



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	RANCANGAN <small>Catatan: Dokumen ini dikendalikan penyebarannya oleh Program Studi Teknik Elektro ITB</small>
Nomor Dokumen	B300-05-TA1516.01.003
Nomor Revisi	05
Nama File	B300-05-TA1516.01.003.docx
Tanggal Penerbitan	4 Mei 2016
Unit Penerbit	Program Studi Teknik Elektro Institut Teknologi Bandung
Jumlah Halaman	35 <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
RANCANGAN	6
2.1 Skenario Penggunaan	6
2.2 Rancangan Perangkat Lunak	9
2.2.1 Pendahuluan	9
2.2.2 Rancangan Proses Latar Depan (<i>Foreground Process</i>)	10
2.2.3 Rancangan Proses Latar Belakang (<i>Background Process</i>)	13
2.3 Rancangan Perangkat Keras	17
2.3.1 Pendahuluan	17
2.3.2 <i>Beacon</i>	18
2.3.3 Fungsi	19
2.3.4 Denah Pemasangan <i>Beacon</i> di Lingkungan Kampus	20
2.3.5 Cubeacon Tools Application	21
2.3.6 Deskripsi Rancangan	23
2.4 Rancangan <i>Cloud Computing</i>	26
2.4.1 Pendahuluan	26
2.4.2 IBM Presence Insights.....	28
2.4.3 IBM Cloudant NoSQL untuk Manajemen Basis Data	34

CATATAN SEJARAH PERBAIKAN

Tabel I – Catatan Sejarah Perbaikan

Versi	Tanggal	Penyunting	Perbaikan
01	6 November 2015	Astari Purnomo Rizky Indra Syafrian Adirga Khairy Ibrahim	Rilis dokumen versi 01
02	18 Desember 2015	Astari Purnomo Rizky Indra Syafrian Adirga Khairy Ibrahim	Pengubahan blok diagram pada Gambar 1 di bagian rancangan perangkat lunak Penambahan fitur <i>send feedback</i> pada bagian rancangan perangkat lunak Penambahan konfigurasi IBM Presence Insight pada bagian rancangan <i>cloud computing</i> Penambahan rancangan penempatan <i>beacon</i> dan integrasinya pada IBM Presence Insight pada bagian rancangan <i>cloud computing</i>
03	8 April 2016	Astari Purnomo Rizky Indra Syafrian Adirga Ibrahim Khairy	Perubahan pada bagian <i>cloud computing</i> Penambahan penjelasan SDK for Node.js pada bagian <i>cloud computing</i>
04	4 Mei 2016	Astari Purnomo Rizky Indra Syafrian Adirga Ibrahim Khairy	Penambahan <i>flowchart</i> dari fitur-fitur pada sistem Penambahan detail rancangan koneksi MQTT Penambahan penjelasan tentang <i>front-end</i> dan <i>administrator</i> Penambahan penjelasan tentang AJAX

PENGANTAR

1.1 Ringkasan Isi Dokumen

Dokumen ini berisi 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 I Teknik Elektro ITB tahun 2014-2015. Dokumen ini berisi rancangan sistem yang akan digunakan, pengembangan perangkat lunak, pengembangan perangkat keras, alasan pemilihan perangkat lunak dan keras, serta hubungan antar sub-sistem.

1.2 Tujuan Penulisan, Aplikasi, dan Fungsi Dokumen

Penulisan dokumen B300 ini memiliki tujuan sebagai berikut.

1. Sebagai acuan dalam implementasi sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.
2. Sebagai acuan dalam melakukan pengembangan sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.
3. Memberikan gambaran mengenai desain untuk perancangan dan pengembangan sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.
4. Sebagai salah satu dokumentasi proyek tugas akhir Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon.

1.3 Referensi

1. Eyro Digital Teknologi, Ltd. 2015. *Cubeacon Datasheet*. Eyro Digital Teknologi, Ltd. (Online). http://cubeacon.com/docs/Cubeacon%20-%20iBeacon%20Datasheet-V_2.3.pdf. Diakses 7 Oktober 2015 03:00.
2. Google, Inc. 2015. *Android Developers: Getting Started*. Android (Online). <http://developer.android.com/training/index.html>. Diakses 1 November 2015 06:00.
3. Google, Inc. 2015. *Android Developers: API Guide Bluetooth*. Android (Online). <http://developer.android.com/guide/topics/connectivity/bluetooth.html>. Diakses 1 November 2015 07:05.
4. IBM, Corp. 2015. *IBM Mobile First: Native Android Development*. IBM Developer (Online). <https://developer.ibm.com/mobilefirstplatform/documentation/getting-started-native-android-development-6-3/>. Diakses 1 November 2015 07:05.
5. Kontakt Micro-Location LLC. 2014. *Kontakt.io Datasheet v2.0*. Kontakt Micro-Location LLC (Online). Tersedia: <http://developer.kontakt.io/beacon/kontakt-beacon-v2.pdf>. Diakses 2 November 2015 17:05.
6. Estimote, Inc. 2014. *Beacon Tech*. Estimote, Inc (Online). <http://developer.estimote.com/>. Diakses 3 November 2015 14:05.
7. Creative Common Attribution-ShareAlike 4.0 International Licence. 2011. *Learn REST : A RESTful Tutorial*. (Online). <http://www.restapitutorial.com/>. Diakses 3 November 2015 15:14.
8. Quinstreet, Inc. 2015. *API-Application Program Interface*. Quinstreet, Inc (Online). <http://www.webopedia.com/TERM/A/API.html>. Diakses 3 November 2015 18:02

9. IBM Corporation. 2014. *Bluetooth Beacon*. IBM Corporation (Online). <https://presenceinsights.ibmcloud.com/pidocs/setup/beacons>. Diakses 4 November 2015 15:00.
10. 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

RANCANGAN

2.1 Skenario Penggunaan

Pada skenario penggunaan akan dibahas kasus penggunaan sistem dengan berbagai macam kemungkinan. Skenario penggunaan ini mengasumsikan bahwa pengguna telah mengunduh aplikasi yang digunakan sebagai antarmuka dari sistem pada *smartphone*-nya dan telah melakukan registrasi dalam sistem.





Skenario penggunaan dibagi berdasarkan keberadaan pengguna dan *smartphone*-nya terhadap *beacon*. Keberadaan pengguna dibagi menjadi empat yaitu ketika tidak ada *beacon* di sekitar pengguna, ada satu *beacon* di sekitar pengguna, ada lebih dari satu *beacon* di sekitar pengguna, dan ketika pengguna bergerak menjauhi *beacon*.

Skenario penggunaan juga dibagi berdasarkan kemungkinan kondisi *bluetooth*, internet, dan aplikasi Android pada *smartphone*. Kemungkinan tersebut adalah aktif atau tidaknya *bluetooth*, ada atau tidaknya koneksi internet, dan aktif atau tidaknya aplikasi Android.



Secara umum, ketika *bluetooth smartphone* dalam posisi tidak aktif, tidak banyak yang bisa dilakukan oleh aplikasi karena aplikasi tidak bisa memindai *beacon*. Pada kondisi *bluetooth* tidak aktif, keberadaan pengguna terhadap *beacon* menjadi tidak berpengaruh.

Berikut ini akan diberikan beberapa tabel skenario penggunaan dalam beberapa kemungkinan kondisi.



Tabel III – Skenario penggunaan pada kondisi *bluetooth* tidak aktif

Hasil pindai <i>beacon</i> :			<u>Berlaku untuk beberapa kondisi: Tidak ada <i>beacon</i> terdeteksi, <i>beacon</i> terdeteksi, beberapa <i>beacon</i> terdeteksi, pengguna meninggalkan jangkauan <i>beacon</i></u>		
	↑↓		Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
Off	Off	Off	⊖ Tidak ada peringatan	⊖ Tidak ada peringatan	⊖ Tidak ada notifikasi
Off	Off	On	☑ Peringatan “ <i>Bluetooth is disabled</i> ” Meminta pengguna untuk mengaktifkan <i>bluetooth</i>	☑ Peringatan “ <i>No internet connection</i> ”	⊖ Tidak ada informasi yang ditampilkan
Off	On	Off	⊖ Tidak ada peringatan	☑ Terkoneksi internet	⊖ Tidak ada notifikasi
Off	On	On	☑ Peringatan “ <i>Bluetooth is disabled</i> ” Meminta pengguna untuk meaktifkan <i>bluetooth</i>	☑ Terkoneksi internet	⊖ Tidak ada informasi yang ditampilkan
 <i>Bluetooth</i>			↑↓ Internet	 <i>Application</i>	



Tabel IV – Skenario penggunaan pada kondisi tidak ada *beacon* terdeteksi

Hasil pindai <i>beacon</i> :			Tidak ada <i>beacon</i> terdeteksi		
	↑↓		Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
On	Off	Off	⊖ Tidak ada peringatan	⊖ Tidak ada peringatan	⊖ Tidak ada notifikasi
On	Off	On	☒ Peringatan " <i>No beacon nearby</i> "	☒ Peringatan " <i>No internet connection</i> "	⊖ Tidak ada informasi yang ditampilkan
On	On	Off	⊖ Tidak ada peringatan	☑ Terkoneksi internet	⊖ Tidak ada notifikasi
On	On	On	☒ Peringatan " <i>No beacon nearby</i> "	☑ Terkoneksi internet	⊖ Tidak ada informasi yang ditampilkan

Tabel V – Skenario penggunaan pada kondisi *beacon* terdeteksi

Hasil pindai <i>beacon</i> :			<i>Beacon</i> terdeteksi		
	↑↓		Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
On	Off	Off	☑ <i>Beacon</i> terdeteksi	⊖ Tidak ada peringatan	☑ Notifikasi " <i>Beacon detected</i> " ⊖ Tidak ada informasi yang ditampilkan
On	Off	On	☑ <i>Beacon</i> terdeteksi	☒ Peringatan " <i>No internet connection</i> "	⊖ Tidak ada informasi yang ditampilkan
On	On	Off	☑ <i>Beacon</i> terdeteksi	☑ Terkoneksi internet	☑ Notifikasi " <i>Beacon detected</i> " ☑ Informasi singkat ditampilkan
On	On	On	☑ <i>Beacon</i> terdeteksi	☑ Terkoneksi internet	☑ Informasi ditampilkan

Tabel VI – Skenario penggunaan pada kondisi tidak ada *beacon* terdeteksi

Hasil pindai <i>beacon</i> :			Beberapa <i>beacon</i> terdeteksi		
	↑↓		Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
On	Off	Off	☑ Beberapa <i>beacon</i> terdeteksi	⊖ Tidak ada peringatan	☑ Notifikasi " <i>Multiple beacon detected</i> " ⊖ Tidak ada informasi yang ditampilkan
On	Off	On	☑ Beberapa <i>beacon</i> terdeteksi	☒ Peringatan " <i>No internet connection</i> "	⊖ Tidak ada informasi yang ditampilkan
On	On	Off	☑ Beberapa <i>beacon</i> terdeteksi	☑ Terkoneksi internet	☑ Notifikasi " <i>Multiple beacon detected</i> " ☑ Informasi singkat dari <i>beacon</i> terdekat ditampilkan

On	On	On	<input checked="" type="checkbox"/> Beberapa <i>beacon</i> terdeteksi	<input checked="" type="checkbox"/> Terkoneksi internet	<input checked="" type="checkbox"/> Informasi ditampilkan
----	----	----	-----------------------------------------------------------------------	---------------------------------------------------------	-----------------------------------------------------------

Tabel VII – Skenario penggunaan pada kondisi pengguna bergerak menjauhi jangkauan *beacon*

Hasil pindai <i>beacon</i> :			Pengguna meninggalkan jangkauan <i>beacon</i>		
	↑↓		Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
On	Off	Off	⊖ Tidak ada peringatan	⊖ Tidak ada peringatan	⊖ Tidak ada notifikasi
On	Off	On	<input checked="" type="checkbox"/> Peringatan " <i>No beacon nearby</i> "	<input checked="" type="checkbox"/> Peringatan " <i>No internet connection</i> "	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir ditampilkan
On	On	Off	⊖ Tidak ada peringatan	<input checked="" type="checkbox"/> Terkoneksi internet	⊖ Tidak ada notifikasi
On	On	On	<input checked="" type="checkbox"/> Peringatan " <i>No beacon nearby</i> "	<input checked="" type="checkbox"/> Terkoneksi internet	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir ditampilkan

Tabel VIII – Skenario penggunaan pada kondisi khusus

Kondisi khusus	Peringatan <i>bluetooth</i>	Peringatan internet	Aksi aplikasi
<i>Bluetooth</i> dimatikan secara tiba-tiba	<input checked="" type="checkbox"/> Peringatan " <i>Bluetooth is disabled</i> "	⊖ Tidak ada peringatan	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir ditampilkan
Internet dimatikan secara tiba-tiba	⊖ Tidak ada peringatan	<input checked="" type="checkbox"/> Peringatan " <i>No internet connection</i> "	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir ditampilkan
Informasi di- <i>refresh</i> ketika tidak ada <i>beacon</i> terdeteksi	⊖ Tidak ada peringatan	⊖ Tidak ada peringatan	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir ditampilkan
Informasi di- <i>refresh</i> ketika <i>beacon</i> yang berbeda terdeteksi	⊖ Tidak ada peringatan	⊖ Tidak ada peringatan	<input checked="" type="checkbox"/> Informasi dari <i>beacon</i> terakhir dihapus, informasi dari <i>beacon</i> terbaru ditampilkan

Bluetooth ↑↓ Internet *Application*

2.2 Rancangan Perangkat Lunak

2.2.1 Pendahuluan

Rancangan perangkat lunak pada sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon digunakan untuk menerima UUID dari *beacon*, mengirim UUID ke *cloud server*, dan menampilkan informasi yang sesuai dengan *beacon*. Semua aktivitas tersebut dilakukan pada perangkat *smartphone* yang berbasis Android.

