);
344
mLocationRequest.setPriority(LocationRequest.PRIORITY_BALANCED_POWER_A
CCURACY);
345         if (ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION)
== PackageManager.PERMISSION_GRANTED) {
346
LocationServices.FusedLocationApi.requestLocationUpdates(mGoogleApiClient,
mLocationRequest, this);
347         }
348     }
349
350 }
351
352     @Override
353     public void onConnectionSuspended(int i) {
354
}
355
356     @Override
357     public void onLocationChanged(Location location) {
358
}

```

```

359         mLastLocation = location;
360         if (mCurrLocationMarker != null) {
361             mCurrLocationMarker.remove();
362         }
363     }
364     //Place current location marker
365     LatLng latLng = new LatLng(location.getLatitude(),
366     location.getLongitude());
367     //move map camera
368     mMap.moveCamera(CameraUpdateFactory.newLatLng(latLng));
369     mMap.animateCamera(CameraUpdateFactory.zoomTo(11));
370
371     //stop location updates
372     if (mGoogleApiClient != null) {
373
374         LocationServices.FusedLocationApi.removeLocationUpdates(mGoogleApiClient,
375         this);
376     }
377
378     @Override
379     public void onConnectionFailed(ConnectionResult
380     connectionResult) {
381
382         public static final int MY_PERMISSIONS_REQUEST_LOCATION = 99;
383         public boolean checkLocationPermission() {
384             if (ContextCompat.checkSelfPermission(this,
385                 Manifest.permission.ACCESS_FINE_LOCATION)
386                 != PackageManager.PERMISSION_GRANTED) {
387
388                 // Asking user if explanation is needed
389                 if
390                     (ActivityCompat.shouldShowRequestPermissionRationale(this,
391                         Manifest.permission.ACCESS_FINE_LOCATION)) {
392
393                     //Prompt the user once explanation has been shown
394                     ActivityCompat.requestPermissions(this,
395                         new
396                         String[]{Manifest.permission.ACCESS_FINE_LOCATION},
397                         MY_PERMISSIONS_REQUEST_LOCATION);
398
399                     } else {
400                         // No explanation needed, we can request the
401                         permission.
402                         ActivityCompat.requestPermissions(this,
403                             new
404                             String[]{Manifest.permission.ACCESS_FINE_LOCATION},
405                             MY_PERMISSIONS_REQUEST_LOCATION);
406                         }
407
408             @Override

```

```

409     public void onRequestPermissionsResult(int requestCode,
410                                         String permissions[],
411                                         int[] grantResults) {
412         switch (requestCode) {
413             case MY_PERMISSIONS_REQUEST_LOCATION: {
414                 // If request is cancelled, the result arrays are
415                 // empty.
416                 if (grantResults.length > 0
417                     && grantResults[0] ==
PackageManager.PERMISSION_GRANTED) {
418                     // permission was granted. Do the
419                     // contacts-related task you need to do.
420                     if (ContextCompat.checkSelfPermission(this,
421                         Manifest.permission.ACCESS_FINE_LOCATION)
422                             == PackageManager.PERMISSION_GRANTED) {
423                         if (mGoogleApiClient == null) {
424                             buildGoogleApiClient();
425                         }
426                         mMap.setMyLocationEnabled(true);
427                     }
428
429                 } else {
430                     // Permission denied, Disable the functionality
431                     // that depends on this permission.
432                     Toast.makeText(this, "permission denied",
Toast.LENGTH_LONG).show();
433                 }
434                 return;
435             }
436         }
437     }
438
439
440     private class ReverseGeocodingTask extends AsyncTask<LatLng,
Void, String> {
441         Context mContext;
442
443         public ReverseGeocodingTask(Context context) {
444             super();
445             mContext = context;
446         }
447
448         // Finding address using reverse geocoding
449         @Override
450         protected String doInBackground(LatLng... params) {
451
452             double latitude = params[0].latitude;
453             double longitude = params[0].longitude;
454
455             List<Address> addresses = null;
456             String addressText="";
457
458             try {

```

```

459             addresses = geocoder.getFromLocation(latitude,
longitude,1);
460         } catch (IOException e) {
461             e.printStackTrace();
462         }
463
464         if(addresses != null && addresses.size() > 0 ){
465             Address address = addresses.get(0);
466
467             addressText = String.format("%s, %s, %s",
468                                         address.getMaxAddressLineIndex() > 0 ?
address.getAddressLine(0) : "",
469                                         address.getLocality(),
470                                         address.getCountryName());
471         }
472
473         return addressText;
474     }
475
476     @Override
477     protected void onPostExecute(String addressText) {
478         markerOptions.title(addressText);
479         mMap.addMarker(markerOptions);
480         etPlace.setText(addressText);
481     }
482 }
483
484 private String downloadUrl(String strUrl) throws IOException{
485     String data = "";
486     InputStream iStream = null;
487     HttpURLConnection urlConnection = null;
488     try{
489         URL url = new URL(strUrl);
490         // Creating an http connection to communicate with url
491         urlConnection = (HttpURLConnection)
url.openConnection();
492
493         // Connecting to url
494         urlConnection.connect();
495
496         // Reading data from url
497         iStream = urlConnection.getInputStream();
498
499         BufferedReader br = new BufferedReader(new
InputStreamReader(iStream));
500
501         StringBuffer sb = new StringBuffer();
502
503         String line = "";
504         while( ( line = br.readLine()) != null){
505             sb.append(line);
506         }
507
508         data = sb.toString();
509         br.close();
510
511     }catch(Exception e){
512

```

```

513         }finally{
514             iStream.close();
515             urlConnection.disconnect();
516         }
517         return data;
518     }
519     /** A class, to download Places from Geocoding webservice */
520     private class DownloadTask extends AsyncTask<String, Integer,
String>{
521
522         String data = null;
523
524         // Invoked by execute() method of this object
525         @Override
526         protected String doInBackground(String... url) {
527             try{
528                 data = downloadUrl(url[0]);
529             }catch(Exception e){
530                 Log.d("Background Task",e.toString());
531             }
532             return data;
533         }
534
535         // Executed after the complete execution of
doInBackground() method
536         @Override
537         protected void onPostExecute(String result){
538
539             // Instantiating ParserTask which parses the json data
from Geocoding webservice
540             // in a non-ui thread
541             ParserTask parserTask = new ParserTask();
542
543             // Start parsing the places in JSON format
544             // Invokes the "doInBackground()" method of the class
ParseTask
545             parserTask.execute(result);
546         }
547     }
548
549     /** A class to parse the Geocoding Places in non-ui thread */
550     class ParserTask extends AsyncTask<String, Integer,
List<HashMap<String, String>>>{
551
552         JSONObject jObject;
553
554         // Invoked by execute() method of this object
555         @Override
556         protected List<HashMap<String, String>>
doInBackground(String... jsonData) {
557
558             List<HashMap<String, String>> places = null;
559             GeocodeJSONParser parser = new GeocodeJSONParser();
560
561             try{
562                 jObject = new JSONObject(jsonData[0]);
563
564                 /** Getting the parsed data as a an ArrayList */

```

