



# 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	<b>TUGAS AKHIR TEKNIK ELEKTRO:</b> <i>Perancangan Sistem Kecerdasan dan Keamanan pada Perangkat Internet-of-Things Rumah Cerdas</i>
Jenis Dokumen	<b>PERANCANGAN</b> <small>Catatan: Dokumen ini dikendalikan penyebarannya oleh Prodi Teknik Elektro ITB</small>
Nomor Dokumen	<b>B300-02-TA1617.01.056</b>
Nomor Revisi	<b>02</b>
Nama File	<b>B300</b>
Tanggal Penerbitan	<b>2 Mei 2017</b>
Unit Penerbit	<b>Prodi Teknik Elektro - ITB</b>
Jumlah Halaman	<b>94</b> (termasuk lembar sampul ini)

Data Pengusul				
Ditulis Oleh	Nama	Bryan Tandiawan	Jabatan	Anggota Kelompok
	Tanggal	02 Mei 2017	Tanda Tangan	
	Nama	Billy Austen	Jabatan	Anggota Kelompok
	Tanggal	02 Mei 2017	Tanda Tangan	
	Nama	Revie Marthensa	Jabatan	Anggota Kelompok
	Tanggal	02 Mei 2017	Tanda Tangan	
Disetujui Oleh	Nama	Trio Adiono, S.T., M.T., Ph.D.	Jabatan	Pembimbing
	Tanggal	02 Mei 2017	Tanda Tangan	

## 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.....	5
1.5 DAFTAR GAMBAR.....	6
1.6 DAFTAR TABEL .....	8
2. PENDAHULUAN DAN RUANG LINGKUP KERJA.....	10
2.1 PENDAHULUAN.....	10
2.2 RUANG LINGKUP KERJA PROYEK.....	11
3. DESKRIPSI DAN PEMILIHAN ELEMEN-ELEMEN DALAM SISTEM .....	13
3.1 SISTEM INDOOR .....	13
3.1.1 Access Point .....	13
3.1.2 Host .....	14
3.1.3 Node .....	17
3.1.4 Aktuator .....	20
3.2 SISTEM OUTDOOR .....	21
3.2.1 Server .....	21
3.2.2 Mobile Application .....	22
3.2.3 Pemilihan Protokol Komunikasi .....	22
4. PERANCANGAN SUBSISTEM INDOOR DAN OUTDOOR.....	24
4.1 PENDAHULUAN.....	24
4.2 PERANCANGAN SUBSISTEM INDOOR .....	25
4.2.1 Perancangan Sistem Komunikasi dalam Subsistem Indoor.....	25
4.2.2 Perancangan Remote Controller .....	28
4.2.2.1 Perancangan Hardware Remote Contro .....	28

4.2.2.2	<i>Perancangan Software Remote Control</i> .....	29
4.3	PERANCANGAN SUBSISTEM OUTDOOR.....	30
4.3.1	Perancangan Basis Data di Server dan Host .....	31
4.3.1.1	<i>Basis Data di Server Pusat</i> .....	31
4.3.1.2	<i>Basis Data di Host</i> .....	33
4.3.2	Keamanan Jaringan .....	36
4.3.3.	Perancangan Mekanisme Antrian Data Dalam Protokol AMQP .....	39
4.3.4	Perancangan Paket Data untuk Komunikasi pada Subsistem Outdoor .....	42
4.3.5	Perancangan Aplikasi Android .....	44
4.3.6	Perancangan Program pada Server.....	83
4.3.7	Perancangan Program pada Host .....	89

## Catatan Sejarah Perbaikan Dokumen

VERSI	TANGGAL	PENYUNTING	PERBAIKAN
2	1 Mei 2017	Billy	Revisi penjelasan pada bagian <i>mobile application</i>
		Bryan	Penambahan penjelasan protokol <i>hardware</i> dan perancangan <i>remote control</i>
		Revie	Penambahan penjelasan skema komunikasi, <i>back end</i> pada aplikasi, dan penjelasan program pada server dan <i>host</i>

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

## 1. Pengantar

### 1.1 RINGKASAN ISI DOKUMEN

B300 ini berisi desain proyek rumah cerdas berbasis IoT. Dokumen ini merupakan perkembangan dari dokumen sebelumnya yaitu B100 dan B200. Pada dokumen ini, dijelaskan perancangan sistem rumah cerdas secara spesifik dan mendetail dari segi *software*, *hardware* maupun komunikasi antara *software* dan *hardware*. Dijabarkan juga Data Flow Diagram (DFD) maupun Flowchart dari setiap unit pada sistem rumah cerdas untuk mempermudah pemahaman isi dokumen.

### 1.2 TUJUAN PENULISAN DAN APLIKASI/KEGUNAAN DOKUMEN

Tujuan penulisan dokumen ini adalah:

1. Memberikan gambaran dan menjelaskan sistem rumah cerdas berbasis IoT secara spesifik dan mendetail.
2. Sebagai desain acuan dalam merancang sistem rumah cerdas.
3. Sebagai dokumentasi tugas akhir.

### 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. <https://www.rabbitmq.com/tutorials/>

## 1.5 DAFTAR GAMBAR

Gambar 1 Desain Blok Sistem Rumah Cerdas .....	11
Gambar 2 Block Diagram <i>Access Point</i> .....	14
Gambar 3 Diagram Block Raspberry Pi .....	17
Gambar 4 Mesh Topology .....	19
Gambar 5 Block Diagram Modul Zigbee .....	20
Gambar 6 Block Diagram STM32.....	21
Gambar 7 Skema Komunikasi RabbitMQ yang Paling Sederhana .....	23
Gambar 8 DFD Sistem Level 0.....	25
Gambar 9 DFD Sistem Indoor Level 1 .....	25
Gambar 10 DFD Sistem Indoor Level 2 .....	25
Gambar 11 Struktur perangkat keras <i>remote</i> IrDA-based .....	29
Gambar 12 Diagram alir perangkat <i>remote</i> .....	30
Gambar 13 DFD Sistem Outdoor Level 1 .....	30
Gambar 14 DFD Sistem Outdoor Level 2 .....	31
Gambar 15 Skema Penggabungan Enkripsi Simetrik dan Asimetrik .....	38
Gambar 16 Skema RPC .....	41
Gambar 17 Halaman Utama.....	45
Gambar 18 Halaman Sign In.....	46
Gambar 19 Diagram Alir Proses Sign In .....	47
Gambar 20 Halaman Forget Password .....	48
Gambar 21 Halaman Sign Up .....	49
Gambar 22 Diagram Alir Sign Up .....	51
Gambar 23 Dashboard .....	52
Gambar 24 Dashboard Navigation Drawer.....	53
Gambar 25 Keadaan dashboard sedang terkunci .....	54

Gambar 26 Dashboard Room.....	55
Gambar 27 Dashboard Scenario .....	56
Gambar 28 Tombol Floating Button.....	58
Gambar 29 Menu curtain .....	59
Gambar 30 Menu door.....	59
Gambar 31 Menu fan .....	59
Gambar 32 Menu lamp .....	59
Gambar 33 Menu switch.....	59
Gambar 34 Menu temperature .....	59
Gambar 35 Halaman Room .....	62
Gambar 36 Halaman Add Device .....	63
Gambar 37 Halaman Add Room.....	65
Gambar 38 Halaman Add Scenario .....	66
Gambar 39 Halaman Notifikasi .....	68
Gambar 40 Halaman Pesan.....	68
Gambar 41 Halaman Analisis Data.....	69
Gambar 42 Halaman Member.....	71
Gambar 43 Halaman Add Member.....	73
Gambar 44 Diagram Alir Register .....	74
Gambar 45 Halaman Getting Started.....	75
Gambar 46 Halaman Get Invited .....	76
Gambar 47 Diagram Alir Proses Pendaftaran Seorang <i>User</i> ke Suatu Rumah .....	77
Gambar 48 Halaman Setting.....	78
Gambar 49 Halaman User.....	80
Gambar 50 Halaman Create New .....	80

Gambar 51 Halaman About .....	82
Gambar 52 Halaman Help .....	83

## 1.6 DAFTAR TABEL

Tabel 1 Tabel Perbandingan Arduino Uno, Raspberry Pi, dan BeagleBone .....	16
Tabel 2 Perbandingan ZigBee, Bluetooth, dan WiFi [1] .....	17
Tabel 3 Susunan Paket Data yang Dialirkan pada Subsistem Indoor .....	27
Tabel 4 Jenis Aktuator Beserta Interpretasi Data Payload yang Dikirim oleh Host.....	28
Tabel 5 Kolom-kolom Tabel users di <i>database</i> general_data .....	32
Tabel 6 Kolom-kolom Tabel users di <i>database</i> general_data .....	32
Tabel 7 Kolom-kolom Tabel info di <i>database</i> homeXXXXXX .....	33
Tabel 8 Kolom-kolom Tabel devices di <i>database</i> homeXXXXXX.....	33
Tabel 9 Kolom-kolom Tabel rooms di <i>database</i> homeXXXXXX.....	34
Tabel 10 Kolom-kolom Tabel scenarios di <i>database</i> homeXXXXXX.....	34
Tabel 11 Kolom-kolom Tabel dataMMYYYY di <i>database</i> homeXXXXXX.....	35
Tabel 12 Kolom-kolom Tabel users di <i>database</i> homeXXXXXX.....	35
Tabel 13 Daftar Header Pesan .....	44
Tabel 14 Data JSON Sign In.....	46
Tabel 15 Data JSON Forget Password.....	49
Tabel 16 Data JSON Sign In.....	50
Tabel 17 Data JSON Delete dan Edit Room.....	55
Tabel 18 Data JSON Edit dan Delete Scenario .....	56
Tabel 19 Data JSON Pengaktifan Skenario .....	57
Tabel 20 Data JSON Loading Data dari Host.....	58
Tabel 21 Data JSON untuk Perintah ke Perangkat .....	60
Tabel 22 Data JSON Delete dan Edit Perangkat .....	61



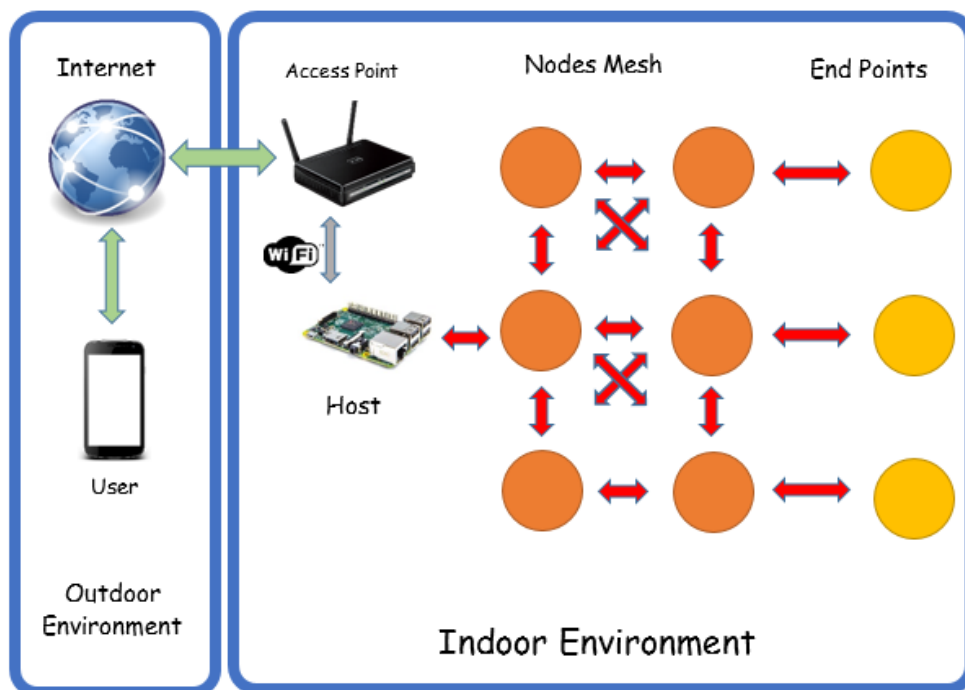
Tabel 23 Data JSON Add Device .....	64
Tabel 24 Data JSON Add Room.....	65
Tabel 25 Data JSON Add Scenario .....	67
Tabel 26 Data JSON Permintaan Data Penggunaan Perangkat .....	70
Tabel 27 Data JSON Permintaan List User .....	72
Tabel 28 Data JSON untuk Permintaan Register.....	74
Tabel 29 Data JSON untuk Konfirmasi User .....	77
Tabel 30 Data JSON Edit Profil <i>User</i> .....	79
Tabel 31 Data JSON untuk <i>List</i> Data Rumah .....	81
Tabel 32 Data JSON untuk Edit Data Rumah .....	81
Tabel 33 Data JSON untuk Edit <i>Privilege</i> Perangkat.....	82
Tabel 34 Penjelasan Paket Data .....	84
Tabel 35 Rangkuman Pelayanan Host .....	90

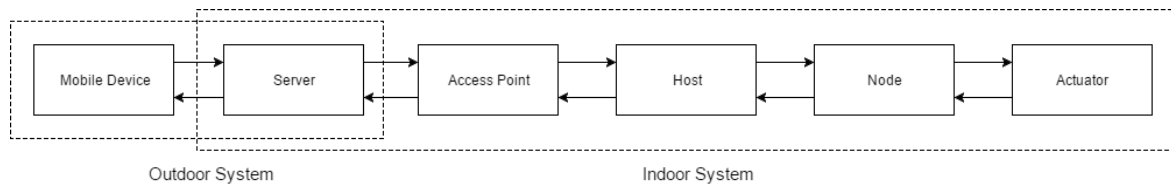
## 2. Pendahuluan dan Ruang Lingkup Kerja

### 2.1 Pendahuluan

Sistem rumah cerdas yang dirancang ini menggunakan konsep *Internet of Things* sehingga pengguna dapat melakukan monitoring dan mengendalikan sistem menggunakan *mobile application* yang terhubung dengan koneksi internet. Sistem rumah cerdas dibagi menjadi dua subsistem yaitu sistem *indoor* dan sistem *outdoor*. Sistem *indoor* berisi jaringan nirkabel yang mengintegrasikan perangkat-perangkat keras di dalam rumah. Ada beberapa elemen dalam sistem *indoor*, yaitu *Access Point*, *Host*, dan *Node* yang terhubung ke perangkat-perangkat keras aktuator di dalam rumah (*end-device*).

Lingkungan *outdoor* terdiri dari *cloud* dan *mobile device* (ponsel). *Cloud* merupakan server sistem rumah cerdas. Server akan melayani permintaan data yang datang dari pengguna. Server juga merupakan pusat dari keseluruhan data sistem, baik itu data pengguna, maupun data setiap rumah yang terdaftar pada sistem. *Mobile device* merupakan penghubung antara pengguna dengan sistem rumah cerdas. Setiap *mobile device* pengguna akan dipasang suatu aplikasi. Melalui aplikasi ini, pengguna dapat mengakses sistem dengan mudah dan sederhana.





**Gambar 1 Desain Blok Sistem Rumah Cerdas**

## 2.2 Ruang Lingkup Kerja Proyek

Seperti yang telah disebutkan pada dokumen B200, proyek ini merupakan kelanjutan dari proyek yang telah dibangun oleh tim dari Pusat Mikroelektronika ITB yang dikepalai oleh Bapak Trio Adiono, Ph.D. Pada proyek sebelumnya, tim sudah berhasil mengimplementasikan beberapa jenis perangkat keras cerdas sebagai *node*, yaitu:

1. Lampu LED RGB, yaitu perangkat lampu yang bisa menyala dengan berbagai macam warna mengikuti komposisi warna RGB.
2. *Power switcher* (sakelar), yaitu perangkat untuk memutus dan menyambungkan arus listrik.
3. Kipas angin, yaitu perangkat kipas angin yang kecepatan rotasinya bisa diatur, mulai dari 0% sampai 100%
4. Kunci pintu, yaitu perangkat kunci pintu otomatis yang dikendalikan secara elektrik melalui solenoida.
5. Sensor suhu dan kelembapan, yaitu perangkat pengukur suhu dan kelembapan di suatu ruangan.
6. Pengendali tirai, yaitu perangkat yang dapat mengendalikan posisi tirai, apakah ingin dinaikan atau diturunkan.
7. *Host* yang berfungsi sebagai koordinator keseluruhan perangkat dan sebagai sumber perintah yang akan disampaikan kepada setiap perangkat.

Semua jenis perangkat di atas sudah bisa berkomunikasi dengan topologi jaringan *mesh*. Keunggulan jaringan berbentuk *mesh* adalah skema komunikasi yang fleksibel. Jalur komunikasi tidak harus selalu dari *host* ke *node* seperti pada topologi *star*, namun data bisa datang dari *node* manapun dan bisa disampaikan kepada elemen apapun di dalam sistem. Dengan konsep demikian, data yang diberikan oleh *host* bisa saja tidak dikirimkan langsung kepada *node* tertuju, namun disampaikan dulu kepada *node* perantara yang jaraknya lebih dekat. Setelah itu, barulah data diteruskan kepada *node* yang ingin dituju. Konsep perantaraan data ini membuat jangkauan komunikasi sistem bisa menjadi sangat luas.

Protokol komunikasi yang digunakan adalah protokol Zigbee. Zigbee merupakan protokol komunikasi nirkabel menggunakan sinyal *Radio Frequency* (RF) yang terdaftar sebagai IEEE 802.15.4. Zigbee dirancang untuk memfasilitasi *personal area network* dengan ruang kerja yang terbatas. Keunggulan yang dimiliki oleh Zigbee adalah konsumsinya yang sangat rendah dibandingkan dengan beberapa protokol komunikasi pada umumnya seperti Bluetooth (IEEE 802.15.1) dan WiFi (IEEE 802.11). Namun, *trade off* yang dialami adalah rendahnya *data rate* selama komunikasi berlangsung. Dengan keadaan demikian, Zigbee sangat cocok untuk jaringan komunikasi yang tidak memerlukan *data rate* yang tinggi. Sistem rumah cerdas merupakan salah satu sistem yang cocok menggunakan protokol ini karena *data rate* komunikasi yang diperlukan tidak besar. Sistem rumah cerdas hanya perlu mengkomunikasikan data-data perintah dan data status pada setiap perangkat.

Tim pengembang sudah membuat aplikasi Android sederhana untuk mengakses sistem. Aplikasi ini baru mewadahi fitur-fitur sederhana dalam sistem, misalnya menyalakan atau mematikan perangkat. Akses ke sistem juga baru bisa dilakukan melalui jaringan Bluetooth. Dengan demikian, akses hanya bisa dilakukan dari jarak yang sangat pendek, yaitu hanya puluhan meter saja.

Untuk melanjutkan apa yang telah dicapai di atas, maka ruang lingkup kerja Proyek Tugas Akhir ini adalah sebagai berikut.

1. Perancangan aplikasi Android yang lebih mewadahi fitur-fitur sistem rumah cerdas yang lebih lengkap dan mudah digunakan sehingga menarik bagi pengguna dan bisa mencapai tujuan awal, yaitu membantu penghuni rumah dalam mengendalikan perangkat-perangkat di dalam rumahnya.
2. Perancangan *cloud server* untuk menjadi pelayan sekaligus penyimpanan data seluruh rumah yang terdaftar. Hal yang perlu dipersiapkan adalah program yang akan dijalankan di komputer server dan struktur basis data (*database*).
3. Penambahan satu jenis perangkat keras, yaitu *remote* untuk menyalakan atau mematikan AC maupun televisi. *Remote* ini harus bisa terhubung dengan jaringan Zigbee bersama-sama dengan perangkat-perangkat lain yang telah diimplementasikan sebelumnya oleh tim dari PME.
4. Perancangan protokol komunikasi antar perangkat, baik di daerah *indoor* maupun *outdoor*. Protokol komunikasi ini harus efisien dan bisa mendukung seluruh fitur dalam sistem.

5. Perancangan sistem keamanan jaringan yang melindungi sistem dari serangan-serangan pihak-pihak tidak bertanggung jawab.

Dengan demikian, tujuan akhir dari Proyek Tugas Akhir ini adalah membuat sistem rumah cerdas yang terhubung ke jaringan internet, memiliki *cloud server*, serta bisa diakses melalui aplikasi Android.

### **3. Deskripsi dan Pemilihan Elemen-elemen Dalam Sistem**

#### **3.1 Sistem Indoor**

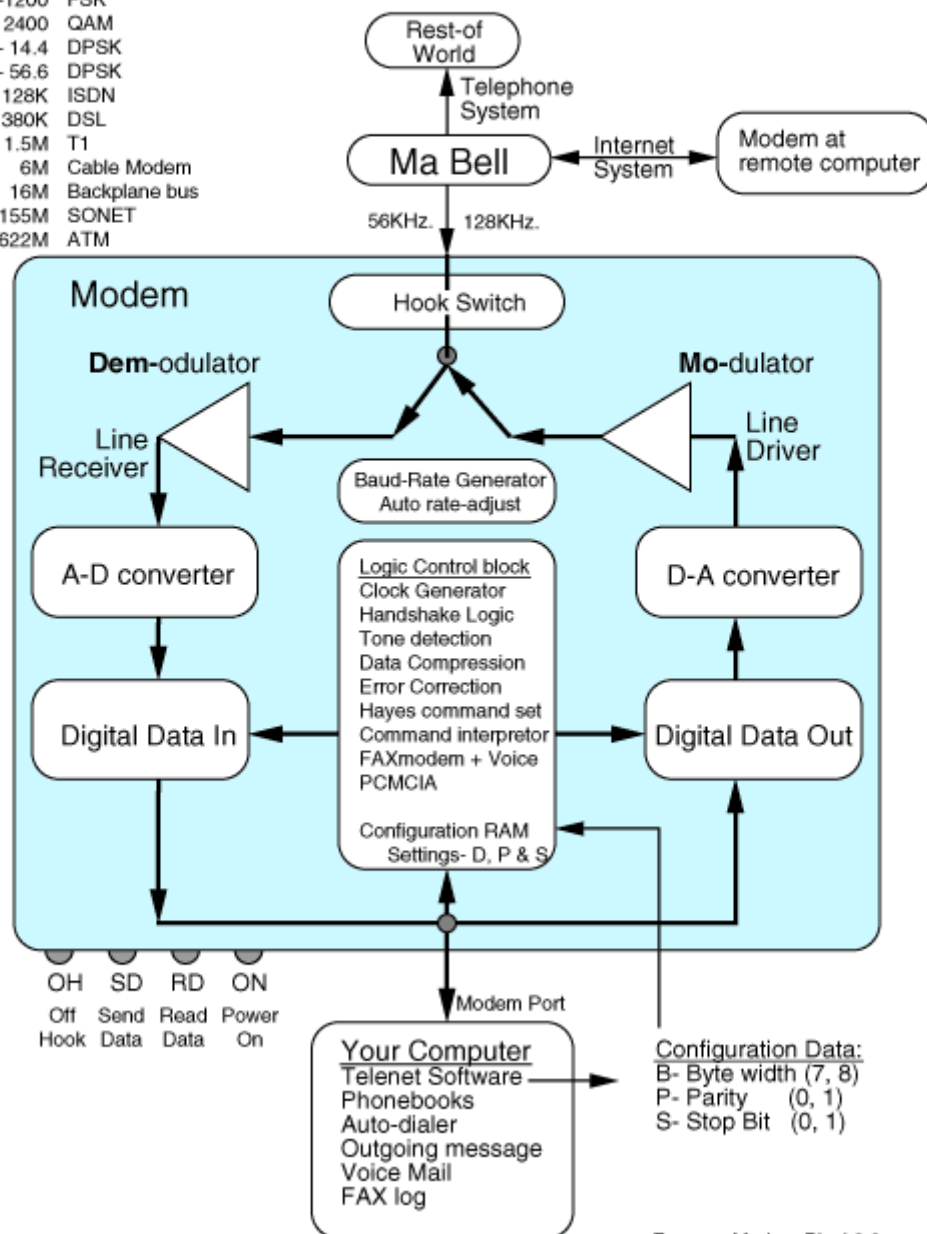
Sistem indoor berisi *Access Point*, *host*, *node* dan aktuator.

