

**APLIKASI ANDROID UNTUK SISTEM KAMPUS PINTAR
DENGAN IBEACON – UBEACON**

TUGAS AKHIR

Oleh

ADIRGA IBRAHIM KHAIRY

NIM : 13212102

Program Studi Teknik Elektro



**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2016**

**APLIKASI ANDROID UNTUK SISTEM KAMPUS PINTAR
DENGAN IBEACON – UBEACON**

oleh :

Adirga Ibrahim Khairy

Tugas Akhir ini telah diterima dan disahkan
sebagai persyaratan untuk memperoleh gelar

SARJANA TEKNIK

di

**PROGRAM STUDI TEKNIK ELEKTRO
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG**

Bandung, 15 Juni 2016

Disetujui oleh :

Pembimbing I

Pembimbing II

Ir. Emir Mauludi Husni, M.Sc., Ph.D.

NIP. 19670707 2006041 016

Andi Sama

CIO PT Sinergi Wahana Gemilang

ABSTRAK

APLIKASI ANDROID UNTUK SISTEM KAMPUS PINTAR DENGAN IBEACON – UBEACON

Oleh

Adirga Ibrahim Khairy

NIM : 13212102

PROGRAM STUDI TEKNIK ELEKTRO

Tugas akhir ini menjelaskan tentang perancangan fungsi aplikasi uBeacon yang terintegrasi dengan sistem informasi di lingkungan kampus dengan iBeacon. Aplikasi uBeacon adalah aplikasi berbasis android yang berfungsi untuk menerima informasi dari server setiap kali *smartphone* pengguna mendeteksi keberadaan beacon. Selain itu, aplikasi uBeacon juga berfungsi untuk mengelompokkan informasi berdasarkan kategori, melakukan fitur presensi kelas perkuliahan, serta memberikan petunjuk keberadaan mahasiswa di lingkungan kampus. Aplikasi ini akan terintegrasi dengan server kampus. Protokol komunikasi yang digunakan pada aplikasi ini adalah *Message Queuing Telemetry Transport (MQTT)* untuk pertukaran informasi antara aplikasi dengan server. Dalam pengujian, aplikasi uBeacon telah berjalan dan terintegrasi dengan baik.

Kata Kunci: android, beacon, *bluetooth low energy*, dan MQTT.

ABSTRACT

ANDROID APPLICATION FOR SMART CAMPUS INFORMATION SYSTEM WITH IBEACON – uBeacon

By

Adirga Ibrahim Khairy

NIM : 13212102

ELECTRICAL ENGINEERING STUDY PROGRAM

This final project describes about the functional design of uBeacon application which is integrated with campus information system with iBeacon. This application, uBeacon, is an android based application which is used to receive information from the server whenever the beacon signal is detected. Moreover, this application is used to categorize each information by its topic categories, mark user presence in the classroom, and to navigate user position inside the campus. This application is integrated with campus server. The communication protocol used in this application is Message Queuing Telemetry Transport (MQTT) for information transaction between uBeacon application and the server. This application has been successfully tested and integrated properly.

Keywords: android, beacon, *bluetooth low energy*, dan MQTT.

PRAKATA

Puji syukur penulis panjatkan ke hadirat Allah SWT, yang atas rahmat dan karunia Nya penulis dapat menyelesaikan tugas akhir yang berjudul “*Aplikasi Android untuk Sistem Kampus Pintar dengan iBeacon – uBeacon*” tepat pada waktunya. Shalawat dan salam tercurah kepada Rasulullah Muhammad SAW beserta keluarganya.

Buku ini adalah laporan pengerjaan tugas akhir yang dilakukan pada tahun ajaran 2015/2016. Buku tugas akhir ini dibuat sebagai upaya pemenuhan syarat kelulusan program sarjana Program Studi Teknik Elektro di Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung.

Selama melaksanakan tugas akhir ini, penulis mendapat bantuan dan dukungan dari berbagai pihak. Untuk itu, penulis ingin mengucapkan terima kasih kepada :

1. Bapak, Ibu, Kakak, Adik, dan keluarga besar yang telah memberikan dukungan kepada penulis selama menuntut ilmu di ITB.
2. Bapak Ir. Emir Mauludi Husni, M.Sc, Ph.D. selaku dosen pembimbing tugas akhir penulis atas bimbingan dan nasihat nya kepada penulis.
3. Bapak Arif Sasongko, ST, M.Sc, Ph.D. selaku ketua program studi Teknik elektro ITB beserta Tim Tugas Akhir Teknik Elektro ITB yang telah memberikan arahan selama pengerjaan tugas akhir.
4. Bapak Andi Sama selaku *Chief Information Officer* dari PT Sinergi Wahana Gemilang yang turut bekerjasama dalam pengerjaan tugas akhir.
5. Rizky Indra Syafrian dan Astari Purnomo selaku rekan satu tim pengerjaan tugas akhir yang telah bersedia bekerjasama dengan penulis selama kegiatan tugas akhir dilaksanakan.
6. Gilang Julian dan Ahmad Shahab yang telah membantu dan memberikan saran kepada penulis dalam mengerjakan tugas akhir.
7. Seluruh teman di Ruang Riset Mandiri yaitu Riksa, Syaiful Andy, Hilmy, Meynard, Adil, Afdal, Adelwin, Irena, Vivi, Andaswara, Arfie, Nadia, Zuhdi, Alfi, dan Aris yang telah bersama-sama melaksanakan kegiatan tugas akhir selama ini.