```

565         places = parser.parse(jObject);
566
567     }catch(Exception e){
568         Log.d("Exception",e.toString());
569     }
570     return places;
571 }
572
573     // Executed after the complete execution of
574     doInBackground() method
575     @Override
576     protected void onPostExecute(List<HashMap<String, String>>
list){
577
578         // Clears all the existing markers
579         mMap.clear();
580
581         for(int i=0;i<list.size();i++){
582
583             // Creating a marker
584             MarkerOptions markerOptions = new MarkerOptions();
585
586             // Getting a place from the places list
587             HashMap<String, String> hmPlace = list.get(i);
588
589             // Getting latitude of the place
590             double lat =
Double.parseDouble(hmPlace.get("lat"));
591
592             // Getting longitude of the place
593             double lng =
Double.parseDouble(hmPlace.get("lng"));
594
595             // Getting name
596             String name = hmPlace.get("formatted_address");
597
598             LatLng latLng = new LatLng(lat, lng);
599
600             // Setting the position for the marker
601             markerOptions.position(latLng);
602
603             // Setting the title for the marker
604             markerOptions.title(name);
605
606             // Placing a marker on the touched position
607             mMap.addMarker(markerOptions);
608
609             // Locate the first location
610             if(i==0)
611
612         }
613     }
614 }
```

Kode pada GeocodeJSONParser.java:

```

1 package com.example.billyausten.newfinal;
2
3 import org.json.JSONArray;
4 import org.json.JSONException;
5 import org.json.JSONObject;
6
7 import java.util.ArrayList;
8 import java.util.HashMap;
9 import java.util.List;
10
11 public class GeocodeJSONParser {
12
13     /** Receives a JSONObject and returns a list */
14     public List<HashMap<String, String>> parse(JSONObject jObject) {
15
16         JSONArray jPlaces = null;
17         try {
18             /** Retrieves all the elements in the 'places' array */
19             jPlaces = jObject.getJSONArray("results");
20         } catch (JSONException e) {
21             e.printStackTrace();
22         }
23         /** Invoking getPlaces with the array of json object
24          * where each json object represent a place
25          */
26         return getPlaces(jPlaces);
27     }
28
29     private List<HashMap<String, String>> getPlaces(JSONArray
jPlaces) {
30         int placesCount = jPlaces.length();
31         List<HashMap<String, String>> placesList = new
ArrayList<HashMap<String, String>>();
32         HashMap<String, String> place = null;
33
34         /** Taking each place, parses and adds to list object */
35         for(int i=0; i<placesCount;i++) {
36             try {
37                 /** Call getPlace with place JSON object to parse
the place */
38                 place = getPlace((JSONObject)jPlaces.get(i));
39                 placesList.add(place);
40
41             } catch (JSONException e) {
42                 e.printStackTrace();
43             }
44         }
45
46         return placesList;
47     }
48
49     /** Parsing the Place JSON object */
50     private HashMap<String, String> getPlace(JSONObject jPlace) {
51
52         HashMap<String, String> place = new HashMap<String,
String>();
53         String formatted_address = "-NA-";
54         String lat="";

```