##### **3.1.1 Access Point**

*Access point* merupakan jembatan antara *host* dengan server. *Access Point* yang digunakan adalah perangkat modem yang dapat menyediakan jaringan internet bagi *host*. *Host* bisa dihubungkan ke *access point* melalui jaringan nirkabel (WiFi) maupun melalui saluran berkabel, misalnya kabel ethernet. Jenis modem yang digunakan di dalam sistem sangat fleksibel dan tergantung kepada penghuni rumah. Selama modem berfungsi dengan baik untuk menghubungkan *host* ke internet, maka modem tersebut dapat digunakan. Salah satu diagram blok modem yang umum digunakan adalah sebagai berikut.

**Modem Block Diagram**Baud Rate Modulation Scheme

300 -1200	FSK
2400	QAM
9.6 - 14.4	DPSK
28.8 - 56.6	DPSK
128K	ISDN
380K	DSL
1.5M	T1
6M	Cable Modem
16M	Backplane bus
155M	SONET
622M	ATM



Fname= Modem Block2.8

**2****Gambar 2 Block Diagram Access Point****3.1.2 Host**

Seperti yang sebelumnya sudah disinggung, *host* adalah koordinator seluruh perangkat *end-device* yang ada di dalam rumah. Selain sebagai koordinator dan pemberi perintah, *host* juga berfungsi sebagai penyimpan data dan status *realtime* setiap perangkat.

Dengan demikian, *host* juga memiliki basis data sederhana untuk menyimpan informasi-informasi *realtime*.

Untuk memenuhi kebutuhan yang ada di dalam sistem, *host* dituntut dapat memenuhi beberapa kriteria berikut.

1. *Host* diharapkan memiliki modul komunikasi serial. Komunikasi serial ini diperlukan supaya *host* dapat dipasang modul untuk komunikasi Zigbee. Dengan demikian, *host* akan terhubung dengan keseluruhan perangkat-perangkat di dalam rumah.
2. *Host* diharapkan memiliki kemampuan komputasi yang cukup baik dan dapat menjalankan beberapa program tertentu secara bersamaan. Hal ini diperlukan karena *host* tidak hanya berfungsi sebagai pemberi perintah kepada perangkat, namun juga sebagai penghubung sistem *indoor* dengan sistem *outdoor*. *Host* akan menerima banyak data dari sistem *outdoor* dan terkadang data-data tersebut harus diolah. Dengan besarnya tanggung jawab *host*, *host* lebih baik memiliki kemampuan *multitasking* untuk mengeksekusi beberapa program secara paralel.
3. *Host* juga diharapkan memiliki modul komunikasi ke jaringan luar, baik itu berkabel maupun nirkabel. Hal ini diperlukan supaya *host* bisa dihubungkan dengan *access point* sehingga terhubung ke internet.
4. *Host* diharapkan memiliki fasilitas penyimpanan data (*Data Storage*) karena fungsinya sebagai penyimpan data *realtime*.
5. *Host* juga diharapkan memiliki harga yang bersaing dan tidak terlalu mahal sehingga bisa meminimalisir biaya pemasangan sistem.

Ada beberapa alternatif perangkat yang bisa dijadikan *host*, yaitu Arduino Uno, Raspberry Pi 3, dan BeagleBone. Berikut adalah perbandingan ketiga perangkat tersebut yang diperoleh dari situs <http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/>.

Tabel 1 Tabel Perbandingan Arduino Uno, Raspberry Pi, dan BeagleBone

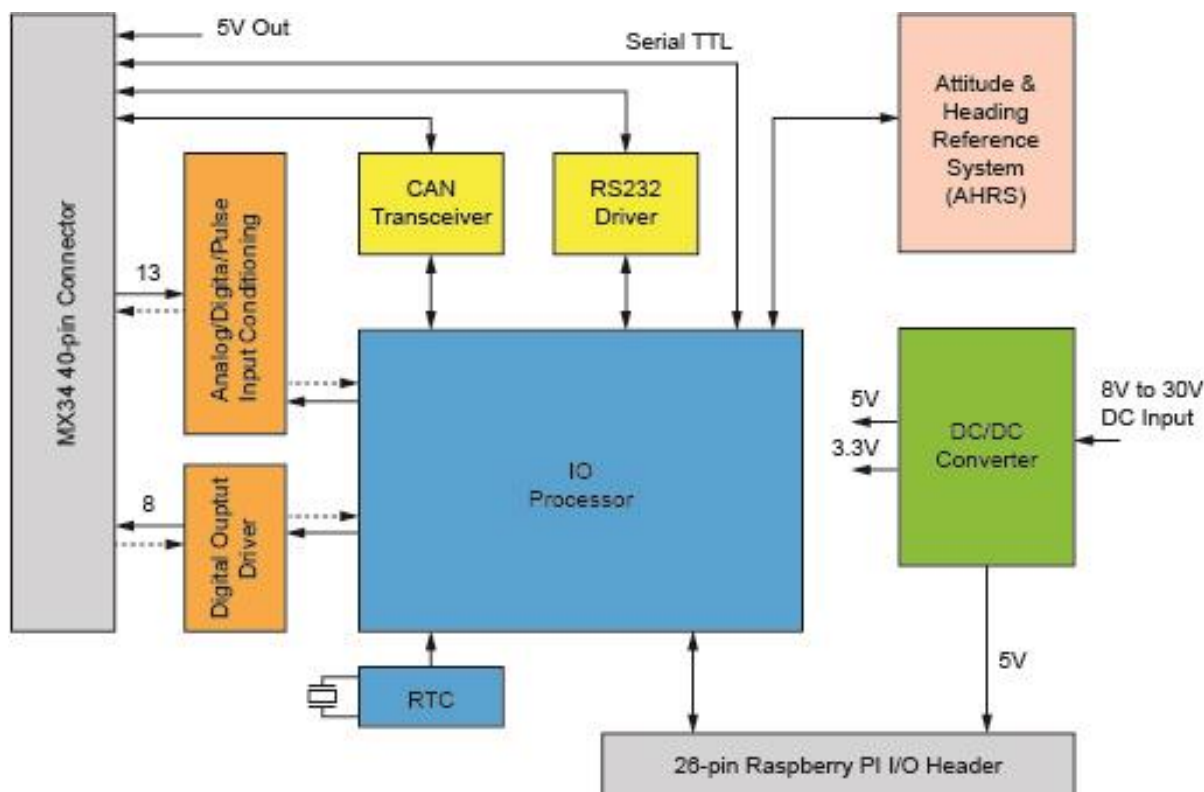
Name	Arduino Uno	Raspberry Pi	BeagleBone
Model Tested	R3	Model B	Rev A5
Price	\$29.95	\$35	\$89
Size	2.95"x2.10"	3.37"x2.125"	3.4"x2.1"
Processor	ATMega 328	ARM11	ARM Cortex-A8
Clock Speed	16MHz	700MHz	700MHz
RAM	2KB	256MB	256MB
Flash	32KB	(SD Card)	4GB(microSD)
EEPROM	1KB		
Input Voltage	7-12v	5v	5v
Min Power	42mA (.3W)	700mA (3.5W)	170mA (.85W)
Digital GPIO	14	8	66
Analog Input	6 10-bit	N/A	7 12-bit
PWM	6		8
TWI/I2C	2	1	2
SPI	1	1	1
UART	1	1	5
Dev IDE	Arduino Tool	IDLE, Scratch, Squeak/Linux	Python, Scratch, Squeak, Cloud9/Linux
Ethernet	N/A	10/100	10/100
USB Master	N/A	2 USB 2.0	1 USB 2.0
Video Out	N/A	HDMI, Composite	N/A
Audio Output	N/A	HDMI, Analog	Analog

Arduino Uno memang memiliki harga yang paling murah dibandingkan dengan yang lain. Namun, perangkat ini tidak memfasilitasi fitur *multitasking* karena hanya berupa mikrokontroler. Kemampuan komputasi dan memori perangkat ini juga termasuk yang paling rendah dibandingkan yang lain. BeagleBone merupakan perangkat yang memiliki harga paling mahal. Namun, bila dilihat kemampuan komputasi dan memorinya, BeagleBone memiliki performa yang relatif sama dengan Raspberry Pi 3. Perbedaan paling mencolok antara Raspberry Pi 3 dan BeagleBone adalah jumlah pin *General Purpose Input Output* (GPIO). Raspberry Pi 3 hanya memiliki pin GPIO sebanyak 8 buah saja, sedangkan BeagleBone 66 buah. Namun, sistem rumah cerdas yang hendak dibangun tidak memerlukan jumlah pin yang banyak. Dengan demikian, jenis perangkat yang dipilih untuk dijadikan *host* adalah Raspberry Pi 3 karena harganya yang jauh lebih murah dan tetap sesuai dengan kriteria yang diinginkan.

Berikut adalah diagram blok dari Raspberry Pi 3.

Nomor Dokumen: B300-01-TA1617.01.056	Nomor Revisi: 00	Tanggal: 02/05/2017	Halaman 16 dari 94
--------------------------------------	------------------	---------------------	--------------------





**Gambar 3 Diagram Block Raspberry Pi**

### 3.1.3 Node

*Node* digunakan agar setiap perangkat rumah cerdas dapat menerima informasi dari *user*. *Node* merupakan modul Zigbee yang terpasang pada setiap peralatan sehingga setiap peralatan dapat membangun jaringan untuk berkomunikasi. Protokol ini telah ditentukan oleh tim dari PME dengan beberapa pertimbangan. Ada beberapa protokol yang bisa digunakan, yaitu Bluetooth, Zigbee, dan WiFi. Perbandingan ketiga teknologi tersebut dijelaskan dalam tabel berikut.

**Tabel 2 Perbandingan ZigBee, Bluetooth, dan WiFi [1]**

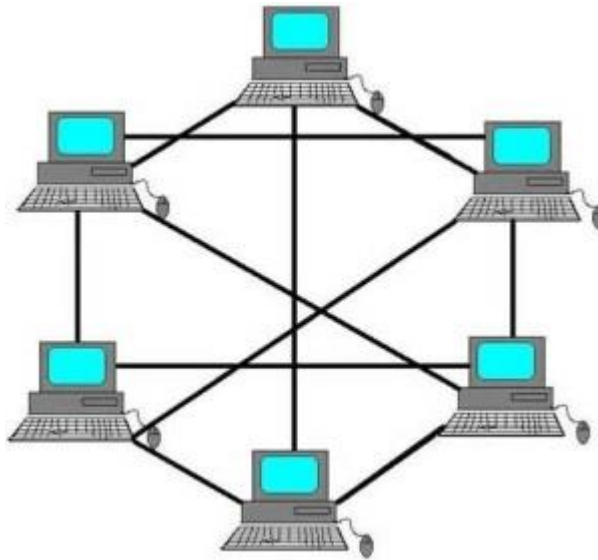
Fitur	ZigBee	Bluetooth	WiFi
Kompleksitas	Simpel	Kompleks	Sangat Kompleks
Jangkauan	300 m	10 m	100 m

<i>Data Rate</i>	250 kbps	1 Mbps	11 Mbps
<b>Konsumsi Daya</b>	Rendah	Menengah	Tinggi
<b>Jumlah Node</b>	6400	7	32

Masing-masing protokol di atas memiliki kelemahan dan kelebihan masing-masing. ZigBee memiliki kelebihan dalam hal kesederhanaan struktur data dan jangkauan, namun memiliki kelemahan pada *data rate*-nya. Dengan demikian, ZigBee adalah teknologi yang cocok untuk diterapkan pada sistem yang dituntut memiliki jangkauan cukup luas namun aliran data yang terjadi dalam sistem tidak terlalu besar. Karena rendahnya *data rate*, maka ZigBee mengkonsumsi daya yang tidak terlalu besar. Bluetooth memiliki keunggulan pada *data rate* dan kompatibilitas ketika dihubungkan dengan *smartphones*. Dengan demikian, Bluetooth cocok digunakan untuk aplikasi *smartphone* yang membutuhkan *data rate* menengah. WiFi memiliki keunggulan pada *data rate*-nya. *Data rate* pada WiFi adalah yang tertinggi bila dibandingkan dengan dua teknologi lainnya. Dengan demikian, WiFi cocok untuk digunakan pada sistem yang memiliki aliran data yang tinggi (biasanya sistem yang memerlukan transmisi media seperti video, musik, dan sebagainya) [2].

Dari analisis di atas, teknologi yang dipilih untuk sistem rumah cerdas ini adalah ZigBee. Pemilihan ini cenderung didasarkan pada kebutuhan *data rate*. Untuk *end-device*, kecepatan transfer data yang diperlukan sangat kecil karena data yang dikirimkan hanya berupa perintah-perintah atau informasi tertentu. Dengan demikian, teknologi ZigBee yang memiliki kesederhanaan dalam struktur datanya dan konsumsi energinya yang rendah adalah yang paling cocok di antara ketiga teknologi di atas. Selain itu, ZigBee juga dapat menangani jumlah *node* yang terbilang banyak. Hal ini sesuai dengan yang diperlukan pada sistem rumah cerdas yang dirancang ini.

Modul untuk komunikasi Zigbee sudah dipilih oleh tim pengembang sebelumnya, yaitu modul XBee Pro. Modul ini memiliki performa yang sangat baik dalam mengkomunikasikan data. Modul ini memfasilitasi topologi jaringan *mesh* sehingga setiap elemen dalam jaringan bisa mengirimkan dan menerima data.

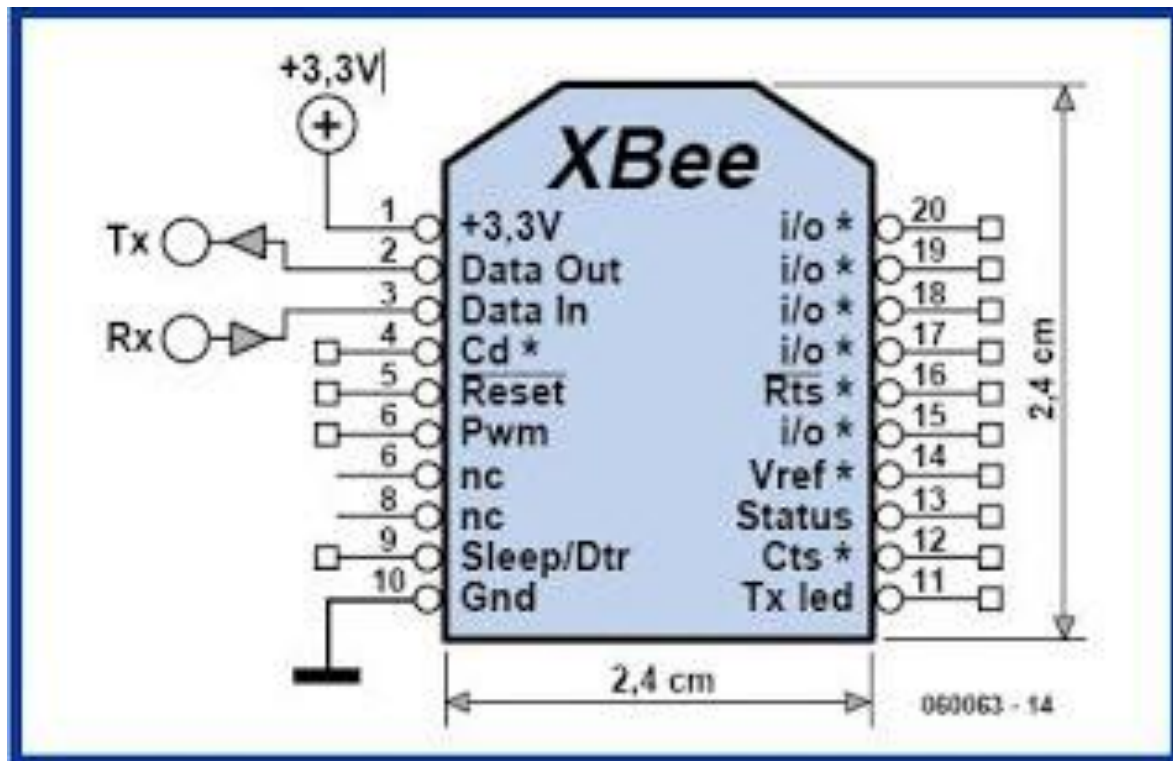


**Gambar 4 Mesh Topology**

(sumber:<http://www.networking-basics.net/wp-content/uploads/2014/09/full-mesh-topology-300x279.jpg>)

Setiap modul XBee memiliki alamat sebagai identitas yang unik dan membedakan satu modul dengan modul yang lain. Dengan demikian, hal yang penting dalam komunikasi adalah alamat *node* yang hendak dituju. Dengan mengetahui informasi tersebut, maka kita dapat mengirimkan data kepada *node* tersebut.

Berikut adalah penjelasan pin-pin dalam modul XBee Pro.

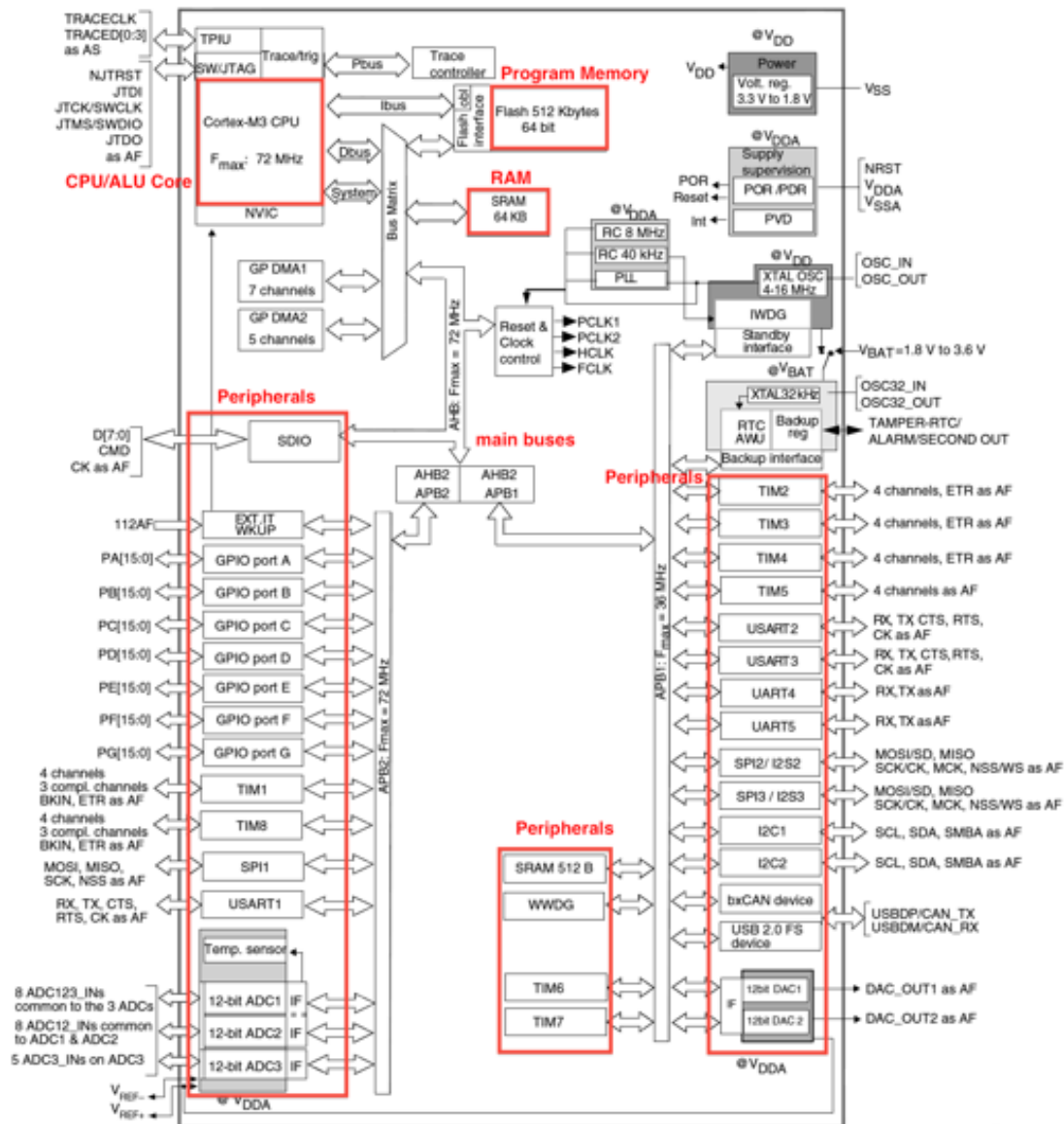


**Gambar 5 Block Diagram Modul Zigbee**

(Sumber: <https://www.elprocus.com/wp-content/uploads/2015/01/Zigbee-Pin-Diagram.jpg>)

### 3.1.4 Aktuator

Aktuator dalam sistem ini adalah perangkat-perangkat keras di dalam rumah. Ketika *host* memberikan perintah kepada *node* mengenai perubahan status sebuah *device*, maka *node* segera meneruskan perintah tersebut kepada aktuator pada *device* tersebut. Aktuator harus bisa menerjemahkan perintah tersebut dan mengaktualisasikannya sesuai dengan jenis aktuator tersebut. Untuk bisa menerjemahkan pesan, maka digunakan mikrokontroler STM32. Pemilihan ini telah dilakukan oleh tim pengembang dari PME ketika perancangan awal. Jika dibandingkan dengan mikrokontroler ATmega, STM32 memiliki harga yang lebih rendah dan lebih hemat energi walaupun penggunaannya tergolong lebih sulit.