8. Teman-teman di Ruang Tugas Akhir lainnya yang saling mendukung dan membantu satu sama lain dalam berbagai bentuk.
9. Teman-teman di Teknik Elektro ITB angkatan 2012 yang telah menempuh kegiatan perkuliahan bersama-sama hingga saat ini.
10. Teman-teman di HME ITB yang telah membantu penulis dalam menuntut ilmu, mendapatkan pengalaman, bermain, dan berorganisasi.
11. Sheila Amalia Saleh yang terus memberikan dukungan dan semangat selama pengerjaan tugas akhir.
12. Seluruh pihak yang telah membantu penulis menyelesaikan tugas akhir namun tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa tugas akhir ini bukanlah tanpa kelemahan dan tanpa cacat. Oleh karena itu, penulis sangat mengharapkan kritik dan saran. Semoga laporan dan tugas akhir ini dapat bermanfaat bagi para pembaca dan bagi perkembangan ilmu pengetahuan dan teknologi.

Bandung, 14 Juni 2016

Penulis

DAFTAR ISI

ABSTRAK	3
ABSTRACT	4
PRAKATA	5
DAFTAR ISI	7
DAFTAR GAMBAR	8
BAB I PENDAHULUAN	9
1.1 Latar Belakang	9
1.2 Rumusan Masalah	9
1.3 Tujuan Tugas Akhir	10
1.4 Lingkup Permasalahan	10
1.5 Metodologi	11
1.6 Sistematika Penulisan	11
BAB II TINJAUAN PUSTAKA	13
2.1 Android Operating System (OS)	13
2.2 Message Queuing Telemetry Transport (MQTT)	13
2.3 Beacon	14
2.4 Estimote SDK	15
BAB III DESAIN DAN IMPLEMENTASI	16
3.1 Aplikasi Sistem Informasi untuk Kampus Pintar	16
3.2 Spesifikasi	18
3.3 Desain	18
3.4 Perancangan Graphical User Interface (GUI)	20
3.5 Implementasi dan Aplikasi	21
BAB IV PEMBAHASAN	26
4.1 Aspek dan Skenario Pengujian	26
4.2 Prosedur dan Hasil Pengujian	27
BAB V PENUTUP	33
5.1 Kesimpulan	33
5.2 Saran	33
DAFTAR PUSTAKA	34

DAFTAR GAMBAR

Gambar 1 Ilustrasi kerja protokol MQTT	14
Gambar 2 Rancangan Umum Sistem	16
Gambar 3 Skema aliran informasi.....	17
Gambar 4 Rancangan tampilan utama aplikasi uBeacon	20
Gambar 5 Rancangan tampilan <i>My Location</i>	20
Gambar 6 Rancangan tampilan <i>Attend Class</i>	21
Gambar 7 Diagram alir Aliran Informasi pada uBeacon	21
Gambar 8 Diagram alir pengisian informasi ListView	23
Gambar 9 Diagram alir fitur Categories.....	23
Gambar 20 Diagram alir fitur Attend Class	24
Gambar 11 Diagram alir fitur My Location.....	25
Gambar 12 <i>logcat</i> pada Android Studio	27
Gambar 13 Notifikasi pada <i>smartphone</i> yang mendeteksi Beacon	28
Gambar 14 Tampilan utama aplikasi uBeacon	28
Gambar 15 Tampilan detil Beacon Information	29
Gambar 16 Tampilan pada <i>Navigation Drawer</i>	29
Gambar 37 Tampilan pada menu kategori	30
Gambar 18 Tampilan Categories	30
Gambar 19 Tampilan pada <i>Attend Class</i>	31
Gambar 20 Tampilan akhir <i>Attend Class</i>	31
Gambar 21 Tampilan pada fitur My Location	32

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada lingkungan kampus, merupakan hal penting bagi mahasiswa untuk menggali berbagai informasi tentang pengumuman, berita kampus, lokasi gedung, serta berbagai kegiatan kampus yang dapat menunjang kegiatan belajar mahasiswa, khususnya bagi mahasiswa baru. Permasalahan yang sering terjadi adalah kesulitan bagi mahasiswa untuk mendapatkan akses informasi yang lengkap dan terbaru sehingga mahasiswa dapat kehilangan kesempatan untuk mengikuti suatu kegiatan ataupun mendapatkan berita penting.

Untuk dapat memenuhi kebutuhan mahasiswa akan informasi, dibutuhkan sebuah media yang tepat sehingga persebaran informasi akan menjadi lebih efektif. Seiring dengan perkembangan teknologi, penggunaan *smartphone* di Indonesia terus meningkat secara signifikan setiap tahun nya. Oleh karena itu, persebaran informasi melalui media *smartphone* dapat menjadi solusi yang tepat bagi permasalahan ini.