Pengembangan aplikasi Android dilakukan dengan menggunakan *software* Android Studio beserta Java Development Kit(JDK). Emulasi aplikasi menggunakan *hardware* berupa *smartphone* OPPO Find 7 dengan sistem operasi Android Kitkat 4.4.2 yang dilengkapi dengan teknologi Bluetooth v4.0(*Bluetooth Low Energy*).

Alternatif lain yang digunakan dalam pengembangan aplikasi Android adalah dengan menggunakan *Integrated Development Environment* (IDE) Eclipse Mars 4.5 dengan *plug-in* JavaScript dan HTML, serta dengan *bundle* Android Development Tools (ADT).

Perangkat lunak ini dipilih karena mudah digunakan dan memiliki bahasa pemrograman yang mudah diimplementasikan. Perangkat keras berupa tablet OPPO Find 7 dipilih karena memiliki API Level yang lebih tinggi dari 18 sehingga memiliki support *bluetooth low energy* dari sistem operasi Android.

Tabel IX – Perangkat Lunak Pengembangan

Perangkat Lunak	Versi	Tipe	Pengembang	Bahasa	Sistem Operasi
Android Studio	1.4	Integrated Development Environment	Google	Java, XML, Gradle	Cross-platform
Eclipse IDE for JavaScript Web Developers	Mars 4.0	Integrated Development Environment	Eclipse Foundation	JavaScript, HTML, CSS, XML	Cross-platform
Java Development Kit (Standard Edition)	7.80	Plug-in	Oracle	Java	Cross-platform
Android Software Development Kit	Marshmallow 6.0 API Level 23	Plug-in	Google	Java, XML, Gradle	Cross-platform
Android Development Tools (ADT)	23.0.0	Plug-in	Google	Java, XML	Cross-platform

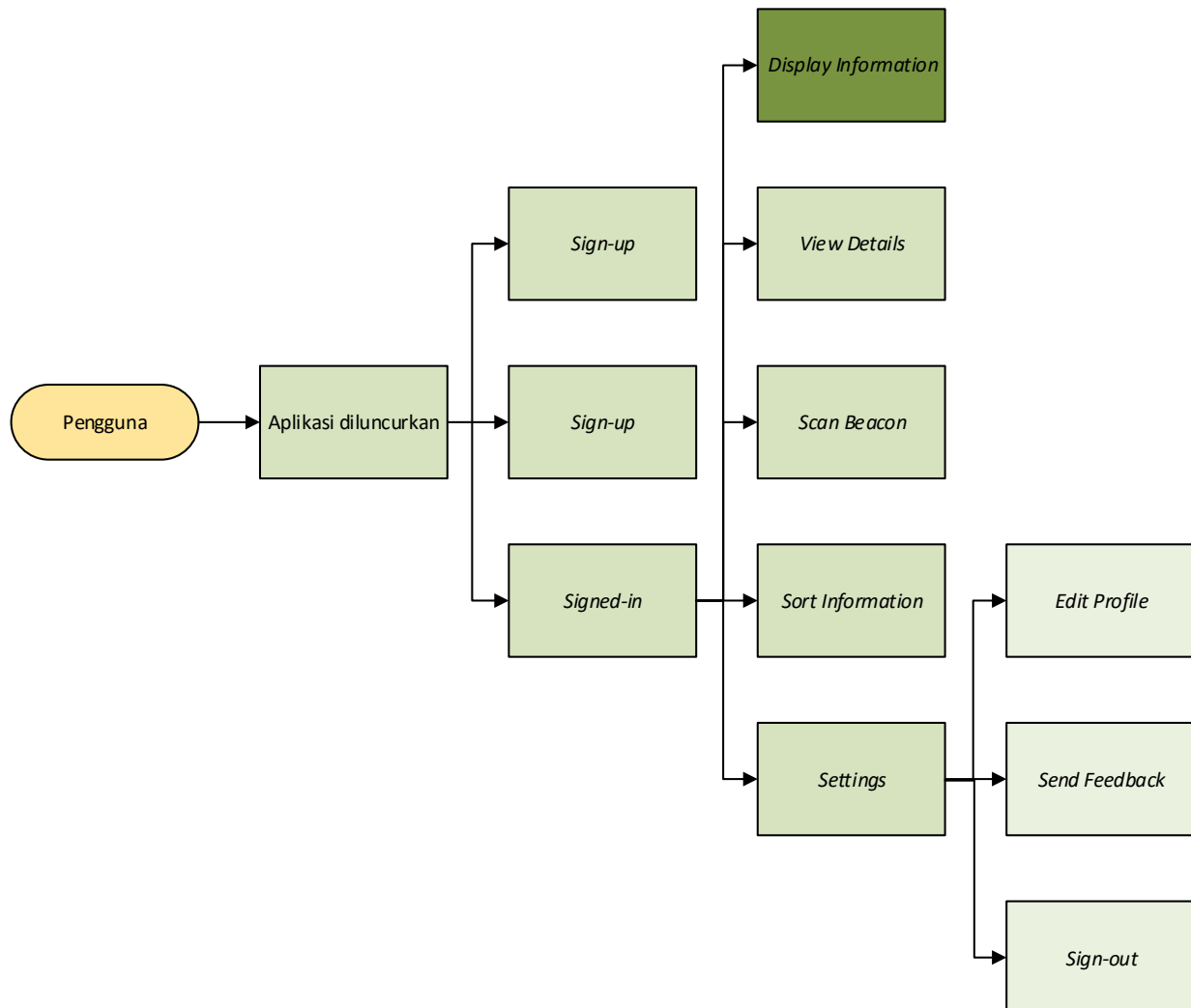
Tabel X – Perangkat Keras Pengembangan

Nama	Spesifikasi
Perangkat	Samsung Galaxy Note 10.1 GT-N8000
Sistem Operasi	Android KitKat 4.4.2 API Level 19
Bluetooth	V4.0 with A2DP

Rancangan aplikasi Android dibagi menjadi dua yaitu rancangan proses latar depan (*foreground process*) dan rancangan proses latar belakang (*background process*). *Foreground process* meliputi rancangan kerja aplikasi yang terlihat oleh pengguna. *Background process* meliputi rancangan kerja aplikasi yang tidak terlihat oleh pengguna.

2.2.2 Rancangan Proses Latar Depan (*Foreground Process*)

Foreground process pada aplikasi Android berkaitan dengan *Graphical User Interface* (GUI). GUI merupakan bagian penting dari aplikasi sebagai jembatan antara pengguna dengan seluruh perintah(*command*) yang ada pada aplikasi. Seluruh proses yang melibatkan interaksi antara pengguna dengan aplikasi akan terjadi pada bagian ini.



Gambar 1 Diagram blok penggunaan aplikasi (*foreground process*). ©Dokumentasi Penulis

Diagram blok di atas adalah skema penggunaan aplikasi Android oleh pengguna saat menggunakan aplikasi yang dibuat. Tampilan awal aplikasi dirancang untuk memasuki bagian *user account system* yang akan melihat status pengguna (*user's state*). Tampilan yang sesuai dengan status pengguna akan ditampilkan yang dapat berupa halaman untuk melakukan *sign-in*. Jika pengguna sudah melakukan *sign in*, maka pengguna akan diteruskan langsung ke bagian tampilan informasi utama aplikasi yang terdiri dari beberapa fungsi seperti pada blok diagram.

Pada tampilan kedua, akan muncul daftar dari berbagai informasi yang sesuai dengan *beacon* yang ada di sekitar *smartphone* dan informasi balikan dari server (*view*

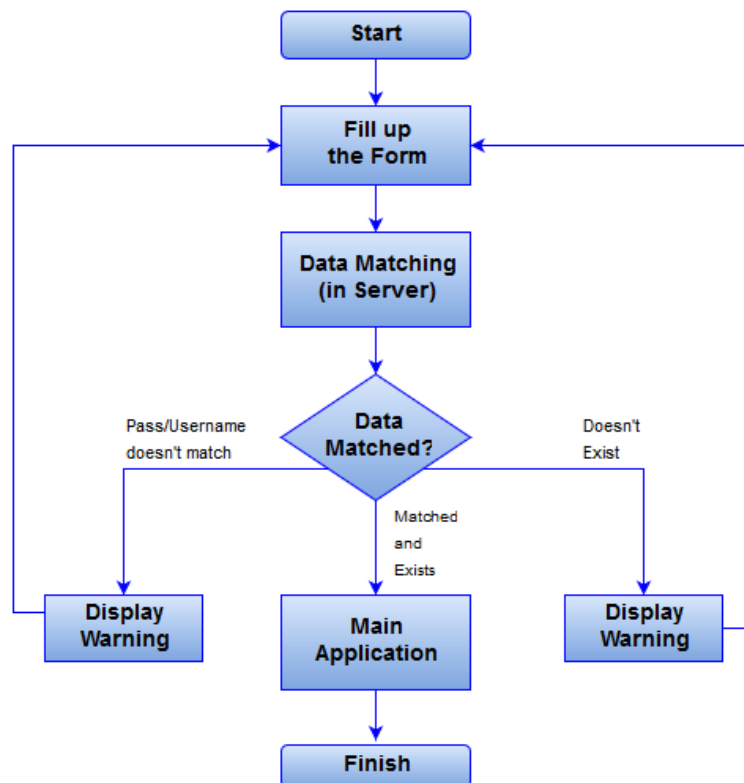
information) pada bagian *homepage*. Pada tampilan ini pula, tersedia beberapa menu untuk melakukan *scan beacon*, *sort information*, *attend class*, *edit profile*, dan *sign-out*. Jika salah satu informasi yang ada pada daftar dipilih pada bagian *homepage*, maka akan muncul tampilan informasi secara detail (*view detail*).

Berikut penjelasan dari setiap bagian dari keadaan dan fungsi dari fitur aplikasi.

1. Sign-In

Jika pengguna menggunakan aplikasi untuk pertama kalinya dan belum melakukan *sign-in* di masa lampau, maka tampilan *sign-in* akan muncul. Jika pengguna belum memiliki akun pribadi, maka pengguna perlu melakukan registrasi akun dengan menekan tombol *sign-up* dan mengisi form yang disediakan. Setelah pengguna telah berhasil melakukan registrasi, maka akan muncul menu *sign-in* kembali untuk diisi dengan data registrasi yang sebelumnya dibuat. Data yang diperlukan untuk melakukan *sign-in* adalah *username* dan *password*.