```

55     String lng="";
56
57     try {
58         // Extracting formatted address, if available
59         if(!jPlace.isNull("formatted_address")){
60             formatted_address =
jPlace.getString("formatted_address");
61         }
62
63         lat =
jPlace.getJSONObject("geometry").getJSONObject("location").getString("lat");
64         lng =
jPlace.getJSONObject("geometry").getJSONObject("location").getString("lng");
65
66         place.put("formatted_address", formatted_address);
67         place.put("lat", lat);
68         place.put("lng", lng);
69
70     }catch (JSONException e) {
71         e.printStackTrace();
72     }
73     return place;
74 }
75

```

Bagian penting pada kode *CreateNewActivity1.java* adalah

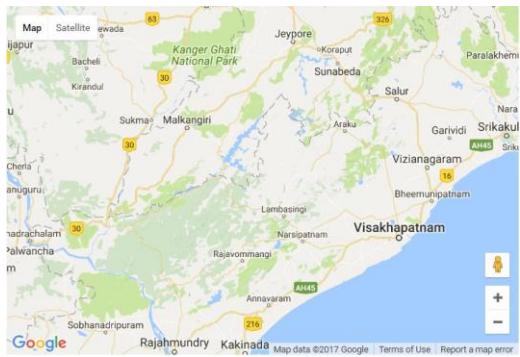
- Baris 106 sampai 133 berisi inisiasi *fragment* untuk Google Maps dan pengecekan versi Android, selain itu *onCreate* diisi dengan pengaturan *font* dan inisiasi komponen-komponen pembentuk *Activity*.
- Baris 136 sampai 165 digunakan untuk mencari alamat pada *EditText* alamat dan kemudian ditampilkan pada Google Maps.
- Baris 169 sampai 206 berisi hal-hal yang dilakukan ketika *Button* “SAVE” ditekan, yaitu mengirimkan perintah dalam bentuk JSON kepada server.
- Baris 209 sampai 222 digunakan untuk memperoleh nama dan alamat rumah yang sudah terdaftar pada server, kemudian menampilkan nama dan alamat tersebut pada halaman House Information.
- Baris 225 sampai 284 merupakan kode untuk mengakses dan mengatur Google Maps. Pada baris 231 sampai 242 merupakan inisiasi Google Play Service untuk mendukung Google Maps. Baris 245 sampai 271 merupakan kode untuk menuliskan alamat dan meletakkan marker pada lokasi yang dipilih oleh pengguna.
- Baris 287 sampai 328 merupakan fungsi dari *rpc* yaitu fungsi untuk mengirimkan perintah dalam bentuk JSON yang inisiasinya dilakukan pada baris 169 sampai

2016. Kode JSON dapat dilihat pada tabel dibawah. Pada bagian ini, terdapat

properti dari pesan, isi dari json, kode untuk mengirim pesan, menunggu dan juga menerima balasan.

- Baris 365 sampai 375 merupakan fungsi ketika lokasi diubah dengan cara menekan lokasi bari pada peta.
- Baris 377 sampai 437 mengurus *permission* penggunaan MAPS.
- Baris 440 sampai 482 merupakan fungsi untuk memberikan *output* sebuah alamat terhadap *input* longitude dan latitude berdasarkan input pengguna pada lokasi yang dipilih. Kemudian menuliskan alamat tersebut pada *EditText* yang tersedia dan meletakkan marker pada lokasi yang dipilih.
- Baris 484 hingga 547 berisi mengenai koneksi yang diperlukan oleh Google Maps.
- Baris 550 hingga 613 merupakan fungsi untuk mengubah *input* berupa alamat menjadi lokasi longitude dan latitude yang dapat diproses lebih lanjut oleh Google Maps.
- Untuk penjelasan lebih lanjut, dapat dilihat sebagai berikut:

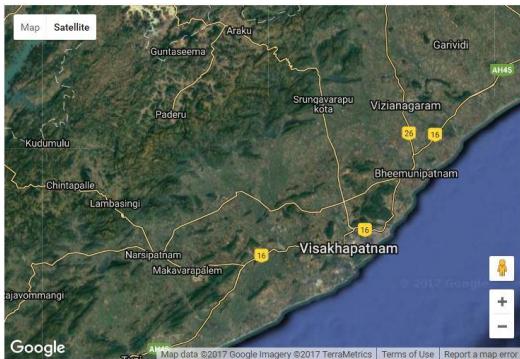
Dalam *onMapReady* terdapat kode *setMapType* yang berfungsi untuk memilih tipe tampilan pada Google Maps, apakah tampilan tipe NORMAL (menampilkan jalan, sungai, laut, danau dan tempat-tempat penting), tipe SATELLITE (menampilkan citra dari satelit Google Earth), tipe HYBRID (menampilkan tampilan dari NORMAL dan SATELLITE) dan tipe TERRAIN (menampilkan tampilan peta berdasarkan informasi medan). Dipilih tipe NORMAL karena tipe ini merupakan tipe yang paling mudah dilihat karena pada umumnya pengguna tidak memerlukan citra dari Google Earth maupun informasi medan dari lokasi rumahnya. Pada *onMapReady* terdapat pengecekan *permission ACCESS_FINE_LOCATION* yang berguna dalam mendeteksi lokasi pengguna. Setelah mendapatkan izin, fungsi *onLocationChanged* dapat mendeteksi lokasi dengan mencari lokasi *latitude* dan *longitude* dari *handphone* pengguna dengan menggunakan kode *getLatitude* dan *getLongitude*. Kemudian posisi peta diarahkan ke lokasi pengguna dengan menggunakan kode *moveCamera*.



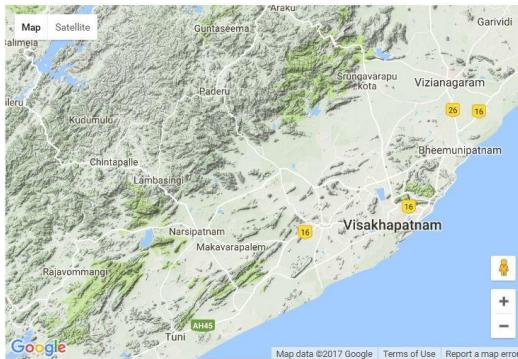
Gambar 45 Google Maps Tipe NORMAL



Gambar 46 Google Maps Tipe SATELLITE



Gambar 47 Google Maps Tipe HYBRID



Gambar 48 Google Maps Tipe TERRAIN

Ketika tidak berada di sekitar rumah, pengguna dapat mengetikan alamat rumahnya pada *EditText* dengan *hint Address* untuk mengubah posisi peta menjadi berada pada lokasi yang diketikkan. Untuk melakukan hal tersebut, diperlukan *GeocodeJSONParser.java* untuk memproses input alamat oleh pengguna menjadi posisi longitude dan latitude pada peta. Setelah itu dengan menggunakan fungsi *onPostExecute*, posisi peta akan diubah dan akan ditampilkan *marker* pada lokasi yang sesuai dengan input dari pengguna. Posisi *marker* dipindahkan dengan menggunakan kode *markerOption.position*. Ketika lokasi yang ditampilkan pada peta tidak sesuai dengan alamat yang sebenarnya, pengguna dapat memindahkan lokasi *marker* pada peta dengan melakukan *tapping* pada lokasi yang dirasa benar. Hal ini dilakukan dengan melakukan *setOnMapClickListener* pada Google Map, memunculkan *marker* pada lokasi tersebut dan melakukan fungsi *ReverseGeocodingTask*. Pada *ReverseGeocodingTask*, digunakan *Geocoder* untuk menghubungkan informasi lokasi pada peta berdasarkan latitude dan longitude dan juga lokasi pada peta berdasarkan alamat jalan. Sehingga ketika pengguna menekan sebuah lokasi pada peta, maka kode *geocoder.getFromLocation* akan membaca lokasi tersebut dan menyimpan lokasi tersebut dalam sebuah variabel. Kemudian akan dibaca alamat lokasi, kota dan juga negara dari

variabel tersebut dengan cara menuliskan `getAddressLine`, `getLocality` dan juga `getCountryName`. Kemudian hasil tersebut disimpan dalam sebuah `string` yang akan ditampilkan dalam `TextView` yang berada dibawah Google Maps. `TextView` tidak diberi nilai awal, sehingga sebelum pengguna melakukan *tapping* pada Google Maps, `TextView` tidak akan tampak. Terdapat `Button` “SAVE” yang dapat digunakan membuka kembali halaman `SettingActivity.java`.

Ketika halaman ini terbuka, aplikasi secara otomatis meminta *request* data rumah dengan paket data *request* sebagai berikut.

Tabel 36 Tabel JSON pada saat request data rumah

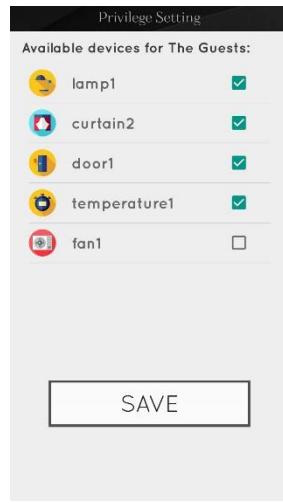
Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MH (Manajemen Home)	list	<ul style="list-style-type: none"> • Home ID 	<i>Host</i>	AES	Nama Rumah dan Posisi atau Koordinat Lintang dan Bujur dari Rumah

Bila *user* melakukan edit terhadap informasi rumah, maka perintah edit yang dikirimkan memiliki bentuk paket data sebagai berikut.

Tabel 37 Tabel JSON pada saat edit informasi rumah

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MH (Manajemen Home)	edit	<ul style="list-style-type: none"> • Home ID • Nama Baru • Koordinat Rumah Baru 	<i>Host</i>	AES	Tidak ada jawaban

6.20 Halaman Privilege (`PrivilegeActivity.java`)



Gambar 55 Halaman Privilege Setting

Kode XML halaman ini sebagai berikut:

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_privilege"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="0dp"
10    android:paddingRight="0dp"
11    android:paddingTop="0dp"
12
13    tools:context="com.example.billyausten.newfinal.PrivilegeActivity"
14    android:orientation="vertical">
15
16        <LinearLayout
17            android:layout_width="match_parent"
18            android:layout_height="33dp"
19            android:layout_gravity="center"
20            android:orientation="vertical"
21            android:background="@drawable/backblack2"
22            android:id="@+id/backtitle">
23
24            <LinearLayout
25                android:layout_width="wrap_content"
26                android:layout_height="wrap_content"
27                android:orientation="horizontal"
28                android:layout_gravity="center">
29
30                <TextView
31                    android:layout_width="290dp"
32                    android:layout_height="wrap_content"
33                    android:text="Privilege Setting"
34                    android:id="@+id/activitytitle"
35                    android:textAlignment="center"
36                    android:layout_gravity="center_vertical"
37                    android:textSize="18dp"
```

```

36         android:textColor="#AAAAAA"
37         android:layout_marginTop="3dp"/>
38
39     </LinearLayout>
40
41 </LinearLayout>
42
43 <TextView
44     android:layout_width="match_parent"
45     android:layout_height="wrap_content"
46     android:textSize="18dp"
47     android:text="Available devices for The Guests:"
48     android:layout_marginTop="10dp"
49     android:layout_marginBottom="10dp"
50     android:layout_marginLeft="15dp"
51     android:textColor="#404040"
52     android:id="@+id/tvsubject"/>
53
54 <ListView
55     android:layout_width="311dp"
56     android:layout_height="380dp"
57     android:id="@+id/listView"
58     android:layout_gravity="center"/>
59
60 <Button
61     android:layout_width="wrap_content"
62     android:layout_height="wrap_content"
63     android:layout_gravity="center"
64     android:text="Save"
65     android:layout_marginTop="30dp"
66     android:textSize="30dp"
67     android:id="@+id/bsave"
68     android:background="@drawable/signin"/>
69
70</LinearLayout>