Gambar 6 Block Diagram STM32

(Sumber: [http://avi.hobby-site.org:8888/wiki/images/e/ec/Stm32\\_block\\_diagram.png](http://avi.hobby-site.org:8888/wiki/images/e/ec/Stm32_block_diagram.png))

## 3.2 Sistem Outdoor

Sistem outdoor berisi *server* dan *mobile application*.

### 3.2.1 Server

Server merupakan pusat penyimpanan seluruh data sistem. Data yang perlu disimpan di server adalah informasi pengguna (*user information*) seperti *email*, *user name*, *password*,

*home ID*, dan lain-lain (akan dijelaskan secara mendetail pada Bab Basis Data). Selain itu, data yang perlu disimpan adalah informasi mengenai perangkat di dalam rumah yang meliputi daftar perangkat (*devices list*), daftar ruangan (*rooms*), daftar skenario (*scenarios*), dan lain-lain. Seluruh data tersebut disimpan dalam basis data. Untuk mengimplementasikan basis data, diperlukan suatu *environment* tertentu dalam pembuatan basis data tersebut. Ada beberapa alternatif yang bisa dipilih, yaitu:

1. MySQL
2. SQLite

Dari hasil penelusuran di banyak situs, diperoleh simpulan bahwa MySQL merupakan *environment* yang paling banyak digunakan sehingga memiliki komunitas yang besar. Komunitas yang besar sangat berguna ketika kita mengalami kesulitan selama proses implementasi karena ada banyak forum yang bisa menjadi sarana untuk bertanya. Dengan demikian, *environment* yang dipilih untuk pembuatan basis data adalah **MySQL**.

### **3.2.2 Mobile Application**

Aplikasi yang dibuat menggunakan *operating system* Android dengan alasan bahwa *operating system* Android lebih banyak digunakan ketimbang *operating system* IOS maupun yang lainnya. Pembuatan *mobile application* menggunakan platform Android Studio 2.2. Dipilih Android Studio 2.2 dibandingkan dengan Eclipse dan Visual Studio karena Android Studio 2.2 memiliki pengaturan *layout* program yang lebih mudah dibandingkan Eclipse dan Android Studio memiliki komunitas yang lebih besar dibandingkan dengan Visual Studio sehingga mempermudah pemrograman *mobile application*.

### **3.2.3 Pemilihan Protokol Komunikasi**

Subsistem outdoor berhubungan dengan dunia luar, yaitu internet. Untuk bisa berkomunikasi di internet, ada dua protokol yang dapat menjadi alternatif untuk berkomunikasi.

1. *Hyper Text Transfer Protocol* (HTTP)

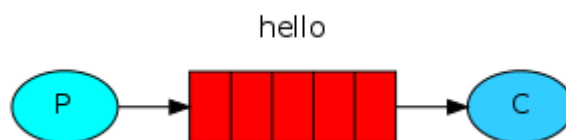
HTTP merupakan protokol komunikasi antara server dan client yang paling umum digunakan. Pada protokol ini, client melakukan request ke server, lalu server akan memberikan response sesuai dengan request yang diberikan oleh server.

Keunggulan protokol ini adalah kemudahan dan kepraktisan penggunaannya karena sudah banyak pihak yang membuat library untuk protokol ini. Kelemahan dari protokol ini adalah client harus melakukan request terlebih dahulu sebelum server mengirimkan data ke client. Dengan kata lain, pada protokol ini tidak ada mekanisme server secara spontan mengirimkan data ke client. Padahal, mekanisme tersebut diperlukan pada sistem rumah cerdas ini. Server harus dapat mengirimkan data tertentu (terutama kepada subsistem indoor) tanpa client mengirimkan permintaan.

## 2. *Advance Message Queueing Protocol (AMQP)*

Sesuai dengan namanya, protokol ini dilengkapi dengan suatu antrian yang dapat menampung sejumlah data tertentu. Dengan demikian, protokol cocok diterapkan pada server yang menangani cukup banyak client. Pada protokol ini, server hanya bertindak sebagai broker pesan, bukan bertindak langsung sebagai pelayan atau pemberi jawaban kepada client. Dengan demikian, skema komunikasi pada AMQP sangatlah fleksibel karena client bisa berkomunikasi dengan server dan client juga bisa berkomunikasi dengan client yang lain.

Untuk sistem rumah cerdas yang dirancang, diperlukan suatu protokol yang bisa memfasilitasi komunikasi asinkronus, artinya komunikasi yang bisa berjalan satu arah. HTTP hanya bisa memfasilitasi komunikasi sinkronus yang selalu dua arah (menunggu *request* dari *client*). AMQP bisa memfasilitasi komunikasi asinkronus. Selain itu, AMQP juga memfasilitasi mekanisme antrian. Dengan demikian, kita tidak perlu khawatir akan adanya kegagalan komunikasi karena setiap data yang dikirimkan pasti akan dieksekusi satu per satu dalam antrian dan tidak akan hilang. *Environment* populer yang menerapkan protokol AMQP adalah RabbitMQ. Skema komunikasi paling sederhana pada RabbitMQ adalah sebagai berikut.



**Gambar 7 Skema Komunikasi RabbitMQ yang Paling Sederhana**

Pada skema di atas, terdapat sebuah *publisher* (P) dan *consumer* (C). *Publisher* adalah pengirim pesan, *consumer* adalah penerima pesan. Dalam suatu sistem yang besar mungkin saja terdapat banyak *publisher* dan *consumer*. Pesan yang dikirimkan oleh *publisher* akan masuk ke dalam suatu antrian milik suatu *consumer*. Setiap *consumer* pasti memiliki minimal satu antrian yang terikat kepadanya. Dalam contoh di atas, antrian yang terikat kepada *consumer* bernama hello.

## 4. Perancangan Subsistem Indoor dan Outdoor

### 4.1 Pendahuluan

Untuk membangun suatu sistem rumah cerdas, maka setiap perangkat di dalam sistem harus bisa berkomunikasi dengan baik. Dengan demikian, diperlukan suatu aturan atau protokol tertentu yang menjadi dasar cara perangkat-perangkat tersebut bisa saling berkomunikasi dan bertukar data. Sistem terbagi menjadi dua subsistem utama, yaitu *indoor* dan *outdoor*. Dengan demikian, protokol komunikasi yang digunakan oleh subsistem *indoor* tentu berbeda dengan subsistem *outdoor* karena protokol ini harus disesuaikan dengan keadaan masing-masing subsistem. Subsistem *indoor* cukup menggunakan protokol komunikasi sederhana yang sifatnya lokal dan memiliki jangkauan yang pendek. Subsistem *outdoor* memerlukan protokol komunikasi yang lebih kompleks dan memiliki ketahanan tinggi karena berhubungan dengan internet dan *cloud*.

Selain protokol berkomunikasi, sistem rumah cerdas ini juga harus dilengkapi dengan sistem keamanan agar sistem rumah cerdas dapat terlindungi dari retasan pihak-pihak yang tidak bertanggung jawab. Sistem keamanan yang diperlukan adalah berupa enkripsi data-data, terutama data-data yang sedang ditransmisikan. Dengan demikian, sistem harus memiliki suatu enkriptor dan dekriptor supaya data-data dapat ditransmisikan dan diterima dengan baik tanpa ada kesalahan pengiriman data.

Secara umum, aliran data di dalam sistem dijelaskan dalam *Data Flow Diagram Berikut*.





#### 4.2.1 Perancangan Sistem Komunikasi dalam Subsisitem Indoor

```

graph LR
    Server[Server] --> HS((House Selecting))
    HS --> AP[Access Point]
    AP <--> Host[Host]
    Host --> DS((Device Selecting))
    DS --> Node[Node]
    Node <--> Actuator[Actuator]
    Actuator --> Node
    Node --> Host
    Host --> AP
    AP --> Server
  
```

The diagram illustrates the system architecture. It consists of a linear sequence of components: Server, Access Point, Host, Node, and Actuator. Between the Server and Access Point, and between the Host and Node, there are circular nodes labeled 'House Selecting' and 'Device Selecting' respectively. The flow of data and control is as follows:
 

- Server sends data to House Selecting, which then sends it to the Access Point.
- Access Point and Host are connected via bidirectional communication.
- Host sends data to Device Selecting, which then sends it to the Node.
- Node and Actuator are connected via bidirectional communication.
- Actuator sends data back to the Node.
- Node sends data back to the Host.
- Host sends data back to the Access Point.
- Access Point sends data back to the Server.

```

graph LR
    Server[Server]
    AP[Access Point]
    Host[Host]
    Node[Node]
    Actuator[Actuator]
    DS1((Data Saving))
    DCI1((Data Checking and Interpretation))
    ED1((Encrypting Data))
    HS((House Selecting))
    DD1((Decrypting data))
    DCI2((Data Checking and Interpretation))
    DS2((Data Saving))
    DD2((Decrypting data))
    DI((Data Interpretation))
    DP1((Data Processing))
    DP2((Data Processing))

    Server --> AP
    AP --> Host
    Host --> Node
    Node --> Actuator
    Actuator --> Node
    Node --> Host
    Host --> AP
    AP --> Server
    Server --> DS1
    DS1 --> Server
    Server --> DCI1
    DCI1 --> Server
    Server --> ED1
    ED1 --> HS
    HS --> AP
    AP --> DD1
    DD1 --> DCI2
    DCI2 --> Server
    Host --> DD2
    DD2 --> DCI2
    DCI2 --> Host
    Host --> DS2
    DS2 --> Host
    Host --> DCI3((Data Checking and Interpretation))
    DCI3 --> DS3((Device Selecting))
    DS3 --> Node
    Node --> DI
    DI --> DP1
    DP1 --> DP2
    DP2 --> Actuator
  
```

**Gambar 10 DFD Sistem Indoor Level 2**

Dari DFD di atas, terlihat bahwa ada satu proses penting yang harus dipikirkan di dalam subsistem *indoor*, yaitu pemilihan perangkat (*device selecting*). Proses ini penting karena inilah yang akan menentukan ke perangkat mana suatu perintah harus dikirimkan oleh *host* ketika ada permintaan dari pengguna. Seperti yang telah dibahas dalam subbab 3.1, modul XBee Pro memiliki identitas unik berupa alamat. Dengan demikian, yang menjadi pembeda antara satu perangkat dengan yang lain adalah alamat tersebut.

Selain *device selecting*, proses lain yang tidak kalah pentingnya adalah proses *home selecting*. Untuk bisa membedakan rumah yang satu dengan yang lain, maka diperlukan suatu identitas unik bagi setiap rumah yang disebut dengan ID Rumah atau *home ID*. Dengan demikian, subsistem *outdoor* bisa membedakan satu rumah dengan rumah yang lain.

Selanjutnya, hal yang perlu dipikirkan adalah bagaimana protokol pengiriman perintah dari *host* ke *node*. Protokol berhubungan dengan bentuk data yang ditransfer dari satu perangkat ke perangkat yang lainnya. Bentuk data ini harus efisien dan umum sehingga dapat digeneralisasi dan diterapkan pada semua jenis perangkat.

Agar dapat membentuk suatu paket data yang efisien dan umum, maka ada beberapa hal yang harus terkandung di dalam paket data tersebut.

1. Home ID, yaitu bagian awal yang merupakan identitas unik untuk satu rumah. Bagian ini harus ada untuk memastikan bahwa data yang diterima oleh suatu perangkat merupakan data yang memang valid untuk rumah tersebut. Data ini berupa gabungan dari enam buah karakter sehingga ukurannya 6 byte. Sebagai contoh, paket data dalam suatu rumah diberi ID “123456”. Maka, setiap data yang memiliki Home ID bukan “123456” tidak akan dihiraukan oleh perangkat-perangkat yang ada di rumah tersebut. Data yang datang ke perangkat bisa saja bukan data yang valid, namun data sembarang yang masuk ke dalam sistem, misalnya dari rumah tetangga, atau dari usaha peretas yang ingin meretas sistem rumah tersebut.
2. Alamat Perangkat, yaitu alamat unik setiap perangkat XBee yang adalah di dalam subsistem indoor suatu rumah. Data yang dianggap valid untuk suatu perangkat adalah data yang memiliki ID atau alamat yang cocok untuk perangkat.

3. Payload, yaitu bagian inti dari paket data yang dikirimkan. Bagian ini berisi informasi yang ingin disampaikan dari suatu perangkat ke perangkat lainnya. Informasi tersebut bisa berupa perintah maupun status suatu perangkat.

Dengan tiga komponen di atas, maka paket data yang hendak diterapkan pada subsistem indoor adalah sebagai berikut.

**Tabel 3 Susunan Paket Data yang Dialirkan pada Subsistem Indoor**

<b><i>Home ID</i></b> <b>(6 B)</b>	<b><i>Alamat Perangkat</i></b> <b>(8 B)</b>	<b><i>Payload</i></b> <b>(1 B)</b>
---------------------------------------	--	---------------------------------------

### **Home ID (ID)**

Home ID berisi data 6 byte berupa karakter dengan fungsi seperti dijelaskan sebelumnya, yaitu sebagai penanda bahwa paket data yang diterima oleh suatu perangkat memang untuk rumah tersebut atau bukan. Home ID ini berisi karakter-karakter alfanumerik. Total keseluruhan kemungkinan alfanumerik kapital adalah 36 karakter (alfabet kapital dan angka). Dengan demikian, jumlah ID rumah unik yang bisa digenerasi dengan 6 digit karakter adalah :

$$\text{Jumlah ID Rumah Unik} = (36)^6 = 2.176 \times 10^9$$

### **Alamat Perangkat (Address - AD)**

Kode perangkat merupakan kode unik bagi setiap perangkat dalam subsistem indoor. Kode perangkat berukuran 8 byte. Ukuran ini disesuaikan dengan alamat pada modul XBee.

### **Data Payload (PL)**

Payload merupakan inti dari paket data ini. Payload berisi perintah atau status yang diberikan dari atau ke host. Setiap perangkat memiliki interpretasi yang berbeda-beda terhadap nilai payload ini. Seperti yang telah dijelaskan pada subbab 2.2, tim dari PME telah berhasil mengimplementasikan enam jenis aktuator. Selain itu, ada satu tambahan jenis aktuator yang diharapkan dapat diimplementasikan dalam proyek ini. Dengan demikian,

secara keseluruhan, terdapat tujuh jenis aktuator. Masing-masing aktuator memiliki interpretasi perintah yang berbeda-beda yang dijelaskan sebagai berikut.

**Tabel 4 Jenis Aktuator Beserta Interpretasi Data Payload yang Dikirim oleh Host**

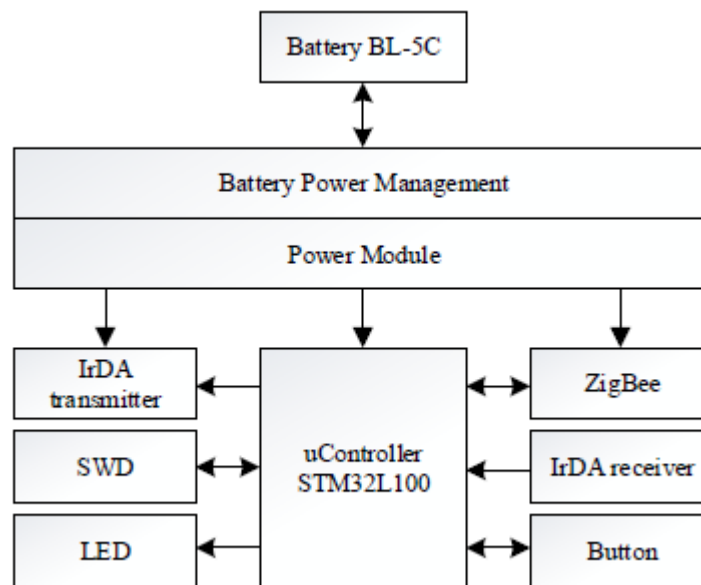
No	Jenis <i>End-Device</i>	Payload	Makna
1	Sakelar	0x00	OFF
		0x64	ON
2	Temperature	0xC8	Monitoring
		0x00-0xFF	Measuring
	Humidity	0xC8	Monitoring
		0x00-0xFF	Measuring
3	RGB Lamp	0x00-0xFF (3byte)	Color Changing
4	Curtain	0x00-0xFF	Dimming (0% - 100%)
5	Fan	0x00-0xFF	Dimming (0% - 100%)
6	Door Lock	0x00	Open
		0x64	Close
7	Remote	0x00	OFF
		0x64	ON

## 4.2.2 Perancangan Remote Controller

### 4.2.2.1 Perancangan Hardware Remote Contro

Untuk dapat mengatur peralatan elektronik yang membutuhkan sinyal infrared, maka dibuat sebuah *remote infrared*. Komponen utamanya adalah IrDA *transmitter* dan IrDA *receiver*.

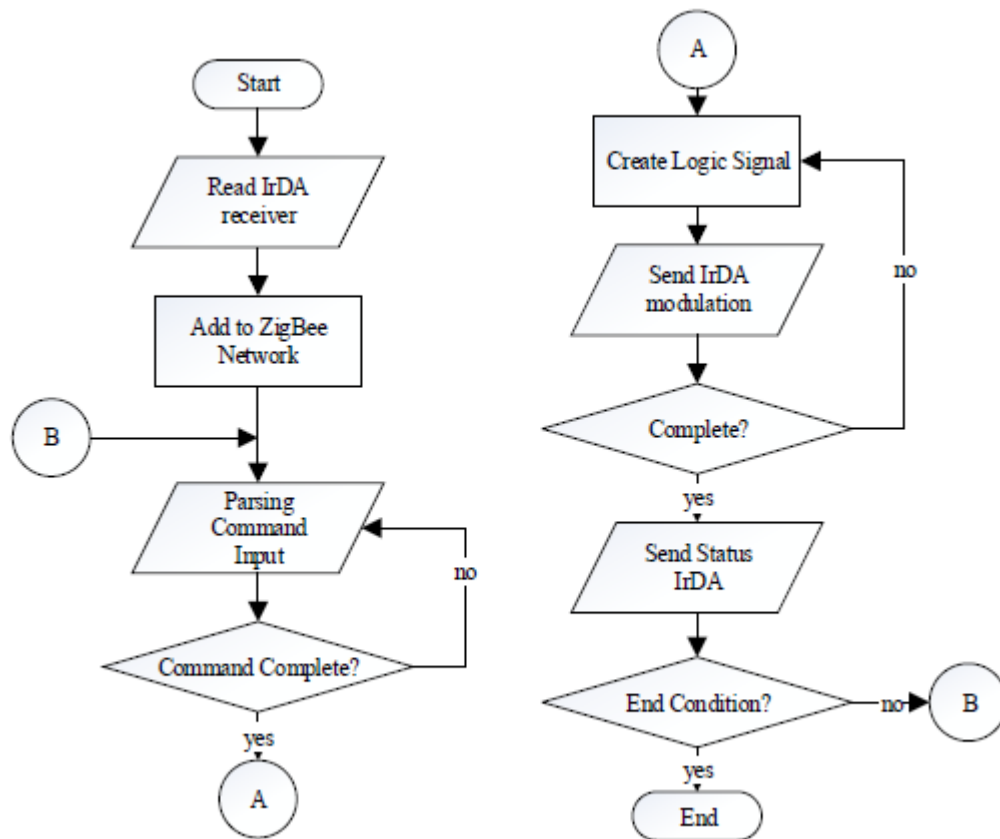
IrDA transmitter yang digunakan berupa IR LED dengan mode *wide* sehingga dapat mengirim sinyal IR yang lebih luas. IrDA receiver yang digunakan adalah TSOP 1838 yang menerima sinyal dengan frekuensi 38 kHz. Selain komponen *transmitter* dan *receiver*, mikrokontroller yang digunakan untuk pemrosesan sinyal adalah STM32L100RC yang memiliki memory yang cukup besar serta pemrosesan yang cepat dengan daya *ultra-low power*. Remote juga akan dilengkapi kabel USB Micro sebagai sumber daya, serta memiliki power backup berupa Battery BL-5C. Untuk proses penukaran informasi, remote akan dipasang modul XBEE Pro.