Berikut flowchart dari fitur Sign-in pada sistem ini.



Gambar 2 Flowchart Sign In Activity. ©Dokumentasi Penulis

2. Sign-Up

Sign-up merupakan mekanisme registrasi awal pengguna agar terdaftar pada sistem. Pada menu *sign-up* terdapat isian berupa nama, *e-mail*, *password*, tanggal lahir, dan minat (*interest*), pekerjaan(dosen atau mahasiswa), dan unit kegiatan mahasiswa jika merupakan mahasiswa.

3. Signed-In

Signed-in adalah kondisi ketika pengguna telah melakukan *sign-in*. Jika pengguna telah *signed-in*, maka *sign-in* tidak perlu dilakukan lagi setiap kali aplikasi diluncurkan sampai dengan pengguna memutuskan untuk melakukan *sign-out*.

4. View Information

View information merupakan tampilan *default* dari aplikasi. Pada tampilan ini, beberapa informasi yang sesuai dengan *beacon* yang terpindai diunduh dari *cloud server* dan ditampilkan. Informasi yang ditampilkan bersifat *brief* atau secukupnya saja seperti nama acara, waktu, dan tempat.

5. View Details

Jika salah satu informasi pada bagian *view information* dipilih, maka tampilan informasi secara detail akan ditampilkan. Pada tampilan ini, informasi akan ditampilkan secara utuh meliputi nama acara, deskripsi, waktu, tempat, harga tiket, dan lain-lain.

6. Scan Beacon

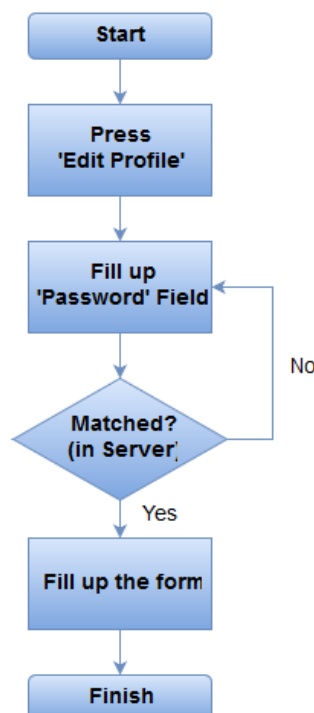
Menu ini digunakan untuk memindai *beacon* terdekat agar didapati informasi dari *beacon* terkait. Menu ini dapat juga digunakan untuk memperbaharui (*refresh*) tampilan informasi hasil pindai *beacon* yang terakhir dilakukan.

7. Sort Information

Pengguna dapat mengurutkan informasi yang muncul berdasarkan jarak *beacon* yang terdekat atau berdasarkan jenis acara yang mereka minati.

8. Edit Profile

Menu ini digunakan untuk mengubah data registrasi pengguna. Pada menu ini, pengguna dapat mengubah nama, tanggal lahir, *password*, dan minat yang digunakan untuk registrasi pada awal pendaftaran. Mekanisme dari edit profile yang akan dibangun pada sistem ini akan dijelaskan pada bagan flowchart berikut ini.



Gambar 3 Flowchart Edit Profile Activity. ©Dokumentasi Penulis

9. Send Feedback

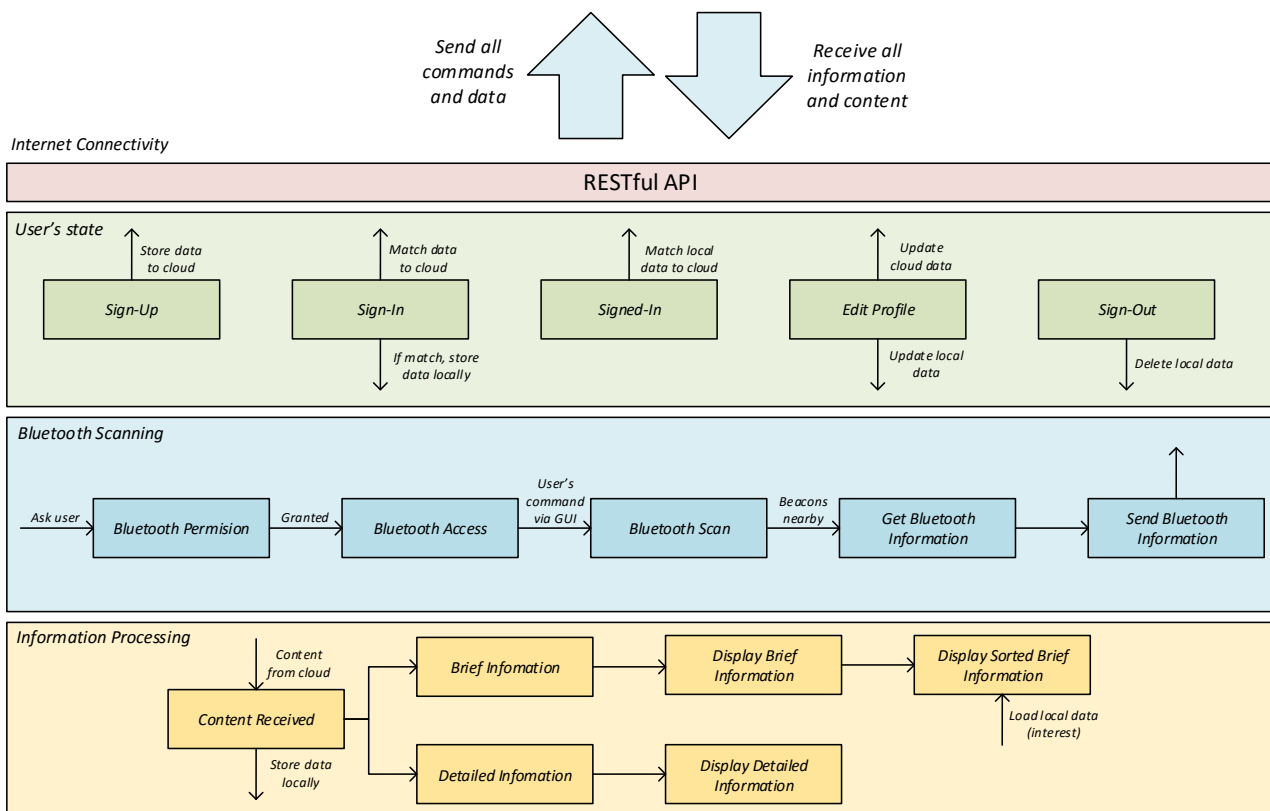
Menu ini digunakan untuk mengirimkan kritik dan saran untuk pengembang sistem ini. Adanya menu ini bertujuan agar pengembang dapat memiliki bahan pembaharuan aplikasi selain dari bahan yang ditemukan dari pencarian *bug* pada aplikasi.

10. Sign-Out

Menu ini digunakan untuk mengeluarkan pengguna dari aplikasi. Menu ini juga digunakan apabila pengguna yang sama berganti *smartphone*. Data yang sudah ada di *cloud server* tidak akan terhapus dengan menggunakan menu ini.

2.2.3 Rancangan Proses Latar Belakang (*Background Process*)

Background process pada aplikasi Android dibagi menjadi *User's State*, *Bluetooth Scanning/Monitoring*, *MQTT Service*, *Information Processing*, dan *Internet Connectivity*. *User's State* adalah keberadaan pengguna pada sistem. *Bluetooth Scanning* adalah proses pemindaian, pengambilan informasi, dan pengiriman dari *beacon* ke *cloud server*. *Information Processing* adalah proses penerimaan informasi dari *cloud server* sampai ditampilkan pada GUI aplikasi. *Internet Connectivity* adalah koneksi internet sebagai media pengirim informasi dan antarmuka yang digunakan untuk menghubungkan aplikasi dengan *cloud server*.

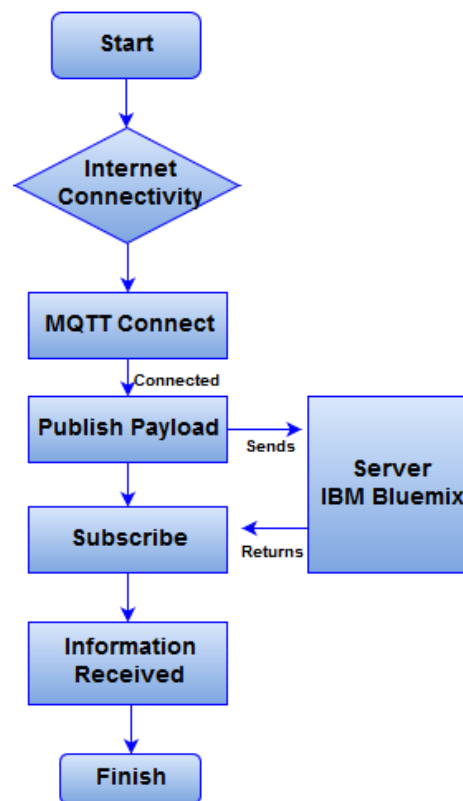


Gambar 4 Diagram blok proses yang tidak terlihat oleh pengguna (*background process*). ©Dokumentasi Penulis

MQTT Service untuk interaksi perangkat lunak dengan server

Pada perancangan perangkat lunak aplikasi yang dibuat, proses transfer data yang terjadi antara perangkat android dengan server berlangsung melalui fitur RESTful API dan dilengkapi dengan fitur MQTT Service yang dapat melakukan pengiriman data via internet dalam bentuk JSON file yang dapat diproses dengan mudah oleh server karena file yang diterima hanya merupakan sebuah file teks.

Adapun, proses yang terjadi pada MQTT Service pada devais android dijelaskan dalam flowchart berikut.



Gambar 5 Connect via MQTT ©Dokumentasi Penulis

Bluetooth Scanning/Monitoring

Bluetooth Scanning merupakan fungsi yang diperlukan untuk melakukan *scanning beacon* pada beacon. Pada fungsi ini, *bluetooth smartphone* perlu diaktifkan terlebih dahulu agar dapat memindai sinyal BLE yang ada di sekitar *smartphone* untuk diambil data-datanya. Data-data yang diambil dari *bluetooth low energy* dari *beacon* antara lain adalah:

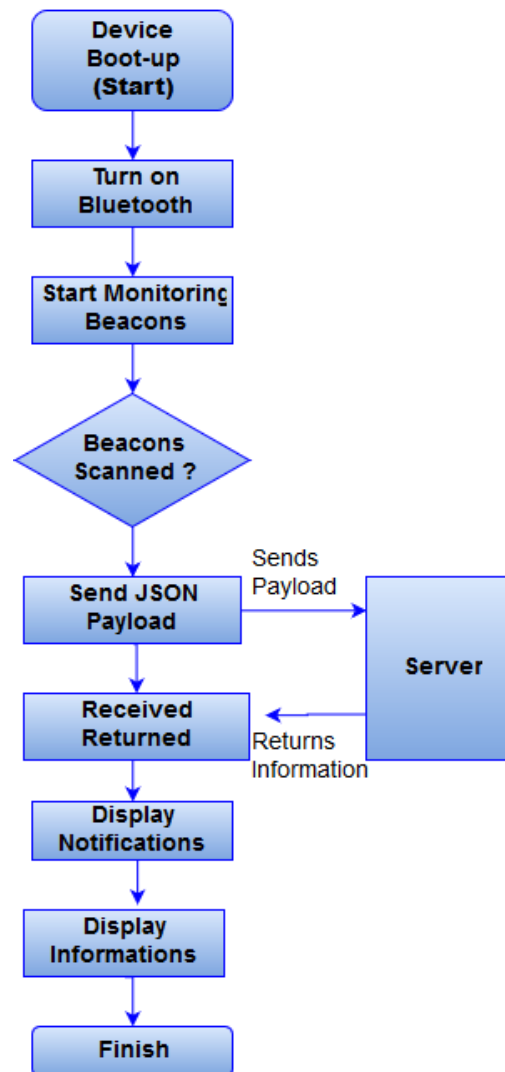
1. Bluetooth MAC Address
2. Bluetooth LE UUID (32 Byte) + Major (4 Byte) + Minor (2 Byte)
3. Bluetooth LE Proximity (number and range)
4. Bluetooth LE RSSI
5. Bluetooth LE Tx Power

Data yang dikirim ke *cloud server* untuk mendapatkan konten yang sesuai adalah Major + Minor dari UUID. Data *proximity* digunakan untuk mengurutkan informasi sesuai dengan jarak relatif *smartphone* pengguna dengan *beacon*. Data *proximity* dapat berupa angka (jarak dalam meter) dan dapat berupa status *range* (*immediate, near, far*).