```

Pada halaman ini, admin dapat mengubah kewenangan *privilege* guest dengan cara *check* atau *uncheck* pada *Checkbox device* yang dipilih. *Device* ditampilkan dengan menggunakan *list* seperti *list* pada penambahan *device room*. *Device* pada *Checkbox* yang bertanda *check* dapat digunakan oleh seluruh *privilege*, sedangkan jika dalam keadaan *uncheck*, *device* tersebut hanya dapat digunakan pengguna dengan kategori *privilege* admin. Setelah menentukan pilihan, pengguna dapat menekan *Button* “SAVE” untuk menyimpan konfigurasi yang diinginkan.

Kode java halaman ini sebagai berikut:

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Intent;
4 import android.graphics.Typeface;
5 import android.support.v7.app.AppCompatActivity;

```

```

6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.AdapterView;
9 import android.widget.Button;
10import android.widget.ListView;
11import android.widget.TextView;
12import com.rabbitmq.client.AMQP;
13import com.rabbitmq.client.DefaultConsumer;
14import com.rabbitmq.client.Envelope;
15import org.json.simple.JSONObject;
16import org.json.simple.parser.JSONParser;
17import java.io.IOException;
18import java.io.StringWriter;
19import java.util.ArrayList;
20import java.util.List;
21import java.util.UUID;
22import java.util.concurrent.ArrayBlockingQueue;
23import java.util.concurrent.BlockingQueue;
24
25public class PrivilegeActivity extends AppCompatActivity {
26
27    ListView listView;
28
29    @Override
30    protected void onCreate(Bundle savedInstanceState) {
31        super.onCreate(savedInstanceState);
32        setContentView(R.layout.activity_privilege);
33
34        //PENGATURAN FONT
35        Typeface font_signin = Typeface.createFromAsset(getAssets(),
36                "fonts/Quicksand-Regular.otf");
37        Typeface font_title = Typeface.createFromAsset(getAssets(),
38                "fonts/Quicksand-Bold.otf");
39        Typeface font_or = Typeface.createFromAsset(getAssets(),
40                "fonts/Quicksand-BoldItalic.otf");
41        Typeface font_activity =
42            Typeface.createFromAsset(getAssets(), "fonts/Foglihten.otf");
43
44        //JUDUL ACTIVITY
45        TextView activitytitle = (TextView)
46        findViewById(R.id.activitytitle);
47        activitytitle.setTypeface(font_activity);
48
49        //TULISAN "AVAILABLE DEVICES FOR GUESTS:"
50        TextView tvsubject = (TextView)
51        findViewById(R.id.tvsubject);
52        tvsubject.setTypeface(font_title);
53
54        //LISTVIEW YANG BERISI DAFTAR BARANG
55        final ListAddDeviceRoom customList = new
56        ListAddDeviceRoom(this, MainActivity.mainList);
57        for (ArrayList<String> element : MainActivity.mainList) {
58            if (element.get(5).equals("Guest")) {
59                ListAddDeviceRoom.checked.add(element);
60            }
61        }
62        listView = (ListView) findViewById(R.id.listView);
63        listView.setAdapter(customList);

```