Gambar 11 Struktur perangkat keras *remote* IrDA-based

#### 4.2.2.2 Perancangan Software Remote Control

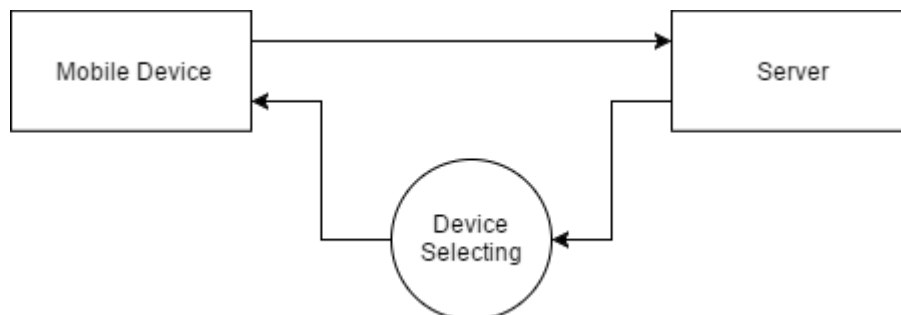
Diagram alir remote dimulai dengan pembacaan karakter sinyal dari remote konvensional. Remote akan memiliki sebuah push button yang berfungsi sebagai penanda bahwa remote sedang dalam mode receive. Setelah membaca karakter sinyal remote konvensional, hasil pembacaan karakter tersebut nantinya yang akan disimpan sebagai suatu variable berupa array yang akan digunakan referensi pembangkitan sinyal yang akan dimodulasi dengan frekuensi pembawa. Frekuensi pembawa yang akan dibangkitkan adalah sebesar 38kHz yang merupakan standar protocol khusus pertukaran data melalui *infrared*. Berikut adalah diagram alir dari perangkat remote:



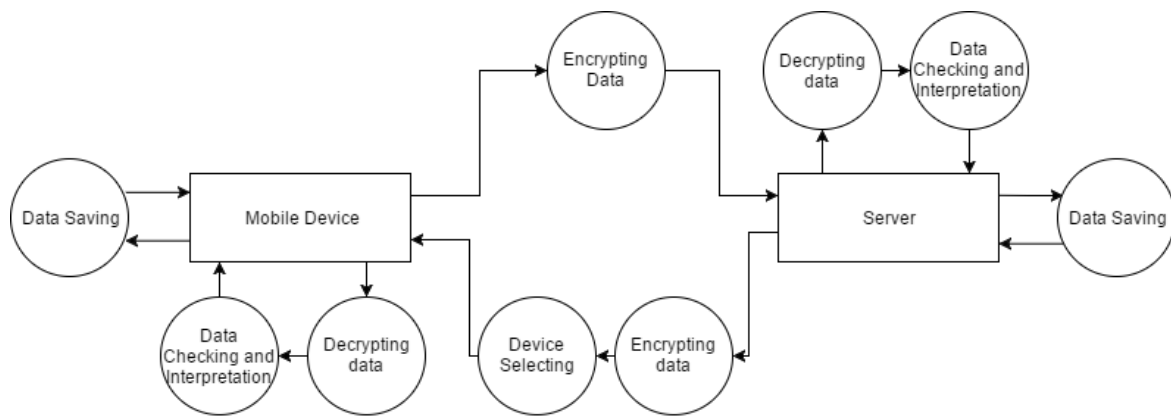
Gambar 12 Diagram alir perangkat *remote*

#### 4.3 Perancangan Subsistem Outdoor

Seperti yang telah dijelaskan pada subbab 3.2, ada dua komponen utama di subsistem *outdoor*, yaitu server dan ponsel. Aliran data dalam subsistem ini dijelaskan melalui DFD berikut.



Gambar 13 DFD Sistem Outdoor Level 1



**Gambar 14 DFD Sistem Outdoor Level 2**

Dari DFD di atas, terlihat bahwa proses yang penting di subsistem ini adalah *device selecting*, yaitu pembedaan satu ponsel dengan ponsel lainnya. Untuk mengatasi hal tersebut, maka perlu dibuat sistem penamaan setiap pengguna dengan identitas unik. Penamaan tersebut berupa pembuatan akun bagi setiap pengguna. Identitas dalam akun tersebut berupa alamat *email*, *password*, nama pengguna, dan nomor ID pengguna (*user ID*). Dengan demikian, setiap pengguna bisa dibedakan satu dengan yang lain.

#### **4.3.1 Perancangan Basis Data di Server dan Host**

Untuk memenuhi seluruh fitur yang telah disebutkan pada dokumen B200, diperlukan suatu basis data yang dirancang secara efektif dan kompak sehingga tidak memboroskan memori. Basis data diletakan di dua tempat, yaitu di server pusat dan di *host*. Data yang disimpan di server pusat adalah data yang berhubungan dengan seluruh pengguna secara umum. Data yang disimpan di *host* rumah secara lokal adalah data yang berhubungan dengan perangkat-perangkat yang ada di rumah tersebut.

##### **4.3.1.1 Basis Data di Server Pusat**

Server harus memiliki suatu basis data yang menyimpan informasi yang berkaitan dengan akun seluruh pengguna yang terdaftar. Dengan demikian, dibuatlah suatu *database* bernama *general\_data* yang berisi dua buah tabel bernama *users* dan *aeskeylist*. Tabel *users* berisi data pribadi seluruh *user* yang terdaftar di sistem. Tabel *aeskeylist* berisi daftar kunci AES untuk setiap *user*. Kolom-kolom tabel *users* dan *aeskeylist* adalah sebagai berikut.

**Tabel 5 Kolom-kolom Tabel users di *database general\_data***

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Isi</b>
<b>email</b>	VARCHAR(50)	Email pengguna
<b>password</b>	VARCHAR(20)	Password pengguna
<b>homeid</b>	VARCHAR(12)	ID rumah di mana pengguna didaftarkan
<b>name</b>	VARCHAR(20)	Nama pengguna
<b>picture</b>	MEDIUMTEXT	Gambar profil pengguna yang telah mengalami Base64 <i>encoding</i>
<b>privilege</b>	VARCHAR(11)	Jenis privilege pengguna (ADMIN atau Guest)
<b>aeskey</b>	VARCHAR(16)	Kunci enkripsi AES untuk pengguna (akan dijelaskan lebih lanjut pada bagian keamanan jaringan)

**Tabel 6 Kolom-kolom Tabel users di *database general\_data***

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Isi</b>
<b>phoneid</b>	VARCHAR(8), PRIMARY KEY	Nomor ID unik ponsel pengguna (diambil dari dalam OS Android)
<b>aeskey</b>	VARCHAR(16)	Kunci enkripsi AES untuk pengguna (akan dijelaskan lebih lanjut pada bagian keamanan jaringan)



#### 4.3.1.2 Basis Data di Host

Host harus menyimpan data-data yang berhubungan dengan perangkat-perangkat di dalam rumah secara lokal. Hal ini dimaksudkan supaya ada pembagian kerja dan lalu lintas data di server sehingga server tidak bekerja terlalu keras. Dengan demikian, dibuatlah sebuah database bernama homeXXXXXX dengan X merupakan 6 digit ID rumah. Isi dari database ini adalah beberapa tabel, yaitu:

- 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

Penjelasan tiap kolom dari masing-masing tabel adalah sebagai berikut.

**Tabel 7 Kolom-kolom Tabel info di database homeXXXXXX**

Nama Kolom	Tipe Data	Isi
<b>name</b>	VARCHAR(20), PRIMARY KEY	Nama rumah
<b>latitude</b>	DOUBLE	Koordinat posisi latitude (lintang) rumah
<b>longitude</b>	DOUBLE	Koordinat posisi longitude (bujur) rumah
<b>Lockstatus</b>	VARCHAR (6)	Status kunci rumah (locked atau unlocked)

**Tabel 8 Kolom-kolom Tabel devices di database homeXXXXXX**

Nama Kolom	Tipe Data	Isi
<b>name</b>	VARCHAR(20)	Nama perangkat
<b>address</b>	VARCHAR(15), PRIMARY KEY	Alamat XBee perangkat
<b>type</b>	VARCHAR(11)	Tipe perangkat (sakelar, lampu, tirai, dsb.)
<b>status</b>	VARCHAR(8)	Status dari perangkat tersebut (ON, OFF, 50%, dsb.)
<b>room</b>	VARCHAR(20)	Ruangan di mana perangkat tersebut diletakan
<b>privilege</b>	VARCHAR(5)	Privilege device, apakah device bisa diakses oleh ADMIN saja atau oleh ADMIN dan Guest

**Tabel 9 Kolom-kolom Tabel rooms di *database homeXXXXXX***

Nama Kolom	Tipe Data	Isi
<b>name</b>	VARCHAR(20), PRIMARY KEY	Nama ruangan

**Tabel 10 Kolom-kolom Tabel scenarios di *database homeXXXXXX***

Nama Kolom	Tipe Data	Isi
<b>name</b>	VARCHAR(20), PRIMARY KEY	Nama skenario
<b>time</b>	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 11 Kolom-kolom Tabel dataMMYYYY di *database homeXXXXXX***

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Isi</b>
<b>address</b>	VARCHAR(8), PRIMARY KEY	Alamat XBee perangkat
<b>lasttime</b>	VARCHAR(14)	Waktu terakhir (jam, menit, detik, hari, bulan, tahun) ketika perangkat terakhir dinyalakan atau diakfikan
<b>d01 sampai d31 (d adalah date)</b>	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 12 Kolom-kolom Tabel users di *database homeXXXXXX***

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Isi</b>
<b>phoneid</b>	VARCHAR(16), PRIMARY KEY	ID pengguna
<b>email</b>	VARCHAR(50)	Email pengguna

<b>privilege</b>	<b>VARCHAR(14)</b>	<b>Privilege pengguna (ADMIN atau Guest)</b>
<b>aeskey</b>	<b>VARCHAR(16)</b>	Kunci enkripsi AES untuk pengguna (akan dijelaskan lebih lanjut pada bagian keamanan jaringan)

Selain tabel-tabel di atas, ada juga tabel dengan nama yang sesuai dengan nama skenario-skenario yang ada di rumah. Tabel bernama skenario ini berisi daftar barang apa saja yang harus diakses melalui skenario ini dan apa statusnya. Dengan demikian, *host* akan mengetahui apa yang harus dilakukan ketika ada suatu permintaan skenario. Bentuk tabel ini sama persisi dengan tabel devices.

Sebagai tambahan, seluruh data yang ada di database homeXXXXXX (kecuali users) juga disimpan di server. Isinya benar-benar disalin persis dengan yang ada di *host* pada jam 12 malam setiap harinya. Dengan demikian, bila ada kejadian yang tidak diinginkan menimpa *host*, misalnya *host* rusak, maka masih ada *back up* data untuk menyelamatkan data-data tersebut.

#### **4.3.2 Keamanan Jaringan**

Karena subsistem *outdoor* menggunakan jaringan internet sebagai sarana komunikasi, maka faktor keamanan menjadi sangat penting. Bila jaringan sistem tidak aman, maka bisa saja sistem rumah cerdas disusupi oleh *hacker* atau pihak-pihak lain yang tidak bertanggung jawab dan bisa berakibat fatal.

Untuk protokol HTTP, kebanyakan orang biasanya menggunakan HTTPS (HTTP secured) untuk mengatasi masalah keamanan jaringan. Namun, karena protokol AMQP yang digunakan, maka harus dicari suatu metode lain untuk mengamankan komunikasi data. Salah satu cara yang paling mudah adalah dengan melakukan proses enkripsi terhadap data-data yang dikirimkan di jaringan internet.

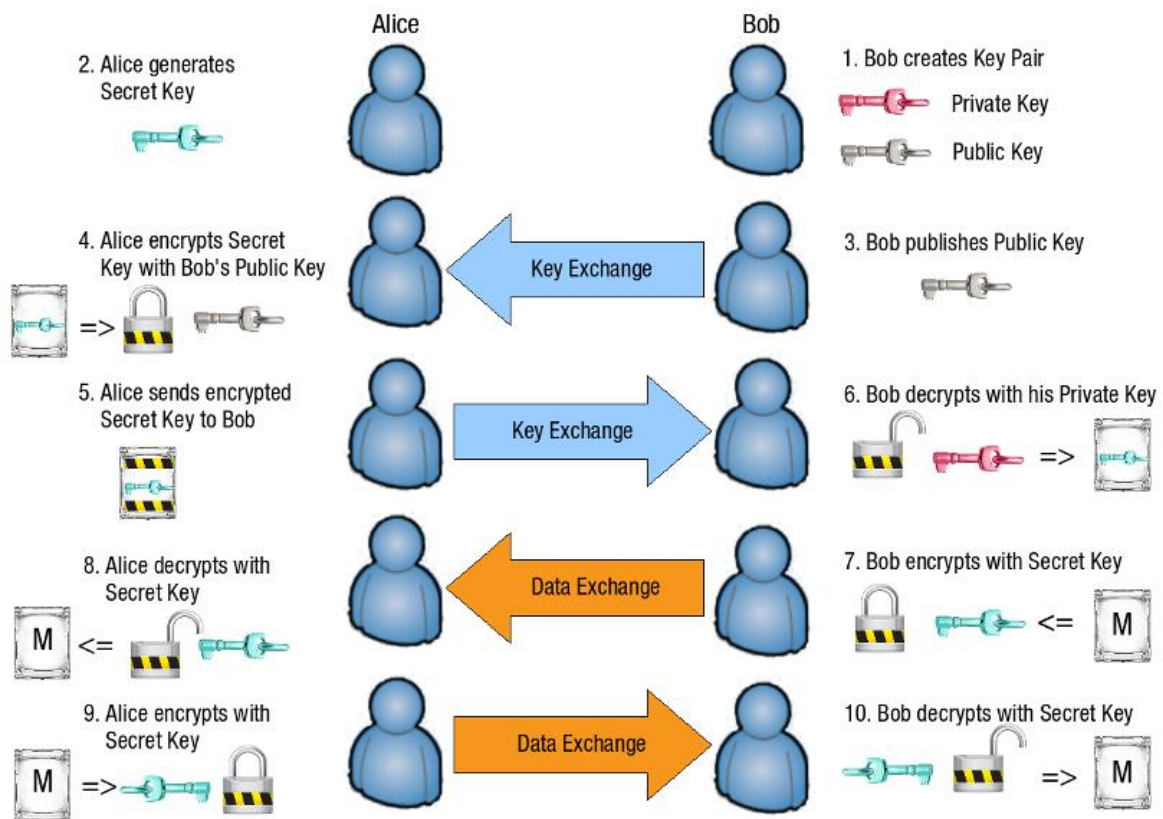
Ada banyak metode enkripsi yang telah ditemukan saat ini. Secara umum, metode-metode tersebut dapat dikategorikan menjadi dua jenis, yaitu enkripsi simetrik dan asimetrik.

Enkripsi simetrik adalah enkripsi di mana kunci (*key*) untuk melakukan enkripsi dan dekripsi sama. Secara umum, enkripsi simetrik memiliki kompleksitas proses yang tidak terlalu tinggi. Namun, dari sisi tingkat keamanan, enkripsi ini lebih rendah karena ketika kunci enkripsinya diketahui, seseorang bisa melakukan enkripsi sekaligus dekripsi.

Enkripsi asimetrik adalah enkripsi di mana kunci untuk melakukan enkripsi dan dekripsi berbeda. Oleh karena itu, pada jenis ini, biasanya dikenal dua jenis kunci, yaitu *public key* dan *private key*. *Public key* adalah kunci untuk melakukan enkripsi. Kunci ini boleh disebarluaskan secara bebas (*public*) karena tidak akan membahayakan. Sebaliknya, *private key* adalah kunci untuk melakukan dekripsi dan sifatnya rahasia (hanya boleh diketahui oleh dekriptor). Secara umum, kompleksitas proses enkripsi asimetrik lebih tinggi. Namun, enkripsi asimetrik relatif lebih aman karena kunci untuk enkripsi dan dekripsi berbeda.

Dari pembahasan di atas, diperoleh simpulan bahwa kedua jenis enkripsi memang memiliki kelemahan dan kelebihan tersendiri. Dengan demikian, cara terbaik yang biasanya banyak dilakukan adalah dengan mengkombinasikan kedua jenis enkripsi di atas. Mekanisme umum untuk menggabungkan keduanya adalah sebagai berikut.

1. Server melakukan generasi *public key* dan *private key* enkripsi asimetrik.
2. Server mengirimkan *public key* ke *client*.
3. *Client* melakukan generasi *key* untuk enkripsi simetrik.
4. *Key* simetrik dikirimkan ke server dengan dienkripsi terlebih dahulu menggunakan *public key*.
5. Server menerima pesan, lalu melakukan dekripsi pesan menggunakan *private key*. Server akan mengetahui apa *key* simetrik yang telah digenerasi oleh *client*.
6. Komunikasi berikutnya menggunakan enkripsi simetrik.



**Gambar 15 Skema Penggabungan Enkripsi Simetrik dan Asimetrik**

(Sumber:

[http://documentation.axsguard.net/manuals/Gatekeeper/8.0.0/output/html\\_chunked/howto\\_guides/ipsec/generated\\_output.chunked/images/ipsec\\_rsa.png](http://documentation.axsguard.net/manuals/Gatekeeper/8.0.0/output/html_chunked/howto_guides/ipsec/generated_output.chunked/images/ipsec_rsa.png))

Dengan demikian, cara inilah yang digunakan untuk mengamankan komunikasi data sistem.

Berikutnya, hal yang perlu dipikirkan adalah pemilihan metode enkripsi. Ada banyak metode yang bisa menjadi pilihan. Beberapa contoh metode enkripsi simetrik adalah:

- AES
- DES
- Triple DES
- Twofish
- Serpent

Dari kelima jenis metode di atas, metode yang paling sering digunakan adalah AES (dengan algoritma dari Rijndael). Metode ini merupakan metode yang digunakan oleh NSA

(*National Security Agency*), biro keamanan nasional Amerika Serikat, untuk mengamankan data-data rahasia milik mereka. Enkripsi ini tergolong sangat aman dan jarang pihak-pihak yang berhasil menyerang enkripsi ini. Selain itu, sudah banyak *library* yang memudahkan penggunaan enkripsi ini sehingga kita tidak perlu membuat algoritma enkripsi ini dari awal. Dengan demikian, dipilihlah AES sebagai metode enkripsi simetrik.

Beberapa contoh enkripsi asimetrik adalah:

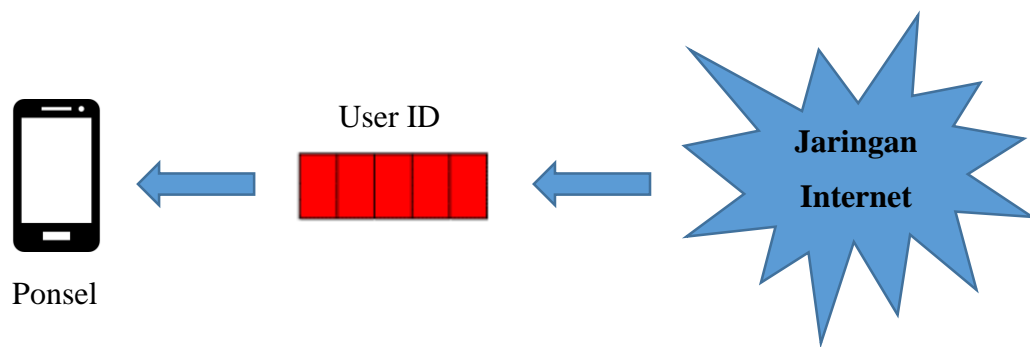
- DSA
- RSA
- PGP

Dari ketiga metode di atas, metode yang paling banyak dan umum digunakan adalah RSA. Mirip dengan AES, RSA juga merupakan salah satu metode yang sulit dirusak. RSA memiliki beberapa kategori berdasarkan jumlah bitnya. Semakin besar jumlah bit yang digunakan, maka enkripsi akan semakin sulit diserang dan string yang bisa dienkripsi juga semakin panjang. Namun, proses kalkulasi juga akan semakin kompleks. Dengan demikian, dipilihlah RSA 2048 bit (256 byte) sebagai metode enkripsi asimetrik. Jumlah bit disesuaikan dengan perkiraan panjang string yang ingin dienkripsi selama komunikasi.

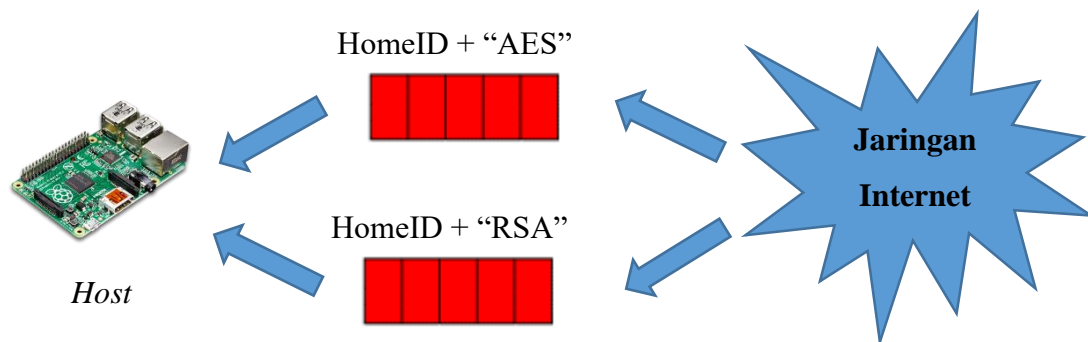
#### **4.3.3. Perancangan Mekanisme Antrian Data Dalam Protokol AMQP**

Seperti yang telah dijelaskan sebelumnya, ada tiga elemen yang saling berkomunikasi di subsistem *outdoor*, yaitu ponsel, server, dan *host*. Dengan demikian, setiap elemen ini harus memiliki antriannya masing-masing di jaringan dengan protokol AMQP.

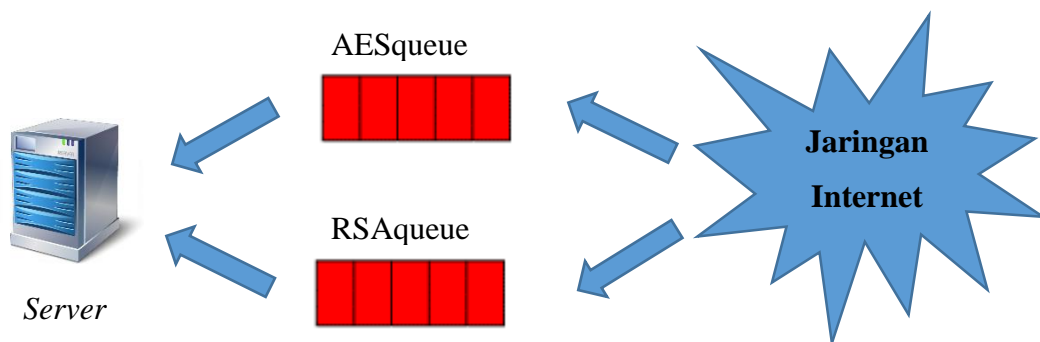
Setiap ponsel pasti sedang digunakan oleh seorang pengguna yang memiliki akun. Setiap akun pasti memiliki nomor identitas (*User ID*). Dengan demikian, untuk memudahkan, setiap ponsel pasti menunggu pesan dari sebuah antrian yang diberi nama sesuai dengan *User ID* akun pada ponsel tersebut.