Selain itu, dilibatkan pula fitur *Bluetooth monitoring* yang memungkinkan pemindaian signal Bluetooth LE pada background application dari aplikasi yang dirancang. Hal tersebut dilakukan agar dapat memudahkan pengguna dalam penggunaan aplikasi yang tidak perlu secara manual melakukan pemindaian *Bluetooth*

untuk mendapatkan informasi yang diinginkan, melainkan mendapatkannya secara otomatis ketika aplikasi mendeteksi keberadaan beacon.

Berikut rancangan flowchart dari fitur Bluetooth monitoring yang dibuat:



Gambar 6 Diagram blok *Beacon Monitoring*. ©Dokumentasi Penulis

User's State

User's State adalah status keadaan(*state*) pengguna dalam sistem. Pengguna dapat berada pada beberapa keadaan(*state*), seperti belum terdaftar pada sistem, sudah terdaftar dan belum masuk ke dalam sistem, sudah terdaftar dan sudah masuk dalam sistem, dan keluar dari sistem. Mekanisme *sign-up*, *sign-in*, dan *sign-out* diperlukan untuk menjelaskan keadaan pengguna dalam sistem. Selain itu, mekanisme ini juga digunakan untuk memastikan pengguna jika sudah terdaftar dalam *database*. Agar mekanisme ini dapat bekerja secara optimal, aplikasi harus mampu melakukan hal-hal di bawah ini.

1. Menyimpan data ke *cloud server*

Hal ini digunakan untuk registrasi pengguna baru (*sign-up*). Setiap kali pengguna baru melakukan *sign-up*, data registrasi tersimpan otomatis di *cloud server*. Untuk melakukan hal ini, dibutuhkan koneksi internet.

2. Membandingkan data dengan *cloud server*

Hal ini digunakan untuk *sign-in* pengguna dan cek data pengguna yang telah *signed-in*. Untuk melakukan hal ini, dibutuhkan koneksi internet.

3. Memperbaharui data di *cloud server*

Hal ini digunakan apabila pengguna ingin mengganti data registrasi. Data yang tersimpan pada penyimpanan lokal juga diperbaharui. Untuk melakukan hal ini dibutuhkan koneksi internet.

4. Menyimpan data di penyimpanan lokal

Hal ini digunakan untuk menyimpan data *sign-in* pengguna pada penyimpanan lokal. Data yang disimpan lokal inilah yang akan dibandingkan oleh data yang ada di *cloud server*.

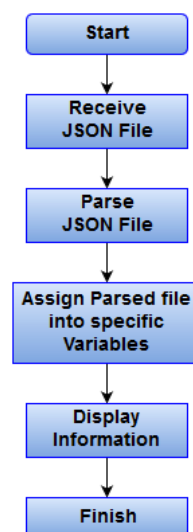
5. Menghapus data pada penyimpanan lokal

Hal ini digunakan ketika pengguna ingin *sign-out* dari sistem sehingga data *sign-in* pada penyimpanan lokal tidak bisa dibandingkan dengan *cloud server*.

Information Processing

Information Processing adalah proses pengolahan data konten yang diterima *smartphone* dari *cloud server*. Data konten yang diterima pada *smartphone* berupa sekumpulan data yang diterima dalam bentuk JSON text. Data ini kemudian dipilah berdasarkan data yang akan ditampilkan pada *brief information* dan *detailed information*. Sebuah data konten tentang sebuah acara di kampus dapat memuat judul acara, penyelenggara, waktu, tempat, dan deskripsi. Pada *brief information* hanya ditampilkan judul, waktu, dan tempat, sedangkan pada *detailed information* semua informasi ditampilkan.

Proses pengolahan informasi juga akan dilakukan untuk menentukan informasi baru dan lama. Penentuan informasi baru dan informasi lama berpengaruh pada skenario penggunaan pada saat adanya koneksi internet atau tidak (kinerja luar jaringan/*offline mode*). Proses yang harus dilakukan adalah penyimpanan informasi baru dan penghapusan informasi lama dari penyimpanan lokal *smartphone*. Penjelasan lebih lanjut tentang kinerja luar jaringan dapat dilihat pada bagian skenario penggunaan dengan tidak adanya koneksi internet. Proses ini dapat dijelaskan dalam flowchart berikut:



Gambar 7 Flowchart Information Processing. ©Dokumentasi Penulis

Internet Connectivity

Konektivitas internet dibutuhkan untuk mengirim semua *command* dan data yang dibutuhkan ke *cloud server*. *Command* yang dikirim berupa *store*, *match*, dan *update*. Data yang dikirim adalah data registrasi pengguna yang juga tersimpan pada penyimpanan lokal. Komunikasi aplikasi pada *smartphone* dengan *cloud server* harus dijumpai oleh sebuah *Application Program Interface* (API). API yang digunakan adalah RESTful API yang dikembangkan oleh Oracle.

2.3 Rancangan Perangkat Keras

2.3.1 Pendahuluan

Rancangan perangkat keras pada sistem Personalisasi Informasi dan Promosi di Lingkungan Kampus dengan iBeacon digunakan sebagai pemicu(*trigger*) bagi aplikasi *smartphone* dalam bentuk beacon. *Beacon* yang akan digunakan pada sistem ini memiliki sistem *open architecture*, yaitu *beacon* sebagai perangkat keras hanya akan memancarkan sinyal *identifier* (UUID, Major, dan Minor) secara terus-menerus. Ketika *smartphone* melakukan pemindaian sinyal radio dari *beacon* maka proses dan *handling* dari sinyal tersebut akan diambil alih oleh *software* dalam *smartphone*.

Seiring dengan perkembangan internet dan *smartphone* di Indonesia, pasar pengembangan *Internet of Things*(IoT) juga meningkat. Hal ini didukung pula dengan beragamnya produsen *beacon* di dunia, mulai dari Kontakt.io, Estimote, Cubeacon, dan lain-lain.

2.3.2 Deteksi Lokasi

Dari berbagai jenis metode deteksi lokasi yang dapat diaplikasikan pada *smartphone*, *beacon* dipilih sebagai metode deteksi lokasi karena ukurannya yang kecil, instalasinya yang mudah, dan perawatannya yang tidak memakan banyak biaya. Berikut ini merupakan tabel yang menunjukkan kelebihan dan kekurangan dari metode deteksi lokasi alternatif selain *beacon*.

Tabel XI – Perbandingan beberapa metode deteksi lokasi

Metode deteksi lokasi	Kelebihan	Kekurangan
Beacon	Ukuran kecil Instalasi mudah Penempatan yang fleksibel Tidak membutuhkan listrik	Butuh penggantian baterai setiap 2 tahun Cukup mahal jika perangkat yang digunakan banyak
Wi-Fi Router	Infrastruktur sudah tersedia di kampus Menyediakan koneksi internet	Membutuhkan instalasi listrik Penempatan tidak fleksibel Cukup mahal jika perangkat yang digunakan banyak
GPS	Tidak membutuhkan perangkat tambahan	Akurasi rendah
BTS Triangulation and Trilateration	Tidak membutuhkan perangkat tambahan	Akurasi rendah (lebih rendah dari GPS)

Beacon dengan sinyal *bluetooth*-nya dipilih sebagai metode deteksi lokasi karena ukurannya yang kecil yaitu sekitar 37x37x37mm. Karena ukurannya yang kecil maka instalasinya pun mudah yaitu dengan memasangnya ditembok, dilangit-langit, atau plafon. Pemasangannya dapat dilakukan dengan *velcro*, *double tape*, atau diletakkan begitu saja. Penempatannya yang fleksibel dan tidak membutuhkan listrik juga

mempermudah instalasinya. Penggantian baterai setiap 2 tahun sekali juga tidak memakan banyak biaya dan tidak terlalu merepotkan.

Wi-Fi *router* juga dapat digunakan sebagai salah satu metode deteksi lokasi dengan sinyal radionya. Wi-Fi *router* merupakan infrastruktur yang sudah tersedia di kampus yang digunakan sebagai penyedia koneksi internet. Karena sudah tersedia di lingkungan kampus maka perangkat ini siap untuk digunakan sebagai metode deteksi lokasi. Akan tetapi, Wi-Fi *router* di lingkungan kampus hanya terdapat pada ruangan tertutup. Penempatan Wi-Fi *router* di luar ruangan tidak fleksibel karena membutuhkan listrik. Harganya yang lebih mahal dibandingkan *beacon* juga tidak memungkinkan untuk instalasi dalam jumlah banyak.

GPS pada *smartphone* juga dapat digunakan sebagai deteksi lokasi. Kelebihan dari metode ini yaitu sudah banyaknya *smartphone* yang memiliki dukungan GPS sehingga deteksi lokasi tidak membutuhkan perangkat tambahan. Akan tetapi, kelemahan dari GPS yaitu akurasi yang rendah, terlebih lagi pada ruangan tertutup. Pada ruangan terbuka GPS dapat digunakan sebagai deteksi lokasi, tetapi pada ruangan tertutup GPS kehilangan akurasi. Karena akurasi yang rendah ini maka GPS tidak dapat digunakan pada ruangan tertutup yaitu untuk sistem monitor absensi. Hal yang sama juga berlaku untuk BTS *triangulation* dan *trilateration* yang memiliki akurasi lebih rendah dari GPS.

2.3.3 Beacon

Pada tabel di bawah akan dijabarkan *datasheet* yang dimiliki oleh Cubeacon, Kontakt.io, dan Estimote.

Tabel XII – Perbandingan antara beacon dari beberapa produsen

Spesifikasi	Cubeacon	Kontakt.io	Estimote
Feature	<ul style="list-style-type: none"> • <i>Broadcast</i> paket data menggunakan Bluetooth LE® (4.0) • Sesuai untuk digunakan di semua perangkat dengan Bluetooth 4.0 (BLE) • Standar Apple iBeacon™ • Parameter konfigurasi <i>major</i> and <i>minor</i> values • Standar Apple Advertising interval • Konsumsi daya rendah • Masa hidup baterai 2 tahun, dapat diganti 	<ul style="list-style-type: none"> • <i>Broadcast</i> paket data menggunakan Bluetooth LE® (4.0) • Kompatibel untuk digunakan di semua perangkat dengan Bluetooth 4.0 (BLE) • Standar Apple iBeacon™ • Parameter konfigurasi UUID, <i>major</i> and <i>minor</i> values, nama perangkat, <i>power level transmission</i>, dan interval iklan • Manajemen melalui <i>mobile phone</i> • Masa hidup baterai 1 tahun, dapat diganti 	<ul style="list-style-type: none"> • <i>Broadcast</i> paket data menggunakan Bluetooth LE® (4.0) • Kompatibel untuk digunakan di semua perangkat dengan Bluetooth 4.0 (BLE) • Standar Apple iBeacon™ • 32 bit ARM MO Cortex, BLE, and <i>motion temperature system</i> • Manajemen dan analisis + RESTful API • Masa hidup baterai 3 tahun, dapat diganti
Power supply	3.3 VDC Max 1.8 VDC Min		

Current	0.1 mA Max		
	50 uA Min		
RF power	-93 dBm	-12 dBm	
Working frequency	2.4 GHz ISM Band		2.4 GHz radio signal
Transmission range	±4 dB		
Operating temp	-5 °C Min		
	65 °C Max		
Firmware	Open beacon	Over-the-air, security upgrade	Over-the-air, security upgrade
Advertising interval	128 ms		
Battery model	CR2477 1000mAh	CR2477 1000mAh	CR2477 1000mAh
Range broadcast	20-80 meters		40-50 meters (default 70m)
Compatible OS (min)	iOS 7 & Android 4.3		iOS 7 & Android 4.3
BLE chip	nRF51822		nRF51822
Bluetooth version	4.0	4.0	4.0
Height	37 mm		
Width	37 mm		
Length	37 mm		
Weight	42 gram		

Dari pemaparan ketiga *datasheet* di atas, digunakan Cubeacon sebagai perangkat keras yang akan digunakan dalam pengembangan sistem. Pemilihan ini dilakukan karena Cubeacon *developer* berpusat di Surabaya, Indonesia sehingga akses dan pembelian perangkat dapat lebih mudah dibandingkan dengan pembelian Estimote atau Kontakt.io yang memerlukan *shipping* dari luar negeri. Selain itu, dengan menggunakan produk lokal ini sudah dapat dibangun sistem yang sangat memadai dengan *beacon* yang dapat di konfigurasi nilai *identifier* nya, serta *broadcast range* yang jauh yaitu mencapai 20-80 meter. Hal ini sangat cocok untuk diimplementasikan di lingkungan kampus yang cukup luas.

