



Dokumentasi Produk Tugas Akhir

Lembar Sampul Dokumen

| | |
|--------------------|---|
| Judul Dokumen | TUGAS AKHIR TEKNIK ELEKTRO: <i>Internet of Things: Personalisasi Informasi dan Promosi</i> di Lingkungan Kampus dengan iBeacon |
| Jenis Dokumen | PENGUJIAN <small>Catatan: Dokumen ini dikendalikan penyebarannya oleh Program Studi Teknik Elektro ITB</small> |
| Nomor Dokumen | B500-03-TA1516.01.003 |
| Nomor Revisi | 03 |
| Nama File | B500-03-TA1516.01.003.docx |
| Tanggal Penerbitan | 4 Mei 2016 |
| Unit Penerbit | Program Studi Teknik Elektro Institut Teknologi Bandung |
| Jumlah Halaman | 39 <small>(termasuk lembar sampul ini)</small> |

| Data Pengusul dan Pembimbing | | | | |
|------------------------------|---------|---|--------------|-------------------------|
| Pengusul | Nama | Astari Purnomo | Jabatan | Anggota |
| | Tanggal | 4 Mei 2016 | Tanda Tangan | |
| | Nama | Rizky Indra Syafrian | Jabatan | Anggota |
| | Tanggal | 4 Mei 2016 | Tanda Tangan | |
| | Nama | Adirga Ibrahim Khairy | Jabatan | Anggota |
| | Tanggal | 4 Mei 2016 | Tanda Tangan | |
| Pembimbing | Nama | Ir. Emir Mauludi Husni, M.Sc., Ph.D. | Jabatan | Dosen Pembimbing |
| | Tanggal | 4 Mei 2016 | Tanda Tangan | |

DAFTAR ISI

| | |
|--|----|
| CATATAN SEJARAH PERBAIKAN | 3 |
| PENGANTAR | 4 |
| 1.1 Ringkasan Isi Dokumen | 4 |
| 1.2 Tujuan Penulisan, Aplikasi, dan Fungsi Dokumen | 4 |
| 1.3 Referensi | 4 |
| 1.4 Daftar Singkatan..... | 5 |
| PENGUJIAN | 7 |
| 2.1 Pengujian Sub-sistem <i>Back-end Application</i> | 7 |
| 2.1.1 Lingkup Pengujian | 7 |
| 2.1.2 Konfigurasi Pengujian | 7 |
| 2.1.3 Syarat Pengujian..... | 7 |
| 2.1.4 Prosedur Pengujian..... | 8 |
| 2.1.5 Hasil Pengujian..... | 8 |
| 2.2 Pengujian Sub-sistem <i>Mobile Application</i> | 16 |
| 2.2.1 Lingkup Pengujian | 16 |
| 2.2.2 Konfigurasi Pengujian | 17 |
| 2.2.3 Syarat Pengujian..... | 17 |
| 2.2.4 Prosedur Pengujian..... | 17 |
| 2.2.5 Hasil Pengujian..... | 18 |
| 2.3 Pengujian Sub-sistem <i>Beacon</i> | 32 |
| 2.3.1 Lingkup Pengujian | 32 |
| 2.3.2 Konfigurasi Pengujian | 32 |
| 2.3.3 Syarat Pengujian..... | 33 |
| 2.3.4 Prosedur Pengujian..... | 33 |
| 2.3.5 Hasil Pengujian..... | 34 |

CATATAN SEJARAH PERBAIKAN

Tabel I – Catatan Sejarah Perbaikan

| Versi | Tanggal | Penyunting | Perbaikan |
|-------|---------------|---|--|
| 01 | 11 Maret 2016 | Astari Purnomo Rizky Indra Syafrian Adirga Ibrahim Khairy | Rilis dokumen versi 01 |
| 02 | 19 April 2016 | Astari Purnomo Rizky Indra Syafrian Adirga Ibrahim Khairy | Rilis dokumen versi 02 Penambahan data pengujian <i>beacon</i> Penambahan hasil pengujian <i>mobile apps</i> Penambahan hasil pengujian <i>back-end application</i> |
| 03 | 4 Mei 2016 | Astari Purnomo Rizky Indra Syafrian Adirga Ibrahim Khairy | Hasil pengujian pemunculan notifikasi Hasil pengujian <i>beacon monitoring</i> Hasil pengujian jarak terjauh <i>beacon</i> Hasil pengujian <i>attend class</i> Hasil pengujian <i>front-end</i> untuk pengiriman notifikasi dan sistem absensi |

PENGANTAR

1.1 Ringkasan Isi Dokumen

Dokumen ini berisi hasil pengujian dari rancangan proyek tugas akhir yang berjudul Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon. Dokumen ini diajukan sebagai salah satu syarat untuk memenuhi proyek Tugas Akhir II Teknik Elektro ITB tahun 2015-2016.

Dokumen ini terdiri dari beberapa bagian yaitu lingkup pengujian, konfigurasi pengujian, syarat pengujian, prosedur pengujian, dan hasil pengujian untuk masing-masing sub-sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon. Pada akhir dokumen terdapat kesimpulan pengujian dari keseluruhan sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.

1.2 Tujuan Penulisan, Aplikasi, dan Fungsi Dokumen

Penulisan dokumen B500 ini memiliki tujuan sebagai berikut.

1. Dokumentasi tahap pengujian Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon berdasarkan rancangan dan spesifikasi yang telah dibuat.
2. Menjadi panduan dalam pelaksanaan kerja dan pengambilan keputusan dalam pengembangan sistem lebih lanjut.
3. Sebagai salah satu dokumentasi proyek tugas akhir Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.

1.3 Referensi

1. International Business Machines (IBM), Corp. 2016. *IBM Bluemix Documentation*. IBM Bluemix. (Online). <https://console.ng.bluemix.net/docs/>. Diakses 8 Maret 2016 16:30.
2. Tutorials Points. 2016. *Node.js Tutorial*. (Online). <http://www.tutorialspoint.com/nodejs/>. Diakses 8 Maret 2016 16:30.
3. Rudd, Adam, Matteo Collina, and Maxime Agor. 2016. *A library for the MQTT protocol*. Neatly Packaged Modules (npm), Inc. (Online). <https://www.npmjs.com/package/mqtt>. Diakses 8 Maret 2016 16:30.
4. StrongLoop, IBM, and other expressjs.com contributors. *Express Framework Node.js Guide*. Node.js Foundation. (Online). <http://expressjs.com/en/guide/>. Diakses 8 Maret 2016 16:30.
5. International Business Machines (IBM), Corp. 2016. *Cloudant Node.js CRUD*. IBM Cloudant. (Online). <https://github.com/cloudant/nodejs-cloudant>. Diakses 8 Maret 2016 16:30.
6. Valerie Lampkin, Weng Tat Leong, Leonardo Olivera, Sweta Rawat, Nagesh Subrahmanyam, Rong Xiang, Gerald Kallas, Neeraj Krishna, Stefan Fassmann, Martin Keen, Dave Locke. 2012. *IBM Redbooks Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry*. USA : International Business Machines Corporation (IBM Corp).
7. Google, Inc. 2015. *Processes and Application Life Cycle*. Android Developer (Online). <http://developer.android.com/guide/topics/processes/process-lifecycle.html>. Diakses 4 Januari 2015 14.00.

8. Google, Inc. 2015. *Processes and Thread*. Android Developer (Online). <http://developer.android.com/guide/components/processes-and-threads.html>. Diakses 4 Januari 2015 14.00.
9. Google, Inc. 2015. *Connectivity Manager* (Online). <http://developer.android.com/reference/android/net/ConnectivityManager.html>. Diakses 4 Januari 2015 14.00.
10. Google, Inc. 2015. *View* (Online). [http://developer.android.com/reference/android/view/View.html#isInEditMode\(\)](http://developer.android.com/reference/android/view/View.html#isInEditMode()). Diakses 4 Januari 2015 14.00.
11. IBM Corporation. 2015. *IA92: WBI Brokers - Java implementation of WebSphere MQ Telemetry transport*. IBM Corporation (Online). http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24006006&loc=en_US&cs=utf-8&lang=en. Diakses 4 November 2015 16:00.
12. IBM Corporation. 2015. *Using MQ Telemetry Transport protocol in IBM Worklight mobile applications*. IBM Corporation (Online). http://www-01.ibm.com/support/docview.wss?rs=171&uid=swg24033580&loc=en_US&cs=utf-8&lang=en. Diakses 4 November 2015 16:00.
13. GitHub, Inc. 2016. *IoT Starter for Android*. GitHub, Inc (Online). <https://github.com/ibm-messaging/iot-starter-for-android>. Diakses pada 12 Januari 2016 17.00
14. GitHub, Inc. 2016. *Android MQTT Demo*. GitHub, Inc (Online). <https://github.com/jeffprestres/AndroidMQTTDemo>. Diakses pada 12 Januari 2016 17.00
15. Dc-square GmbH. 2016. *MQTT Client Library Encyclopedia – Paho Android Service*. HiveMQ Enterprise MQTT Broker (Online). <http://www.hivemq.com/blog/mqtt-client-library-encyclopedia-paho-android-service>. Diakses pada 12 Januari 2016 17.00
16. Dc-square GmbH. 2016. *MQTT Security Fundamentals: Authorization*. HiveMQ Enterprise MQTT Broker (Online). <http://www.hivemq.com/blog/mqtt-security-fundamentals-authorization/>. Diakses pada 12 Januari 2016 17.00
17. Auth0, Inc. 2013-2015. *Authenticating & Authorizing Devices using MQTT with Auth0*. Auth0, Inc (Online). <https://auth0.com/docs/scenarios/mqtt>. Diakses pada 12 Januari 2016 17.00
- IBM Corporation. 2015. *Learning Center-Introducing IBM Cloudant*. IBM Corporation (Online). <https://cloudant.com/learning-center/>. Diakses 4 November 2015 16:00.

1.4 Daftar Singkatan

Berikut ini diberikan tabel referensi dari singkatan-singkatan yang digunakan dalam dokumen ini.

Tabel II – Daftar Singkatan

| Singkatan | Arti |
|-----------|------------------------------------|
| IoT | Internet of Things |
| BLE | Bluetooth Low Energy |
| UUID | Universally Unique Identifier |
| GUI | Graphical User Interface |
| SDK | Software Development Kit |
| IDE | Integrated Development Environment |

| | |
|------|-------------------------------------|
| IBM | International Business Machines |
| REST | Representative State Transfer |
| API | Application Program Interface |
| MAC | Media Access Control |
| RSSI | Radio Signal Strength Indicator |
| PI | Presence Insights |
| HTTP | Hypertext Transfer Protocol |
| URL | Uniform Resource Locator |
| URI | Uniform Resource Identifier |
| SQL | Structured Query Language |
| PCB | Printed Circuit Board |
| IDR | Indonesia Rupiah |
| ITB | Institut Teknologi Bandung |
| AJAX | Asynchronous Javascript and XML |
| XML | Extensible Markup Language |
| MQTT | Message Queuing Telemetry Transport |
| JSON | JavaScript Object Notation |

PENGUJIAN

2.1 Pengujian Sub-sistem *Back-end Application*

Dalam melakukan pengujian terhadap sub-sistem *back-end application*, perlu dilakukan pengamatan pada *console* di *server* serta perubahan yang ada pada *database*. Pada *source code* telah diberikan sebuah *syntax* sederhana untuk mengamati bahwa setiap bagian tugas *back-end application* berjalan dengan baik.

2.1.1 Lingkup Pengujian

Pengujian pada sub-sistem *back-end application* dilakukan pada lingkup pengujian sebagai berikut.

1. Pengujian penerimaan data dari aplikasi Android berupa data *sign-in*, *sign-up*, dan data *beacon*.
2. Pengujian pengolahan data *beacon* yaitu dikirimkannya kembali informasi *beacon* yang sesuai dengan *database*.
3. Pengujian data pengiriman notifikasi dari *front-end* oleh *administrator* sampai ke *smartphone* pengguna
4. Pengujian sistem absensi dengan ditampilkannya daftar hadir kelas kuliah pada *front-end* pada *administrator dashboard*.

2.1.2 Konfigurasi Pengujian

Pengujian pada sub-sistem *back-end application* dilakukan pada konfigurasi *software* dan *hardware* sebagai berikut.

1. Personal Computer Intel™ Core® i7 3770K 3.5 GHz, RAM 8GB, dengan OS Windows 10 dan koneksi internet.
2. IBM Bluemix sebagai *server* tempat *back-end application* berjalan. Lokasi server berada di Sydney, Australia.
3. Smartphone LG Nexus 5 dengan OS Android Marshmallow 6.0.1.

2.1.3 Syarat Pengujian

Terdapat beberapa batasan yang digunakan untuk melakukan pengujian sub-sistem *back-end application* ini, diantaranya adalah sebagai berikut.

1. Data yang dikirimkan ke *back-end application* harus berbentuk *plain text* dengan format JSON seperti yang sudah dibahas pada dokumen sebelumnya.
2. Status/informasi yang dikembalikan oleh *back-end application* ke aplikasi Android harus sesuai dengan yang *di-request*.
3. Keseluruhan proses yang dilakukan oleh mulai *back-end application* dari menerima data sampai mengembalikan status/informasi tidak boleh terlalu lama.
4. Pengiriman notifikasi dari *form* yang ada pada *front-end* harus penuh tidak ada isian yang kosong.
5. Pengujian sistem absensi dilakukan dengan menggunakan akun buatan dengan nama dan NIM *dummy* serta menggunakan *database* peserta kuliah *dummy*.

2.1.4 Prosedur Pengujian

Dalam melakukan implementasi sub-sistem *back-end application* dilakukan beberapa tahapan prosedur sebagai berikut.

1. Melakukan *sign-up* pada aplikasi Android, melihat data *sign-up* yang sampai ke *back-end application* merupakan data yang sama dengan yang digunakan pada saat *sign-up*. Melihat pada *database* bahwa data yang diregisterasikan telah masuk.
2. Melakukan *sign-in* pada aplikasi Android, melihat data *sign-in* yang sampai ke *back-end application* merupakan data yang sama dengan yang digunakan pada saat *sign-in*. Melihat pada *database* bahwa data yang digunakan untuk *sign-in* sesuai.
3. Melakukan *edit profile* pada aplikasi Android dan melihat pada *database* bahwa data telah berubah sesuai dengan data baru yang diberikan pengguna.
4. Melakukan *scanning beacon* dengan aplikasi Android, melihat data *beacon* yang sampai ke *back-end application* merupakan data yang sesuai, melihat apakah informasi yang dikirimkan kembali dari *back-end application* ke aplikasi Android merupakan data yang sesuai pada *database*.
5. Melakukan pengiriman notifikasi dari *front-end* dan melihat sampainya pesan notifikasi pada aplikasi Android.
6. Melakukan pengiriman data *beacon* untuk sistem absensi dan melihat pada *front-end* bahwa pengguna yang termasuk dalam sistem absensi muncul pada daftar hadir.