```

57
58     //TOMBOL SAVE
59     Button bsave = (Button) findViewById(R.id.bsave);
60     bsave.setTypeface(font_signin);
61     bsave.setOnClickListener(new View.OnClickListener() {
62
63         @Override
64         public void onClick(View v) {
65
66             try {
67                 JSONObject obj = new JSONObject();
68
69                 //data yang dikirimkan
70                 obj.put("homeid", MainActivity.homeid);
71                 obj.put("head", "MD");
72                 obj.put("type", "editprivilege");
73                 obj.put("checked", ListAddDeviceRoom.checked);
74                 StringWriter toSent = new StringWriter();
75                 obj.writeJSONString(toSent);
76                 String message = toSent.toString();
77                 AMQP.BasicProperties props = new
AMQP.BasicProperties
78                     .Builder()
79                     .correlationId(MainActivity.phoneid)
80                     .build();
81                 MainActivity.channel.basicPublish("", MainActivity.homeid + "AES", props,
MainActivity.AESencrypt(message).getBytes("UTF-8"));
82
83             } catch (Exception e) {
84
85             }
86             ListAddDeviceRoom.checked.clear();
87             Intent afterloginIntent = new
Intent(PrivilegeActivity.this,MainActivity.class);
88             PrivilegeActivity.this.startActivity(afterloginIntent);
89         }
90     });
91 }
92

```

Pada kode java, bagian yang penting adalah:

- Baris 34 sampai 46 menginisiasi *TextView* dan mengatur *font* yang digunakan.
- Baris 48 sampai 56 mengatur isi dari *list device* yang ditampilkan pada *ListView*, jika *device* merupakan *privilege guest*, maka *Checkbox* akan diberi tanda centang, jika bukan *privilege guest*, maka *Checkbox* tidak diberi tanda centang.
- Baris 58 sampai 90 mengatur hal-hal apa saja yang dilakukan ketika *Button* “SAVE” ditekan. Baris 70 sampai 85 menentukan perintah yang dikirimkan kepada server untuk menyimpan *device* untuk *privilege guest* dalam data JSON seperti berikut:

Tabel 38 Tabel JSON Privilege Setting

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	editprivilege	<ul style="list-style-type: none"> • Home ID • Daftar seluruh device beserta privilegenya 	<i>Host</i>	AES	Tidak ada jawaban

6.21 Halaman About (AboutActivity.java)



Gambar 56 Halaman About (AboutActivity.java)

Kode XML halaman ini sebagai berikut:

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:id="@+id/activity_about"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="0dp"
10    android:paddingRight="0dp"
11    android:paddingTop="0dp"
12    tools:context="com.example.billyausten.newfinal.AboutActivity"
13    android:orientation="vertical">
14        <LinearLayout

```

```

15     android:layout_width="match_parent"
16     android:layout_height="33dp"
17     android:layout_gravity="center"
18     android:orientation="vertical"
19     android:background="@drawable/backblack2"
20     android:id="@+id/backtitle">
21     <LinearLayout
22         android:layout_width="wrap_content"
23         android:layout_height="wrap_content"
24         android:orientation="horizontal"
25         android:layout_gravity="center">
26
27         <TextView
28             android:layout_width="290dp"
29             android:layout_height="wrap_content"
30             android:text="About"
31             android:id="@+id/activitytitle"
32             android:textAlignment="center"
33             android:layout_gravity="center_vertical"
34             android:textSize="18dp"
35             android:textColor="#AAAAAA"
36             android:layout_marginTop="3dp"/>
37     </LinearLayout>
38 </LinearLayout>
39
40     <ImageView
41         android:layout_width="250dp"
42         android:layout_height="250dp"
43         android:src="@drawable/mindsabout"
44         android:layout_gravity="center"
45         android:layout_marginTop="10dp"/>
46
47</LinearLayout>

```

Pada halaman ini ditampilkan beberapa hal mengenai MINDS baik dari sisi aplikasi maupun dari sisi pengembang aplikasi MINDS. Pada halaman ini pengguna dapat melihat *credit* dari MINDS. Halaman ini dibuat dengan menggunakan gabungan beberapa *ImageView* ataupun *TextView*.

Kode java halaman ini sebagai berikut:

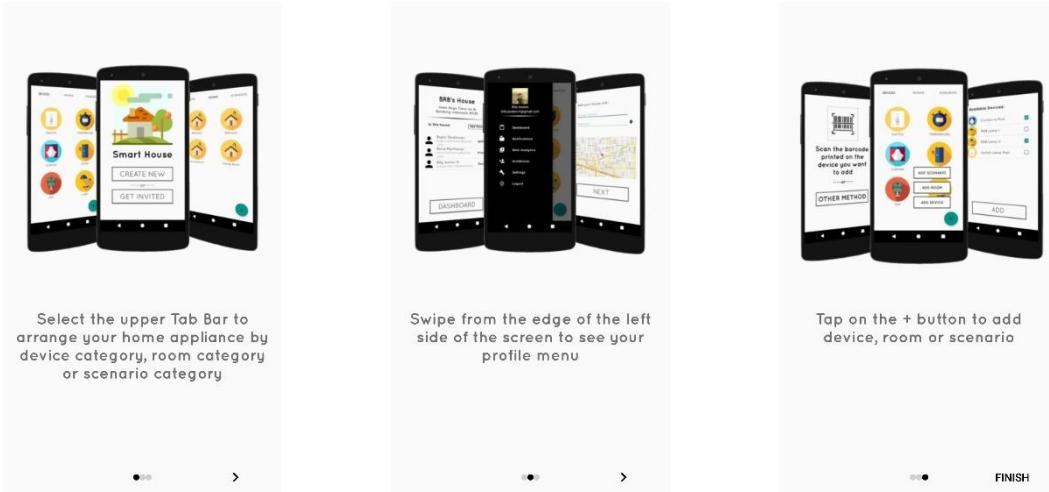
```

1 package com.example.billyausten.newfinal;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class AboutActivity extends AppCompatActivity {
7
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_about);
12     }

```

Pada kode java, hanya digunakan `setContentView` untuk menampilkan halaman yang sesuai dengan pengaturan pada kode xml.