Cubeacon sudah memiliki standar Apple iBeacon protocol yang menjadi jaminan dalam hal kesetaraan fungsi dengan *beacon* merek lainnya. Selain itu, *battery lifetime* dari Cubeacon juga dapat bertahan lama dengan daya transmisi yang rendah. Jika dibandingkan dengan dua produsen yang lain, merek Cubeacon memiliki spesifikasi yang sesuai dengan kebutuhan sistem yang akan dibangun.

2.3.4 Fungsi

Cubeacon adalah perangkat berbasis *bluetooth low energy* (BLE) yang memancarkan sinyal *identifier* (UUID) secara terus-menerus untuk menjadi *trigger* bagi perangkat portabel di area *broadcast*-nya menggunakan standar Apple.

Saat *smartphone* menangkap sinyal *identifier* tersebut, maka akan terdeteksi beberapa parameter yaitu:

- Bluetooth MAC Address
- Bluetooth Low Energy UUID, Major and Minor value
- Bluetooth Low Energy Proximity

- Bluetooth RSSI
- Bluetooth Tx Power

Parameter tersebut (nilai *major* dan *minor*) kemudian akan ditangkap oleh aplikasi Android melalui SDK *cloud-service* dan dikirim ke *cloud server* untuk dipastikan dan selanjutnya digunakan sebuah *beacon connector* untuk menampilkan informasi dari sinyal data kembali ke *smartphone*.

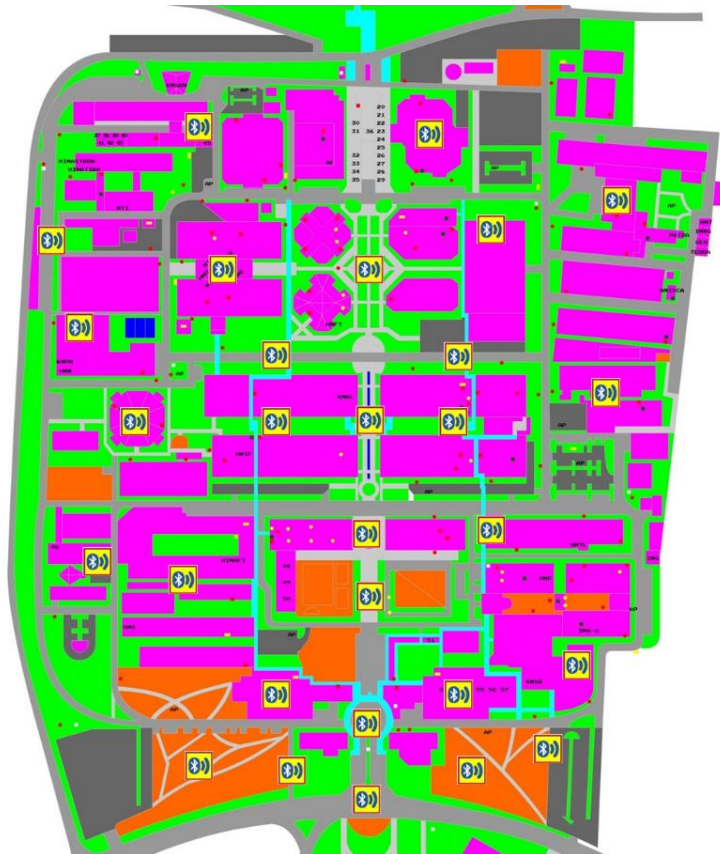
2.3.5 Denah Pemasangan *Beacon* di Lingkungan Kampus

Beacon akan ditempatkan pada lingkungan kampus dengan posisi *indoor-positioning system* dan *outdoor-positioning system*. Pemasangan dirancang dengan penempatan pada kampus ITB. Dengan mempertimbangkan luas area kampus dengan jarak yang dapat dijangkau oleh *beacon* pada posisi *outdoor*, maka didapatkan jumlah yang dibutuhkan sebagai berikut:

Tabel XIII – Perhitungan luas area kampus dan *beacon* yang dibutuhkan

Area Kampus	
Luas Tanah ITB	286830 m ²
Beacon	
Jarak Maksimum	70 m
Area Terjangkau dalam 1 Beacon	15393.8 m ²
Jumlah Beacon Dibutuhkan	19 buah

Sedangkan, penempatan pada *indoor-positioning system* dirancang untuk dapat mengakomodasi seluruh gedung perkuliahan di lingkungan kampus. Berikut penempatan *beacon* yang akan dirancang :



Gambar 8 Peta penempatan *beacon* di ITB. ©Institut Teknologi Bandung

Pada rancangan sistem ini, modul-modul *beacon* akan diletakkan di 27 titik strategis di lingkungan kampus yaitu:

- | | |
|------------------------------|---------------------------|
| 1. Jalan Ganesha | 15. DPR (2) |
| 2. Lapangan Sipil (2) | 16. GKU Barat |
| 3. Lapangan Seni Rupa | 17. GKU Timur |
| 4. Tempat parkir Seni Rupa | 18. Perpustakaan |
| 5. Aula Barat | 19. BSC-A |
| 6. Aula Timur | 20. BSC-B |
| 7. Gedung Arsitektur | 21. Oktagon |
| 8. Boulevard | 22. Labtek Biru |
| 9. Information Center | 23. Gedung SBM |
| 10. Gedung Teknik Lingkungan | 24. Gerbang Matematika |
| 11. Gedung Teknik Sipil (2) | 25. Gedung Hanggar |
| 12. Indonesia Tenggara | 26. Gedung FTTM |
| 13. Labtek VIII | 27. Gedung Mekanika Tanah |
| 14. Labtek V | |

Jumlah *beacon* yang dibutuhkan dalam rancangan sistem ini ditunjukkan oleh tabel berikut :

Tabel XIV – Jumlah *beacon* yang diperlukan untuk ditempatkan pada setiap kelas

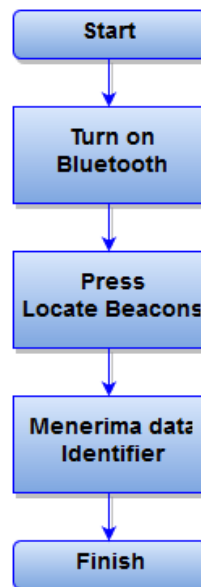
Jumlah Gedung Perkuliahan	27 Gedung
Rata-Rata Jumlah Ruang Kelas Setiap Gedung (Asumsi)	10 Ruang
Total Beacon dibutuhkan	270 buah

Oleh karena itu total *beacon* yang dibutuhkan adalah sebanyak 289 buah untuk mengakomodasi seluruh lingkungan kampus.

2.3.6 *Cubeacon Tools Application*

Cubeacon memiliki antarmuka berupa *developer kit* yang dapat digunakan untuk melakukan *monitoring* terhadap *beacon* yang diletakkan di lingkungan kampus mulai dari melihat aktif atau tidaknya *beacon*, jarak *beacon*, kalibrasi untuk mengetahui daya transmisi dari masing-masing *beacon* juga melakukan pengaturan nilai UUID, *minor*, *major*, dan RSSI.

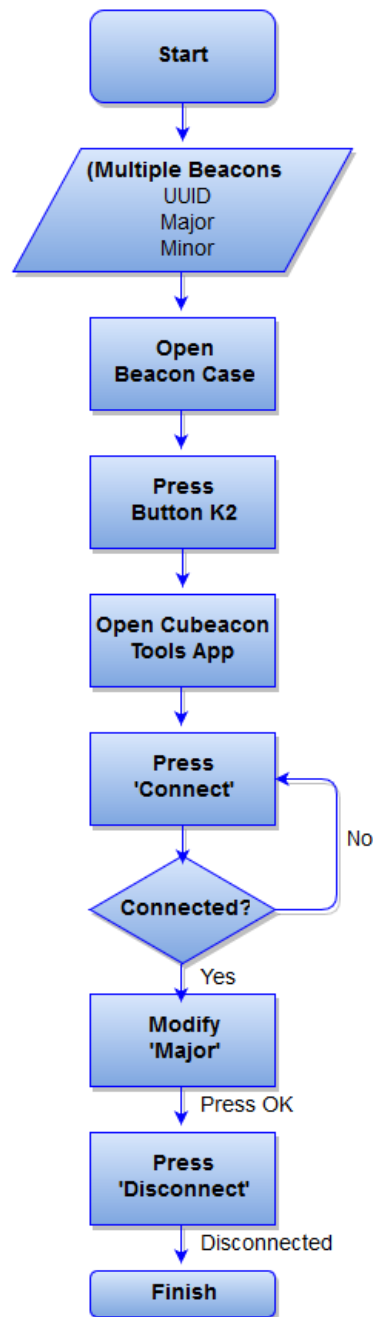
Untuk melakukan *monitoring* Cubeacon dapat dilakukan langkah-langkah berikut:



Gambar 9 Flowchart monitoring beacons. ©Dokumentasi penulis

Sistem personalisasi di lingkungan kampus dengan beacon ini melibatkan sejumlah beacon yang akan diletakkan di berbagai titik di kampus. Sedangkan, secara *default* saat melakukan pembelian cubeacon akan didapatkan 3 buah beacon dengan nilai identifier *major* dengan nilai 1, 2, dan 3. Oleh karena itu, perlu dilakukan konfigurasi dengan cara melakukan kustomisasi pada identifier beacon dengan menggunakan *Cubeacon Tools App*.

Berikut langkah untuk menggunakan Cubeacon Tools App untuk mengubah nilai Major pada setiap beacon.



Gambar 10 Flowchart pengubahan nilai ID Major pada beacon. ©Dokumentasi Penulis

2.3.7 Deskripsi Rancangan

Spesifikasi dan parameter yang dimiliki oleh setiap Cubeacon selain dari *datasheet* yang digunakan untuk perbandingan.