Hal yang berbeda terjadi pada antrian *host*. Antrian pada *host* tidak bisa dibuat hanya satu antrian saja karena *host* bisa menerima data yang dienkripsi baik menggunakan RSA maupun AES. Dengan demikian, harus ada dua antrian yang berbeda untuk membedakan proses dekripsinya. Karena setiap rumah memiliki ID rumah (*Home ID*), maka nama kedua antrian pada *host* yang paling mudah adalah *Home ID* + “AES” dan *Home ID* + “RSA”.



Hal yang mirip terjadi pada server. Server juga akan menerima data yang dienkripsi baik menggunakan RSA maupun AES. Dengan demikian, struktur antrian server sama persis dengan struktur antrian pada *host*. Nama queue tersebut adalah AESqueue dan RSAqueue

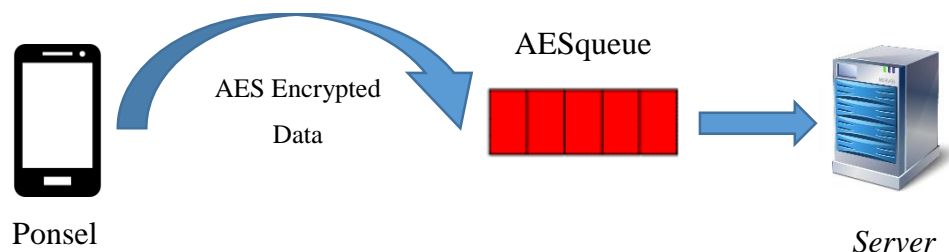




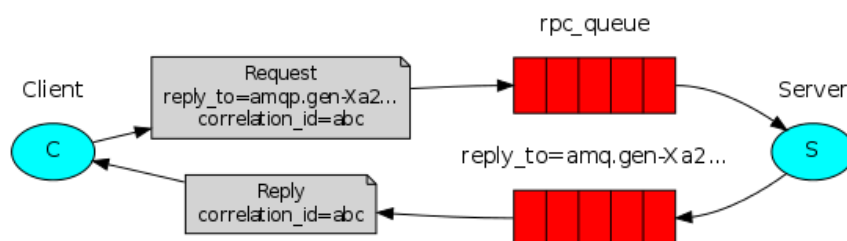
Selama proses komunikasi di dalam sistem, ada dua kemungkinan skema komunikasi yang terjadi, yaitu:

- Skema komunikasi satu arah, misalnya dari ponsel ke *server* atau dari ponsel ke *host*, yang tidak membutuhkan balasan.
- Skema komunikasi dua arah, yaitu komunikasi yang membutuhkan balasan. Misalnya, ponsel mengirimkan suatu pesan ke *server*, kemudian ponsel menunggu sampai *server* memberikan jawaban atas pesan yang baru saja dikirim tersebut.

Dalam kasus yang pertama, pesan cukup dikirimkan oleh pengirim dan ditujukan ke *queue* perangkat tertuju. Misalnya, bila ponsel ingin mengirimkan pesan satu arah ke *server*, data dienkripsi menggunakan AES, maka ponsel cukup mengirimkannya langsung ke AESqueue dan *server* akan segera memproses data tersebut tanpa harus memberikan jawaban. Skema ini disebut skema komunikasi biasa (*common communication scheme*).



Dalam kasus yang kedua, RabbitMQ menyediakan skema khusus bernama RPC (*Remote Procedure Call*). Skema ini memfasilitasi komunikasi dua arah dengan adanya fitur properti pesan. Setiap pesan yang dialirkan di dalam jaringan bisa diberi properti. Salah satu isi properti pesan tersebut adalah properti *reply\_to*. Properti inilah yang digunakan dalam skema RPC.



**Gambar 16 Skema RPC**

Setiap kali pengirim hendak mengirimkan pesan dengan skema RPC, pengirim mendeklarasikan *queue* baru dengan nama sembarang (RabbitMQ juga memfasilitasi pembuatan *queue* baru dengan nama sembarang). *Queue* inilah yang nantinya akan menjadi *queue* untuk pesan balasan dari pihak tertuju. Nama *queue* sembarang ini dituliskan atau disimpan di dalam properti *reply\_to* milik pesan yang hendak dikirimkan pengirim. Dengan demikian, ketika pesan sampai di pihak tertuju, pihak tertuju ini akan mengetahui ke mana ia harus mengirimkan balasannya.

Pesan balasan tidak dikirimkan ke *default queue* pengirim (untuk ponsel misalnya *queue* dengan nama *phone ID*) supaya pesan balasan tidak bercampur dengan pesan-pesan lain yang datang dari skema komunikasi biasa. Dengan demikian, pihak pengirim tau bahwa pesan balasan ini memang betul-betul balasan untuk pesan yang telah dikirimkan sebelumnya.

Selain adanya properti *reply\_to*, RabbitMQ juga memfasilitasi suatu properti lain bernama *correlation\_id*. Properti ini bisa diibaratkan seperti penanda atau *tag* pesan. *Correlation\_id* bisa diisi dengan string sembarang. Dalam sistem komunikasi rumah cerdas ini, *correlation\_id* akan diisi oleh *phone ID* pengirim. Dengan demikian, setiap pesan yang mengalir di dalam jaringan akan diketahui asalnya dari mana. Hal ini juga yang nantinya akan membantu *server* dan *host* ketika hendak melakukan dekripsi AES. Karena setiap ponsel memiliki kunci AES masing-masing, maka *host* dan *server* perlu tau dari mana setiap pesan ini dikirim supaya bisa menggunakan kunci yang tepat untuk dekripsi.

#### **4.3.4 Perancangan Paket Data untuk Komunikasi pada Subsistem Outdoor**

Agar ketiga pihak (*mobile device*, *server*, *host*) dapat saling berkomunikasi, maka diperlukan suatu perjanjian paket data yang hendak dikirimkan sehingga seluruh pihak dapat memahami informasi atau instruksi apa yang dialirkan dalam sistem. Perancangan paket data ini harus didasarkan pada kebutuhan komunikasi di dalam sistem. Kebutuhan ini kurang lebih terangkum dalam fitur-fitur sistem yang telah dijelaskan pada dokumen B200 dan tampilan aplikasi yang telah dijelaskan pada subbab 3.2. Proses-proses yang membutuhkan komunikasi data dalam sistem antara lain sebagai berikut.

- Proses *sign up* dan *sign in* setiap akun

- Proses manajemen pengguna, misalnya penambahan atau pengurangan pengguna di dalam suatu rumah, ganti *password*, ganti nama, dll.
- Proses manajemen perangkat, misalnya penambahan atau pengurangan perangkat di dalam rumah, kategori ruangan, dan skenario.
- Proses loading data dari server atau *host* ke ponsel
- Proses pengiriman notifikasi ke ponsel
- Proses pemberian perintah untuk perangkat
- Proses perekaman durasi pemakaian setiap perangkat di rumah

Untuk memenuhi seluruh kebutuhan itu, maka paket data dirancang untuk memiliki *header* dan *type* tertentu sesuai dengan kategori komunikasinya. Dengan demikian, bentuk paket data secara umum adalah sebagai berikut.

HEADER (2 CHAR)	TYPE	ISI PESAN
--------------------	------	-----------

Setelah dibentuk, paket data disatukan dan dengan menggunakan *library* JSON (*JavaScript Object Notation*). Dengan menggunakan *library* ini, maka setiap elemen-elemen di dalam pesan akan menyatu menjadi sebuah string panjang. Contoh serialisasi JSON adalah sebagai berikut.

Paket data sebelum serialisasi:

- Header = “SI”
- Email = “aaa@gmail.com”
- Password = “1234”

Data setelah serialisasi:

```
{“Header” = “SI”, “Email” = “aaa@gmail.com”, “Password” = “1234”}
```

Data yang telah diserialisasi inilah yang akan dikirimkan dan dikomunikasikan di antara perangkat di subsistem *outdoor*. Dengan demikian, pesan tidak lagi terpisah-pisah, melainkan satu kesatuan string yang kompak dan fleksibel untuk dikirimkan kemanapun.

Untuk memudahkan seluruh fitur, header-header pesan ditetapkan sebagai berikut.

**Tabel 13 Daftar Header Pesan**

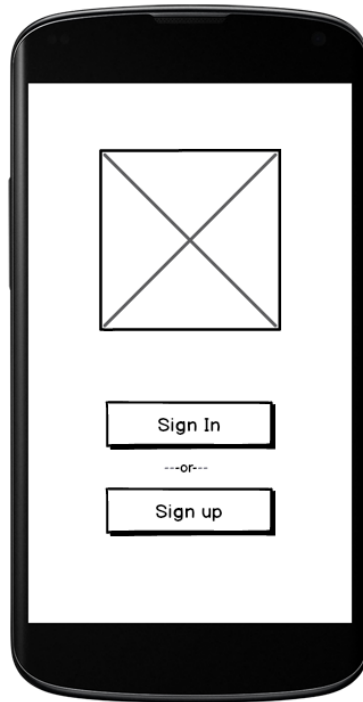
Header	Kepanjangan	Isi pesan
SI	Sign In	Permintaan sign in
SU	Sign Up	Permintaan sign up
MH	Management–Home	Pesan yang berhubungan dengan manajemen rumah
MU	Management–Users	Pesan yang berhubungan dengan manajemen pengguna
MD	Management–Devices	Pesan yang berhubungan dengan manajemen perangkat, room, dan skenario
CO	Command	Perintah kepada perangkat, misalnya perintah menyalakan atau mematikan perangkat
NO	Notifications	Pemberitahuan adanya notifikasi baru
DA	Data	Permintaan data analitik penggunaan perangkat
UP	Update	Update data analitik
LO	Load	Permintaan daftar perangkat, room, dan skenario
SC	Scenario	Perintah mengaktifkan skenario

### 4.3.5 Perancangan Aplikasi Android

#### 4.3.5.1 Halaman Awal

Halaman ini hanya ditampilkan pada *user* yang belum terdaftar. Jika *user* sudah pernah melakukan *sign in*, *user* segera masuk ke dalam aplikasi. Pada halaman ini, ditampilkan logo dan nama aplikasi dalam bentuk gambar untuk mengenalkan kepada *user* mengenai aplikasi Rumah Cerdas. Pada bagian di bawah gambar terdapat dua buah tombol

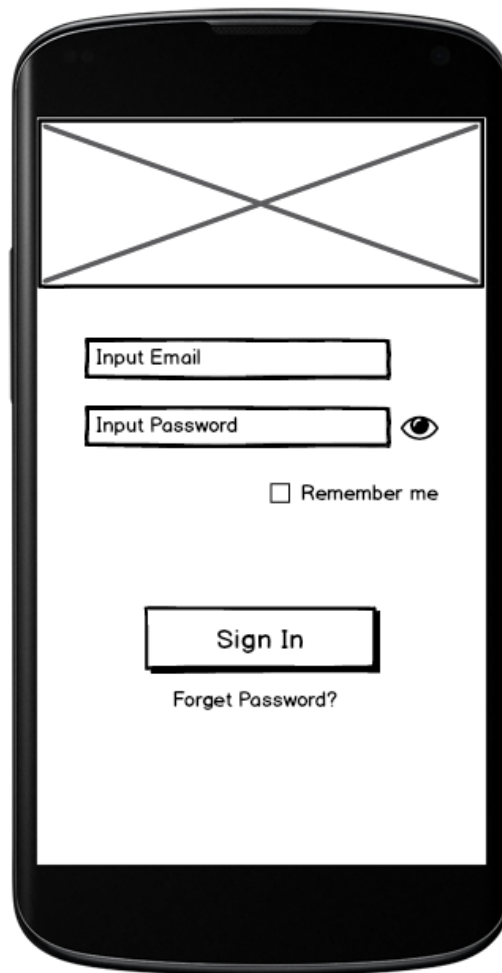
untuk menuju dua halaman yang berbeda. Tombol “SIGN IN” jika ditekan akan menampilkan halaman yang dapat digunakan *user* untuk memasukan identifikasi agar dapat *sign in* ke dalam aplikasi. Sedangkan jika tombol “SIGN UP” ditekan akan menampilkan halaman untuk mendaftarkan *user* kedalam *database* sistem Rumah Cerdas.



**Gambar 17 Halaman Utama**

#### 4.3.4.2 Halaman Sign In

Ketika *user* menekan tombol “SIGN IN” pada halaman sebelumnya, *user* akan masuk kedalam halaman ini. Bagian-bagian dari halaman ini adalah sebagai berikut.



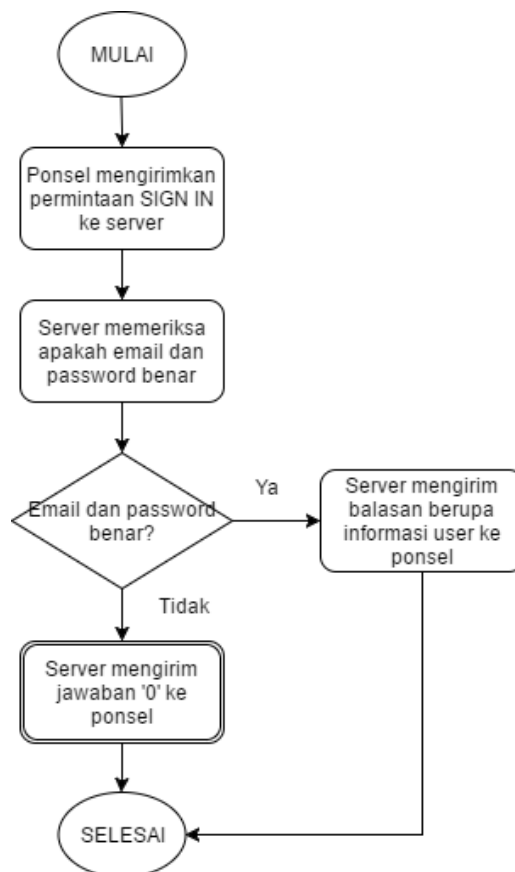
**Gambar 18 Halaman Sign In**

- Ada 2 kolom isian yang bisa diisi oleh user. Kolom pertama dapat digunakan untuk memasukan *email user*, sedangkan kolom kedua digunakan untuk memasukan *password user*. Kolom *password* akan menampilkan karakter yang berbentuk lingkaran agar tidak ada orang yang dapat membaca *password user* pada saat *password* sedang diketikan. Pada saat mengetikan *password*, *user* dapat menekan tombol bergambar mata untuk menampilkan karakter-karakter *password* agar tidak terjadi kesalahan pada saat *user* melakukan *input password*.
- *User* dapat menekan *checkbox* “*remember me*” agar pengguna terus terdaftar di dalam aplikasi pada *smartphone* walaupun aplikasi di-*destroy*.
- Terdapat tombol “SIGN IN” di bagian bawah *checkbox*. Ketika *user* menekan tombol “SIGN IN” yang terletak dibawah *checkbox*, maka aplikasi akan mengirimkan permintaan *sign in* kepada server dengan data JSON sebagai berikut.

**Tabel 14 Data JSON Sign In**

Nomor Dokumen: B300-01-TA1617.01.056	Nomor Revisi: 00	Tanggal: 02/05/2017	Halaman 46 dari 94
--------------------------------------	------------------	---------------------	--------------------

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
SI	-	<ul style="list-style-type: none"> <li>Email</li> <li>Password</li> <li>Kunci AES</li> </ul>	Server	RSA	<p>Jika email dan password sesuai dan pengguna sudah terdaftar di suatu rumah:</p> <ul style="list-style-type: none"> <li>Home ID</li> <li>Nama pengguna</li> <li>Gambar profil</li> <li>Privilege pengguna (ADMIN atau Guest)</li> <li>User ID</li> </ul> <p>Jika email dan password tidak sesuai:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '0'</li> </ul> <p>Jika email dan password sesuai, tapi pengguna belum terdaftar di suatu rumah tertentu: Jawaban berupa string 'none'</p>



**Gambar 19 Diagram Alir Proses Sign In**

Kunci AES digenerasi sebelum data ini dikirimkan. Kunci ini akan digunakan oleh akun pengirim. Dengan demikian, setiap kali pengiriman pesan berikutnya (setelah berhasil *sign in*) akan dienkripsi menggunakan AES menggunakan kunci ini. Pesan ini masih dienkripsi menggunakan RSA karena kunci AES baru saja digenerasi.

- Di bawah tombol “SIGN IN” terdapat tulisan “*Forget Password?*”, jika tulisan ini ditekan, *user* dapat menuju ke halaman yang akan membantu *user* dalam mengetahui *password* aplikasi.

#### 4.3.5.3 Forget Password Activity

Seperti yang sebelumnya sudah dijelaskan, halaman ini berguna untuk mengingatkan *user* akan *password* aplikasi ini.



**Gambar 20 Halaman Forget Password**

Pada halaman ini, terdapat sebuah gambar dan keterangan halaman. Dibawah keterangan halaman, terdapat kolom *input email user*. Dengan memasukan *email user*, aplikasi dapat mengirimkan *password* aplikasi sang *user* melalui *email user* tersebut. Di bawah kolom *input*, terdapat tombol untuk melakukan eksekusi permintaan *password*. Ketika tombol “SEND REQUEST” ditekan, aplikasi akan mengirimkan permintaan ini ke server. Bentuk paket data *request* yang dimaksud adalah sebagai berikut.



**Tabel 15 Data JSON Forget Password**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	forgetpwd	<ul style="list-style-type: none"> <li>Email</li> </ul>	Server	RSA	'1' jika email terdaftar '0' jika email tidak terdaftar

#### 4.3.5.4 Halaman Sign Up

Halaman ini digunakan untuk mendaftarkan *email* serta *password* yang digunakan untuk aplikasi Rumah Cerdas.

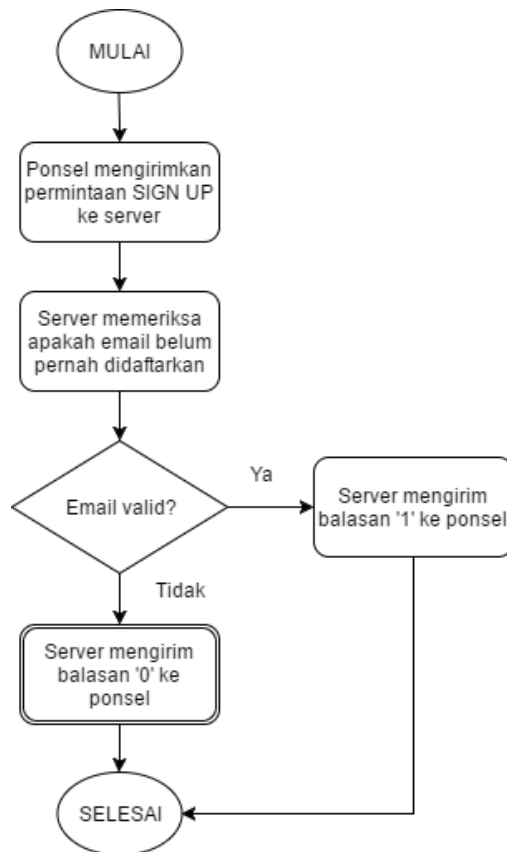
**Gambar 21 Halaman Sign Up**

Isi dari halaman ini dijelaskan sebagai berikut.

- Seperti pada halaman *sign in*, halaman ini memuat gambar berupa logo aplikasi dan terdapat tiga buah kolom input.
- Untuk mendaftar, *user* perlu memasukan data untuk identifikasi berupa nama, *email* dan *password* serta konfirmasi dari *password* yang sudah dimasukan sebelumnya untuk memastikan bahwa *password* yang diketikan benar. Ketika *password* dan konfirmasi *password* yang diketikan berbeda, akan terdapat peringatan yang memperingatkan bahwa *password* dan konfirmasi *password* tersebut berbeda. Selain dengan menggunakan konfirmasi *password*, terdapat juga tombol bergambar mata yang dapat menampilkan *password* seperti yang sudah dijelaskan sebelumnya untuk memastikan bahwa *password* yang dimasukan benar.
- Dengan menekan *Button* “SIGN UP” *user* dapat mendaftarkan data yang sebelumnya sudah diisi. Penekanan tombol ini akan memerintahkan aplikasi untuk mengirim data JSON sebagai berikut.

**Tabel 16 Data JSON Sign In**

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

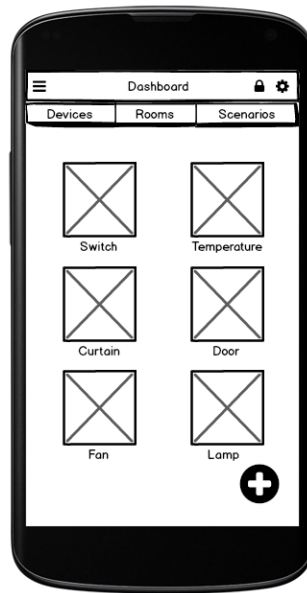


**Gambar 22 Diagram Alir Sign Up**

Keterangan: sama seperti pada proses *sign in*, kunci AES digenerasi sebelum pendaftaran dan akan digunakan setelah proses *sign up* ini terlewati. Karena kunci AES baru digenerasi pada tahap ini, maka enkripsi yang digunakan untuk pesan ini adalah RSA.

#### 4.3.5.5 Halaman Utama

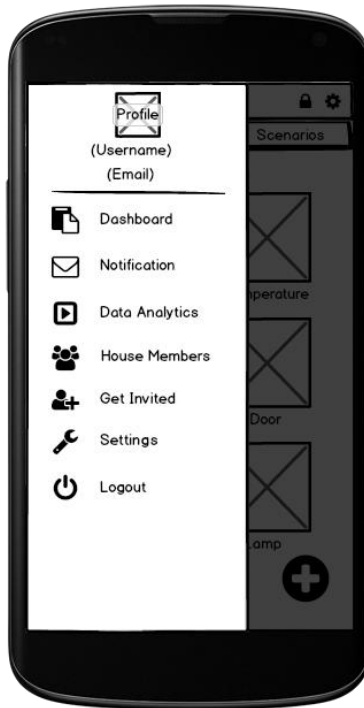
Menu utama dalam aplikasi ini memuat berbagai pengaturan *device* yang dikemas dalam berbagai kategori. *User* dapat mengakses seluruh *device* rumah cerdas yang dikelompokkan berdasarkan jenis *device*, lokasi *device* maupun skenario aktivasi *device*.