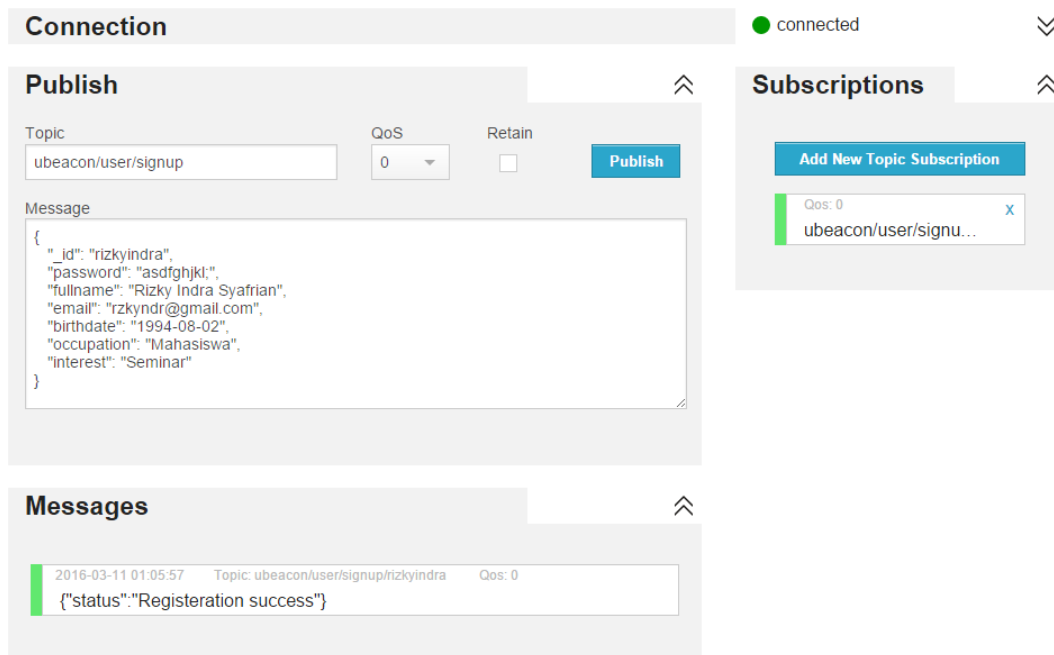
2.1.5 Hasil Pengujian

Berikut adalah hasil pengujian yang dilakukan terhadap sub-sistem *back-end application*.

Back-end Application

1. *Sign-up*

Ketika dilakukan *sign-up* pada aplikasi Android, terlihat dari status balasan bahwa *back-end application* telah menerima data *sign-up*. Selanjutnya adalah penambahan data tersebut kedalam *database*. Terlihat pada *database* data *sign-up* telah masuk dan menjadi bagian dari *database*.

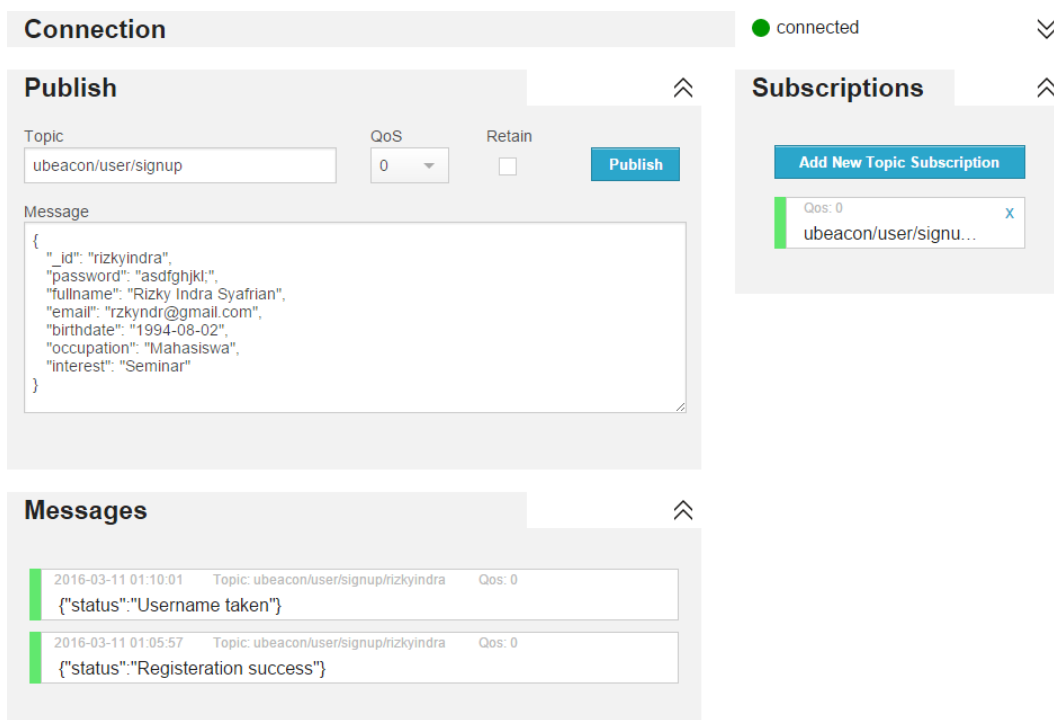


Gambar 1 Proses *sign-up* berhasil dibuktikan dengan status balasan yang diberikan *back-end application*. ©Dokumentasi Penulis

```
1 {
2   "_id": "rizkyindra",
3   "_rev": "5-512ef5691cc2b4c3765e2cbfe414118b",
4   "password": "asdfghjkl;",
5   "fullname": "Rizky Indra Syafrian",
6   "email": "rzkyndr@gmail.com",
7   "birthdate": "1994-08-02",
8   "occupation": "Mahasiswa",
9   "interest": "Seminar"
10 }
```

Gambar 2 Data registrasi yang telah masuk ke dalam *database*. ©Dokumentasi Penulis

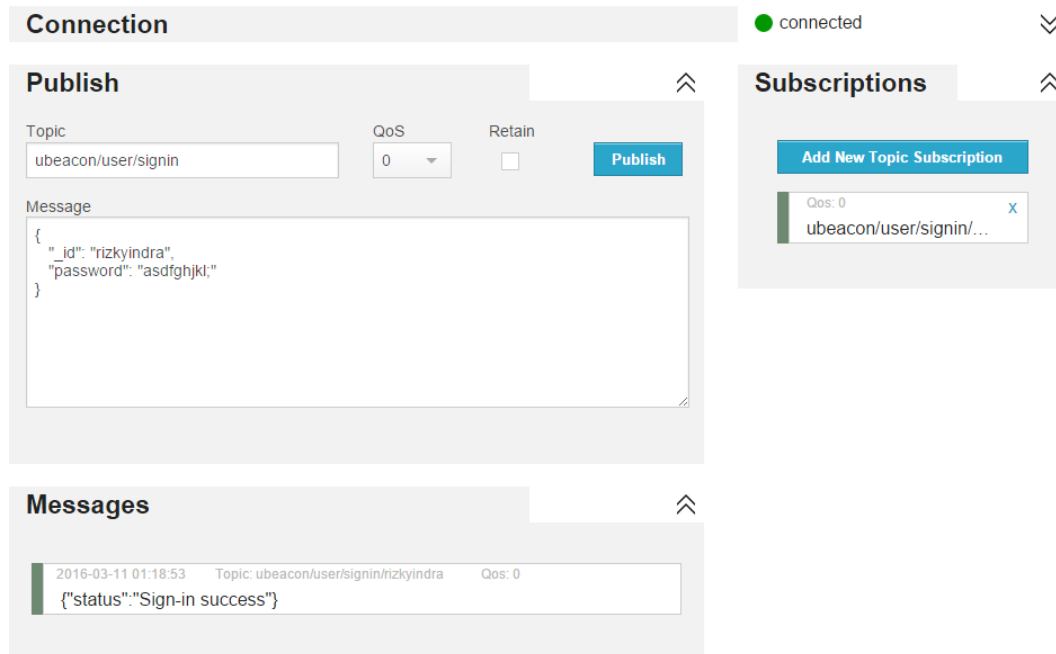
Apabila dilakukan *sign-up* kembali dengan data registrasi yang sama maka *back-end application* akan menolak. Dapat dilihat pada status balasan yang diberikan *back-end application*.



Gambar 3 Proses *sign-up* tidak berhasil karena *username* telah digunakan. ©Dokumentasi Penulis

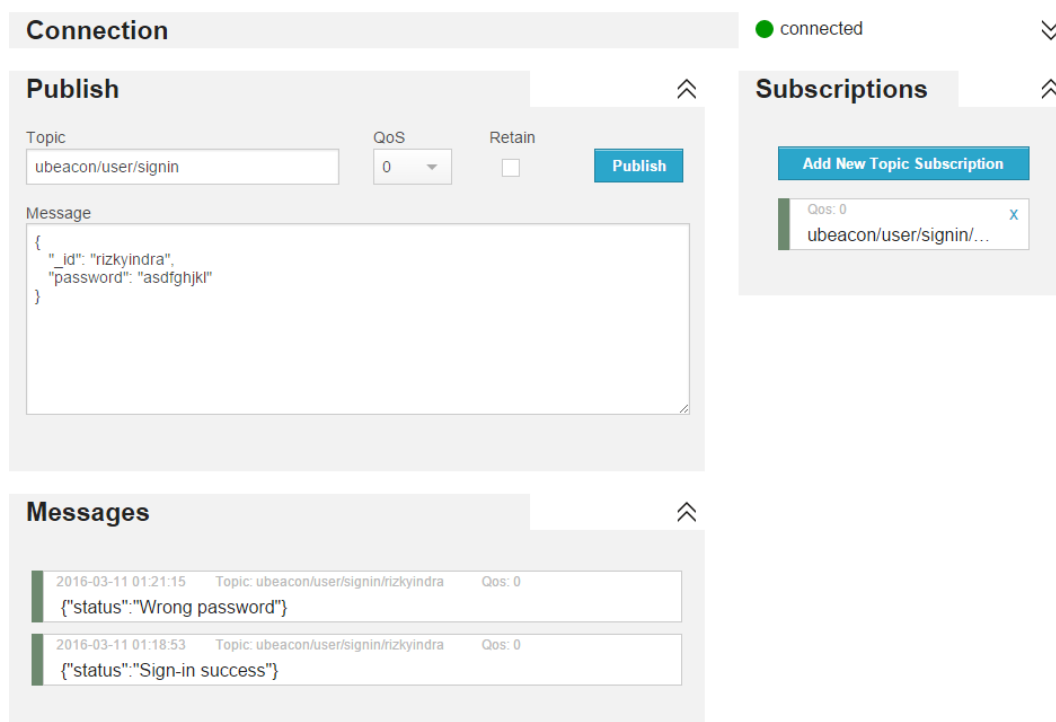
2. Sign-in

Ketika dilakukan *sign-in* pada aplikasi Android, terlihat dari status balasan bahwa *back-end application* telah menerima data *sign-in*. Selanjutnya adalah mencocokkan data *sign-in* dengan *database*. Status balasan akan menunjukkan keberhasilan *sign-in*.

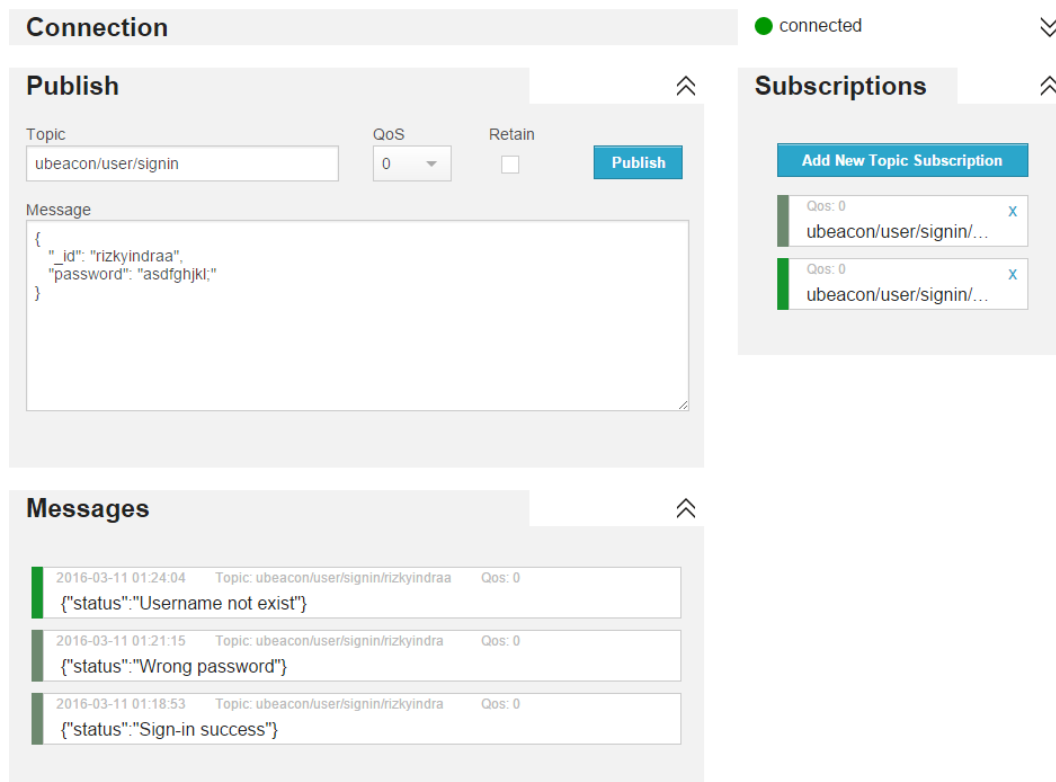


Gambar 4 Proses *sign-in* berhasil dibuktikan dari balasan yang diberikan *back-end application*. ©Dokumentasi Penulis

Ketika *sign-in* dilakukan dengan *password* yang tidak sesuai dengan *database*, maka *back-end application* akan menolak proses *sign-in*. Jika *sign-in* dilakukan dengan *username* yang tidak terdaftar, maka *back-end application* akan memberikan status bahwa *username* tidak terdaftar.