Tabel XV – Operation State & Power Consumption

State	DC/DC Enable OFF (mA)	DC/DC Enable ON (mA)
Wake-up	7.1	5.4
Pre-processing	8.4	6.9
Pre-receiving	11.6	10.7
Receiving	18.9	15.4
Rx changed	9.2	5.8
Transmitting	18.3	15.4

Tabel XVI – Parameter Cubeacon (Red)

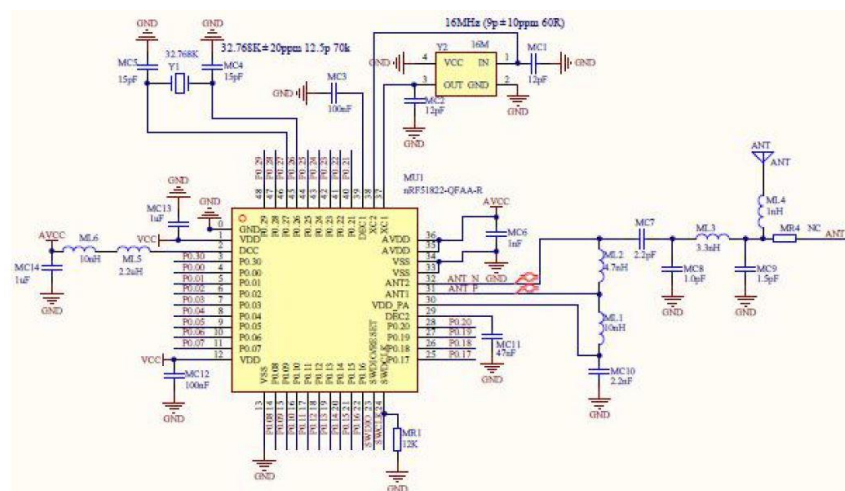
Parameter	Default value
Proximity UUID	CB10023F-A318-3394-4199-A8730C7C1AEC
Major	0x0001
Minor	0x0284
MAC Address	DC:2A:1D:0C:BE:1A

Tabel XVII – Parameter Cubeacon (Green)

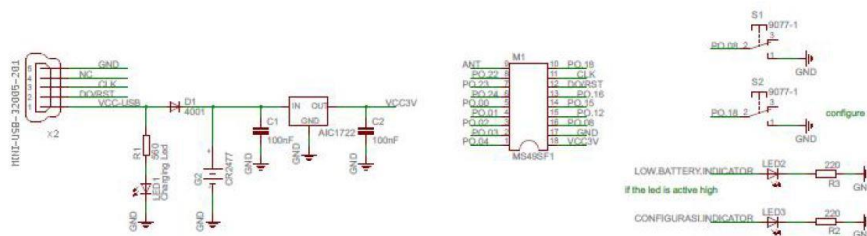
Parameter	Default value
Proximity UUID	CB10023F-A318-3394-4199-A8730C7C1AEC
Major	0x0002
Minor	0x0284
MAC Address	F8:A4:2F:5A:36:BB

Tabel XVIII – Parameter Cubeacon (Blue)

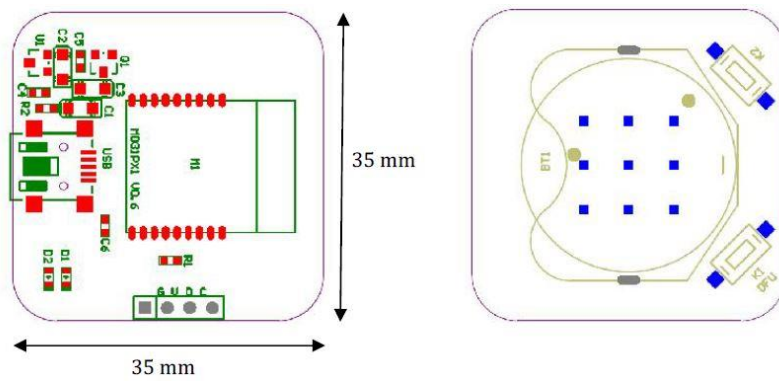
Parameter	Default value
Proximity UUID	CB10023F-A318-3394-4199-A8730C7C1AEC
Major	0x0003
minor	0x0284
MAC Address	DE:EF:66:D1:9B:B1



Gambar 11 Skema rangkaian internal *beacon*. ©Eyro Digital Teknologi, Ltd.



Gambar 12 Skema rangkaian internal *beacon*. ©Eyro Digital Teknologi, Ltd.



Gambar 13 Skema PCB *beacon*. ©Eyro Digital Teknologi, Ltd.



Gambar 14 Bentuk fisik *beacon*. ©Eyro Digital Teknologi, Ltd.

2.4 Rancangan *Cloud Computing*

2.4.1 Pendahuluan

Dalam perancangan sistem *cloud computing* akan digunakan *server* IBM Bluemix. IBM Bluemix digunakan karena merupakan *server* yang menyediakan *runtime* yang lengkap dan beberapa aplikasi analisis yang siap pakai. Salah satu aplikasi siap pakai ini adalah IBM Presence Insights yang akan membantu pembuatan analisis keramaian pengunjung kampus dan menampilkannya dalam bentuk grafik dan peta keramaian. Untuk *database* akan digunakan IBM Cloudant NoSQL sebagai aplikasi penyedia *database* untuk menyimpan data pengguna, informasi *beacon*, dan daftar peserta kelas untuk sistem absensi.

Fungsi utama dari *cloud computing* adalah untuk mengolah data pengguna, mengolah data *beacon* dan informasi *beacon*, mengirimkan notifikasi kepada pengguna, dan menjadi sistem absensi kelas. Pengolahan data pengguna termasuk registrasi akun, ubah profil akun, dan penghapusan akun. Pengolahan data *beacon* dan informasinya termasuk menambahkan data *beacon* ke *database* dan menghubungkannya dengan IBM Presence Insight. Pengiriman notifikasi dan pemantauan sistem absensi kelas dilakukan dengan *front-end* yang akan dibangun.

Administrator akan bertugas sebagai pengelola *database* pengguna, pengelola *database* informasi *beacon*, pengirim notifikasi, pemantau sistem absensi, dan pemantau hasil analisis IBM Presence Insight. *Administrator* bertugas juga untuk melayani pemasangan iklan dan pengiriman notifikasi ketika ada pihak yang mengajukan. *Administrator* juga dapat menyampaikan daftar hadir kelas ke program studi, dosen, atau pihak akademik terkait.

Sebuah *front end* akan dibangun untuk memudahkan pekerjaan-pekerjaan *administrator*. *Front-end* akan dibuat di atas *back-end* dan berkomunikasi langsung dengan *back-end* menggunakan AJAX (Asynchronous Javascript and XML). *Front-end* dibuat agar *administrator* tidak perlu melakukan pekerjaannya langsung di web IBM Bluemix atau dengan mengubah program *back-end*. Tetapi untuk beberapa pekerjaan masih harus dilakukan manual pada web IBM Bluemix atau IBM Presence Insights seperti penghapusan/penambahan *database* informasi *beacon* dan pengelolaan *database* pengguna dan daftar peserta kelas. Karena *front-end* yang dibangun hanya akan digunakan oleh *administrator*, maka yang diutamakan pada *front-end* adalah fungsionalitas di samping tampilan dan *user experience*.

2.4.2 SDK for Node.js, JSON, dan MQTT

Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

Pick a plan Monthly prices shown are for country or region: [Indonesia](#)

Plan	Features
✓ Default	Run one or more apps free for 30 days (375 GB-hours free). \$0.0735 USD/GB-Hour

This is a service plan for the IBM Bluemix Platform runtime.

Start with a runtime:

Space: Rizky Indra Syafrian

Name: Enter new app name

Host: Enter host

Domain: au-syd.mybluemix.net

Selected Plan: Default

CREATE

Gambar 15 Tampilan SDK for Node.js pada katalog IBM Bluemix. ©Eyro Digital Teknologi, Ltd.

SDK for Node.js adalah sebuah *runtime environment* yang disediakan IBM Bluemix untuk membuat *back-end application*. Untuk membuat sebuah *back-end application*, program ditulis dalam JavaScript dengan Node.js Module dan kemudian di-deploy ke *server*. Setelah program di-deploy program akan berjalan pada *server* untuk melayani semua *request* yang datang dari aplikasi Android dari seluruh pengguna terdaftar.

Node.js dipilih sebagai *runtime environment* dari *back-end application* yang dibuat karena merupakan *framework/platform* yang dianggap paling mumpuni. Node.js digunakan untuk mengembangkan *back-end application* yang akan menangani *input/output* yang intensif. Ketika *back-end application* perlu menangani *request* dari aplikasi Android yang digunakan banyak pengguna secara sekaligus, Node.js dapat dengan mudah menangani hal tersebut.

Back-end application yang dibuat dapat di-deploy sementara pada *server* lokal (pada PC yang digunakan untuk proses pengembangan). Setelah siap, maka kemudian dapat di-deploy ke *server* agar dapat diakses oleh aplikasi Android melalui internet dengan *bandwidth* besar yang telah disediakan oleh IBM Bluemix.

Node.js sebagai *runtime environment* dari *back-end application* yang dibuat melakukan pertukaran informasi dengan aplikasi Android dengan bentuk paket data JSON (*JavaScript Object Notation*). JSON merupakan format standar untuk transmisi data asinkron yang biasa digunakan untuk komunikasi *browser/server*. JSON berisikan data yang mudah di baca dengan format pasangan atribut-nilai. Paket-paket data JSON kemudian ditransmisikan melalui API (*Application Programming Interface*).

Untuk komunikasi antara *back-end application* dengan aplikasi Android melalui internet digunakan protokol MQTT (*Message Queuing Telemetry Transport*). MQTT merupakan protokol pengiriman pesan yang bersifat *light-weight* digunakan pada *internet protocol suite*. Protokol ini dapat digunakan untuk pengiriman pesan meskipun *bandwidth* internet yang digunakan terbatas. Pengiriman pesan pada protokol ini menggunakan pola *publish-subscribe* dengan *publisher* sebagai pengirim

pesan dan *subscriber* sebagai penerima pesan. Pola *publish-subscribe* membutuhkan sebuah *broker* yang berperan sebagai distributor pesan ke *client* berdasarkan *topic*-nya. *Topic* berperan sebagai kanal tempat pesan dikirimkan dan diterima oleh *publisher* dan *subscriber*.

2.4.3 IBM Presence Insights

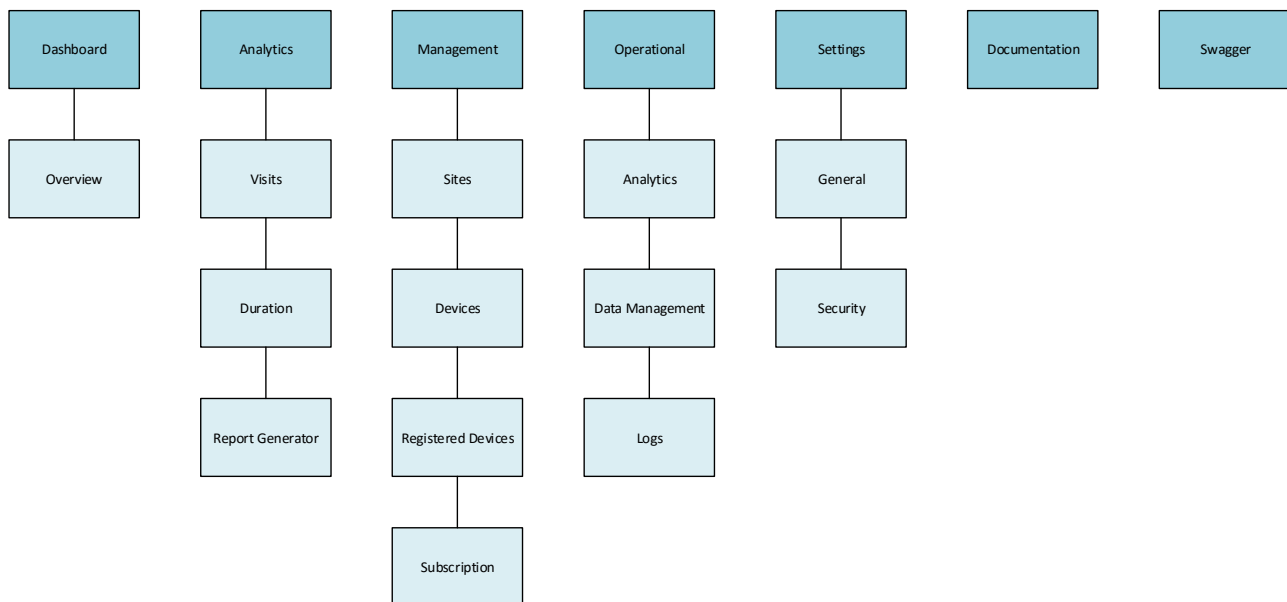
The screenshot shows the IBM Presence Insights service page. On the left, there is a sidebar with the IBM logo, the service name 'Presence Insights', and metadata: 'PUBLISH DATE 04/04/2016', 'AUTHOR IBM', 'TYPE Service', and 'LOCATION Sydney'. A 'VIEW DOCS' button is at the bottom of the sidebar. The main content area has a heading 'IBM Presence Insights helps you understand mobile activity in and around a physical location, such as malls, stadiums, airports and retail establishments. Presence Insights extends customer service, inventory promotions, and support through your customer's mobile devices.' Below this, there are two bullet points: '• Deepen customer engagement' and '• Optimize onsite staff operations'. Each bullet point has a brief description of the service's capabilities. At the bottom of the main content area, there are three small screenshots showing the service's interface. On the right, there is a 'Add Service' form with fields for 'Space' (Rizky Indra Syafrian), 'App' (Leave unbound), 'Service name' (Presence Insights-zj), and 'Credential name' (Credentials-1). There is also a 'Selected Plan' dropdown set to 'Free' and a 'CREATE' button.