Gambar 23 Dashboard

Isi dari halaman ini adalah sebagai berikut.

- Pada saat *user* memasuki halaman ini, *user* dapat melihat *title bar* dan *tab bar* kategori pengaturan *device* pada bagian atas. Pada *title bar*, terdapat nama halaman yang sedang dibuka. Khusus pada halaman ini, *title bar* juga memuat tiga buah tombol. Tombol berbentuk tiga garis dibagian kiri dapat digunakan untuk memunculkan *navigation drawer*, tombol dibagian kanan yang berbentuk gembok dapat digunakan untuk mengunci aplikasi sedangkan tombol terakhir dapat dapat digunakan untuk membuka halaman *settings*.
- *Navigation drawer* digunakan untuk menampilkan informasi mengenai *user* baik *profile picture user*, *email* dan nomor *smartphone user*. Dibawah informasi mengenai *user*, terdapat enam buah menu yaitu menu *dashboard*, *notification*, *data analytics*, *house members*, *get invited*, *settings*, dan juga *logout*. Setiap menu akan membuka halamannya tersendiri ketika ditekan kecuali menu *logout* yang akan mengeluarkan *user* dari aplikasi dan melakukan *clear* terhadap informasi di dalam aplikasi Rumah Cerdas. Ketika tombol *logout* ditekan, aplikasi segera menampilkan halaman awal dari aplikasi Rumah Cerdas.



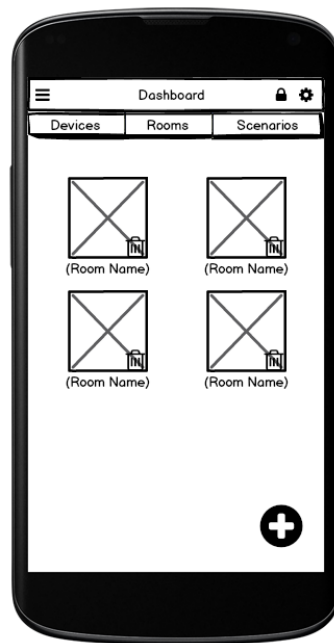
**Gambar 24 Dashboard Navigation Drawer**

- Tombol gembok dapat digunakan untuk mengaktifkan *scenario lock*. *Scenario lock* dapat dimodifikasi untuk mengunci akses utama sebuah rumah. Ketika tombol gembok ditekan, seluruh aplikasi Rumah Cerdas pada setiap member rumah agar *device* maupun pengaturan lainnya tidak dapat diubah kecuali gembok tersebut dibuka dengan cara menekan tombol yang sama. Ketika gembok dibuka, seluruh member akan mengetahui member mana yang membuka gembok melalui fitur *notification*.



**Gambar 25** Keadaan dashboard sedang terkunci

- Pada halaman *dashboard*, secara *default* menu yang ditampilkan merupakan menu *tab bar devices* yaitu enam buah menu yang dikelompokkan berdasarkan jenis *device* yaitu “Switch”, “Temperature”, “Curtain”, “Door”, “Fan”, dan “Lamp” sehingga ketika *user* menekan salah satu menu dalam tampilan tersebut, *user* dapat mengakses halaman yang mengatur seluruh *device* dengan jenis yang sesuai dengan menu tersebut.
- Ketika *user* menekan *tab bar rooms*, *user* dapat melihat menu yang ditampilkan berdasarkan lokasi setiap *device* sehingga ketika salah satu menu ruangan ditekan, akan ditampilkan seluruh *device* yang berada di ruangan tersebut. *User* dapat menekan tombol bergambar tempat sampah yang terletak diujung kanan bawah dari setiap menu *rooms* untuk melakukan *edit* maupun menghapus menu *room* tersebut.



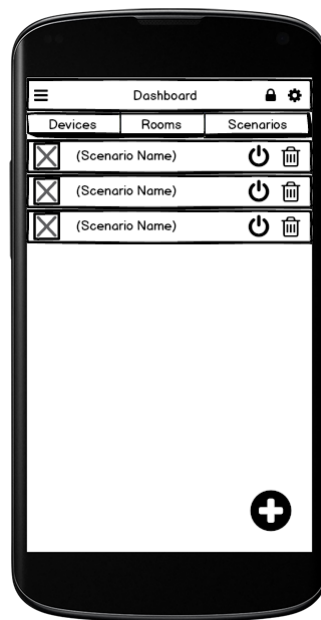
**Gambar 26 Dashboard Room**

Untuk menghapus atau mengedit ruangan, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 17 Data JSON Delete dan Edit Room**

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

- *Tab bar* terakhir adalah *tab bar scenarios*, pada *tab bar* ini *user* dapat mengaktifkan berbagai skenario yang sebelumnya sudah didaftarkan dengan cara menekan tombol bergambar *power* yang terletak dibagian kanan *scenario*, pengguna juga dapat melakukan *edit* ataupun *delete* terhadap suatu menu *scenario* dengan cara menekan tombol bergambar tempat sampah yang terletak disebelah tombol dengan gambar *power*.



**Gambar 27 Dashboard Scenario**

Untuk melakukan penghapusan atau pengeditan skenario, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 18 Data JSON Edit dan Delete Scenario**

Header	Type	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	deletescenario	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama skenario yang hendak dihapus</li> </ul>	Host dan Server	AES	Tidak ada jawaban
	editscenario	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama skenario</li> </ul>	Host dan Server	AES	Tidak ada jawaban



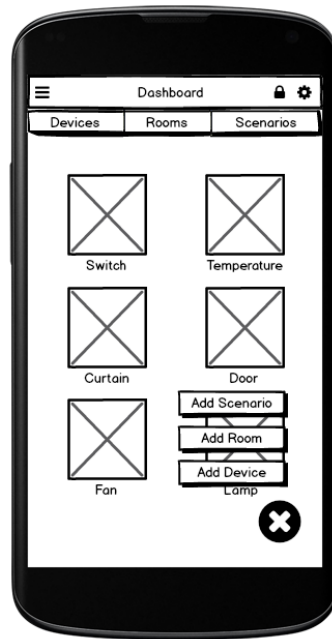
		yang hendak diedit <ul style="list-style-type: none"> <li>• Nama Baru</li> <li>• Daftar device baru di dalam skenario berikut statusnya</li> </ul>			
--	--	--	--	--	--

Untuk mengaktifkan skenario, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 19 Data JSON Pengaktifan Skenario**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
SC (Scenario)	-	<ul style="list-style-type: none"> <li>• Home ID</li> </ul>	Host	AES	Tidak ada jawaban

- Terdapat juga tombol berupa *Floating Button* bergambar tanda tambah yang akan tetap berada pada tempat yang sama walaupun *user* mengganti menu pada *tab bar*. Tombol ini berguna untuk menampilkan menu tambahan untuk menambah *device*, *room* maupun *scenario*.



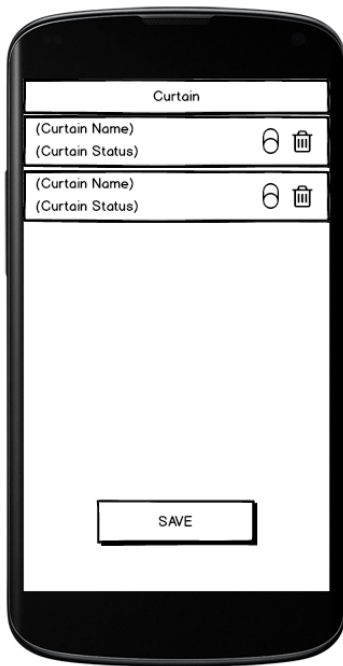
**Gambar 28 Tombol Floating Button**

Ketika halaman utama ini pertama kali dibuka, aplikasi akan secara otomatis meminta *request* daftar device, daftar room, dan daftar skenario dari *host*. Dengan daftar ini, maka isi dari masing-masing *tab* bisa didefinisikan. Pesan *request* ini adalah sebagai berikut.

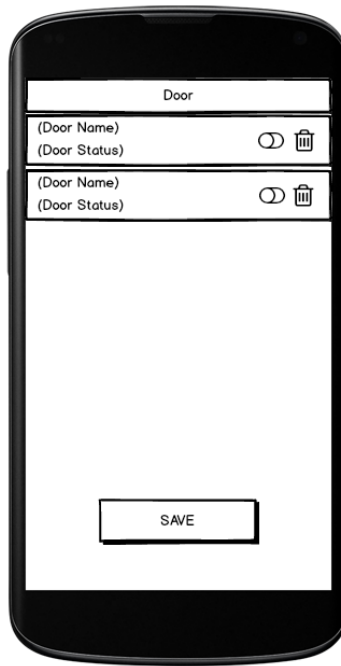
**Tabel 20 Data JSON Loading Data dari Host**

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

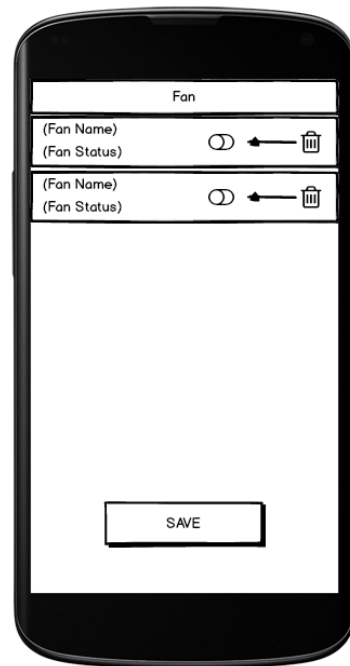
#### 4.3.5.6 Halaman Device



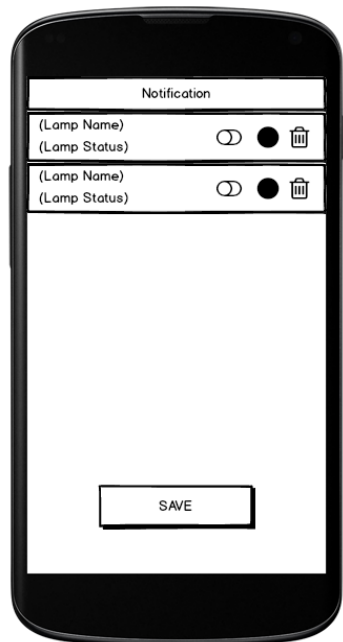
Gambar 29 Menu curtain



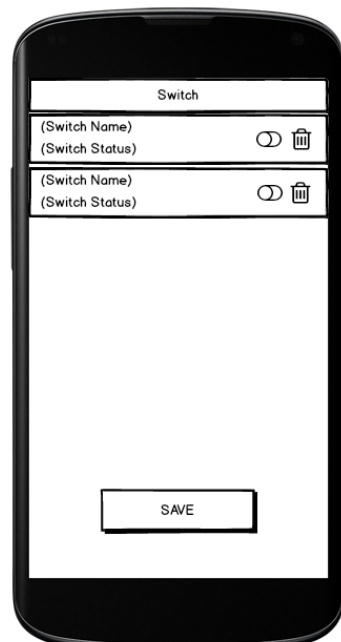
Gambar 30 Menu door



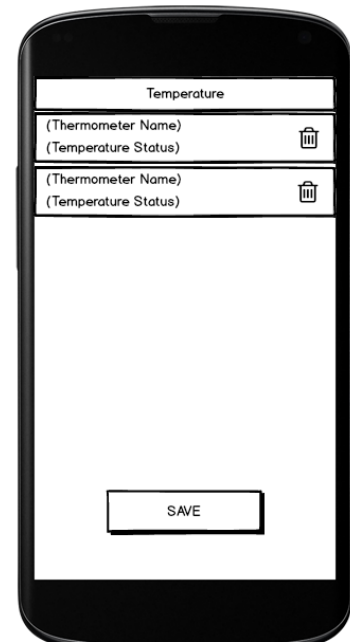
Gambar 31 Menu fan



Gambar 32 Menu lamp



Gambar 33 Menu switch



Gambar 34 Menu temperature

Pada bagian ini dijelaskan enam buah halaman *device* dalam *tab bar devices* yang terdapat pada halaman sebelumnya. Pada saat *user* menekan salah satu menu pada tampilan

menu *device*, pengguna akan langsung diarahkan menuju halaman 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 menaikkan tirai.
- Menu “Door” menampilkan seluruh pintu yang telah terpasang *device* untuk mendukung sistem penguncian dengan menggunakan aplikasi Rumah Cerdas, sehingga tampilan pada menu “Door” serupa dengan tampilan pada menu “Switch” karena hanya mengandalkan *Switch toggle*.
- Menu “Fan” menggunakan *Seek bar* sebagai pengganti *switch* karena 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 tombol berbentuk lingkaran dan menggeser *pointer* dari warna yang ditunjukkan.
- Pada setiap menu terdapat tombol bergambar tempat sampah untuk menghapus *device* yang telah didaftarkan. Pada setiap halaman *device* juga terdapat tombol “SAVE” dibagian bawah untuk menyimpan semua pengaturan *device* yang telah dilakukan.

Untuk memberikan perintah tertentu kepada suatu *device* (*device* jenis apapun), maka aplikasi akan mengirimkan data dengan paket sebagai berikut.

**Tabel 21 Data JSON untuk Perintah ke Perangkat**

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

		dengna jenis perangkat, dijelaskan pada 3.1.4.			
--	--	--	--	--	--

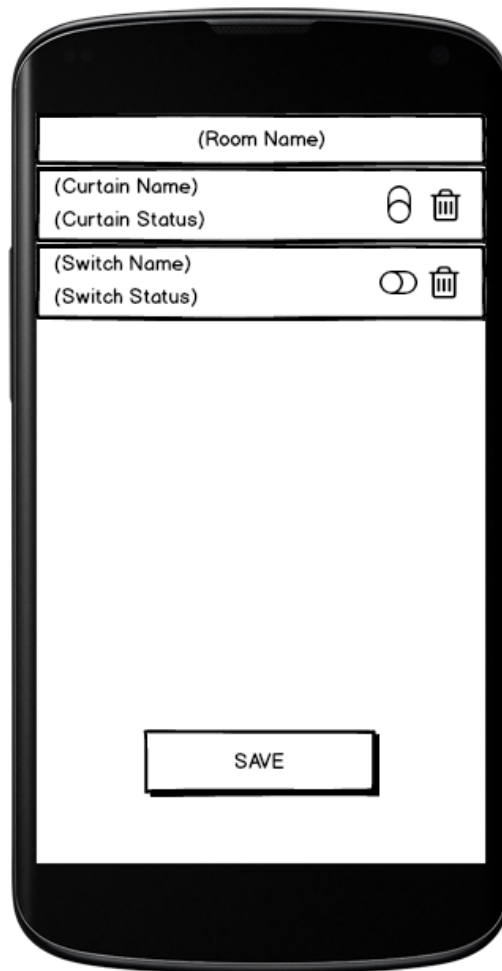
Untuk melakukan penghapusan dan pengeditan nama perangkat, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 22 Data JSON Delete dan Edit Perangkat**

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

#### 4.3.5.7 Halaman Room

Pada halaman ini, *user* dapat memilih *device* yang akan diaktifkan berdasarkan lokasi dari *device* tersebut. *Tab rooms* digunakan untuk mempermudah *user* dalam memilih *device* dengan cepat dan tepat sehingga ketika *user* berada di suatu ruangan dan ingin mengubah status dari sebuah *device*, *user* tidak perlu mencari *device* tersebut diantara seluruh *device* dalam sistem Rumah Cerdas.



**Gambar 35 Halaman Room**

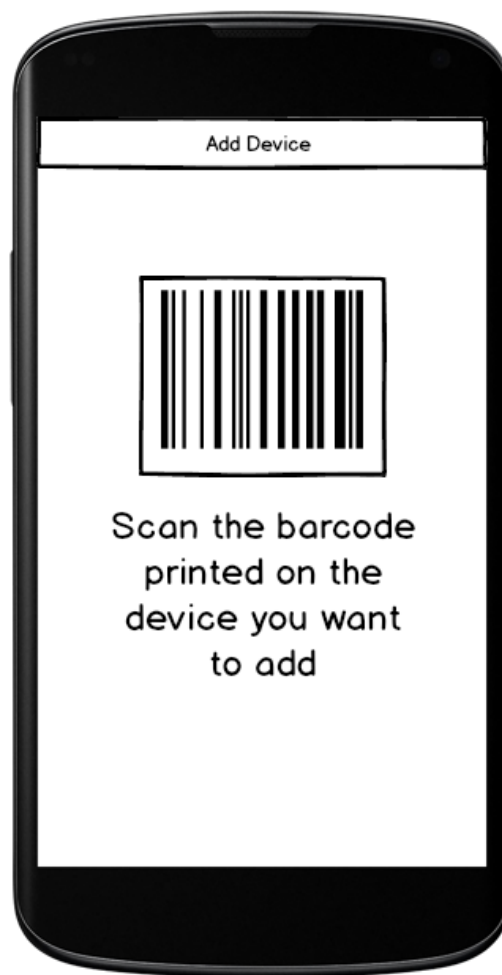
Pada saat *user* menekan salah satu ruangan pada halaman utama, *user* akan menuju halaman *room* tersebut. Halaman *room* akan menyesuaikan tampilan dan konten dari halaman sesuai dengan ruangan yang dipilih, sehingga halaman *room* dapat menampilkan *list* dari *device* yang sesuai pada ruangan tersebut. *User* dapat segera mengganti status dari *device* pada *list device* tersebut seperti pada tampilan *device* pada halaman *device* sebelumnya, hanya saja seluruh jenis *device* dijadikan satu. Paket data yang dikirimkan untuk memberikan perintah sama persis dengan yang ada pada halaman *device*.

#### 4.3.5.8 Halaman Add Device

Halaman ini dapat diakses dari menu pada *Floating Button* yang terdapat pada halaman utama. Pada halaman ini, *user* dapat mendaftarkan *device* baru yang sudah dipasang 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 di-*scan*. Untuk meningkatkan keamanan, *QR Code* merupakan ID *device* yang sudah terenkripsi. Enkripsi dilakukan menggunakan metode RSA. Dengan demikian, tidak akan ada pihak yang mengetahui berapa *address* dari *device* ini.

*User* dapat melakukan *scan* dengan cara menekan gambar *QR Code* pada halaman ini, dan kamera *smartphone* akan segera diaktifkan. Ketika *scan* berhasil, *smartphone* akan mendaftarkan *device* baru tersebut kepada server. Pada kode setiap *device* mengandung informasi kategori *device* tersebut sehingga *user* tidak perlu menuliskan kategori *device* yang ingin didaftarkan karena sistem Rumah Cerdas dapat secara otomatis mendaftarkan *device* tersebut sesuai dengan kategorinya. Pada bagian bawah gambar *QR Code*, terdapat tulisan untuk memperjelas tugas halaman dan tombol “BACK” untuk kembali ke halaman sebelumnya.



**Gambar 36 Halaman Add Device**

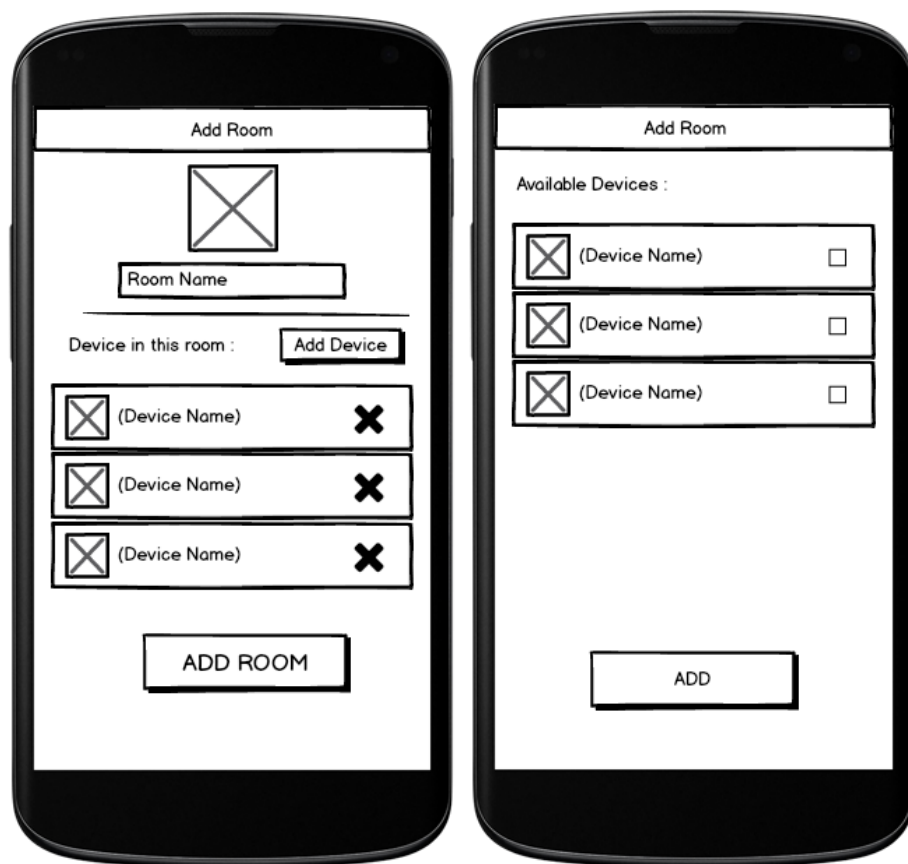
Untuk melakukan penambahan *device*, perintah yang dikirimkan adalah:

**Tabel 23 Data JSON Add Device**

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

#### 4.3.5.9 Halaman Add Room

Seperti pada halaman sebelumnya, halaman ini dapat diakses melalui *Floating Button* pada halaman utama dengan menekan tombol “Add Room”.