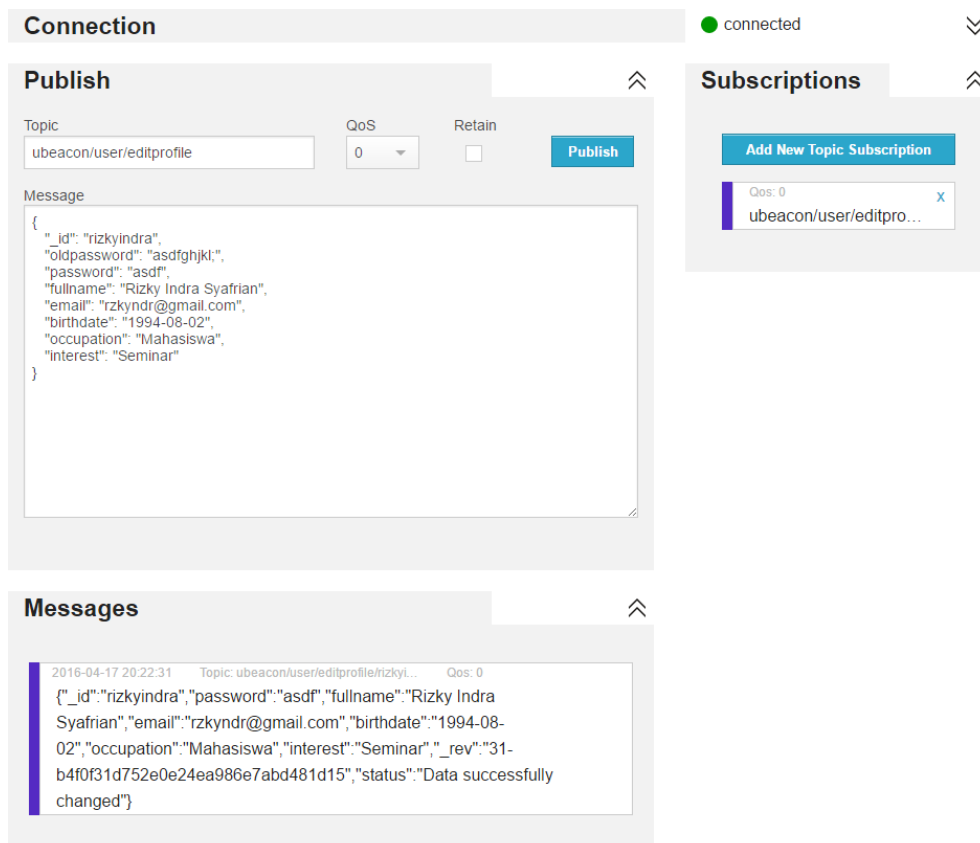
Gambar 5 Proses *sign-in* gagal karena *password* tidak sesuai dengan yang ada pada *database*. ©Dokumentasi Penulis



Gambar 6 Proses *sign-in* gagal karena *username* tidak ada pada *database*. ©Dokumentasi Penulis

3. Edit profile

Ketika dilakukan *edit profile* pada aplikasi Android, *back-end application* akan menyesuaikan *password* dan *username* dengan data yang ada pada *database*. Apabila sesuai, maka status balasan bahwa *edit profile* berhasil akan dikirim.



Gambar 7 Proses *edit-profile* berhasil dan data pada *database* telah diubah. ©Dokumentasi Penulis

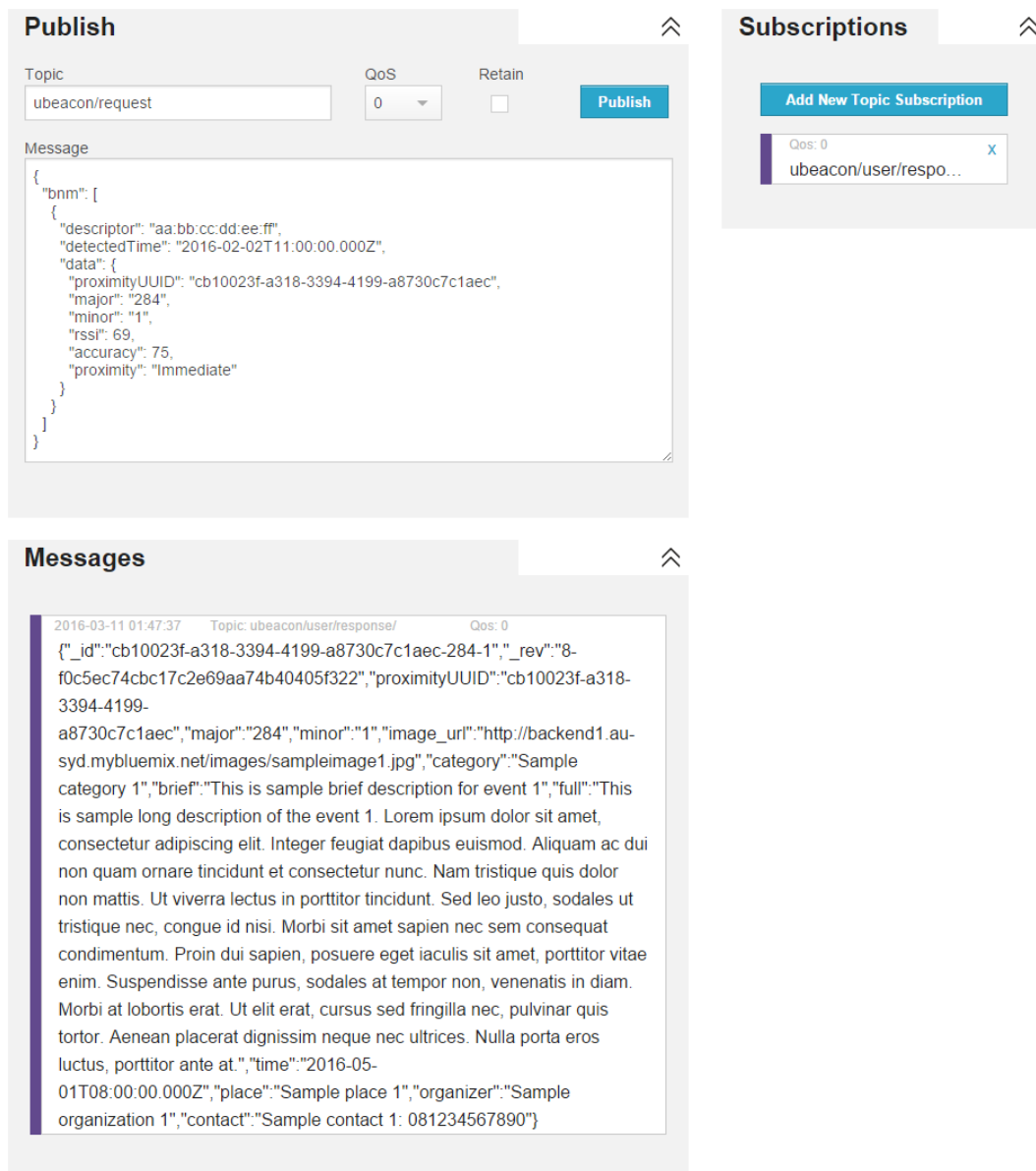
Jika pengguna tidak berkehendak untuk mengganti *password* ketika *edit profile* digunakan, maka pengguna dapat mengosongkan bagian *new password* pada aplikasi Android. *Password* lama akan tetap dapat digunakan dan data lain yang diganti akan berubah pada *database*.

The screenshot displays an MQTT client interface with three main sections: Connection, Publish, and Subscriptions. The Connection status is 'connected'. The Publish section shows a topic 'ubeacon/user/editprofile', QoS '0', and a 'Publish' button. The Message field contains a JSON object: { "_id": "rizkyindra", "oldpassword": "asdf", "password": "", "fullname": "Rizky Indra Syafrian", "email": "rzkyndr@gmail.com", "birthdate": "1994-08-02", "occupation": "Mahasiswa", "interest": "Seminar" }. The Subscriptions section shows a subscription for the same topic with QoS '0'. The Messages section shows a received message with a timestamp of 2016-04-17 20:24:57, the same topic, and a JSON object that includes an additional field: { "_id": "rizkyindra", "password": "asdf", "fullname": "Rizky Indra Syafrian", "email": "rzkyndr@gmail.com", "birthdate": "1994-08-02", "occupation": "Mahasiswa", "interest": "Seminar", "_rev": "32-b236612715a53c44e1fd234b646b0f2d", "status": "Data successfully changed" }.

Gambar 8 Proses *edit-profile* berhasil dengan *password* lama yang tidak diubah. ©Dokumentasi Penulis

4. Data *beacon* dan informasi *beacon*.

Keberhasilan pengiriman data *beacon* untuk melakukan *request* informasi *beacon* dapat terlihat dari balasan *back-end application* yang berupa informasi *beacon* yang dikembalikan. Balasan ini kemudian dicocokkan dengan informasi *beacon* yang tersimpan pada *database*. Berikut ini adalah percobaan *request* informasi *beacon*.



Gambar 9 Bagian *message* merupakan informasi *beacon* yang dikembalikan oleh *back-end application*. ©Dokumentasi Penulis

Apabila dicocokkan dengan *database* informasi *beacon* maka dapat dilihat bahwa data yang dikirimkan kembali ke aplikasi Android oleh *back-end application* sesuai.

```

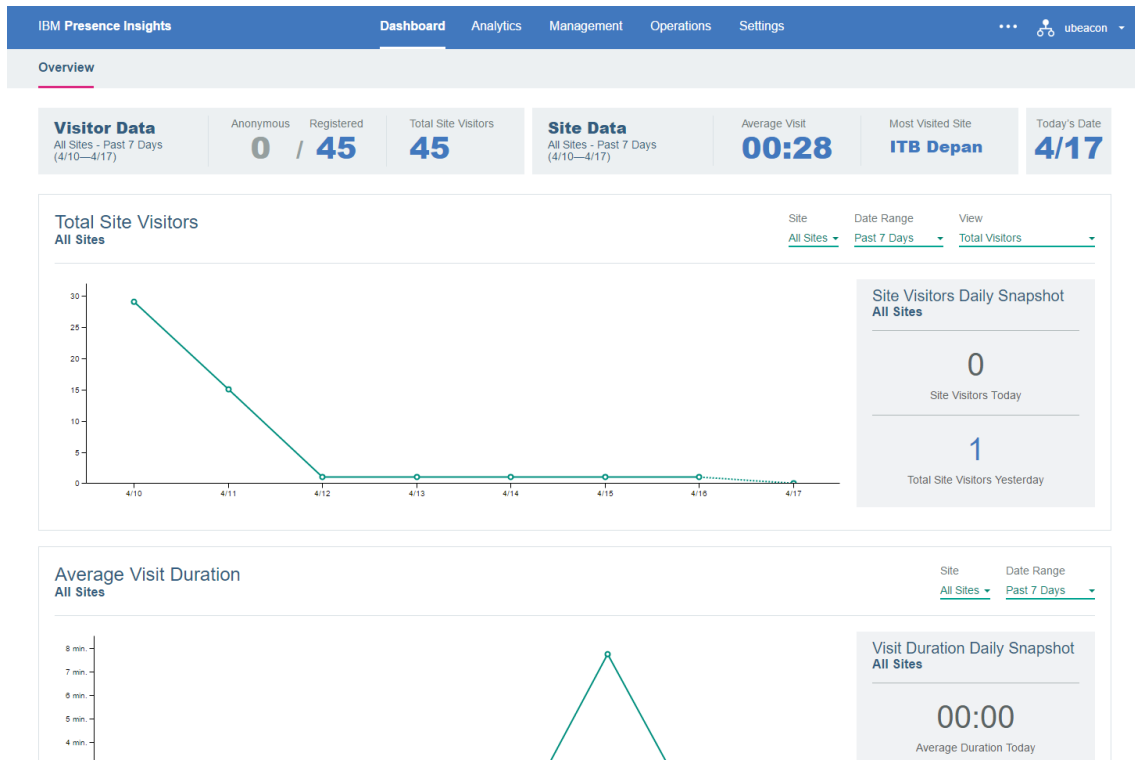
1 {
2   "_id": "cb10023f-a318-3394-4199-a8730c7c1aec-284-1",
3   "_rev": "8-f0c5ec74cbc17c2e69aa74b40405f322",
4   "proximityUUID": "cb10023f-a318-3394-4199-a8730c7c1aec",
5   "major": "284",
6   "minor": "1",
7   "image_url": "http://backend1.au-syd.mybluemix.net/images/sampleimage1.jpg",
8   "category": "Sample category 1",
9   "brief": "This is sample brief description for event 1",
10  "full": "This is sample long description of the event 1. Lorem ipsum dolor sit amet, consectetur adi
11  "time": "2016-05-01T08:00:00.000Z",
12  "place": "Sample place 1",
13  "organizer": "Sample organization 1",
14  "contact": "Sample contact 1: 081234567890"
15 }
```

Gambar 10 Informasi *beacon* yang ada pada *database*. ©Dokumentasi Penulis

IBM Presence Insights

Pengujian IBM Presence Insights (PI) dilakukan dengan mengirimkan *beacon payload* ke *back-end application*. Dengan mengirimkan *beacon payload*, *back-end application* akan mengirimkan balasan berupa informasi *beacon* dan juga meneruskan *payload* tersebut ke IBM Presence Insights.

Karena mekanisme pengujian yang dilakukan hampir sama dengan pengujian *back-end application* untuk *request beacon information*, maka hasil pengujian yang ditunjukkan hanya berupa data yang ditampilkan oleh PI.



Gambar 11 Tampilan grafik hasil pengujian IBM Presence Insights. ©Dokumentasi Penulis

Sistem Notifikasi

Pengujian sistem notifikasi dilakukan dengan pengiriman pesan khusus untuk anggota unit dan mahasiswa dari program studi tertentu. Pengiriman pesan dilakukan dari *front-end* yang telah dibuat. Berikut ini adalah tampilan pesan yang akan dikirimkan pada *front-end*.

Gambar 12 Tampilan *front-end* untuk mengirimkan notifikasi bagi anggota unit. ©Dokumentasi Penulis

Setelah pesan dikirim, aplikasi Android dengan pengguna yang terdaftar pada unit yang dituju akan mendapatkan pesan notifikasi. Pesan notifikasi juga dapat dikirimkan pada mahasiswa dari program studi tertentu. Notifikasi serupa juga akan ditampilkan ketika anggota program studi dikirimkan pesan dari *front-end*.

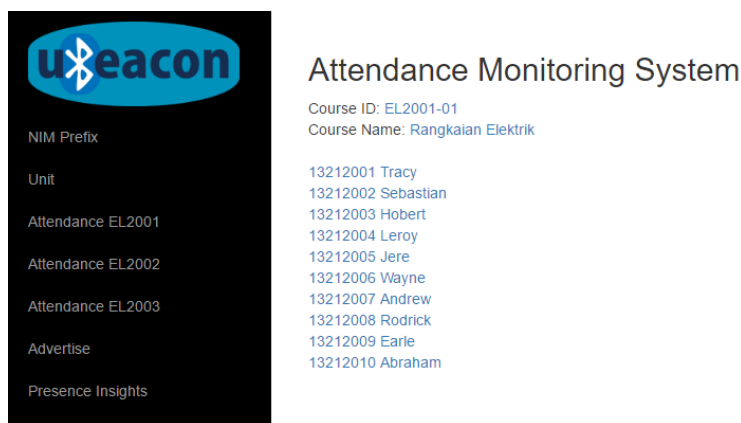


Gambar 13 Tampilan notifikasi pada aplikasi Android. ©Dokumentasi Penulis

Sistem Absensi

Pengujian sistem absensi dilakukan dengan pengiriman *beacon payload* ke *topic* yang berbeda dengan *request beacon information*. *Topic* yang digunakan untuk sistem adalah “*ubeacon/user/presence*”.

Setelah *beacon payload* dikirimkan ke *topic* tersebut, daftar hadir yang telah dibuat pada *front-end* kemudian bertambah sesuai dengan nama pengguna yang terdeteksi. Jika *beacon payload* dikirimkan lagi dengan *username* yang berbeda, maka pengguna yang berbeda tersebut akan ditambahkan dalam absen yang digunakan. Jika pengguna tidak mengirimkan *beacon payload* selama lebih dari 5 menit, maka pengguna akan dihapuskan dari daftar hadir.