6.22 Halaman Help (PagerActivity.java)



Gambar 57 Halaman Pager 1

Gambar 58 Halaman Pager 2

Gambar 59 Halaman Pager 3

Kode XML halaman ini adalah sebagai berikut (activity_pager.xml) :

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.design.widget.CoordinatorLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:id="@+id/main_content">
7
8     <android.support.v4.view.ViewPager
9         android:id="@+id/container"
10        android:layout_width="match_parent"
11        android:layout_height="match_parent"
12        android:paddingBottom="?attr/actionBarSize" />
13
15     <View
16         android:layout_width="match_parent"
17         android:layout_height="1dp"
18         android:layout_gravity="bottom"
19         android:layout_marginBottom="?attr/actionBarSize"
20         android:alpha="0.12"
21         android:background="@android:color/white" />
22     <FrameLayout
23         android:layout_width="match_parent"
24         android:layout_height="?attr/actionBarSize"
25         android:layout_gravity="bottom"
26         android:paddingEnd="@dimen/activity_horizontal_margin"

```

```

27     android:paddingLeft="@dimen/activity_horizontal_margin"
28     android:paddingRight="@dimen/activity_horizontal_margin"
29     android:paddingStart="@dimen/activity_horizontal_margin">
30
31     <LinearLayout
32         android:layout_width="wrap_content"
33         android:layout_height="wrap_content"
34         android:layout_gravity="center"
35         android:orientation="horizontal">
36         <ImageView
37             android:id="@+id/intro_indicator_0"
38             android:layout_width="8dp"
39             android:layout_height="8dp"
40             android:layout_marginEnd="0dp"
41             android:layout_marginRight="0dp"
42             android:background="@drawable/indicator_unselected"
43         />
44
45         <ImageView
46             android:id="@+id/intro_indicator_1"
47             android:layout_width="8dp"
48             android:layout_height="8dp"
49             android:layout_marginEnd="0dp"
50             android:layout_marginRight="0dp"
51             android:background="@drawable/indicator_unselected"
52         />
53
54         <ImageView
55             android:id="@+id/intro_indicator_2"
56             android:layout_width="8dp"
57             android:layout_height="8dp"
58             android:layout_marginEnd="0dp"
59             android:layout_marginRight="0dp"
60             android:background="@drawable/indicator_unselected"
61         />
62
63         <Button
64             android:id="@+id/intro_btn_finish"
65             style="@style/Widget.AppCompat.Button.Borderless"
66             android:layout_width="wrap_content"
67             android:layout_height="wrap_content"
68             android:layout_gravity="end|center"
69             android:text="Finish"
70             android:textColor="#000000"
71             android:visibility="gone" />
72         <ImageButton
73             android:id="@+id/intro_btn_next"
74             style="@style/Widget.AppCompat.Button.Borderless"
75             android:layout_width="wrap_content"
76             android:layout_height="wrap_content"
77             android:layout_gravity="end|center"
78             android:padding="@dimen/activity_horizontal_margin"
79             android:src="@drawable/ic_chevron_right_black_24dp"
80             android:tint="#000000" />
81     </FrameLayout>

```

```
82
83</android.support.design.widget.CoordinatorLayout>
```

Kode XML halaman ini adalah sebagai berikut (fragment_page.xml) :

```
1 <LinearLayout
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:tools="http://schemas.android.com/tools"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"
6     android:paddingBottom="@dimen/activity_vertical_margin"
7     android:paddingLeft="@dimen/activity_horizontal_margin"
8     android:paddingRight="@dimen/activity_horizontal_margin"
9     android:orientation="vertical"
10    android:paddingTop="@dimen/activity_vertical_margin"
11
12    <ImageView
13        android:layout_width="350dp"
14        android:layout_height="280dp"
15        android:layout_gravity="center"
16        android:layout_marginTop="50dp"
17        android:id="@+id/section_img"/>
18
19    <TextView
20        android:id="@+id/section_label"
21        android:layout_width="wrap_content"
22        android:layout_height="wrap_content"
23        android:layout_gravity="center"
24        android:layout_marginTop="50dp"
25        android:textSize="20dp"
26        android:textAlignment="center"/>
27
28</LinearLayout>
```

Pada halaman ini, pengguna diberitahu beberapa fitur pada aplikasi MINDS yang mungkin tidak pengguna sadari seperti tampilan *tab menu*, *swipe profile menu* dan *add device/room/scenario*. Oleh karena itu, halaman ini dibuat dengan harapan pengguna baru dapat memahami cara penggunaan aplikasi MINDS dengan melihat keterangan pada *PagerActivity.java*. Pengguna dapat mengganti tampilan *onboarding* dengan cara di-swipe karena *PagerActivity.java* berupa *fragment* sehingga tidak perlu menggunakan *intent* yang biasa digunakan untuk berpindah *activity*. Untuk membuat tampilan seperti ini dibutuhkan dua buah *java class* dan dua buah *layout xml class*. *PagerActivity.java* mengatur jalan dan tampilan dari halaman pager tersebut sedangkan *Utils.java* digunakan untuk melakukan desain terhadap tampilan *Activity*. Dua buah *layout xml* digunakan untuk mendesain tampilan dari *PagerActivity.java*, yaitu *activity_page.xml* dan *fragment_page.xml*.

fragment_page.xml berfungsi untuk mendesain tampilan utama *PagerActivity.java* yaitu peletakan *ImageView* dan *TextView* yang menampilkan tampilan dari *smartphone* dan keterangan pada gambar sedangkan *activity_page.xml* mendesain tampilan keseluruhan dari Pager Activity, halaman ini juga digunakan untuk mendesain *toolbar* yang berada di bagian bawah *Pager*.

Pada tampilan *onboarding* terdapat tiga buah halaman yang dapat di-swipe. Pada bagian toolbar terdapat *TextView* “SKIP”, *ImageView* berbentuk lingkaran berwarna hitam atau abu-abu sebagai penanda halaman yang sedang terbuka dan juga tombol “>” ataupun “finish”. *TextView* “SKIP” dapat digunakan untuk melewatkkan *PagerActivity.java* dan segera pergi ke *Activity* selanjutnya yaitu *GettingStartedActivity.java*. Pada halaman yang terbuka akan ditandai dengan lingkaran berwarna hitam, sedangkan halaman lainnya berupa lingkaran berwarna abu-abu. Perpindahan halaman tersebut dapat juga dilakukan dengan menekan *Button* “>” pada *toolbar* hingga halaman terakhir, pada halaman terakhir, *Button* “>” berubah menjadi *TextView* “finish” yang berfungsi untuk memindahkan *PagerActivity.java* menuju *GettingStartedActivity.java*. *Button* “>” dapat berubah menjadi *TextView* “finish” dengan bantuan kode yang dapat mengatur *visibility* dari setiap komponen.

Kode java sebagai berikut (*PagerActivity.java*):