### Gambar 37 Halaman Add Room

Pada halaman ini, *user* dapat menambah kategori *room* yang akan ditampilkan pada *tab bar rooms*. Untuk menambah sebuah *room*, dibutuhkan nama *room* dan *device* apa saja yang berada di dalam *room* tersebut. Pada halaman ini, *user* dapat memasukan nama *room* dalam kolom *input* yang berada dibawah gambar. Setelah itu terdapat tombol “Add Device” yang berguna untuk mendaftarkan *device* ke dalam suatu *room*. Tombol “Add Device” akan mengantarkan *user* ke dalam halaman baru untuk menambahkan *device* sehingga *user* dapat memilih *device* dari *list device* yang sudah tersedia dengan cara menekan *check box* yang terletak disebelah kanan *list device*. Ketika sudah selesai memilih *device*, *user* dapat menekan tombol “ADD” sehingga aplikasi segera menambahkan *device* yang dipilih dan *user* segera kembali pada tampilan halaman sebelumnya. Jika *user* ingin menghapus *device* yang sudah dipilih, *user* dapat menekan tombol “x” yang terletak di sebelah nama *device* pada halaman *add room*. Pada saat tombol “Add Room” ditekan, seluruh pengaturan penambahan ruangan segera disimpan kedalam server dan *user* dapat melihat adanya *room* baru pada *tab bar rooms*.

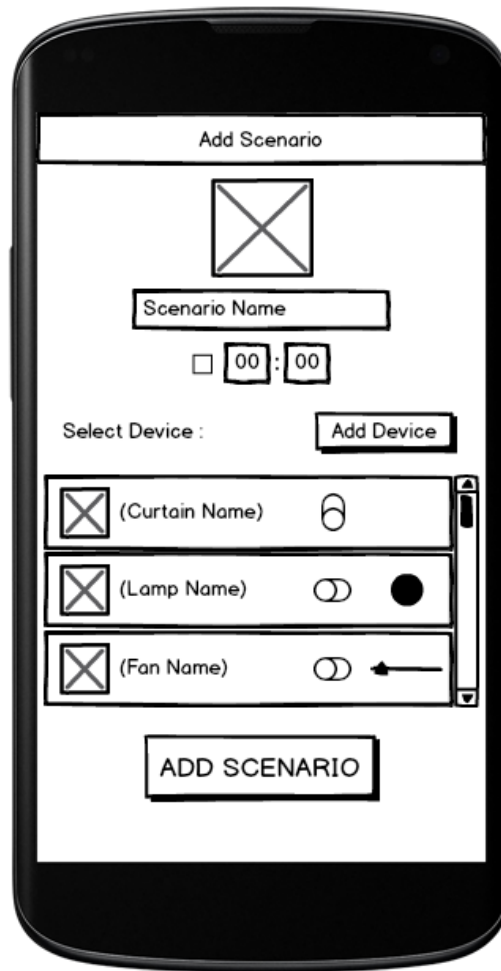
Untuk menambah sebuah ruangan, paket data yang dikirimkan adalah sebagai berikut.

Tabel 24 Data JSON Add Room

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

#### 4.3.5.10 Halaman Add Scenario

Halaman ini dapat diakses dengan menekan tombol “Add Scenario” pada *Floating Button* di halaman utama.



**Gambar 38 Halaman Add Scenario**

Pada halaman ini *user* dapat menambahkan *scenario* penggunaan *device* pada sistem Rumah Cerdas. *Scenario* yang dimaksudkan adalah mode penggunaan *device*, contohnya adalah ketika *user* memilih *scenario* “Sunday Morning” maka lampu taman, lampu balkon dan lampu teras dimatikan, seluruh tirai dibuka dan pintu depan terbuka. Contoh lainnya adalah pada saat *user* 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, *user* tidak perlu menghabiskan banyak waktu untuk mengaktifkan *device* satu per satu, cukup dengan mengaktifkan *scenario* yang didaftarkan saja. Untuk mendaftarkan *scenario*, *user* perlu memasukan nama *scenario* dan *device* apa saja yang perlu diaktifkan. Jika *device* berupa *fan* pengguna dapat melakukan konfigurasi dengan mengatur *SeekBar* pada *device* tersebut. Begitu juga dengan *lamp*, pengguna dapat memilih warna lampu terlebih dahulu. Tampilan pada menu ini serupa dengan tampilan pada menu *add room*, hanya saja pada halaman ini terdapat kolom *input* berupa waktu yang

dapat digunakan *user* untuk otomatisasi pengubahan status *device*. *List device* pada halaman ini juga berbeda dengan halaman *add room* karena pada halaman ini, *list* menampilkan status *device* yang sudah diatur sebelumnya. Dengan melakukan pengaturan *scenario* berdasarkan waktu, *user* dapat mengganti status sebuah maupun beberapa *device* secara bersamaan dengan otomatis.

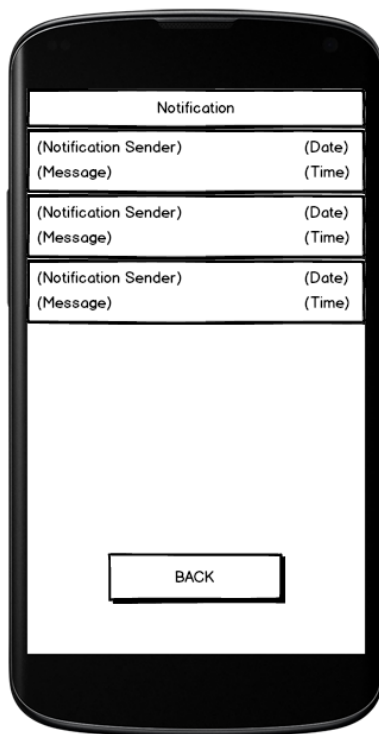
Untuk menambah skenario, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 25 Data JSON Add Scenario**

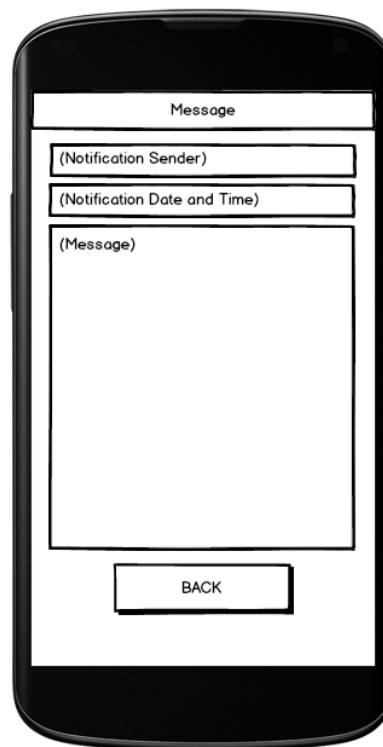
Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	addscenario	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama skenario</li> <li>• Daftar device yang ada dalam skenario dan aksinya</li> </ul>	Host dan Server	AES	Tidak ada jawaban

#### 4.3.5.11 Halaman Notifikasi dan Pesan

*User* dapat menerima berbagai informasi mengenai aplikasi Rumah Cerdas pada halaman notifikasi.



**Gambar 39 Halaman Notifikasi**



**Gambar 40 Halaman Pesan**

Notifikasi akan muncul bilamana salah satu dari hal berikut terjadi.

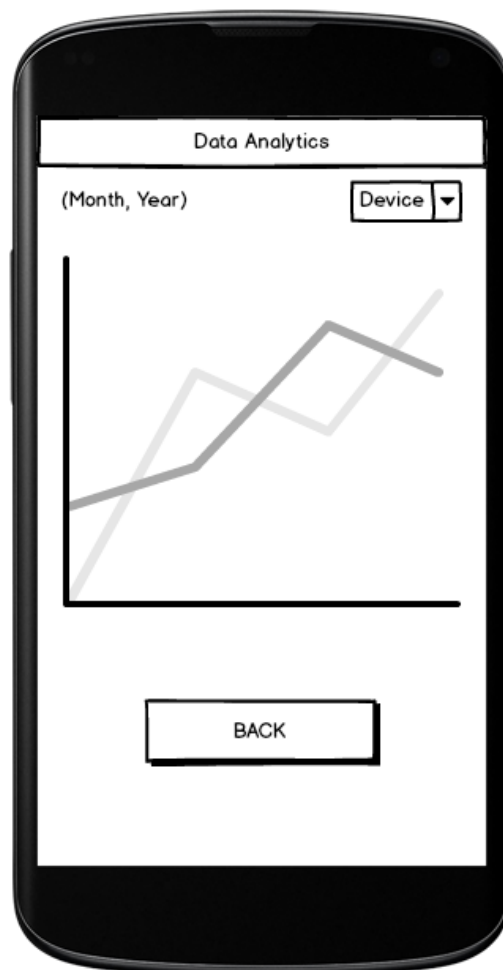
- Ada *user* baru yang diundang oleh ADMIN ke dalam rumah. Ketika ada orang baru yang mendaftarkan diri pada aplikasi Rumah Cerdas dalam suatu rumah, Admin rumah segera mendapatkan pesan permintaan penambahan member seperti yang sebelumnya sudah dibahas. Admin akan menerima enam buah angka yang dapat dimasukan pengguna baru ke dalam halaman *get invited* untuk menyelesaikan tahap pendaftaran.
- Selain urusan pendaftaran, halaman notifikasi 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.

Pada halaman notifikasi ditampilkan nama pengirim, waktu pengiriman dan potongan dari pesan yang diterima. Untuk melihat pesan secara utuh, *user* dapat menekan notifikasi sehingga *user* beralih ke halaman pesan. Pada tampilan halaman pesan, terdapat tiga buah tulisan untuk menampilkan informasi pesan yaitu tulisan yang memuat nama pengirim

notifikasi, tanggal dan waktu notifikasi dikirim dan isi notifikasi tersebut. Pada bagian bawah dari halaman pesan terdapat tombol “BACK” untuk kembali pada halaman notifikasi.

#### 4.3.5.12 Halaman Analisis Data

Untuk memantau penggunaan keseluruhan *device* dalam rumah, *user* dapat melihat grafik penggunaan *device* yang ditampilkan pada halaman ini. Halaman ini tergolong penting untuk membantu *user* melakukan efisiensi terhadap penggunaan energi di dalam rumah. Ketika terjadi peningkatan penggunaan energi, *user* dapat segera mematikan *device-device* yang dirasa tidak perlu untuk digunakan.



**Gambar 41 Halaman Analisis Data**

Grafik yang ditunjukkan merupakan grafik waktu penggunaan dalam satuan menit atau jam terhadap tanggal pada bulan tersebut pada *device* yang dipilih oleh *user* melalui *spinner* dibagian kanan. Diatas grafik terdapat tulisan yang menunjukkan bulan dan tahun dari grafik yang sedang ditampilkan. Terdapat juga tombol “Back” yang berfungsi untuk

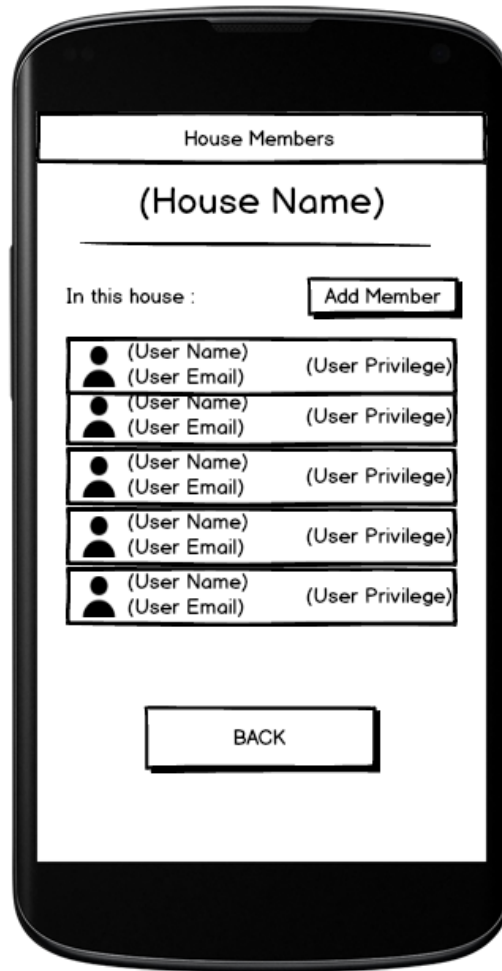
mengembalikan tampilan menjadi halaman utama. Ketika halaman ini dibuka, maka secara otomatis aplikasi akan mengirimkan *request* data dari *host* dengan paket data sebagai berikut.

**Tabel 26 Data JSON Permintaan Data Penggunaan Perangkat**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
DA (Data)	-	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Address device yang ingin dilihat datanya</li> </ul>	<i>Host</i>	AES	Data penggunaan device selama 1 bulan

#### 4.3.5.13 Halaman Member

Halaman ini dapat dibuka dengan menekan menu “House Members” pada *navigation drawer* bila *user* adalah Admin.



**Gambar 42 Halaman Member**

Pada halaman ini *user* dapat melihat nama rumah yang telah didaftarkan sebelumnya dan ditampilkan pada bagian atas halaman ini. Setelah itu, *user* juga dapat mendaftarkan member baru dengan cara menekan tombol “Add Member” yang terletak di bawah pembatas informasi rumah. Pada halaman ini, *user* dapat melihat seluruh *user* lainnya yang terdaftar pada rumah yang sama pada aplikasi Rumah Cerdas. Dalam *list member* tersebut, *user* 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 diatur 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. 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 tombol

“Add Member”, pengguna diarahkan menuju halaman *add member* sedangkan ketika pengguna menekan tombol “BACK”, Pengguna diarahkan menuju halaman utama.

Ketika halaman ini dibuka, aplikasi secara otomatis mengirimkan *request* daftar seluruh *user* di dalam rumah dari server. Daftar inilah yang menjadi acuan isi dari *users list* pada halaman ini. Paket data *request* yang dikirimkan adalah sebagai berikut.

**Tabel 27 Data JSON Permintaan List User**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen Users)	list	<ul style="list-style-type: none"><li>Home ID</li></ul>	<i>Server</i>	AES	Daftar seluruh <i>users</i> yang terdaftar di rumah yang bersangkutan

#### 4.3.5.14 Halaman Add Member

Halaman ini diakses dengan menekan tombol “Add Member” pada halaman member.





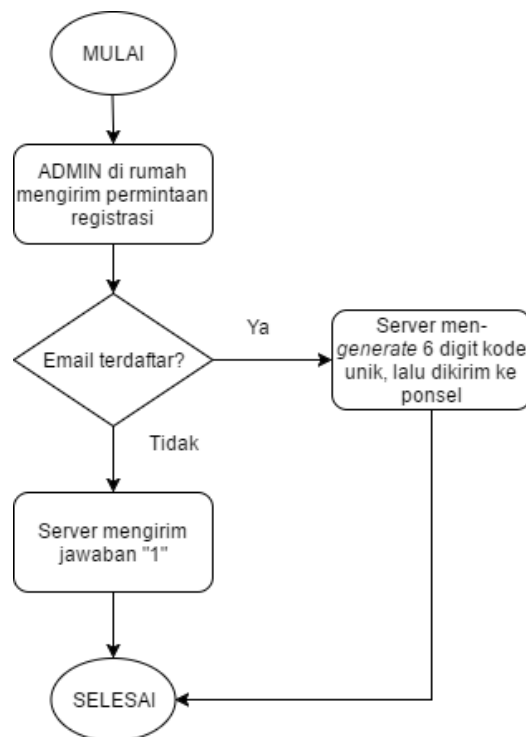
**Gambar 43 Halaman Add Member**

Halaman ini memiliki fungsi sebagai sarana bagi tuan rumah untuk mengirimkan invitasi kepada *user* lain yang akan ikut mengakses rumahnya tersebut. Pada activity ini, terdapat sebuah kolom *input* yang penting, yaitu *email* dari *user* yang ingin diundang. Selain itu, terdapat pula sebuah *Spinner* yang berisi jenis *privilege* apa yang hendak diberikan kepada *user* tersebut. Ada 2 jenis *privilege*, yaitu *admin* dan *guest*. Masing-masing memiliki perbedaan pada kedalaman akses rumah. Admin dapat melakukan manajemen perangkat, misalnya menambah perangkat, menambah *room*, atau menambah *scenario*. Admin juga bisa mengakses seluruh perangkat yang ada, sedangkan *guest* hanya bisa mengakses perangkat tertentu saja yang telah ditentukan oleh Admin.

Ketika tombol “SEND INVITATION” ditekan, maka aplikasi akan mengirimkan pesan kepada server untuk mendaftarkan *user* tersebut masuk ke dalam rumah. Paket pesan yang dikirimkan adalah sebagai berikut.

**Tabel 28 Data JSON untuk Permintaan Register**

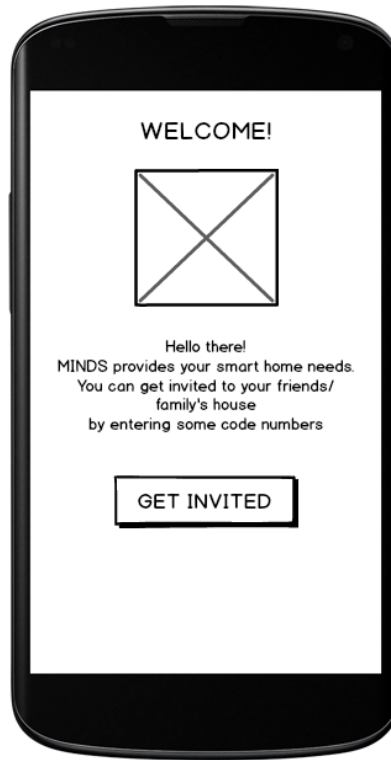
Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	register	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Email yang ingin diundang</li> <li>• Privilege</li> </ul>	Server	AES	Bila email terundang valid (belum terdaftar di rumah lain): <ul style="list-style-type: none"> <li>• 6 digit kode registrasi</li> </ul> Bila email terundang sudah ada di rumah lain: <ul style="list-style-type: none"> <li>• Karakter '1'</li> </ul>



**Gambar 44 Diagram Alir Register**

#### 4.3.5.15 Halaman *GettingStarted*

Halaman ini hanya akan ditampilkan kepada *user* ketika *user* menekan tombol “Get Invited” pada *navigation drawer*. Halaman ini hanya dapat dibuka oleh *user* yang belum teregistrasi di rumah manapun. Bila *user* telah terdaftar di salah satu rumah, maka *user* tidak akan memasuki halaman ini ketika tombol pada drawer ditekan, melainkan akan muncul pesan *toast* “You are already registered to a home”.

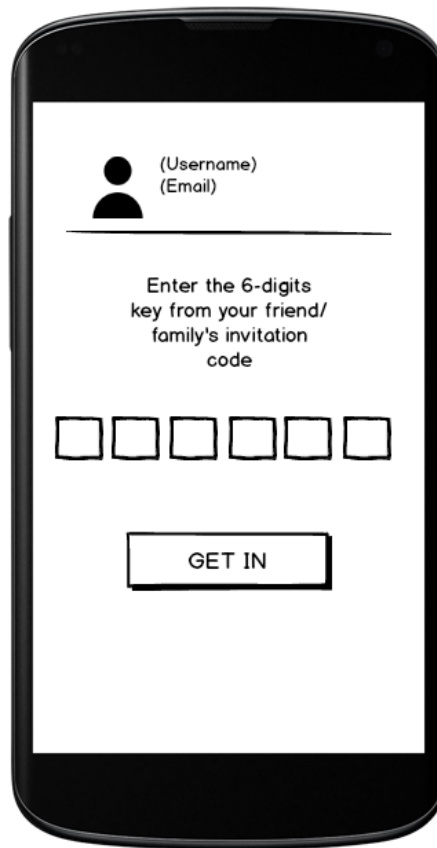


**Gambar 45 Halaman Getting Started**

Pada halaman ini terdapat beberapa tulisan untuk memberitahu *user* mengenai aplikasi ini dan sebuah tombol “GET INVITED” untuk membawa *user* kepada halaman yang dapat digunakan untuk *input invitation code* (halaman *get invited*).

#### *4.3.5.16 Halaman Get Invited*

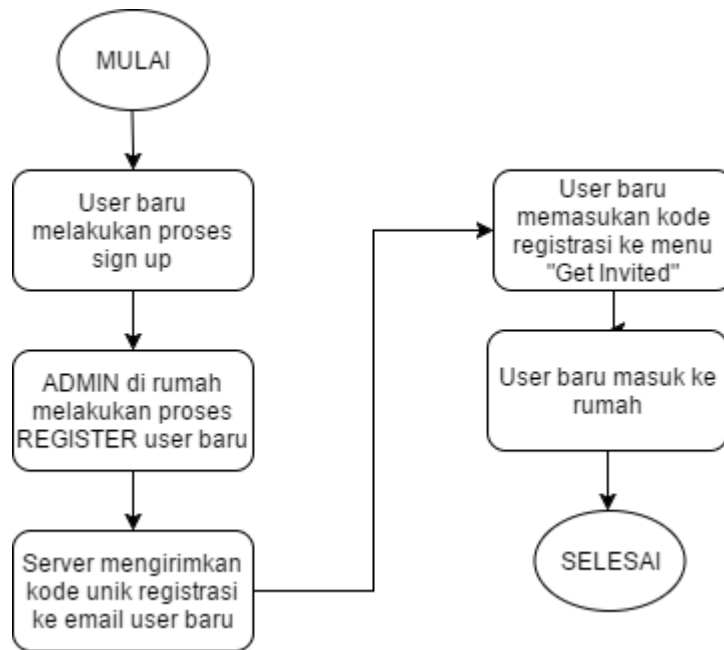
Halaman ini hanya dapat diakses oleh *user* baru setelah menekan tombol “GET INVITED” pada halaman sebelumnya. *User* baru dapat menggunakan halaman ini untuk memasukan *invitation code* dari admin rumah agar dapat bisa menggunakan aplikasi Rumah Cerdas untuk mengatur *device* pada rumah yang terdaftar.



**Gambar 46 Halaman Get Invited**

Untuk mendapatkan *invitation code*, seorang admin di suatu rumah perlu melakukan *request* kepada server dengan cara mengisi identitas *user* baru pada Halaman Member. Setelah mengajukan penambahan member baru, admin akan mendapatkan notifikasi berupa enam digit angka *invitation code* yang dapat diisi oleh *user* baru pada halaman ini. Pada halaman ini terdapat gambar *profile* dan identitas *user* baik *email*, *username* maupun nomor *smartphone* dibagian atas. Sedangkan dibagian bawah terdapat enam buah kolom *input* untuk memasukan *invitation code*. Dibawah kolom *input invitation code*, terdapat tombol untuk memverifikasi kode. *User* baru yang berhasil mendaftarkan diri menggunakan *invitation code* yang dikirimkan oleh admin langsung dapat menggunakan aplikasi Rumah Cerdas dengan konfigurasi yang sudah diatur oleh admin. Sedangkan ketika *invitation code* salah, akan ditampilkan peringatan dan *user* tetap berada pada halaman yang sama.