Gambar 14 Tampilan sistem absensi pada *front-end*. ©Dokumentasi Penulis

2.2 Pengujian Sub-sistem *Mobile Application*

Dalam melakukan pengujian terhadap sub-sistem mobile application maka dilakukan pengujian per-blok dari sub-sistem ini. Pengujian per-blok ini bertujuan untuk melakukan pengecekan dan perbaikan pada setiap bagian yang menyusun sistem ini. Dengan dilakukannya pengujian ini diharapkan sub-sistem dapat diintegrasikan dengan baik.

2.2.1 Lingkup Pengujian

Pengujian yang dilakukan pada sub-sistem mobile application akan dilakukan dalam lingkup pengujian sebagai berikut:

1. Menambahkan fitur *Bluetooth Low Energy*(BLE) untuk pemindaian(*scan*) beacon.
2. Membangun interaksi *mobile application* dengan server IBM Bluemix melalui MQTT.
3. Membangun fitur penyimpanan dan pengelolaan data pada *local memory* smartphone.
4. Membangun *Graphic User Interface*(GUI) sebagai interaksi antarmuka dengan pengguna.
5. Membangun fitur personalisasi penggunaan aplikasi dalam bentuk akun pengguna (*user account*).
6. *Push-Notification* sebagai fitur personalisasi pengguna.
7. Personalisasi dengan menambahkan fitur presensi kehadiran mahasiswa dalam kelas perkuliahan.
8. Personalisasi dengan menambahkan fitur presensi kehadiran mahasiswa dalam kelas perkuliahan.

2.2.2 Konfigurasi Pengujian

Pengujian pada sub-sistem mobile application dilakukan pada konfigurasi software dan hardware sebagai berikut:

1. Personal computer Intel Core i7-4720HQ up to 3.6 GHz, memori RAM 4GB dengan Window 8.1 dan koneksi internet.
2. GUI dijalankan pada Smartphone Oppo Find 7a-x9006 dengan serial number 96cf428d dengan OS Android version 4.3.
3. Hardware *beacon* berupa Cubeacon.

2.2.3 Syarat Pengujian

Terdapat beberapa batasan yang digunakan untuk melakukan pengujian sub-sistem mobile application, diantaranya sebagai berikut:

1. Aplikasi dirancang pada sebuah project manager berupa software Android Studio 1.5.1 dengan JRE dan JDK 1.8.0_72
2. Pengguna mengunduh aplikasi pada smartphone melalui running project pada project manager yang digunakan.
3. Pengguna memasukkan input data berupa nama, username, password, e-mail, jenis kelamin, tanggal dan tahun lahir, pekerjaan, dan minat sesuai dengan kolom yang telah disediakan pada layout aplikasi. Berikut batasan inputan dari pengguna:
 - Nama, jenis kelamin, pekerjaan, dan minat dimasukkan dalam kombinasi huruf.
 - Username, password, e-mail boleh dimasukkan dengan menggunakan kombinasi huruf dan angka.
 - Tanggal dan tahun lahir dimasukkan dengan menggunakan kombinasi angka.
4. Smartphone yang digunakan oleh pengguna terkoneksi dengan internet.
5. Smartphone yang digunakan oleh pengguna telah diaktifkan fitur bluetoothnya.
6. Smartphone yang digunakan oleh pengguna memiliki Bluetooth minimal versi 4.0.

2.2.4 Prosedur Pengujian

Pada pengujian sub-sistem mobile application dilakukan beberapa prosedur pengujian sebagai berikut:

1. Pengujian fitur *Bluetooth Low Energy*(BLE) untuk pemindaian(*scan*) beacon.
2. Pengujian interaksi *mobile application* dengan server IBM Bluemix melalui MQTT.
3. Pengujian fitur penyimpanan dan pengelolaan data pada *local memory* smartphone.
4. Pengujian *Graphic User Interface*(GUI) sebagai interaksi antarmuka dengan pengguna.
5. Pengujian fitur personalisasi penggunaan aplikasi dalam bentuk akun pengguna (*user account*).
6. Pengujian *Push-Notification* sebagai fitur personalisasi pengguna.
7. Pengujian personalisasi dengan menambahkan fitur presensi kehadiran mahasiswa dalam kelas perkuliahan.

8. Pengujian Personalisasi dengan menambahkan fitur presensi kehadiran mahasiswa dalam kelas perkuliahan.

2.2.5 Hasil Pengujian

Berikut adalah hasil pengujian yang dilakukan terhadap sub-sistem mobile:

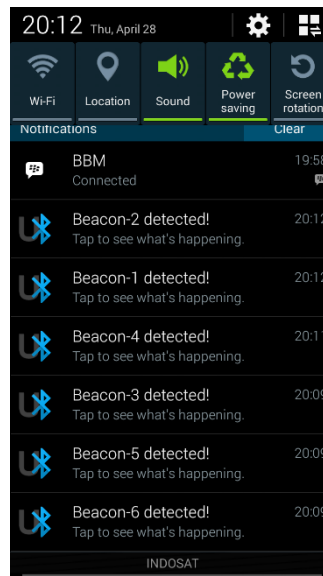
2.2.5.1 Fitur *Bluetooth Low Energy*(BLE) untuk pemindaian(*scan*) beacon.

Fitur Bluetooth Low Energy yang digunakan dalam menjalankan aplikasi uBeacon adalah Beacon Monitor. Fitur tersebut akan terus melakukan pemindaian terhadap sinyal Bluetooth dalam interval tertentu yang dapat di atur. Hasil implementasi pemindaian sinyal Bluetooth terlihat dalam hasil pada *logcat* software Android Studio sebagai berikut.

```
D/BluetoothAdapter: onScanResult () - Device=D4:51:B6:9B:8A:EA RSSI=-63
D/BluetoothAdapter: onScanResult () - Device=DC:2A:1D:0C:BE:1A RSSI=-68
D/BluetoothAdapter: onScanResult () - Device=F8:A4:2F:5A:36:BB RSSI=-69
D/BluetoothAdapter: onScanResult () - Device=D4:51:B6:9B:8A:EA RSSI=-64
D/BluetoothAdapter: onScanResult () - Device=FF:49:BB:4C:5F:DA RSSI=-83
D/BluetoothAdapter: onScanResult () - Device=F8:A4:2F:5A:36:BB RSSI=-59
D/BluetoothAdapter: onScanResult () - Device=DC:2A:1D:0C:BE:1A RSSI=-77
D/BluetoothAdapter: onScanResult () - Device=FF:49:BB:4C:5F:DA RSSI=-80
D/BluetoothAdapter: onScanResult () - Device=DE:EF:66:D1:9B:B1 RSSI=-67
D/BluetoothAdapter: onScanResult () - Device=F8:A4:2F:5A:36:BB RSSI=-62
D/BluetoothAdapter: onScanResult () - Device=D4:51:B6:9B:8A:EA RSSI=-59
D/BluetoothAdapter: onScanResult () - Device=DC:2A:1D:0C:BE:1A RSSI=-71
D/BluetoothAdapter: onScanResult () - Device=FF:49:BB:4C:5F:DA RSSI=-83
D/BluetoothAdapter: onScanResult () - Device=EE:9A:D3:FD:4B:B7 RSSI=-62
D/BluetoothAdapter: onScanResult () - Device=DE:EF:66:D1:9B:B1 RSSI=-67
D/BluetoothAdapter: onScanResult () - Device=F8:A4:2F:5A:36:BB RSSI=-63
D/BluetoothAdapter: stopLeScan ()
```

Script di atas menunjukkan adanya beacon terdeteksi dengan MAC-Address seperti yang ditampilkan pada gambar di atas, serta dilengkapi dengan nilai RSSI yang menunjukkan kekuatan signal yang terdeteksi.

Aplikasi uBeacon diatur agar dapat memberitahu pengguna saat memasuki daerah(region) Beacon yang telah terdefinisi dengan cara memberikan notifikasi. Berikut tampilan notifikasi ketika smartphone menemukan beacon tertentu berdasarkan hasil pengujian.



Gambar 15 Notifikasi saat beacon terdeteksi ©Dokumentasi Penulis

Pada pengujian di atas, terdeteksi 6 buah beacon yang berbeda sehingga pengguna mendapatkan 6 notifikasi yang berbeda pula.

2.2.5.2 Interaksi *mobile application* dengan server IBM Bluemix melalui MQTT

Selanjutnya, setelah pengguna menemukan beacon dengan daerah(region) tertentu, maka aplikasi akan menerima identifier beacon berupa UUID, Major, dan Minor yang selanjutnya digunakan sebagai payload yang akan dikirimkan ke server IBM Bluemix via MQTT. Hal tersebut dilakukan untuk mendapatkan informasi balikan selagi melakukan *publish* dan *subscribe*. Berikut hasil yang terdapat pada bagian *logcat* software Android Studio yang menandakan pengiriman payload kepada server telah berhasil.

```
I/com.example.astari: Service onCreate
I/System.out: -----subscribe
I/System.out: Sending Data-2 Completed
I/System.out: Sending Data-3 Completed
I/System.out: Sending Data-4 Completed
I/System.out: Sending Data-5 Completed
I/System.out: Sending Data-6 Completed
```

Dalam hal ini, salah satu beacon yang terdeteksi memiliki nilai variabel Major 6 seperti gambar berikut.

```
D/BluetoothAdapter: onClientRegistered() - status=0 clientIf=7
D/BluetoothAdapter: onScanResult() - Device=EE:9A:D3:FD:4B:B7 RSSI=-78
I/System.out: Variabel Major = 6
```

Hal tersebut ditandai dengan tampilan *logcat* menunjukkan log “Variabel Major = 6”. Selain itu, saat payload telah berhasil dikirimkan via MQTT, maka akan muncul tanda pada smartphone pengguna berupa *Toast message* yang bertuliskan “*payload is successfully sent*” sebagai bentuk pengujian pengiriman. Berikut tampilan yang terjadi ketika payload JSON telah dikirimkan.



Gambar 16 Pesan saat JSON payload terkirimkan ke server ©Dokumentasi Penulis

Setelah aplikasi berhasil mengirimkan payload kepada server, maka pengguna akan mendapatkan informasi balikan dengan cara *subscribe* berupa JSON file. Balikan tersebut akan disimpan di dalam sebuah ArrayAdapter dalam kelas(class) *Events.java*. Informasi balikan tersebut akan menjadi informasi yang akan ditampilkan kepada pengguna aplikasi uBeacon. Berikut hasil penerimaan informasi dari server IBM Bluemix pada *logcat* software Android Studio.

```

I/System.out: -----
I/System.out: (Hasil Subscribe nya)
I/System.out: 7. UUID      = cb10023f-a318-3394-4199-a8730c7c1aec
I/System.out: 7. Major     = 6
I/System.out: 7. Minor    = 284
I/System.out: [Ads-1]
I/System.out: 7.1. URL      = http://www.bandungactivities.com/images/poster/%20-%20Oddisey%202016%20Ayaskara%
I/System.out: 7.1. Category   = Festival
I/System.out: 7.1. Title      = Oddisey 2016: Ayaskara
I/System.out: 7.1. Brief Info = The Greatest Performance of Art Festival
I/System.out: 7.1. Full Info  = Oddisey 2016 Presents, The Greatest Performance of Art Festival Performed by SBM
I/System.out: AYASKARA: The Journey Begins
I/System.out: Ticket: Presale 25K, On The Spot 30K
I/System.out: Event Included: Performances, Bazaar, Amusement
I/System.out: Visit www.oddisey2016.com for more information.
I/System.out: 7.1. Time       = Open Gate 17.00 Local Time
I/System.out: 7.1. Location  = Institut Teknologi Bandung, campus front area
I/System.out: 7.1. Organizer = SBM-ITB 2018
I/System.out: 7.1. Contact   = www.oddisey2016.com
I/System.out: -----Batas MQTT Diterima-----

```

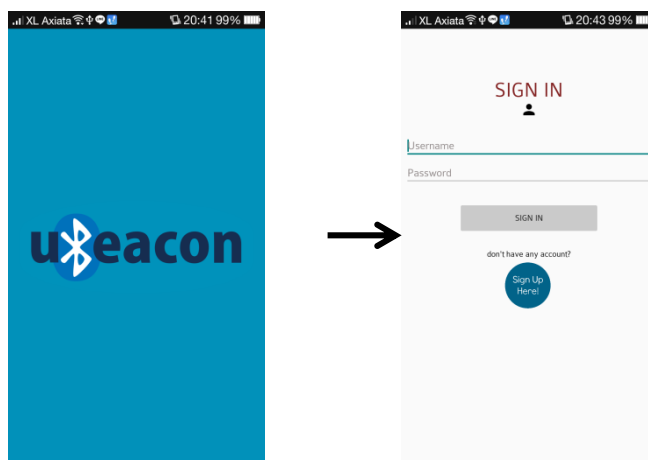
Gambar 17 logcat Android Studio saat menerima informasi via Mqtt ©Dokumentasi Penulis

Dalam hal ini, karena payload yang dikirimkan kepada server memiliki Variabel Major =6, maka informasi balikan yang didapatkan juga sesuai dengan identifier beacon yang dikirimkan. Informasi balikan yang akan didapatkan pengguna antara lain berupa *UUID*, *Major*, *Minor*, *URL*, *Category*, *Title*, *Brief Info*, *Full Info*, *Time*, *Location*, *Organizer*, dan *Contact*. Seluruh informasi tersebut akan ditampilkan pada tampilan homepage pada kelas Navigation.java dan informasi detil yang akan dijelaskan pada bagian selanjutnya.