```
1 package com.example.billyausten.newfinal;
2
3 import android.animation.ArgbEvaluator;
4 import android.content.Intent;
5 import android.graphics.Color;
6 import android.graphics.Typeface;
7 import android.os.Build;
8 import android.os.Bundle;
9 import android.support.design.widget.CoordinatorLayout;
10 import android.support.v4.app.Fragment;
11 import android.support.v4.app.FragmentManager;
12 import android.support.v4.app.FragmentPagerAdapter;
13 import android.support.v4.content.ContextCompat;
14 import android.support.v4.view.ViewPager;
15 import android.support.v7.app.AppCompatActivity;
16 import android.view.LayoutInflater;
17 import android.view.Menu;
18 import android.view.MenuItem;
19 import android.view.View;
20 import android.view.ViewGroup;
21 import android.widget.Button;
22 import android.widget.ImageButton;
23 import android.widget.ImageView;
24 import android.widget.TextView;
```

```

25
26 public class PagerActivity extends AppCompatActivity {
27
28     //DEKLARASI VARIABEL
29     SectionsPagerAdapter mSectionsPagerAdapter;
30     ViewPager mViewPager;
31     ImageButton mNextBtn;
32     Button mFinishBtn;
33     ImageView zero, one, two;
34     ImageView[] indicators;
35     CoordinatorLayout mCoordinator;
36     int page = 0;    // Menentukan halaman yang sedang dikerjakan
37
38     @Override
39     protected void onCreate(Bundle savedInstanceState) {
40         super.onCreate(savedInstanceState);
41
42         // Melakukan verifikasi sistem operasi yang digunakan
43         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.LOLLIPOP)
44         {
45             getWindow().getDecorView().setSystemUiVisibility(
46                 View.SYSTEM_UI_FLAG_LAYOUT_STABLE
47                 |
48             View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN);
49
50             // Melakukan setting tampilan java
51             setContentView(R.layout.activity_pager);
52
53             // Membuat adapter untuk memunculkan ketiga fragment fragment
54             mSectionsPagerAdapter = new
55             SectionsPagerAdapter(getSupportFragmentManager());
56
57             // Melakukan pendaftaran tombol ">" dengan drawable
58             mNextBtn = (ImageButton) findViewById(R.id.intro_btn_next);
59             if (Build.VERSION.SDK_INT <= Build.VERSION_CODES.LOLLIPOP)
60                 mNextBtn.setImageDrawable(
61
62                 Utils.tintMyDrawable(ContextCompat.getDrawable(this,
63                     R.drawable.ic_chevron_right_black_24dp), Color.WHITE)
64             );
65
66             // Melakukan pendaftaran tombol finish
67             mFinishBtn = (Button) findViewById(R.id.intro_btn_finish);
68
69             // Melakukan pendaftaran indikator lingkaran
70             zero = (ImageView) findViewById(R.id.intro_indicator_0);
71             one = (ImageView) findViewById(R.id.intro_indicator_1);
72             two = (ImageView) findViewById(R.id.intro_indicator_2);
73
74             mCoordinator = (CoordinatorLayout)
findViewById(R.id.main_content);
75
76             indicators = new ImageView[]{zero, one, two};

```

```

75      // Melakukan set up ViewPager terhadap adapter yang dipilih
76      // Set up the ViewPager with the sections adapter.
77      mViewPager = (ViewPager) findViewById(R.id.container);
78      mViewPager.setAdapter(mSectionsPagerAdapter);
79      mViewPager.setOffscreenPageLimit(3);
80
81      mViewPager.setCurrentItem(page);
82      updateIndicators(page);
83
84      // Melakukan inisiasi warna background untuk pager
85      final int color1 = ContextCompat.getColor(this,
R.color.grey);
86      final int color2 = ContextCompat.getColor(this,
R.color.grey);
87      final int color3 = ContextCompat.getColor(this,
R.color.grey);
88
89      final int[] colorList = new int[]{color1, color2, color3};
90
91      final ArgbEvaluator evaluator = new ArgbEvaluator();
92
93      // Pengaturan ketika halaman pager diganti
94      mViewPager.addOnPageChangeListener(new
ViewPager.OnPageChangeListener() {
95          @Override
96          public void onPageScrolled(int position, float
positionOffset, int positionOffsetPixels) {
97
98              int colorUpdate = (Integer)
evaluator.evaluate(positionOffset, colorList[position],
colorList[position == 2 ? position : position + 1]);
99              mViewPager.setBackgroundColor(colorUpdate);
100
101         }
102
103         @Override
104         public void onPageSelected(int position) {
105
106             page = position;
107
108             updateIndicators(page);
109
110             switch (position) {
111                 case 0:
112                     mViewPager.setBackgroundColor(color1);
113                     break;
114                 case 1:
115                     mViewPager.setBackgroundColor(color2);
116                     break;
117                 case 2:
118                     mViewPager.setBackgroundColor(color3);
119                     break;
120             }
121
122             mNextBtn.setVisibility(position == 2 ? View.GONE :
View.VISIBLE);
123

```

```

124             mFinishBtn.setVisibility(position == 2 ? View.VISIBLE : View.GONE);
125
126         }
127     }
128
129     @Override
130     public void onPageScrollStateChanged(int state) {
131
132     }
133 }
134
135 // Menambah jumlah halaman ketika button next ditekan
136 mNextBtn.setOnClickListener(new View.OnClickListener() {
137
138     @Override
139     public void onClick(View v) {
140
141         page += 1;
142         mViewPager.setCurrentItem(page, true);
143     }
144
145 // Keluar dari pager activity ketika button finish ditekan
146 mFinishBtn.setOnClickListener(new View.OnClickListener() {
147
148     @Override
149     public void onClick(View v) {
150
151         finish();
152     }
153 }
154
155 // Mengubah tampilan dari indikator lingkaran dibagian bawah pager
156 // ketika halaman pager diganti
157 void updateIndicators(int position) {
158
159     for (int i = 0; i < indicators.length; i++) {
160         indicators[i].setBackgroundResource(
161             i == position ? R.drawable.indicator_selected :
162             R.drawable.indicator_unselected
163         );
164     }
165 }
166
167 // Menambahkan action bar dibagian bawah pager
168 @Override
169 public boolean onCreateOptionsMenu(Menu menu) {
170     getMenuInflater().inflate(R.menu.menu_pager, menu);
171     return true;
172 }
173
174 // Melakukan kontrol dari action bar
175 @Override
176 public boolean onOptionsItemSelected(MenuItem item) {
177     int id = item.getItemId();
178     //noinspection SimplifiableIfStatement
179     if (id == R.id.action_settings) {
180         return true;
181     }

```