Secara sederhana, proses registrasi *user* ke suatu rumah digambarkan dalam diagram alir berikut.



**Gambar 47 Diagram Alir Proses Pendaftaran Seorang *User* ke Suatu Rumah**

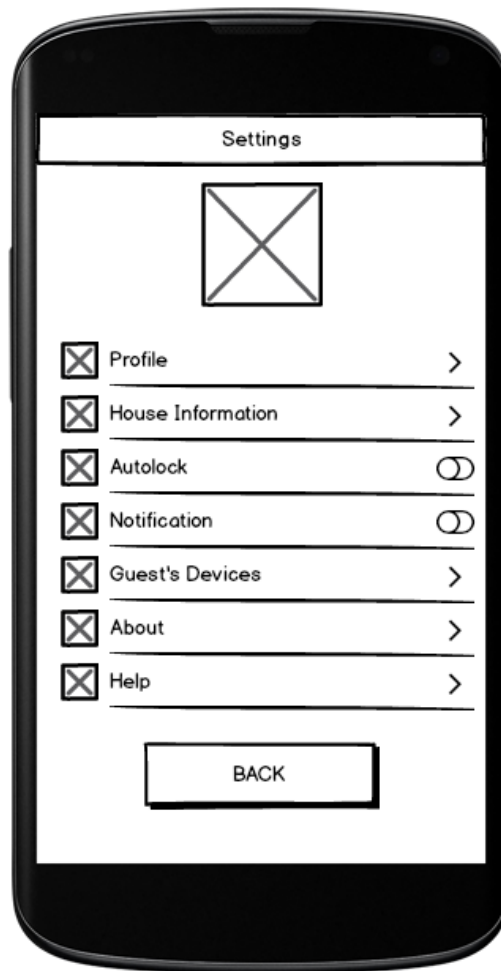
Ketika tombol “Get In” ditekan, maka aplikasi akan mengirimkan paket data sebagai berikut.

**Tabel 29 Data JSON untuk Konfirmasi User**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	Confirm	<ul style="list-style-type: none"> <li>6 digit kode konfirmasi</li> </ul>	Server	AES	Bila kode benar: <ul style="list-style-type: none"> <li>Home ID</li> </ul> Bila kode salah:           Jawaban berupa karakter ‘0’

#### 4.3.5.17 Halaman Setting

*User* dapat mengatur beberapa hal penting pada aplikasi Rumah Cerdas melalui halaman ini. Hal yang dapat diubah melalui halaman ini adalah profil, informasi rumah, pengaturan *privilege*, *autolock*, dan notifikasi. Selain itu, terdapat informasi tambahan mengenai aplikasi Rumah Cerdas yang dapat dilihat dengan menekan menu “About” dan “Help”. Setiap menu mengarahkan pengaturan yang lebih lanjut melalui halamannya masing-masing ketika menu tersebut ditekan, kecuali untuk *autolock* dan *notification* yang statusnya dapat diubah secara langsung pada halaman ini.



**Gambar 48 Halaman Setting**

Fitur *autolock* digunakan untuk mengaktifkan *scenario lock* 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, sistem Rumah Cerdas akan mengaktifkan *scenario lock* pada rumah tersebut. Sistem *autolock* mengantisipasi keadaan ketika *user* lupa mengunci rumah ketika berpergian. *Switch bar* pada *notification* menandakan status notifikasi *user*. Ketika *Switch bar* berada pada posisi ON, maka *user* memperbolehkan aplikasi Rumah Cerdas untuk mengirimkan notifikasi kepada *smartphone user*, begitu juga sebaliknya. *Notification* selalu dikirimkan kepada *smartphone* setiap saat walaupun aplikasi Rumah Cerdas dalam keadaan tertutup sekalipun. Pada bagian bawah, terdapat tombol “Back” yang dapat mengarahkan pengguna menuju halaman utama ketika ditekan.

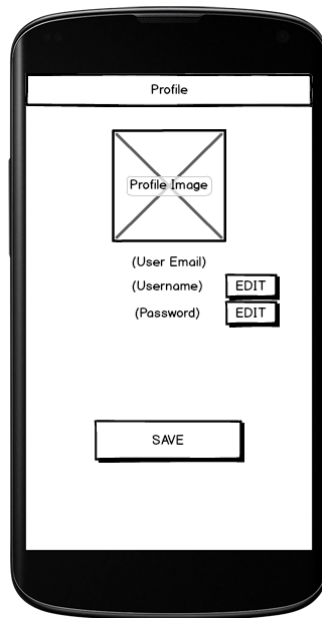
#### 4.3.5.18 Halaman User

Pada Halaman ini, *user* dapat mengganti beberapa informasi pada profil seperti nama dan *password* akun aplikasi Rumah Cerdas. Nama dan *password* yang sukses diganti akan langsung ditampilkan pada *Navigation Drawer* sebagai *preview* dari profil *user*. *User* 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. Untuk melakukan edit profil, paket data yang dikirimkan ke server adalah sebagai berikut.

**Tabel 30 Data JSON Edit Profil User**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MU (Manajemen User)	editprof	<ul style="list-style-type: none"><li>• Email user</li><li>• Nama baru</li><li>• Foto baru</li><li>• Password baru</li></ul>	Server	AES	Tidak ada jawaban

Jika *user* memiliki *privilege* admin, maka *user* dapat mengubah daftar *device* yang dapat digunakan oleh member lain dengan jenis *privilege* guest dengan cara menekan tombol “EDIT” yang berada disebelah kanan tulisan *privilege*. Setelah tombol “EDIT” tersebut ditekan, *user* segera menuju halaman *privilege* dan dapat segera mengganti daftar *device* yang dapat digunakan oleh guest dengan cara melakukan *check* ataupun *uncheck* pada *device* yang dipilih. Setelah menentukan pilihan, *user* dapat menekan tombol “SAVE” untuk menyimpan konfigurasi yang diinginkan.



**Gambar 49 Halaman User**

#### *4.3.5.19 Halaman Informasi Rumah*

Halaman ini digunakan untuk membantu *user* dalam mendaftarkan rumah mereka. *User* dapat memasukan nama dan alamat rumah mereka pada kolom *input* yang tersedia. Nama rumah dapat diisi sesuai dengan keinginan *user* untuk keperluan identifikasi alamat.



**Gambar 50 Halaman Create New**



Pada halaman ini, terdapat fitur sinkronisasi antara kolom *input* untuk alamat dan fitur Google Maps sehingga ketika kolom tersebut diisi dengan alamat, Google Maps dapat mencari posisi alamat tersebut dan *user* hanya perlu melakukan verifikasi terhadap alamat yang ditunjuk oleh *pointer* dengan cara menggeser letak *selected point* pada Google Maps. Setelah dilakukan verifikasi, Kolom *input* tersebut akan berisi alamat baru yang telah diverifikasi. Terdapat tombol “BACK” untuk membawa *user* untuk kembali ke halaman *settings*.

Ketika halaman ini terbuka, aplikasi secara otomatis meminta *request* data rumah dengan paket data *request* sebagai berikut.

**Tabel 31 Data JSON untuk List Data Rumah**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MH (Manajemen Home)	list	<ul style="list-style-type: none"> <li>• Home ID</li> </ul>	<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 32 Data JSON untuk Edit Data Rumah**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MH (Manajemen Home)	edit	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama Baru</li> <li>• Koordinat Rumah Baru</li> </ul>	<i>Host</i>	AES	Tidak ada jawaban

#### 4.3.5.20 Halaman Privilege

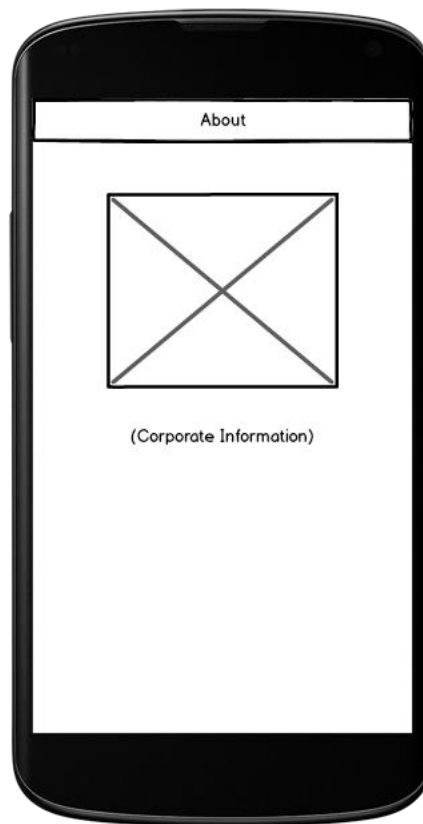
Pada halaman ini ditampilkan pengaturan *privilege* guest di dalam rumah. Jika *user* memiliki *privilege* admin, maka *user* dapat mengubah daftar *device* yang dapat digunakan oleh member lain dengan jenis *privilege* guest dengan cara melakukan *check* ataupun *uncheck* pada *device* yang dipilih. Setelah menentukan pilihan, *user* dapat menekan tombol “SAVE” untuk menyimpan konfigurasi yang diinginkan. Untuk melakukan perubahan *privilege* pada perangkat, paket data yang dikirimkan adalah sebagai berikut.

**Tabel 33 Data JSON untuk Edit *Privilege* Perangkat**

Header	Tipe	Isi Paket Data	Tujuan	Enkripsi	Jawaban dari Tujuan
MD (Manajemen Device)	editprivilege	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Daftar seluruh device beserta privilegenya</li> </ul>	<i>Host</i>	AES	Tidak ada jawaban

#### 4.3.5.21 Halaman *About*

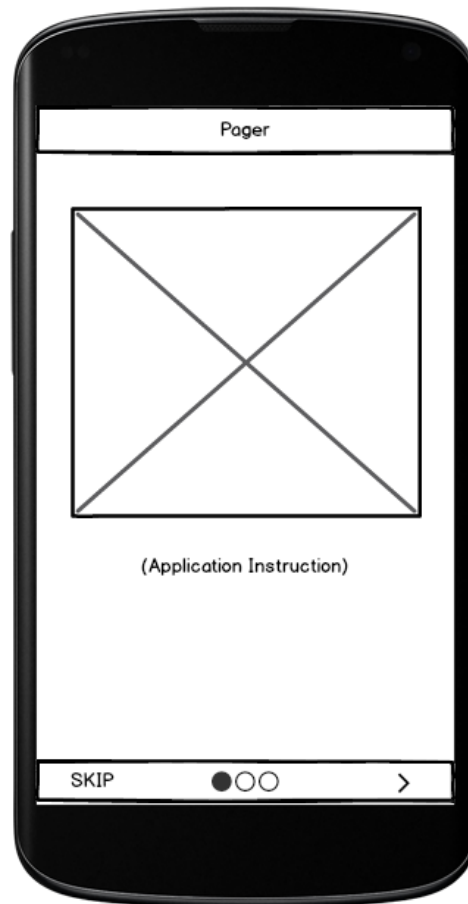
Pada halaman ini ditampilkan beberapa hal mengenai Rumah Cerdas baik dari sisi aplikasi maupun dari sisi pengembang aplikasi MINDS. Pada halaman ini *user* dapat melihat *credit* dari MINDS. Halaman ini dibuat dengan menggunakan gabungan beberapa gambar maupun tulisan.



**Gambar 51 Halaman *About***

#### 4.3.5.21 Halaman *Help*

Pada halaman ini, *user* diberitahu beberapa fitur pada aplikasi MINDS yang mungkin tidak *user* 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 halaman ini. *User* dapat mengganti tampilan halaman ini dengan cara *swipe* ataupun menekan tombol panah kanan pada *bar* di bagian bawah halaman ini.



**Gambar 52 Halaman Help**

#### **4.3.6 Perancangan Program pada Server**

Untuk menjawab seluruh permintaan dan pengiriman data dari aplikasi, maka server memerlukan suatu program untuk mewadahi semua paket data kiriman tersebut. Setiap pesan dengan header apapun harus bisa ditanggapi dengan baik. Rangkuman tanggapan server terhadap seluruh paket yang dikirimkan oleh aplikasi disajikan dalam tabel berikut.

**Tabel 34 Penjelasan Paket Data**

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"> <li>Email pengguna</li> <li>Password</li> </ul>	<p>Server mengecek email dan password ke database, apakah sesuai.</p> <p>Jika email dan password sesuai dan pengguna sudah terdaftar di suatu rumah, server menjawab:</p> <ul style="list-style-type: none"> <li>Home ID</li> <li>Nama pengguna</li> <li>Gambar profil</li> <li>Privilege pengguna (ADMIN atau Guest)</li> <li>User ID</li> </ul> <p>Jika email dan password tidak sesuai, server menjawab:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '0'</li> </ul> <p>Jika email dan password sesuai, tapi pengguna belum terdaftar di suatu rumah tertentu, server menjawab:</p> <ul style="list-style-type: none"> <li>Jawaban berupa string 'none'</li> </ul>

SU (Sign Up)	Ponsel	-	Permintaan untuk sign up atau mendaftarkan akun tertentu	<ul style="list-style-type: none"> <li>Email pengguna</li> <li>Password</li> <li>Nama pengguna</li> </ul>	<p>Server mengecek apakah email sudah terdaftar. Jika email belum pernah didaftarkan sebelumnya:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '1'</li> </ul> <p>Jika email sudah terdaftar:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '0'</li> </ul>
MU (Manajemen Users)	Ponsel	List	Permintaan daftar seluruh akun yang terdaftar untuk suatu rumah	<ul style="list-style-type: none"> <li>Home ID</li> </ul>	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"> <li>Home ID</li> <li>Email terundang</li> </ul>	<p>Server mengecek apakah email terundang belum terdaftar di rumah manapun. Bila email terundang belum terdaftar di rumah lain, server menjawab:</p> <ul style="list-style-type: none"> <li>Enam digit kode yang harus dimasukan di ponsel terundang supaya bisa bergabung dengan rumah (digenerasi secara acak oleh server)</li> </ul>

					<p>Bila email terundang sudah terdaftar di rumah lain:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '1'</li> </ul>
		Confirm	Konfirmasi dari akun terundang yang didaftarkan melalui 'Register'	<ul style="list-style-type: none"> <li>Enam digit kode konfirmasi</li> </ul>	<p>Bila kode benar, server menjawab:</p> <ul style="list-style-type: none"> <li>Home ID</li> </ul> <p>Bila kode salah, server menjawab:</p> <ul style="list-style-type: none"> <li>Jawaban berupa karakter '0'</li> </ul>
		Editprof	Permintaan edit profile dari suatu akun	<ul style="list-style-type: none"> <li>Email peminta</li> </ul>	Server mengubah data profil pada tabel <u>general_data.users</u>
		Forgetpwd	Permintaan reset password dari pengguna yang lupa password	<ul style="list-style-type: none"> <li>Email peminta</li> </ul>	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.
		Deleteuser	Permintaan pengeluaran satu akun dari rumah	<ul style="list-style-type: none"> <li>Email yang hendak dikeluarkan dari rumah</li> </ul>	Server mengubah Home ID email tersebut menjadi 'none' dan privilege-nya juga menjadi 'none'.
MH (Manajamen Home)	Ponsel	edit	Permintaan edit profil rumah	<ul style="list-style-type: none"> <li>Home ID</li> <li>Nama rumah baru</li> <li>Koordinat rumah baru</li> </ul>	Server mengubah nama dan koordinat rumah di tabel <u>homeXXXXXX.info</u> .

MD (Manajemen Device)	Ponsel	adddevice	Permintaan menambah perangkat baru	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama Perangkat</li> <li>• Address</li> <li>• Tipe</li> </ul>	Server menyimpan nama, address, dan tipe perangkat ke database homeXXXXXX.devices.
		deletedevice	Permintaan menghapus perangkat	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Address perangkat yang ingin dihapus</li> </ul>	Server menghapus perangkat dengan address yang sesuai.
		editdevice	Permintaan edit nama perangkat	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Address perangkat</li> <li>• Nama baru</li> </ul>	Server mengganti nama perangkat yang sesuai menjadi nama yang baru
		addroom	Permintaan menambah room	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama room</li> <li>• Daftar perangkat yang terkandung pada room itu</li> </ul>	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"> <li>• Home ID</li> <li>• Nama room</li> </ul>	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"> <li>• Home ID</li> <li>• Nama room baru</li> <li>• Daftar perangkat baru</li> </ul>	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"> <li>• Home ID</li> <li>• Nama scenario</li> <li>• Waktu scenario</li> <li>• Daftar perangkat dalam skenario tersebut</li> </ul>	Server menambahkan nama dan waktu skenario ke tabel homeXXXXXX.scenarios. Server juga membuat tabel bernama sesuai dengan nama skenario yang isinya adalah daftar perangkat skenario tersebut beserta statusnya.
		deletesenario	Permintaan menghapus skenario	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama scenario</li> </ul>	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"> <li>• Home ID</li> <li>• Nama scenario</li> <li>• Nama baru</li> <li>• Waktu baru</li> <li>• Daftar perangkat baru</li> </ul>	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"> <li>• Home ID</li> <li>• Lama penggunaan seluruh perangkat di rumah dalam 1 hari</li> </ul>	Server mengubah isi tabel dataMMYYYY untuk tanggal saat itu dengan data dari <i>host</i> .



#### **4.3.7 Perancangan Program pada Host**

Selain berfungsi sebagai sumber perintah di subsistem *indoor*, *host* juga berfungsi sebagai pelayan pada subsistem *outdoor*. Dengan demikian, pada *host* juga perlu dirancang sebuah program yang melayani permintaan-permintaan dalam sistem. Berikut adalah tabel yang merangkum seluruh jenis pelayanan *host*

**Tabel 35 Rangkuman Pelayanan Host**

Header	Pengirim	Tipe	Fungsi	Isi Paket Permintaan	Aksi Server
SI	Server	-	Permintaan sign in dan perubahan kunci enkripsi	<ul style="list-style-type: none"> <li>Email pengirim</li> <li>Kunci enkripsi baru</li> </ul>	Host menyimpan kunci enkripsi yang baru untuk user yang bersangkutan.
LO (Manajemen Home)	Ponsel	-	Permintaan data perangkat, room, dan scenario	<ul style="list-style-type: none"> <li>Home ID</li> </ul>	Host mengakses tabel homeXXXXXX.info, kemudian mengambil nama dan koordinat rumah, lalu dikirimkan sebagai jawaban.
MH (Manajemen Home)	Ponsel	list	Permintaan data identitas rumah	<ul style="list-style-type: none"> <li>Home ID</li> </ul>	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"> <li>Home ID</li> <li>Nama rumah baru</li> <li>Koordinat rumah baru</li> </ul>	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"> <li>Home ID</li> <li>Email pengguna</li> <li>Privilege</li> </ul>	Host menyimpan data pengguna baru di homeXXXXXX.users.
	Ponsel	deleteuser	Permintaan penghapusan anggota rumah	<ul style="list-style-type: none"> <li>Home ID</li> <li>Email pengguna</li> </ul>	Host menghapus user tersebut dari basis data.
MD (Manajemen Device)	Ponsel	adddevice	Permintaan menambah perangkat baru	<ul style="list-style-type: none"> <li>Home ID</li> <li>Nama Perangkat</li> <li>Address</li> <li>Tipe</li> </ul>	Host menyimpan nama, address, dan tipe perangkat ke database homeXXXXXX.devices.

		deletedevice	Permintaan menghapus perangkat	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Address perangkat yang ingin dihapus</li> </ul>	Host menghapus perangkat dengan address yang sesuai.
		editdevice	Permintaan edit nama perangkat	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Address perangkat</li> <li>• Nama baru</li> </ul>	Host mengganti nama perangkat yang sesuai menjadi nama yang baru
		addroom	Permintaan menambah room	<ul style="list-style-type: none"> <li>• Home ID</li> <li>• Nama room</li> <li>• Daftar perangkat yang terkandung pada room itu</li> </ul>	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"> <li>• Home ID</li> <li>• Nama room</li> </ul>	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"> <li>• Home ID</li> <li>• Nama room baru</li> <li>• Daftar perangkat baru</li> </ul>	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"> <li>• Home ID</li> </ul>	Host menambahkan nama dan waktu skenario ke tabel

				<ul style="list-style-type: none"> <li>Nama scenario</li> <li>Waktu scenario</li> <li>Daftar perangkat dalam scenario tersebut</li> </ul>	homeXXXXXX.scenarios. Host juga membuat tabel bernama sesuai dengan nama skenario yang isinya adalah daftar perangkat skenario tersebut beserta statusnya.
		deletesenario	Permintaan menghapus skenario	<ul style="list-style-type: none"> <li>Home ID</li> <li>Nama scenario</li> </ul>	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"> <li>Home ID</li> <li>Nama scenario</li> <li>Nama baru</li> <li>Waktu baru</li> <li>Daftar perangkat baru</li> </ul>	Host mengubah nama dan waktu skenario di tabel scenarios, lalu mengubah isi tabel dengan nama skenario tersebut dengan daftar perangkat yang baru.
		editprivilege	Permintaan edit privilege perangkat	<ul style="list-style-type: none"> <li>Home ID</li> <li>Daftar privilege baru</li> </ul>	Host mengubah seluruh privilege perangkat di tabel homeXXXXXX.devices.
CO (Command)	Ponsel	-	Perintah terhadap perangkat (menyala, mati, dll)	<ul style="list-style-type: none"> <li>Home ID</li> <li>Alamat perangkat yang ingin diberi perintah</li> <li>Isi perintah</li> </ul>	Host mengubah status perangkat pada tabel homeXXXXXX.devices. 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"> <li>• Home ID</li> <li>• Pengirim notifikasi</li> <li>• Isi pesan notifikasi</li> </ul>	Host menyebarkan isi notifikasi ke seluruh <i>user</i> 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"> <li>• Home ID</li> <li>• Nama scenario</li> </ul>	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"> <li>• Home ID</li> <li>• Alamat perangkat</li> </ul>	Host melihat isi tabel data, kemudian mengirimkan data penggunaan perangkat yang dimaksud sesuai dengan alamat yang diberikan ke ponsel peminta.