2.2.5.3 Fitur penyimpanan dan pengelolaan data pada *local memory* smartphone

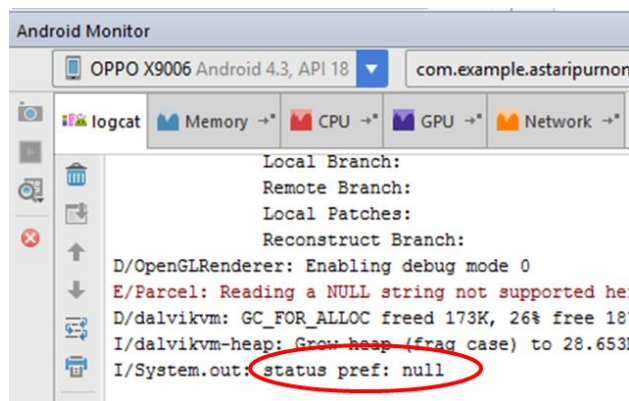
User's state

Untuk melakukan pengujian pada status pengguna akan digunakan sebuah skenario. Pada mulanya setelah aplikasi berhasil diinstal pada smartphone pengguna, tampilan yang diperoleh oleh pengguna adalah sebagai berikut:



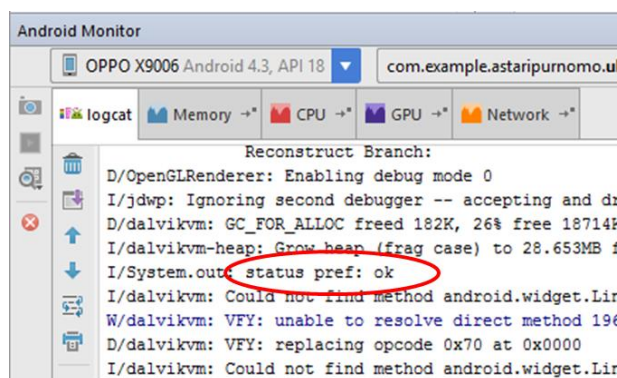
Gambar 18 Tampilan awal ©Dokumentasi Penulis

Berikut adalah status pengguna dalam file SharedPreference:



Gambar 19 Status awal *Shared Preference* ©Dokumentasi Penulis

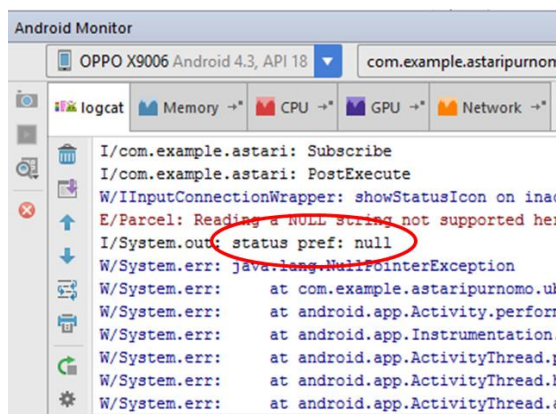
Pada kondisi ini, diperoleh bahwa pengguna masih berada pada state awal sehingga dapat dilihat bahwa file *SharedPreference* menampilkan value “null”. Selanjutnya saat Sign-In berhasil dilakukan, logcat menunjukkan tampilan:



Gambar 20 Sign in berhasil ©Dokumentasi Penulis

Saat Sign-In berhasil dilakukan, maka tampilan logcat pada file *SharedPreference* berubah seperti gambar diatas. Selanjutnya saat aplikasi di-destroy dan kembali diluncurkan, tampilan logcat masih menunjukkan value dari file *SharedPreference* yang sama. Hal ini membuktikan bahwa status pengguna pada Sign-In telah berjalan dengan baik.

Saat log out dilakukan oleh pengguna, tampilan pada logcat menunjukkan:



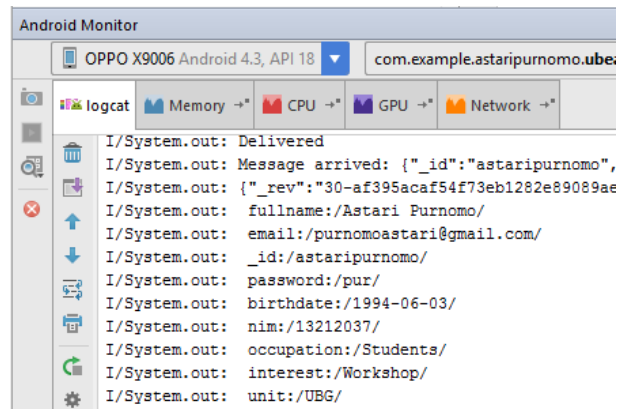
Gambar 21 log saat logout ©Dokumentasi Penulis

Log out menampilkan perubahan dari file *SharedPreference* kembali ditulis dengan “null”. Hal ini menunjukkan bahwa alur dari user’s state telah berjalan dengan baik.

Parsing data dari server

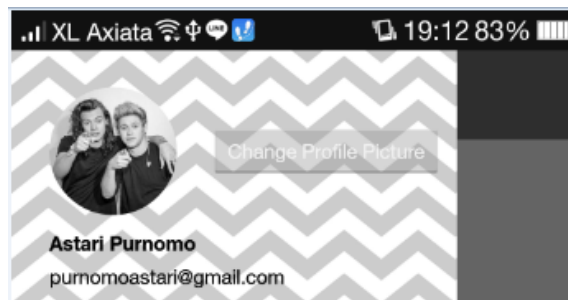
Pengujian penyimpanan data pengguna di file SharedPreference ditunjukkan dalam logcat sebagai berikut:

- Penyimpanan data diri pengguna saat Sign-In berhasil dilakukan

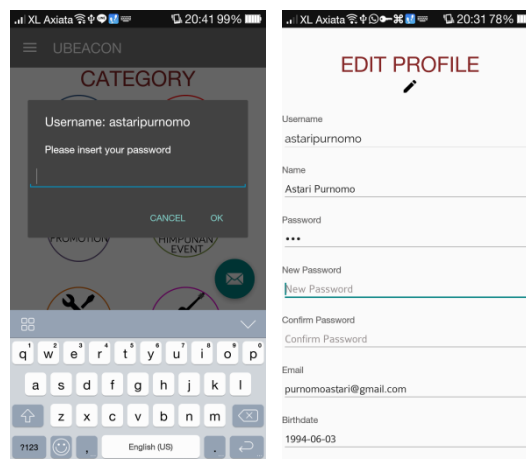


Gambar 22 Penyimpanan data diri ©Dokumentasi Penulis

Dari tampilan logcat ditunjukkan bahwa setelah diterima balasan dari server, setiap value dari key yang dibawa oleh server disimpan dalam file SharedPreference. Tampilan logcat menunjukkan bahwa restorasi data diri pengguna berhasil dilakukan. Selanjutnya akan ditunjukkan beberapa tampilan yang menunjukkan kemampuan aplikasi dalam melakukan parsing data dari file SharedPreference sebagai berikut:



Gambar 23 Hasil parsing data ©Dokumentasi Penulis

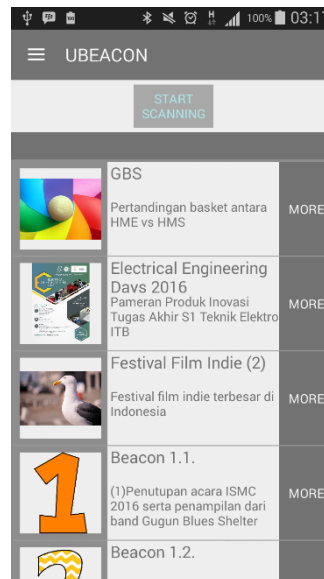


Gambar 24 Edit Profile ©Dokumentasi Penulis

Beberapa tampilan diatas menunjukkan bahwa parsing data dari informasi yang sebelumnya disimpan dalam file SharedPreference dan kemudian di-parsing telah berhasil dilakukan dengan baik.

2.2.5.4 Graphic User Interface(GUI) sebagai interaksi antarmuka dengan pengguna

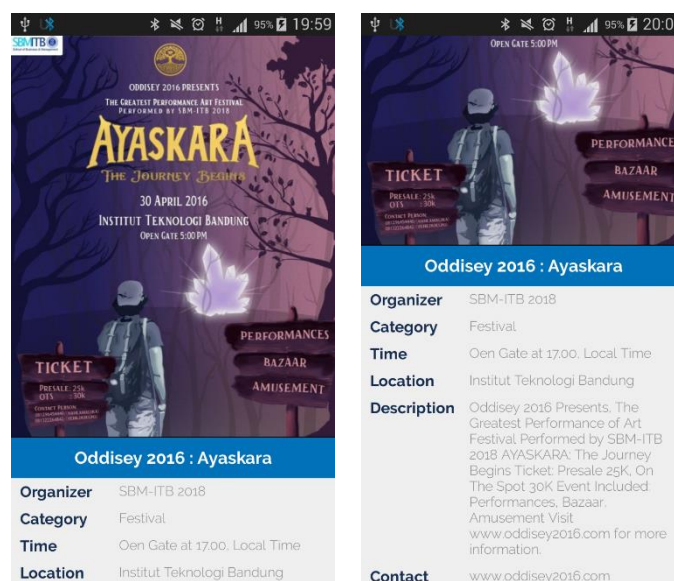
Aplikasi uBeacon akan menampilkan informasi sesuai dengan beacon identifier yang telah terdeteksi oleh proses beacon monitoring yang terjadi pada background application. Ketika informasi telah didapatkan dari server, maka akan ditampilkan tampilan sebagai berikut.



Gambar 25 Tampilan homepage uBeacon ©Dokumentasi Penulis

Tampilan ini ditampung pada fragment ShowFragment.java yang ditempatkan pada container fragment_container aktivitas(*activity*) Navigation.java. Tampilan utama ini terdiri atas gambar iklan, judul iklan, serta keterangan pendek dari iklan. Hal itu di assign berdasarkan informasi yang diperoleh dari balasan MQTT yang ditampung pada kelas *Events.java*. Untuk dapat mempermudah aktivitas pengguna, maka tampilan disusun dalam bentuk ListView seperti gambar di atas.

Jika salah satu tombol ditekan, maka akan ditampilkan sebuah iklan yang sesuai dengan informasi yang dipilih dan berdasarkan lokasi beacon yang terdeteksi juga. Salah satu contoh tampilan detail yang dipilih ditunjukkan oleh hasil sebagai berikut.



Gambar 26 Tampilan detail iklan ©Dokumentasi Penulis

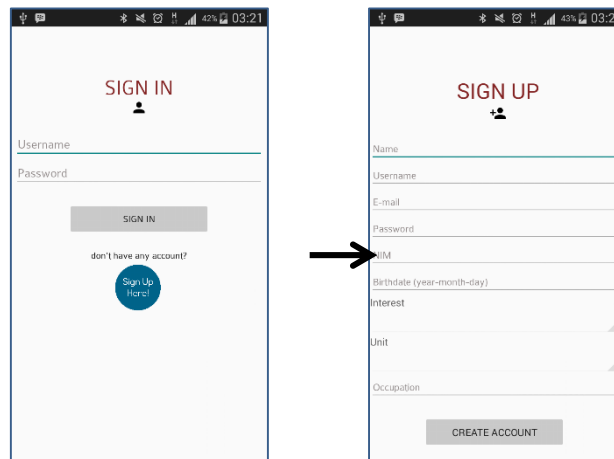
Pada gambar di atas, informasi yang telah ditampung hasil beacon monitoring pada beacon monitor ditampilkan secara detil. Tampilan disusun dengan menggunakan ScrollView agar tampilan lebih fleksibel dan dapat dilihat secara keseluruhan.

2.2.5.5 Fitur personalisasi penggunaan aplikasi dalam bentuk akun pengguna (*user account*)

Pengujian pada fitur personalisasi pengguna dilakukan kedalam beberapa bagian, yaitu:

Sign-Up

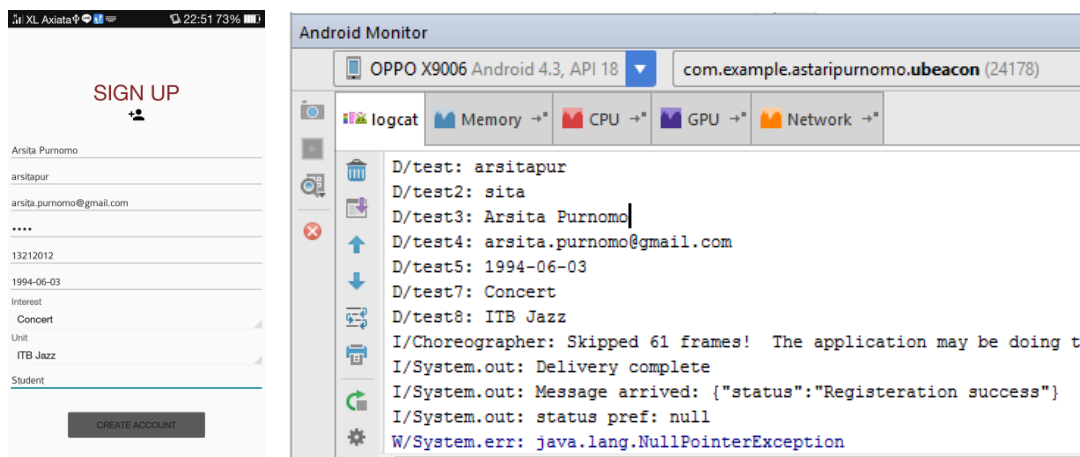
Pada bagian ini, pengguna belum memiliki akun pada aplikasi uBeacon, sehingga pengguna perlu untuk melakukan pendaftaran atau registrasi akun yang dapat langsung dilakukan pada aplikasi uBeacon. Cara untuk mendaftarkan akun dapat dilakukan dengan cara menekan tombol Sign Up sesuai pada gambar di berikut :



Gambar 27 Halaman Sign in Aplikasi Ubeacon. ©Dokumentasi Penulis

Adapun, untuk membuat akun diperlukan koneksi internet pada smartphone pengguna untuk mengirimkan dan mencocokkan data dengan database pada server Ubeacon.

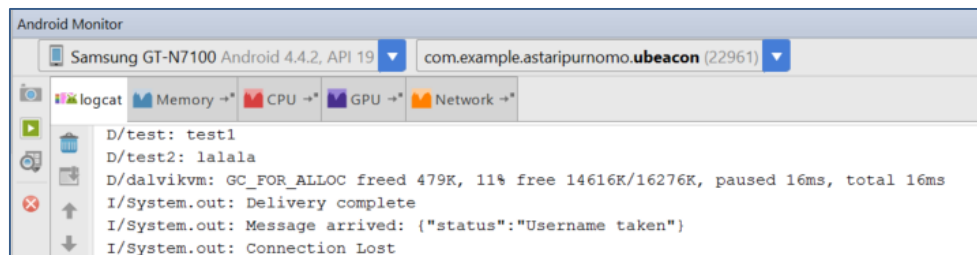
Pada tahap ini, pengguna diminta untuk mengisi sebuah form yang berisi antara lain *Name*, *Username*, *E-mail*, *Password*, *NIM*, *Birthday*, *Interest*, *Unit*, dan *Occupation*. Selanjutnya setelah tombol *CREATE ACCOUNT* diklik, maka data informasi akan dikirimkan dan dicocokkan dengan database pada server yang telah disediakan. Logcat beserta contoh pengisian pada bagian ini adalah sebagai berikut:



Gambar 28 Log pada bagian sign up ©Dokumentasi Penulis

Pada gambar tersebut dilakukan input username berupa “arsitapur” dan beberapa data input lainnya dan berhasil dengan ditandai nilai *status* nya adalah *Registraton success*. Jika dilakukan pengujian dengan melakukan input user pada saat Sign Up berupa “arsitapur”

lagi, maka registrasi akan gagal dan akan ditampilkan log pada software Android Studio sebagai berikut :



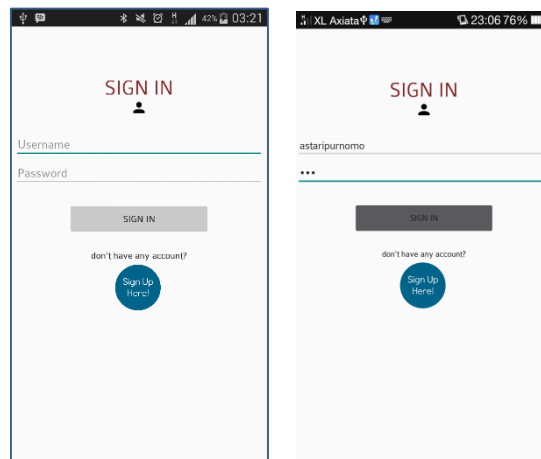
Gambar 29 Kegagalan registrasi ©Dokumentasi Penulis

Berdasarkan hal di atas, jika username sudah terdaftar pada database, maka data *status* akan didapatkan *Username taken*. Jika akun sudah terdaftar, maka pengguna dapat masuk ke keadaan selanjutnya, yaitu keadaan dimana user sudah memiliki akun namun belum melakukan Sign In di dalam aplikasi Ubeacon. Pada kondisi ini, setelah Sign Up berhasil dilakukan, state layout akan berubah dari Sign Up ke state layout Sign In.