```

179     return super.onOptionsItemSelected(item);
180 }
181
182 // Mengatur setting tampilan java
183 public static class PlaceholderFragment extends Fragment {
184
185     private static final String ARG_SECTION_NUMBER =
186         "section_number";
187     ImageView img;
188
189     int[] bgs = new int[]{R.drawable.pager1, R.drawable.pager2,
190     R.drawable.pager3};
190     String[] args = new String[]{
191         "Select the upper Tab Bar to arrange your home
appliance by device category, room category or scenario category",
192         "Swipe from the edge of the left side of the screen
to see your profile menu",
193         "Tap on the + button to add device, room or
scenario"
194     };
195
196     public PlaceholderFragment() {
197
198         public static PlaceholderFragment newInstance(int
sectionNumber) {
199             PlaceholderFragment fragment = new
PlaceholderFragment();
200             Bundle args = new Bundle();
201             args.putInt(ARG_SECTION_NUMBER, sectionNumber);
202             fragment.setArguments(args);
203             return fragment;
204         }
205
206         @Override
207         public View onCreateView(LayoutInflater inflater, ViewGroup
container,
208                         Bundle savedInstanceState) {
209             View rootView =
inflater.inflate(R.layout.fragment_pager, container, false);
210             TextView textView = (TextView)
rootView.findViewById(R.id.section_label);
211             Typeface font_title =
Typeface.createFromAsset(getContext().getAssets(), "fonts/Quicksand-
Bold.otf");
212             textView.setTypeface(font_title);
213
214             textView.setText(args[getArguments().getInt(ARG_SECTION_NUMBER) - 1]);
215             img = (ImageView)
rootView.findViewById(R.id.section_img);
216             img.setBackgroundResource(bgs[getArguments().getInt(ARG_SECTION_NUMBER)
- 1]);
217             return rootView;
218         }
219     }
220 // Mengatur fragment pada pager

```

```

221     public class SectionsPagerAdapter extends FragmentPagerAdapter
{
222
223     public SectionsPagerAdapter(FragmentManager fm) {
224         super(fm);
225     }
226
227     @Override
228     public Fragment getItem(int position) {
229         return PlaceholderFragment.newInstance(position + 1);
230     }
231
232     @Override
233     public int getCount() {
234         return 3;
235     }
236
237     @Override
238     public CharSequence getPageTitle(int position) {
239         switch (position) {
240             case 0:
241                 return "SECTION 1";
242             case 1:
243                 return "SECTION 2";
244             case 2:
245                 return "SECTION 3";
246             }
247         }
248         return null;
249     }
250 }
251 }
```

Kode java sebagai berikut (Utils.java):

```

1 package com.example.billyausten.newfinal;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.graphics.PorterDuff;
6 import android.graphics.drawable.Drawable;
7 import android.support.v4.graphics.drawable.DrawableCompat;
8
9 public class Utils {
10
11     private static final String PREFERENCES_FILE =
"materialsample_settings";
12
13     public static int getToolbarHeight(Context context) {
14         int height = (int)
context.getResources().getDimension(R.dimen.abc_action_bar_default_height_material);
15         return height;
16     }
17
18     public static int getStatusBarHeight(Context context) {
19         int height = (int)
context.getResources().getDimension(R.dimen.statusbar_size);
```

```

20         return height;
21     }
22
23     public static Drawable tintMyDrawable(Drawable drawable, int
color) {
24         drawable = DrawableCompat.wrap(drawable);
25         DrawableCompat.setTint(drawable, color);
26         DrawableCompat.setTintMode(drawable,
PorterDuff.Mode.SRC_IN);
27         return drawable;
28     }
29
30     public static String readSharedSetting(Context ctx, String
settingName, String defaultValue) {
31         SharedPreferences sharedPref =
ctx.getSharedPreferences(PREFERENCES_FILE, Context.MODE_PRIVATE);
32         return sharedPref.getString(settingName, defaultValue);
33     }
34
35     public static void saveSharedSetting(Context ctx, String
settingName, String settingValue) {
36         SharedPreferences sharedPref =
ctx.getSharedPreferences(PREFERENCES_FILE, Context.MODE_PRIVATE);
37         SharedPreferences.Editor editor = sharedPref.edit();
38         editor.putString(settingName, settingValue);
39         editor.apply();
40     }

```

Pada kode *PagerActivity.java*, bagian yang merupakan bagian yang penting adalah:

- Baris 50 sampai 91 merupakan inisiasi dari tampilan yang akan ditampilkan.
- Baris 93 sampai 153 merupakan pengaturan halaman pager ketika *fragment*-nya diganti, baik pengaturan pager yang ditampilkan, warna dari *background*, *visibility* dari komponen pada *action bar*, *counter* jumlah halaman, dan *setting* fungsi button.
- Baris 155 sampai 180 merupakan pengaturan *action bar* dari pager, baik tampilan indikator lingkaran, penampilan *action bar* dan kontrol dari *action bar* itu sendiri
- Baris 183 sampai 218 merupakan pengaturan dari tampilan java yaitu gambar dan kata kata yang ditampilkan pada *fragment* termasuk jenis *font* yang digunakan.

Pada kode *Utils.java*, bagian yang merupakan bagian yang penting adalah:

- Baris 13 sampai 21 merupakan fungsi untuk mendapatkan dimensi tinggi dari *tool bar* maupun *status bar* pada halaman pager.
- Baris 23 sampai 28 merupakan fungsi untuk mengatur warna pada drawable yang digunakan

- Baris 30 sampai 40 merupakan fungsi untuk membaca dan menulis *shared preferences* untuk menyimpan data pada memori *smartphone*.