Gambar 16 Tampilan Presence Insights *service* pada katalog IBM Bluemix. ©Eyro Digital Teknologi, Ltd.

Salah satu fitur utama dari sistem ini adalah fitur promosi yang dapat langsung disampaikan kepada pengguna perangkat Android yang sedang berada di sekitar perangkat *beacon*. Agar fitur promosi dapat berjalan secara efektif, maka perlu dilakukan analisis terhadap perilaku pengguna Android di setiap perangkat *beacon* sehingga didapatkan data aktivitas dan *traffic* dari setiap perangkat *beacon*, serta beberapa data penting lainnya untuk selanjutnya dilakukan tindakan lebih lanjut. Hal tersebut dapat dilakukan dengan menggunakan fitur IBM Presence Insights pada IBM Bluemix.

IBM Presence Insights (PI) pada dasarnya dapat mendeteksi aktivitas pengguna perangkat Android pada lokasi tertentu menggunakan sebuah perangkat yang dalam hal ini adalah *beacon*. Implementasinya dalam sistem ini adalah untuk mengetahui aktivitas dan *traffic* dari kegiatan di setiap gedung yang dipasang perangkat *beacon*. Selanjutnya, dapat dilakukan pengolahan data dengan cara memperhatikan lokasi, kegiatan yang berlangsung, serta jumlah aktivitas yang terjadi sehingga penempatan iklan dan promosi akan menjadi lebih efektif dan sistematis. Adapun, tampilan Presence Insights dalam IBM Bluemix ditunjukkan oleh gambar berikut.

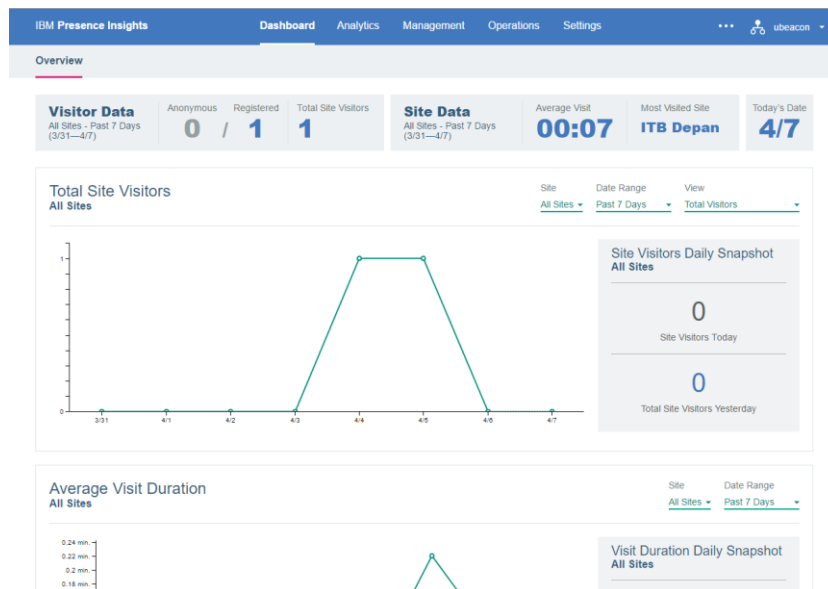
PI menawarkan beberapa fitur dan fungsi yang berguna untuk keperluan pengambilan data dan analisisnya dari perangkat yang mendeteksi *beacon*. Berikut ini akan dijelaskan fitur dan fungsi yang ada pada PI.



Gambar 17 Menu dan fungsi yang terdapat di IBM Presence Insights. ©Eyro Digital Teknologi, Ltd.

1. Dashboard Overview

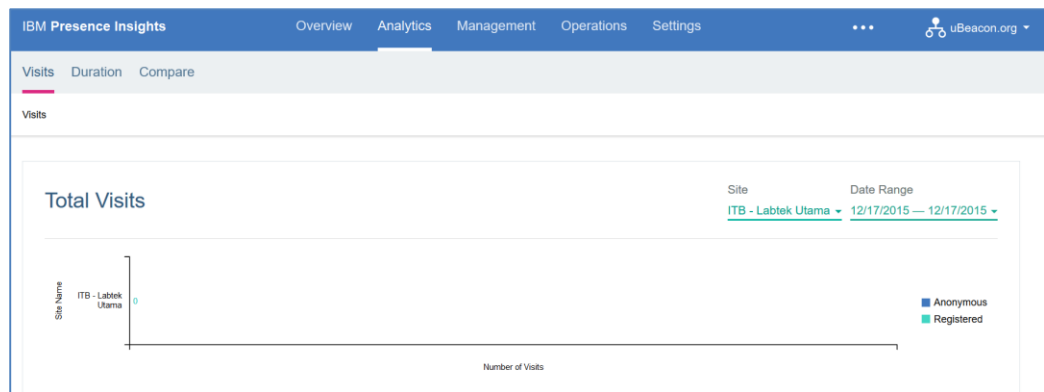
Pada tampilan ini, ditampilkan jumlah pengunjung dari semua *sites* yang telah dibuat. Jumlah pengunjung dipisahkan berdasarkan pengunjung tidak terdaftar (*anonymous*) dan terdaftar (*registered*). Jumlah data yang ditampilkan bisa diatur berdasarkan tujuh hari terakhir, satu bulan terakhir, tiga bulan terakhir dan seterusnya. Ditampilkan *sites* yang paling banyak dikunjungi. Ditampilkan grafik pengunjung dari semua *sites* berdasarkan durasi kunjungan. Ditampilkan grafik rata-rata durasi kunjungan dari semua *sites*.



Gambar 18 Dashboard Overview pada IBM Presence Insights. Data yang ditampilkan merupakan percobaan implementasi ©Eyro Digital Teknologi, Ltd.

2. Visits Analytics

Pada tampilan ini, ditampilkan analisis jumlah pengunjung pada setiap *sites*. Ditampilkan analisis jumlah pengunjung berdasarkan waktu kedatangannya. Ditampilkan juga analisis jumlah pengunjung berdasarkan keramaiannya pada peta *floor layout* yang diunggah.



Gambar 19 Tampilan *visits* pada menu *analytics*. ©IBM Corporation.

3. *Duration Analytics*

Pada tampilan ini, ditampilkan analisis rata-rata durasi kunjungan pada setiap *sites*. Ditampilkan analisis rata-rata durasi kunjungan berdasarkan waktu kedatangannya. Ditampilkan analisis rata-rata durasi kunjungan berdasarkan keramaiannya pada peta *floor layout* yang diunggah. Grafik yang ditampilkan kurang lebih sama dengan grafik *visits analytics*.

4. *Report Generator*

Menu ini digunakan menghasilkan dokumen laporan dari data dan analisis yang telah dilakukan oleh PI. Digunakan juga untuk membandingkan data dari dua *sites* untuk kemudian dianalisis lebih lanjut.

5. *Sites*

Menu ini digunakan untuk mengunggah peta *layout* tempat *beacon* akan ditempatkan dan membuat *sites*, *floors*, dan *zones*. Untuk membuat sebuah *sites*, urutan pembuatannya adalah sebagai berikut.

- Membuat sebuah *sites* baru
- Menambahkan satu atau lebih *floors* pada *sites* tersebut
- Menambahkan peta *floor layout* untuk setiap *floors*
- Membuat satu atau lebih *zones* pada setiap *floors*.
- Menempatkan *beacon* pada *floor layout*. *Beacon* dapat ditempatkan di luar atau di dalam *zones*.
- Mengonfigurasi detail setiap *beacon* yang telah ditempatkan pada *floor layout*.

Setelah langkah-langkah tersebut dilakukan, *sites* siap digunakan.

The screenshot displays the 'Sites' table in the IBM Presence Insights Management menu. The table has columns for Name, Total Visitors, New Visitors, and Average Dwell Time. The sites listed are South Area - ITB, Central Area - ITB, and North - ITB.

Name	Total Visitors	New Visitors	Average Dwell Time
South Area - ITB	0	0	00:00
Central Area - ITB	0	0	00:00
North - ITB	0	0	00:00

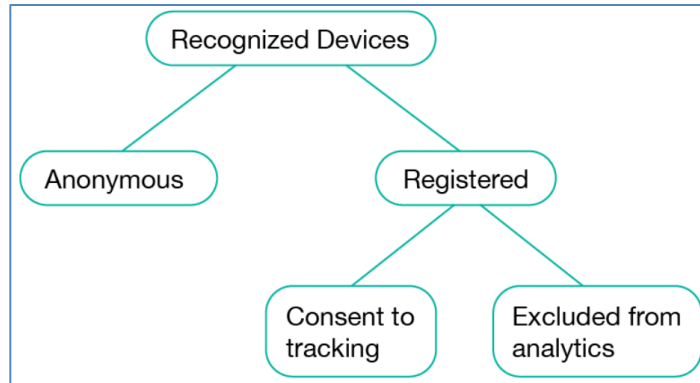
Gambar 19 Tampilan *sites* pada menu *management*. Pada gambar ini telah dibuat tiga *sites* contoh ©IBM Corporation.

6. Zones

Pada menu ini, *zones* yang telah dibuat dalam *sites*. Menu ini digunakan untuk kemudahan melihat data pada setiap *zones* yang telah dibuat.

7. Registered Devices

Menu ini menampilkan daftar perangkat yang dianggap sebagai perangkat yang terdaftar (*registered devices*) dan perangkat yang tidak terdaftar (*anonymous devices*).



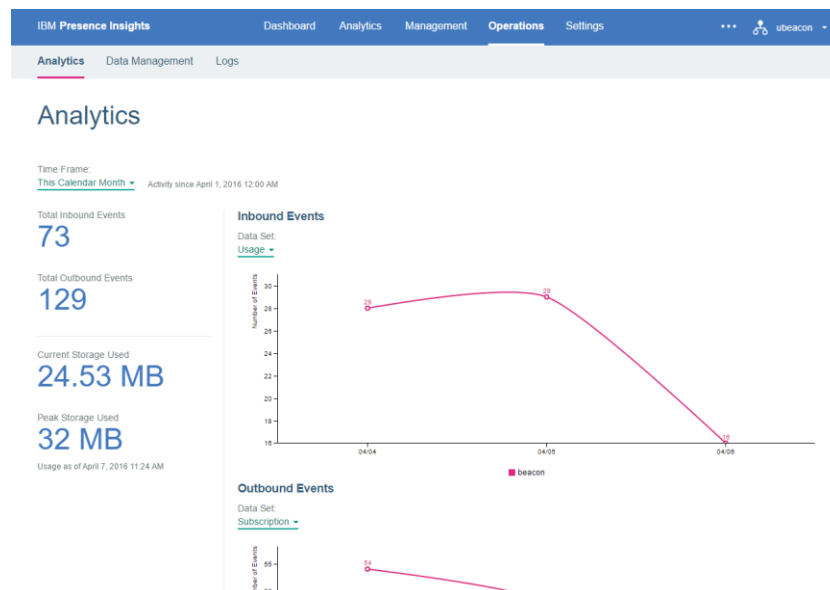
Gambar 20 Pemisahan yang dilakukan PI terhadap perangkat yang terdaftar dengan perangkat yang tidak terdaftar. ©IBM Corporation.

8. Subscription

Menu ini digunakan untuk mendaftarkan *back-end application* yang akan dikirimkan respons hasil olahan PI. Respons ini disebut juga sebagai *event*. Ketika *event* perangkat pengguna mendeteksi *beacon*, *back-end application* yang didaftarkan pada menu ini akan dikirimkan JSON berupa detail *event* tersebut.

9. Analytics

Menu ini digunakan untuk melihat aktivitas kiriman masuk dan kiriman keluar dari seluruh sistem PI.



Gambar 21 Tampilan *analytics* pada menu *operations*. Pada menu ini ditampilkan *traffic* data masuk dan data keluar pada PI ©IBM Corporation.