Proses Sign-Up telah berjalan dengan baik, aplikasi mampu mengubah input dari pengguna menjadi JSON file. Aplikasi juga telah mampu melakukan koneksi ke broker menggunakan MQTT. Aplikasi juga berhasil mengirimkan paket JSON ke alamat topic yang dimaksud. Aplikasi juga telah berhasil menerima balasan dari server dan menampilkan setiap perintah dalam aplikasi dengan baik.

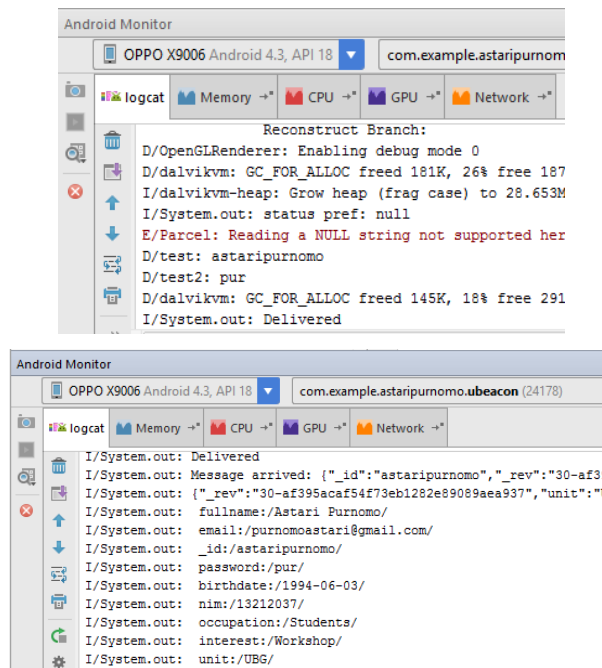
Sign-In

Pada bagian ini, pengguna telah memiliki sebuah akun pada uBeacon dengan kombinasi username dan password yang dapat digunakan untuk melakukan sign in. Username dan password ini menjadi identitas pengguna dalam aplikasi. Selanjutnya, pengguna akan diminta untuk melakukan Sign In ke dalam aplikasi dengan mengisi kombinasi *username* dan *password* yang telah dibuat. Berikut form pengisian dari Sign In yang telah dibuat serta contoh pengisian sesuai dengan akun yang telah dibuat :



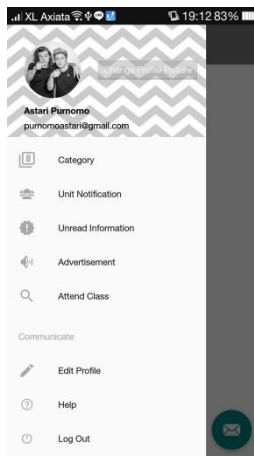
Gambar 30 Melakukan Sign in dengan benar. ©Dokumentasi Penulis

Pada gambar di atas dilakukan pengujian dengan memasukkan input yang tepat dengan kombinasi Username: *astaripurnomo* dan Password: *pur*. Dengan memasukkan input yang benar, maka aplikasi akan masuk ke dalam beranda (*homepage*) utama dari aplikasi serta menghasilkan log pada Android Studio seperti sebagai berikut :



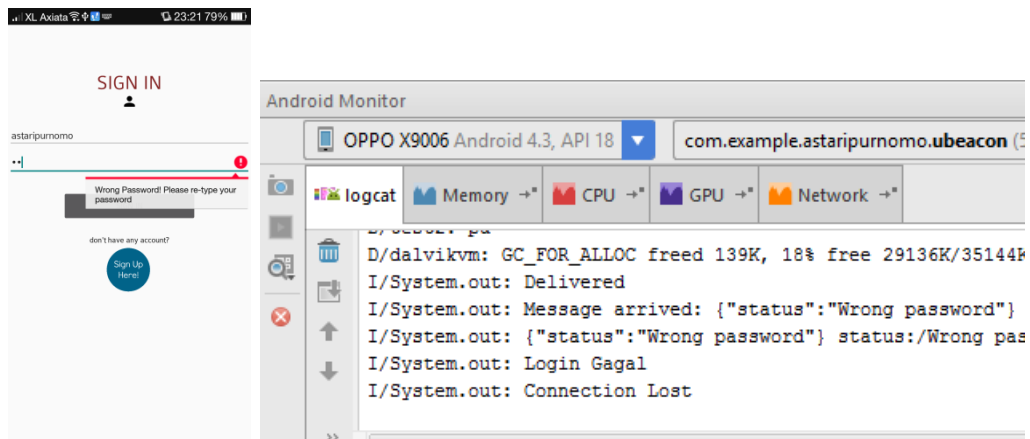
Gambar 31 Hasil logcat ©Dokumentasi Penulis

Pada proses di atas, dengan menekan tombol *Sign In* pengguna akan mengirimkan data tersebut dalam bentuk JSON file ke server melalui MQTT. Setelah pengiriman ke server dengan alamat topic tertentu berhasil dilakukan, ditampilkan log “Delivered” sesuai tampilan diatas. Saat aplikasi berhasil menerima balasan dari server maka akan ditampilkan log “Message arrived....”. Log diatas menunjukkan aplikasi mampu mengambil value dari setiap key yang dibawa oleh backend. Value tersebut selanjutnya disimpan dalam file SharedPreference. Jika prses Sign-In berhasil dilakukan maka tampilan akan berubah menjadi tampilan menu utama, sebagai berikut:



Gambar 32 Navigation bar ©Dokumentasi Penulis

Sedangkan bila sign in gagal dilakukan, maka tampilan log dan aplikasi adalah sebagai berikut:



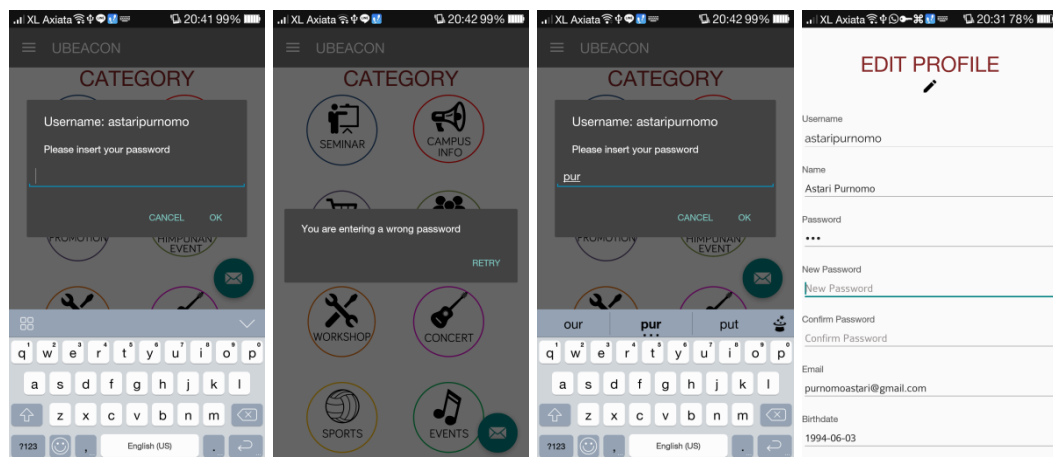
Gambar 33 Login gagal dilakukan ©Dokumentasi Penulis

Jika kombinasi username dan password yang dimasukan oleh pengguna salah, maka server akan memberikan balasan status seperti diatas. Hal ini membuat aplikasi menghasilkan pesan error seperti gambar diatas.

Nama, e-mail, dan gambar yang pengguna pilih akan ditampilkan pada navigation bar pada uBeacon. Saat proses Sign In berhasil dilakukan, state aplikasi akan diubah dan disimpan dalam Shared Preference sehingga saat pengguna menutup aplikasi dan selanjutnya aplikasi di-launch kembali, pengguna tidak perlu melakukan proses Sign In lagi selama proses Log Out belum dilakukan. Selama pengguna tidak melakukan Log Out pada sebuah smartphone, maka pengguna akan terus menjalankan akun yang sama selama belum melakukan Log Out.

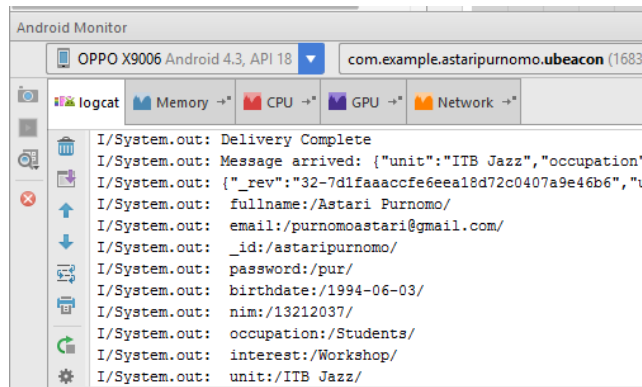
Edit Profile

Pada bagian ini, pengguna dapat melakukan perubahan pada data yang sebelumnya telah disimpan dalam database. Aplikasi akan melakukan perbandingan variable bertipe String saat hendak membuka tampilan Edit Profile. Untuk melakukan tersebut, aplikasi harus mampu memanggil password yang sebelumnya telah disimpan dalam file SharedPreferences.



Gambar 34 Tampilan Edit Profile ©Dokumentasi Penulis

Dengan berhasilnya dilakukan pergantian layout dari Navigation.class menuju Edit Profile maka fungsi SharedPreferences dalam aplikasi berjalan dengan baik. Setelah dilakukan perubahan pada data diri pengguna, dan dikirimkan ke server, diperoleh tampilan log sebagai berikut:

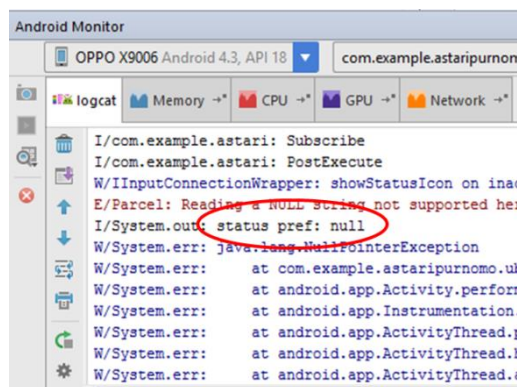


Gambar 35 Pengiriman berhasil. ©Dokumentasi Penulis

Dari hasil log diatas diperoleh bahwa aplikasi berhasil mengirimkan pake JSON ke topic dan berhasil juga memperoleh balasan dari server. Hal ini ditandai dengan berubahnya tampilan aplikasi kembali ke Navigation.class.

Log out

Pada bagian ini, saat pengguna melakukan logout maka aplikasi akan melakukan akses pada file SharedPreferences dan mengubah value dari file menjadi “null”. Hal ini dapat dilihat pda tampilan log Android sebagai berikut:



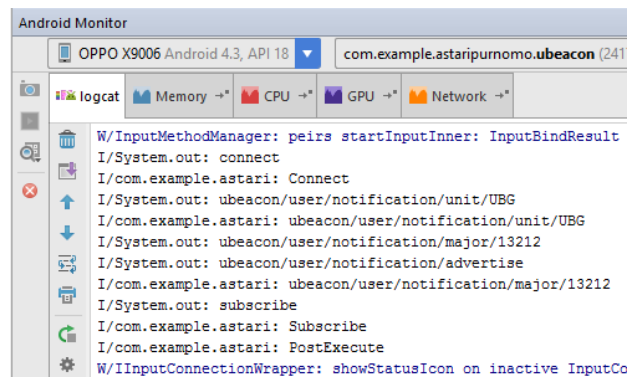
Gambar 36 Logout ©Dokumentasi Penulis

Saat aplikasi di-destroy dan pengguna meluncurkan aplikasi kembali maka aplikasi akan masuk ke state Sign-In, bukan ke menu utama. Fungsi aplikasi dalam menjalankan log out telah berhasil dilakukan dengan baik.

2.2.5.6 *Push-Notification* sebagai fitur personalisasi pengguna

Pada push notification dihasilkan rincian sebagai berikut saat dilakukan pengujian:

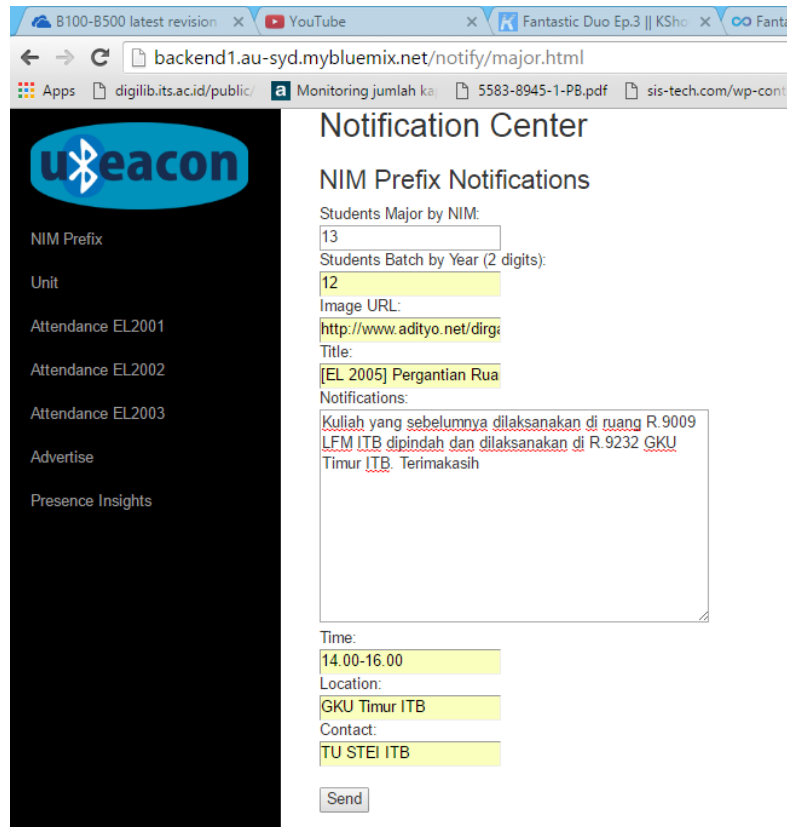
Service



Gambar 37 Service notification ©Dokumentasi Penulis

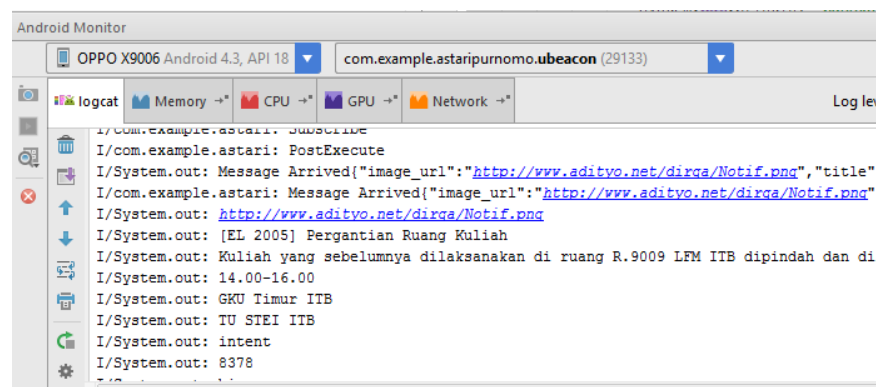
Seperti tampilan diatas, saat Navigation.class berjalan aplikasi akan melakukan subscribe ke tiga alamat topic diatas. Dalam log dapat dilihat bahwa proses koneksi menggunakan MQTT dapat berjalan dengan baik, aplikasi berhasil menempatkan dirinya sebagai subscriber dan siap menerima pesan dari server.

Selanjutnya dapat dilakukan pengetesan melalui frontend uBeacon seperti terlihat pada tampilan dibawah:



Gambar 38 Tampilan Front-end untuk fitur notifikasi. ©Dokumentasi Penulis

Sesaat setelah pesan tersebut dikirim, diperoleh tampilan log sebagai berikut:

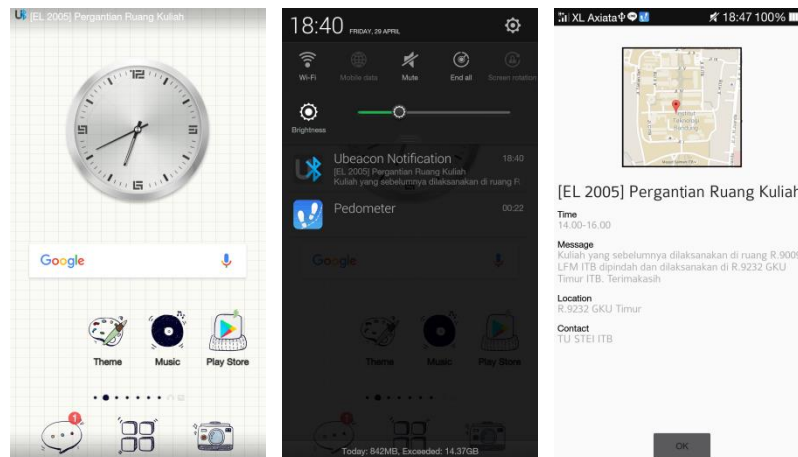


Gambar 39 balikan yang didapatkan ©Dokumentasi Penulis

Pada log ditunjukkan bahwa aplikasi berhasil menerima paket data JSON dari server. Selanjutnya paket tersebut disimpan dalam file SharedPreferences.

Notification

Proses pada service menghasilkan tampilan aplikasi sebagai berikut:

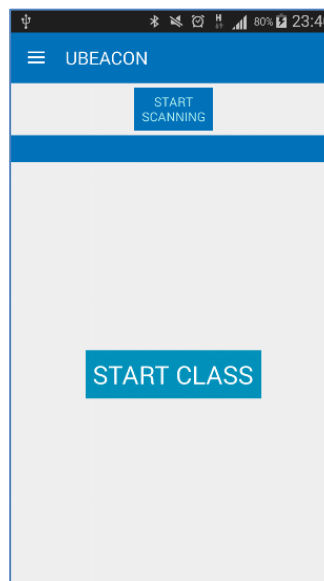


Gambar 40 Tampilan aplikasi saat push notification. ©Dokumentasi Penulis

Pada kondisi ini, aplikasi menarik value dari file `SharedPreferences` dan menampilkannya pada layout `Notification.class` (dilakukan parsing). Dari hasil yang diperoleh diatas, didapati bahwa fitur push-notificaton dapat berjalan dengan baik.

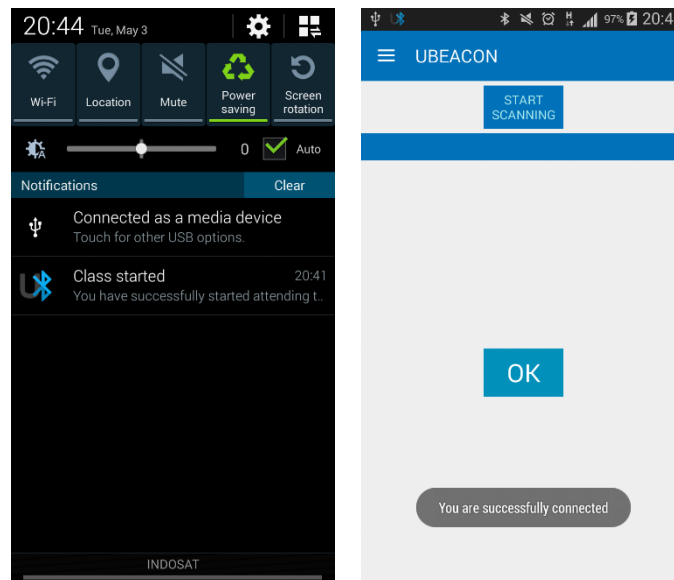
2.2.5.7 Personalisasi dengan menambahkan fitur presensi kehadiran mahasiswa dalam kelas perkuliahan

Pada fitur ini, dilakukan pengujian terhadap fungsi tombol `Start Class` yang terdapat pada fragment `AttendClass.java`. Ketika pengguna menekan pilihan fitur `Attend Class` pada *navigation bar*, maka akan muncul tampilan sebagai berikut.



Gambar 41 Tampilan fragment `AttendClass` ©Dokumentasi Penulis

Ketika pengguna menekan tombol `Start Class`, maka pengguna akan diberikan tanda bahwa pengguna sudah memulai kelas melalui notifikasi, perintah *toast*, serta perubahan tulisan pada tombol sebagai berikut.



Gambar 42 Notifikasi dan pesan saat kelas selesai ©Dokumentasi Penulis

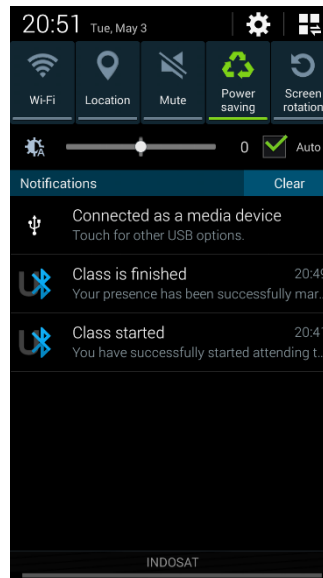
Sedangkan, pada bagian *logcat* pada Android studio ditampilkan log sebagai berikut.

```
I/System.out: Start Attending the class, start counting.
D/TextLayoutCache: Enable myanmar Zawgyi converter
D/TextLayoutCache: Enable myanmar Zawgyi converter
D/TextLayoutCache: Enable myanmar Zawgyi converter
D/TextLayoutCache: Enable myanmar Zawgyi converter
I/System.out: 24
I/System.out: Payload Major = 4
D/Sending Payload:
I/System.out: Sending Data Completed
```

Dalam hal ini, untuk melakukan konfirmasi maka pada bagian *logcat* terdapat tag “Start Attending the class, start counting”. Selanjutnya, aplikasi akan mengirimkan payload setiap 5 menit untuk menandakan keberadaan pengguna sepanjang jam pelajaran berlangsung. Oleh karena itu, dibentuk timer digital dengan interval 300000ms (5 menit). Dalam sistem ini, di asumsikan kelas berlangsung selama 2 jam. Oleh karena itu, jumlah pengiriman payload akan dilakukan sebanyak 24 kali. Hal ini ditunjukkan dengan cara memberi penanda pada bagian logcat sebagai berikut.

```
I/System.out: 23
I/System.out: Payload Major = 4
D/Sending Payload:
I/System.out: Sending Data Completed
I/System.out: 22
I/System.out: Payload Major = 4
D/Sending Payload:
I/System.out: Sending Data Completed
I/System.out: 21
I/System.out: Payload Major = 4
```

Aplikasi akan terus mengirimkan payload selama masih terhubung dengan koneksi internet dan terhubung dengan beacon. Ketika timer berhenti maka akan ditandai dengan notifikasi dan log sebagai berikut.



Gambar 43 Notifikasi yang muncul setelah kelas selesai ©Dokumentasi Penulis

Berikut tampilan payload yang dihasilkan pada bagian logcat Android Studio.

```
I/System.out: 0
I/System.out: Payload Major = 4
D/Sending Payload:
```

Perintah yang “Class is finished” menandakan telah berakhir nya jam pelajaran kuliah yang sedang berlangsung dan server telah menandakan pengguna sebagai hadir di kelas perkuliahan.

2.3 Pengujian Sub-sistem *Beacon*

Dalam melakukan pengujian terhadap sub-sistem perangkat beacon maka dilakukan pengujian pada kemampuan beacon untuk melakukan transmisi sinyal identifier yang berisi proximity UUID, major, dan minor.

2.3.1 Lingkup Pengujian

Pengujian yang dilakukan pada sub-sistem perangkat beacon dilakukan dalam lingkup pengujian sebagai berikut:

1. Pengujian parameter perangkat beacon dalam hal ini digunakan beacon dengan merek Cubeacon.
2. Pengujian kemampuan parameter pada Cubeacon untuk diubah major dan minornya sebagai identifier.
3. Pengujian radius pancar Cubeacon.
4. Pengujian nilai threshold pemancaran beacon terhadap implementasi nya pada ruang *indoor* dan *outdoor*.

2.3.2 Konfigurasi Pengujian

Pengujian pada sub-sistem perangkat beacon dilakukan dengan konfigurasi software dan hardware sebagai berikut:

1. Cubeacon

Digunakan tiga buah Cubeacon dengan parameter sebagai berikut:

Tabel III – Parameter Cubeacon (red)

| Parameter | Default value |
|----------------|--------------------------------------|
| Proximity UUID | CB10023F-A318-3394-4199-A8730C7C1AEC |
| Major | 0x0001 |
| Minor | 0x0284 |
| MAC Address | DC:2A:1D:0C:BE:1A |

Tabel IV – Parameter Cubeacon (green)

| Parameter | Default value |
|----------------|--------------------------------------|
| Proximity UUID | CB10023F-A318-3394-4199-A8730C7C1AEC |
| Major | 0x0001 |
| Minor | 0x0284 |
| MAC Address | DC:2A:1D:0C:BE:1A |

Tabel V – Parameter Cubeacon (blue)

| Parameter | Default value |
|----------------|--------------------------------------|
| Proximity UUID | CB10023F-A318-3394-4199-A8730C7C1AEC |
| Major | 0x0001 |
| Minor | 0x0284 |
| MAC Address | DC:2A:1D:0C:BE:1A |

- Smartphone Oppo Find 7a-x9006 dengan serial number 96cf428d dengan OS Android version 4.3.

- Software Cubeacon Tools App

Software ini dapat diunduh pada Play Store dan merupakan aplikasi yang digunakan untuk melakukan pengaturan pada parameter Cubeacon. Selain itu aplikasi ini juga dapat digunakan untuk scanning beacon dengan parameter proximity, UUID, major, dan minor.

- Software Ubeacon

Untuk mengukur jarak dan keberadaan beacon pada suatu kondisi.

2.3.3 Syarat Pengujian

Terdapat beberapa batasan yang digunakan dalam melakukan pengujian sub-sistem perangkat beacon diantaranya sebagai berikut:

- Pengguna mengaktifkan Bluetooth dengan minimal versi Bluetooth 4.0.
- Untuk melakukan pengujian pada kapasitas Cubeacon untuk diubah parameternya, harus diperhatikan hal-hal berikut:
 - Pengguna harus meletakkan Cubeacon pada radius 1 meter dari smartphone sebagai device yang digunakan untuk melakukan perubahan parameter.
 - Pengguna harus segera memutuskan koneksi sesaat setelah perubahan pada parameter Cubeacon dilakukan.

2.3.4 Prosedur Pengujian

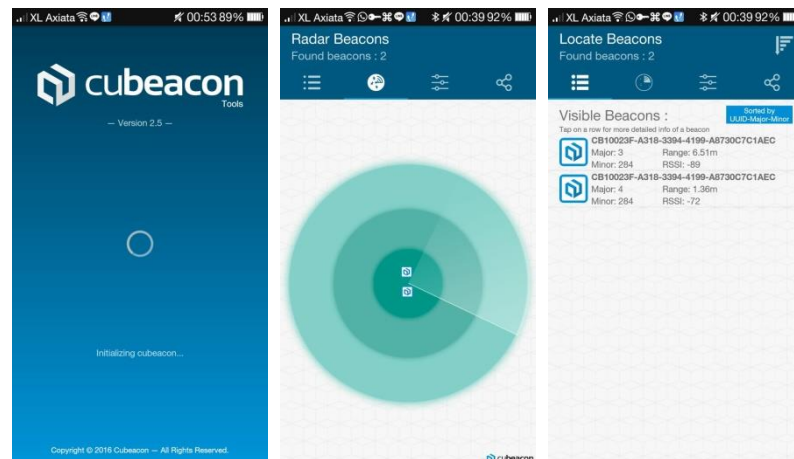
Dalam melakukan pengujian pada sub-sistem perangkat beacon terdapat beberapa prosedur sebagai berikut:

1. Pengguna mengunduh aplikasi yang mampu melakukan scanning dan konfigurasi pada Cubeacon yaitu Cubeacon Tools App pada Play Store.
2. Pengguna mengaktifkan Bluetooth.
3. Pengguna melakukan koneksi antara Cubeacon dengan aplikasi Cubeacon Tools.
4. Pengguna melakukan install aplikasi Ubeacon.
5. Penggunaan aplikasi Ubeacon untuk mengukur nilai threshold pemancaran yang akan menjadi nilai batas seseorang mengikuti kelas pada ruangan indoor.
6. Penggunaan aplikasi Ubeacon untuk mengukur jarak transmisi terjauh Beacon yang dapat dicapai.

2.3.5 Hasil Pengujian

1. Memeriksa parameter hasil scanning Cubeacon oleh smartphone.

Untuk melakukan pengujian ini maka harus dipastikan bahwa aplikasi Cubeacon Tools sedang diluncurkan dan Bluetooth sudah dinyalakan.