10. Data Management

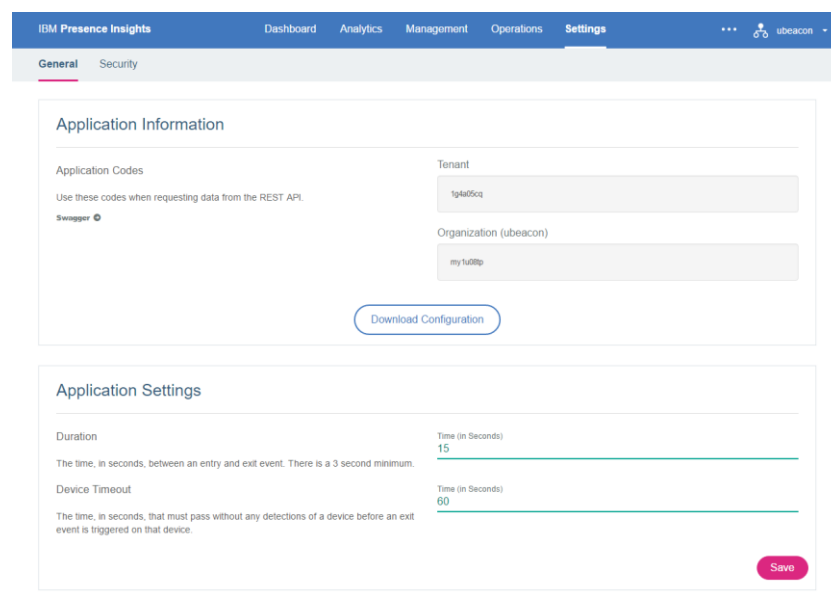
Menu ini digunakan untuk menghapus seluruh data yang masuk dan telah dianalisis oleh PI. Data yang dihapus tidak termasuk konfigurasi *beacon* yang telah dibuat. Data yang dihapus hanya data jumlah pengunjung saja.

11. *Logs*

Menu ini digunakan untuk melihat *error* yang mungkin terjadi pada PI.

12. *General*

Pada menu ini, terdapat pengaturan pembuatan *organization* dan *tenant*. Terdapat juga pengaturan *device timeout*. *Organization* dan *tenant* dibutuhkan untuk pembuatan segala kebutuhan konfigurasi *beacon*, *sites*, dan lain-lain untuk membedakan alamat pengiriman data ke PI dengan PI yang telah dibuat oleh pengguna IBM Bluemix lain. *Device timeout* merupakan pengaturan untuk menentukan waktu yang dibutuhkan untuk sebuah perangkat sudah dianggap tidak lagi di sekitar *beacon* tertentu.



Gambar 22 Hubungan antar perangkat dengan sistem. ©IBM Corporation.

13. *Security*

Pada menu ini, terdapat menu pembuatan *credentials* untuk autentikasi yang dibutuhkan untuk mengirimkan data ke PI.

14. *Documentation*

Pada menu ini, terdapat semua dokumentasi tentang IBM Presence Insights. Dokumentasi berisi cara penggunaan, glosarium, dan aplikasi umum.

15. *Swagger*

Swagger merupakan alat yang dapat digunakan untuk menyimulasikan semua fungsi yang ada pada PI. *Swagger* memuat semua daftar API yang ada pada PI. Pada setiap API yang ditampilkan, simulasi penggunaan API juga dapat dilakukan. *Swagger* merupakan alat yang sangat berguna pada saat pengembangan sistem untuk mencoba API yang akan digunakan pada aplikasi Android maupun *back-end application*.

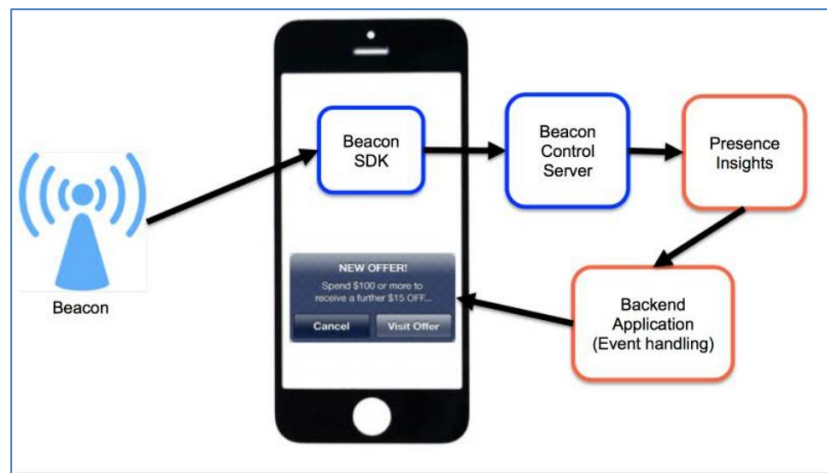
Agar *back-end application* dapat berkomunikasi dengan PI, API PI perlu digunakan untuk keperluan pertukaran data. Dari sekian banyak API yang terdapat di PI, salah

satu API yang utama digunakan untuk sistem yang menggunakan *beacon* sebagai sensor adalah *beacon connector*. API *beacon connector* tersebut adalah

<https://presenceinsights.au-syd.bluemix.net/conn-beacon/v1/tenants/{tenantCode}/orgs/{orgCode}>.

API ini digunakan untuk mengirimkan data *beacon* untuk diolah di PI menjadi data dan analisis yang akan disajikan.

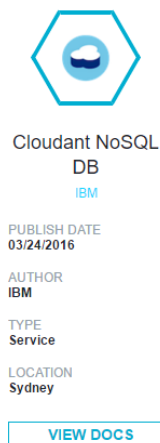
Untuk dapat menggunakan fungsi-fungsi yang ada pada PI, selain *back-end application* yang perlu dikonfigurasi, pada aplikasi Android perlu digunakan Presence Insights SDK. Presence Insights SDK digunakan untuk *beacon proximity positioning* menggunakan Bluetooth pada *smartphone* Android. Pada SDK ini, disediakan fungsi-fungsi untuk mendeteksi *beacon* menggunakan Bluetooth pada *smartphone* Android untuk memudahkan komunikasi dengan *back-end application* dan PI.



Gambar 23 Cara kerja Android SDK dan konfigurasi sistem. ©IBM Corporation.

Aplikasi Android yang dibuat dengan SDK ini berperan sebagai penghubung antara PI, *back-end application*, dan *smartphone* pengguna itu sendiri. Beberapa fungsi pada PI SDK yang akan digunakan pada aplikasi Android tersebut adalah fungsi deteksi *beacon* dan pengiriman data *beacon* sesuai dengan format yang dapat dimengerti PI sehingga sebuah *event* dapat dipicu pada PI. Fungsi penentuan *proximity* dari *beacon* juga digunakan yaitu penentuan *immediate*, *near*, dan *far* dari jarak *beacon* yang terdeteksi. Fungsi lainnya yang juga sangat penting adalah *eventing* yang dapat menentukan apakah perangkat masih berada di sekitar *beacon* atau sudah pindah ke sekitar *beacon* lain.

2.4.4 IBM Cloudant NoSQL



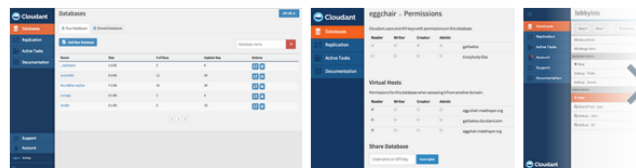
Cloudant NoSQL DB provides access to a fully managed NoSQL JSON data layer that's always on. This service is compatible with CouchDB, and accessible through a simple to use HTTP interface for mobile and web application models

• Ease of Use

Work with self-describing JSON documents through a RESTful API that makes every document in your Cloudant database accessible as JSON via a URL. Documents can be retrieved, stored, or deleted individually or in bulk and can also have files attached. IBM takes care of the provisioning, management, and scalability of the data store, freeing up your time to focus on your application.

• Powerful search, sync and more

With extremely powerful indexing, real time MapReduce and Apache Lucene-based full-text search, Cloudant NoSQL DB makes it easy to add advanced data analytics and powerful data access. Data access can also extend to Cloudant Sync, enabling data access from mobile devices and client apps to run connected or off-line.

The image is a screenshot of the IBM Cloudant NoSQL DB "Add Service" form. It includes fields for "Space" (Rizky Indra Syafrian), "App" (Leave unbound), "Service name" (Cloudant NoSQL DB-vo), and "Credential name" (Credentials-1). There is a "Selected Plan" dropdown set to "Shared" and a green "CREATE" button at the bottom.

Gambar 24 Tampilan Presence Insights service pada katalog IBM Bluemix. ©Eyro Digital Teknologi, Ltd.

IBM Cloudant NoSQL digunakan sebagai *database service* dari *back-end application* yang akan dibuat. Semua bentuk data yang perlu disimpan untuk digunakan kembali harus disimpan pada sebuah *database*. IBM Cloudant NoSQL merupakan layanan *database* yang sesuai karena sudah tergabung dalam paket layanan *cloud computing* yang disediakan oleh IBM Bluemix.

IBM Cloudant NoSQL tidak mengimplementasikan SQL pada *database*-nya. Cloudant menyimpan data pada *database* dalam bentuk dokumen-dokumen JSON yang terpisah. Kelebihan menyimpan data dalam bentuk dokumen-dokumen JSON adalah akses yang mudah untuk setiap *key-value pair* dari dokumen JSON karena pengambilan *database* langsung dalam satu paket dokumen.

Dalam sistem ini, *database* digunakan untuk menyimpan berbagai jenis data. Berikut ini adalah data-data yang disimpan dalam IBM Cloudant NoSQL *database*

1. User database

Database ini memuat semua data pengguna. Data pengguna berupa *username*, *password*, email, dan data-data lain yang dibutuhkan. Ketika pengguna melakukan *sign-in* ke sistem, maka *username* dan *password* yang diberikan pengguna akan dicocokkan ke *database*. Pengguna juga dapat mengganti data-datanya yang sudah diberikan kecuali *username*. *Username* tidak dapat diubah karena merupakan identitas utama data.

2. Beacon database

Database ini memuat semua UUID *beacon* yang aktif. UUID berfungsi sebagai identitas utama data *beacon*. Selain memuat UUID, *database beacon* juga memuat seluruh informasi *beacon* yang dibutuhkan seperti informasi acara, poster, lokasi, dan lain-lain. Untuk sistem absensi, beberapa data *beacon* juga memuat kode kuliah untuk memberitahukan *beacon* tersebut digunakan untuk kelas kuliah tertentu.

3. Class database

Database ini memuat seluruh kelas yang dapat dipantau kehadiran pesertanya. *Database* terpisah berdasarkan kode kuliah dan pesertanya. Kode kuliah menjadi

identitas utama *database* kelas. Pada setiap kelas, terdapat data lainnya seperti nama mata kuliah, dosen pengajar, dan daftar peserta kelas yang mengikuti kuliah. Kehadiran mahasiswa dan *smartphone*-nya menandai kehadirannya sebagai peserta kelas. Kehadiran peserta kelas dapat dipantau melalui *front-end* yang digunakan administrator.

Database dapat diakses oleh *back-end application* ataupun langsung via URL melalui REST API. Untuk mengakses *database* melalui *back-end application*, akses dilakukan dengan modul Nano pada Node.js yang berisi fungsi-fungsi untuk pengaksesan *database*. Untuk mengakses *database* langsung melalui URL maka digunakan REST API pada URL tersebut. Akses *database* melalui *back-end application* maupun URL memerlukan *credentials* agar *database* tidak dapat diakses oleh orang yang tidak memiliki autentikasi.

Pada rancangan sistem ini, *database* digunakan sebagai tempat penyimpanan data registrasi pengguna dan informasi *beacon* yang akan dikirimkan. Akses *database* yang digunakan adalah melalui *back-end application* dan tidak langsung melalui URL. Semua transaksi data dari *database* ke aplikasi Android dijembatani oleh *back-end application*. Skema seperti ini dianggap lebih aman dibandingkan jika aplikasi Android mengakses langsung *database*.