Gambar 44 Proses kompilasi dan instalasi aplikasi Android ke smartphone. ©Dokumentasi Penulis

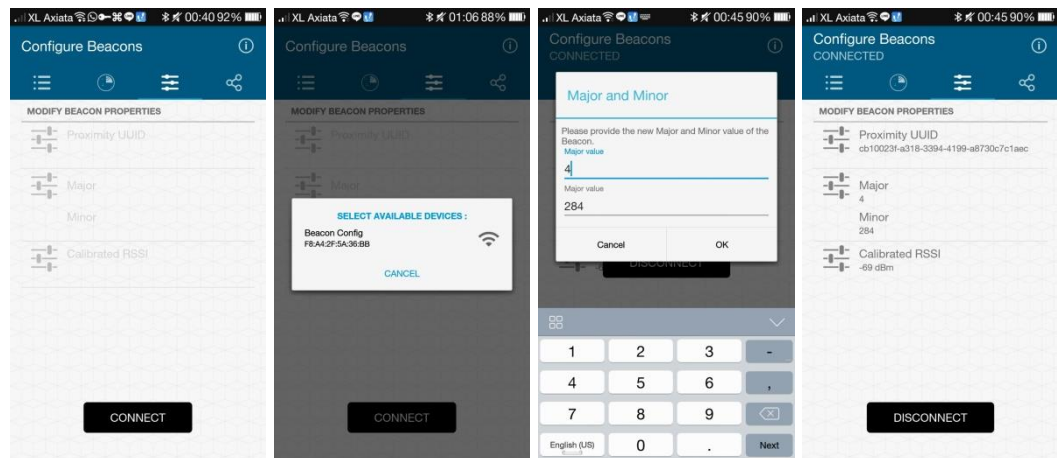
Dari gambar diatas dilihat bahwa Cubeacon terbukti memancarkan sinyal identifier dengan parameter yang sesuai.

2. Melakukan pengujian terhadap kapasitas parameter pada Cubeacon untuk diubah major dan minornya (konfigurasi) sebagai identifier.



Gambar 45 Lokasi switch untuk mengkonfigurasi beacon. ©Dokumentasi Penulis

Untuk melakukan pengujian ini, casing dari Cubeacon harus dibuka. Selanjutnya ditekan tombol K2 pada bagian dalam casing. Setelah itu connect Cubeacon dengan aplikasi Cubeacon Tools sesegera mungkin. Karena bila terlalu lama dibiarkan maka koneksi akan terhenti.



Gambar 46 Proses konfigurasi *beacon* melalui Cubeacon Tools App. ©Dokumentasi Penulis

Dari gambar diatas dapat dilihat bahwa parameter pada Cubeacon dapat diganti sesuai dengan kebutuhan pengguna.

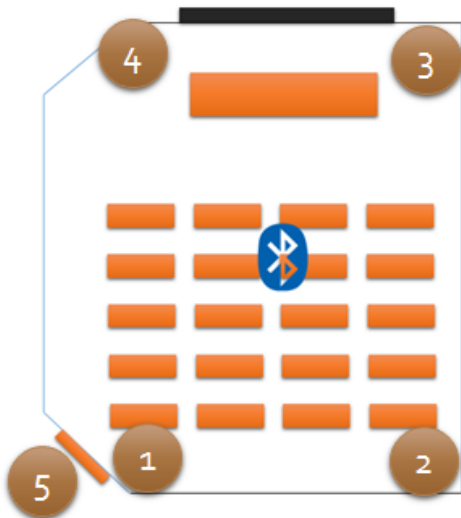
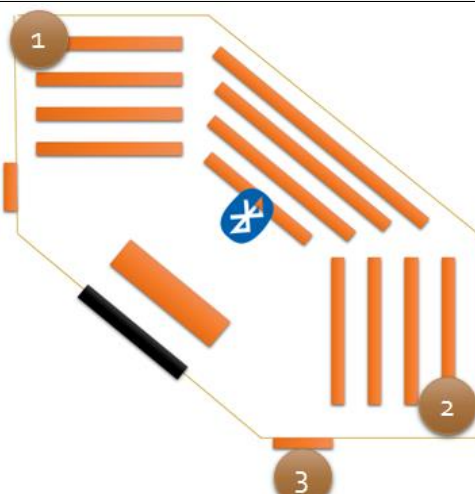
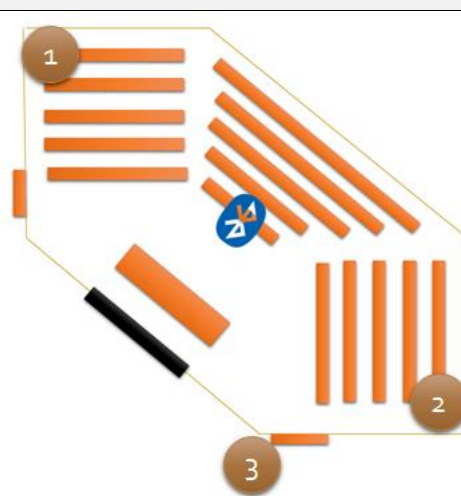
3. Melakukan pengukuran nilai threshold pemancaran yang akan menjadi nilai batas seseorang mengikuti kelas pada ruangan indoor.

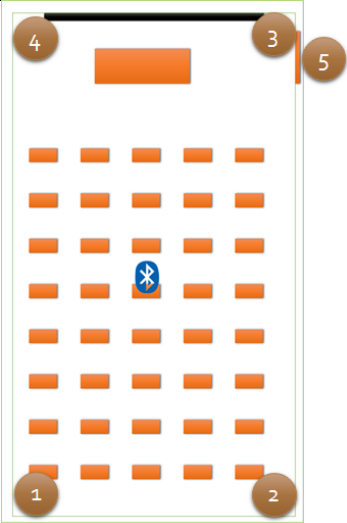
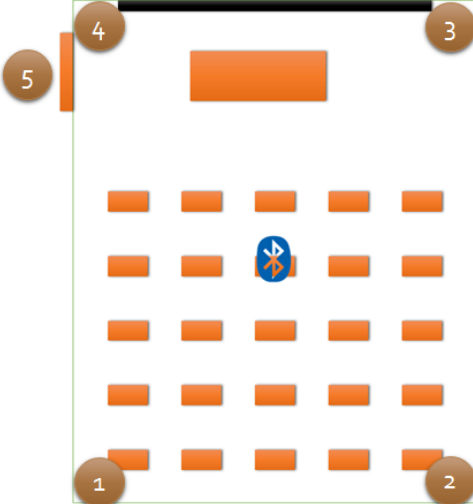
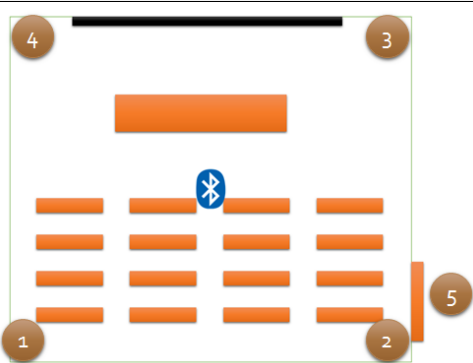
Salah satu fitur dari aplikasi Ubeacon adalah untuk menjadi platform bagi mahasiswa untuk melakukan absensi secara online langsung melalui smartphone masing-masing. Oleh karena itu, dilakukan pengujian pengukuran nilai yang menjadi batas-batas mahasiswa mengikuti sebuah kelas.

Untuk melakukan daftar kehadiran dengan menggunakan Cubeacon, seorang mahasiswa atau dosen perlu untuk masuk ke dalam fitur Attend Class seperti yang dijelaskan sebelumnya. Namun, untuk melakukan pendaftaran kehadiran, tentunya mahasiswa harus berada di dalam kelas dan mendeteksi atribut beacon dalam kelas tersebut.

Jangkauan transmisi beacon dapat dikatakan tergolong cukup jauh(dapat mencapai hingga jarak 80m), maka perlu dilakukan pembatasan nilai tertentu sehingga hanya mahasiswa yang berada di dalam kelas saja yang dapat melakukan daftar kehadiran. Oleh karena itu, dilakukan pengujian terhadap sample 5 ruang kelas dengan ukuran berbeda pada 3 gedung berbeda pada kampus Institut Teknologi Bandung. Percobaan ini dilakukan untuk mengetahui implementasi actual beacon sehingga dapat sesuai dengan fungsi nya dalam mendaftarkan absensi mahasiswa.

Pada pengujian ini diambil beberapa titik terjauh ruangan yang ditandai dengan nomor 1, 2, 3, 4, dst, serta titik lokasi dimana pengguna berada di luar ruangan. Pengukuran didapatkan dengan mengukur nilai atribut RSSI yang dipancarkan oleh Beacon yang mengindikasikan nilai *power* dari beacon yang terdeteksi. Pada pengujian ini diambil 5 sampel nilai yang akan di bentuk nilai rata-rata pada setiap titik lokasi nya. Berikut data yang didapatkan dengan nilai rata-rata dari setiap titik nya:

| No. | Gedung | Oktagon | | |
|---------|---|----------|-------|-----------|
| 1. | Ruangan | 9025 | | |
| |  | Data | | Jarak (m) |
| | | Lokasi-1 | 1.318 | 6.2 |
| | | Lokasi-2 | 1.686 | |
| | | Lokasi-3 | 1.594 | |
| | | Lokasi-4 | 1.222 | |
| | | Lokasi-5 | 3.160 | |
| Ruangan | 9016 | | | |
| 2. |  | Data | | Jarak (m) |
| | | Lokasi-1 | 2.022 | 7.4 |
| | | Lokasi-2 | 1.718 | |
| | | Lokasi-3 | 3.356 | |
| | | | | |
| 3. |  | Data | | Jarak (m) |
| | | Lokasi-1 | 2.536 | 3.856 |
| | | Lokasi-2 | 2.492 | |
| | | Lokasi-3 | 3.856 | |
| | | | | |

| | Gedung | GKU Timur | | |
|----|---|-------------|-----------|------|
| 4. | Ruangan | 9213 | | |
| |  | Data | Jarak (m) | |
| | | Lokasi-1 | 1.56 | 7.1 |
| | | Lokasi-2 | 1.672 | |
| | | Lokasi-3 | 1.618 | |
| | | Lokasi-4 | 1.654 | |
| | | Lokasi-5 | 1.920 | |
| 5. | Ruangan | 9233 | | |
| |  | Data | Jarak (m) | |
| | | Lokasi-1 | 1.4688 | 6.27 |
| | | Lokasi-2 | 1.938 | |
| | | Lokasi-3 | 1.436 | |
| | | Lokasi-4 | 1.696 | |
| | | Lokasi-5 | 2.490 | |
| | Gedung | Labtek VIII | | |
| 6. | Ruangan | 9013 | | |
| |  | Data | Jarak (m) | |
| | | Lokasi-1 | 2.012 | 8.6 |
| | | Lokasi-2 | 2.300 | |
| | | Lokasi-3 | 2.572 | |
| | | Lokasi-4 | 1.856 | |
| | | Lokasi-5 | 2.992 | |

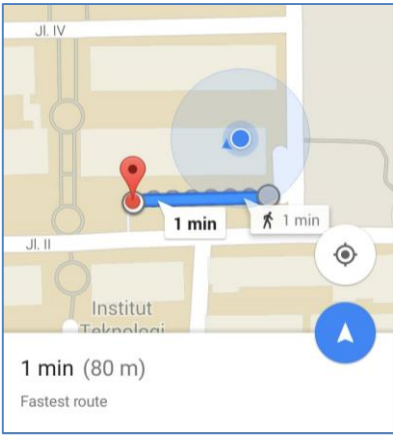
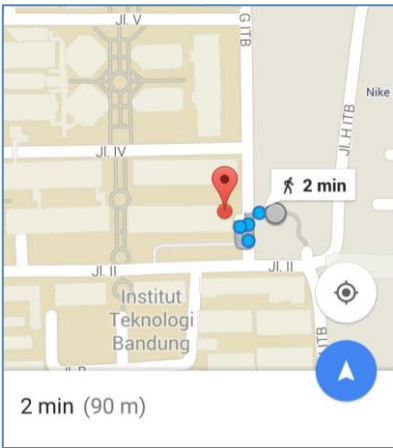
Berdasarkan nilai tersebut, didapatkan nilai pengujian di dalam ruangan indoor memiliki nilai-nilai yang hampir sama saat dilakukan pengukuran di dalam ruangan. Sedangkan, pengukuran di luar ruangan yang hanya berjarak kurang dari

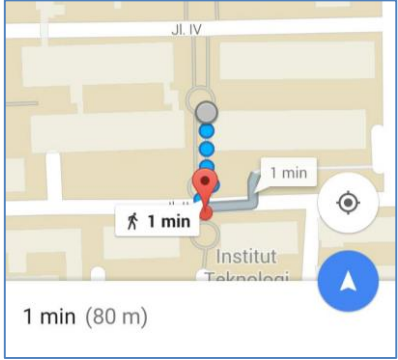
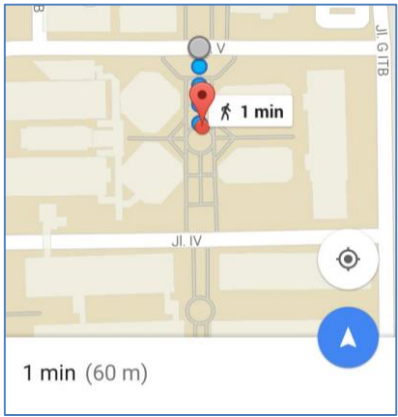
1 meter dari lokasi di dalam ruangan dekat pintu terjadi perbedaan nilai yang cukup signifikan. Hal tersebut terjadi karena terjadi interferensi signal Bluetooth yang dideteksi oleh smartphone.

Berdasarkan data tersebut, nilai terakhir(nilai yang paling besar) pada tabel dapat dijadikan nilai threshold mahasiswa yang tidak dapat melakukan aktivitas pada *Attend Class*. Jika mahasiswa mengirim tombol *Start Class* maka tidak akan ada respon tertentu dari server dan mahasiswa tidak akan dihitung melakukan absensi.

4. Melakukan pengujian terhadap radius Cubeacon.

Dalam melakukan pengujian radius beacon, pengujian dilakukan pada lingkungan terbuka(*outdoor*) untuk mencapai jarak terjauh transmisi signal Bluetooth nya. Beacon akan bekerja secara maksimal jika tidak ada interferensi signal serta berbagai hambatan yang dapat menghambat transmisi signal nya. Oleh karena itu, dipilih 4 buah tempat pengujian di dalam kampus ITB. Berikut data yang didapatkan :

| No. | Lokasi | | Jarak (m) | Gambar |
|-----|---------------------------------|---------------------------------|-----------|---|
| | Awal | Akhir | | |
| 1. | Parkir Labtek VIII (bag. Timur) | Parkir Labtek VIII (bag. Barat) | 80 |  |
| 2. | Parkir GKU Timur | Selasar Labtek VIII | 60 |  |

| | | | | |
|----|---------------------------|---------------|----|--|
| 3. | Kolam Indonesia Tenggelam | Campus Center | 80 |  |
| 4. | Gedung TVST | Sunken Court | 60 |  |

Berdasarkan hasil tersebut, didapatkan bahwa kondisi optimal dari beacon dapat tercapaian. Transmisi signal Bluetooth dapat dirasakan pada radius paling jauh yaitu 80m. Hal tersebut sesuai dengan datasheet yang disediakan oleh modul beacon CUBEACON.