Dalam makalah ini akan dipaparkan sebuah solusi dari permasalahan tersebut, yaitu dengan membuat sebuah sistem kampus pintar dengan ibeacon yang dapat menyebarkan informasi secara kontinu dan bersifat interaktif kepada pengguna. Pengguna dapat berinteraksi melalui sebuah aplikasi uBeacon yang terhubung dengan sistem untuk mendapatkan informasi berdasarkan lokasi, menerima berita kampus, mengetahui keberadaan pengguna, serta dapat melakukan sistem presensi digital yang terintegrasi dengan sistem basis data kampus.

1.2 Rumusan Masalah

Berdasarkan pada latar belakang yang telah dijabarkan pada bagian sebelumnya, maka rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

- Bagaimana cara menciptakan sistem yang efektif dalam persebaran informasi di lingkungan kampus?

- Bagaimana cara menciptakan media yang nyaman dan *user friendly* bagi pengguna?
- Bagaimana cara pengguna dapat berinteraksi dengan sistem informasi di kampus?

1.3 Tujuan Tugas Akhir

Tujuan umum dilakukannya tugas akhir ini adalah sebagai berikut:

1. Pemenuhan salah satu syarat kelulusan pada program sarjana program studi Teknik Elektro. Sekolah Teknik Elektro ITB.
2. Riset untuk pengembangan sistem informasi kampus pintar.
3. Riset untuk implementasi dan aplikasi Internet of Things pada dunia kampus

Tujuan khusus dilakukannya tugas akhir ini adalah sebagai berikut:

1. Menciptakan sistem informasi untuk kampus pintar dengan iBeacon yang dapat menghubungkan mahasiswa dengan berbagai berita kampus, kegiatan kampus, dan sistem presensi kelas.
2. Merancang bentuk komunikasi antara aplikasi android dan penggunaan perangkat beacon yang terintegrasi pada sebuah lingkungan.

1.4 Lingkup Permasalahan

Tugas akhir ini dibuat dengan asumsi dan batasan sebagai berikut.

- Aplikasi hanya dapat dijalankan dengan menggunakan *smartphone* android dengan versi Android 4.2 Jelly Bean ke atas yang memiliki fitur Bluetooth 4.0 (*Bluetooth Low Energy*)
- Aplikasi hanya dapat dijalankan ketika fitur bluetooth dan koneksi internet *smartphone* pengguna dalam keadaan aktif.
- Sistem informasi uBeacon dapat diimplementasikan pada kampus yang telah dipasangkan perangkat Beacon yang memiliki suplai daya berupa baterai dan aktif dan telah di program.
- Sistem informasi uBeacon dapat dijalankan jika informasi telah diatur dalam sistem backend application yang merupakan salah satu bagian dari sistem.

1.5 Metodologi

Seluruh data yang dipaparkan pada buku tugas akhir ini merupakan seluruh data kegiatan selama pengerjaan tugas akhir. Metoda yang digunakan untuk menyelesaikan tugas akhir ini adalah sebagai berikut:

1. Penentuan Topik

Penentuan topik dilakukan dengan memilih topik yang telah ditentukan oleh tim Tugas Akhir berdasarkan pilihan penulis.

2. Penentuan Spesifikasi

Tahap menentukan spesifikasi dari keluaran yang diharapkan.

3. Studi Literatur

Mempelajari desain dari sistem dan algoritma yang sudah ada sebagai dasar perancangan untuk memenuhi spesifikasi yang ingin dicapai.

4. Perancangan dan Implementasi Sistem

Aplikasi yang akan dibuat dirancang berdasarkan masukan, kebutuhan untuk melakukan proses, dan keluaran yang diharapkan. Kemudian aplikasi dibuat dengan menggunakan *environment* pemrograman yang sesuai.

5. Pengujian dan Perbaikan Sistem

Pengujian sistem dilakukan untuk melihat performansi dan kinerja sistem dan melakukan perbaikan untuk *failure/bug* yang dihasilkan.

6. Penarikan Analisis dan Kesimpulan

Berhasil atau tidaknya hasil pengujian sistem dianalisis dan ditarik kesimpulannya.

1.6 Sistematika Penulisan

Buku tugas akhir ini disusun menjadi 5 bab dengan sistematika penulisan sebagai berikut:

1. BAB I PENDAHULUAN

Tahap ini mencakup latar belakang, rumusan masalah, tujuan tugas akhir, lingkup permasalahan, metodologi yang digunakan dalam pengerjaan tugas akhir, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Tahap ini mencakup teori yang mendukung pengerjaan tugas akhir ini.

3. BAB III DESAIN DAN IMPLEMENTASI

Tahap ini mencakup implementasi, mulai dari spesifikasi sistem, desain, hingga implementasi hingga menjadi aplikasi yang dapat digunakan.

4. BAB IV HASIL PENGUJIAN

Tahap ini mencakup hasil pengujian aplikasi yang dibuat.

5. BAB V PENUTUP

Tahap ini berisikan kesimpulan dari pengerjaan tugas akhir ini dan saran untuk para pembaca maupun pengembang topik untuk tahap selanjutnya.

BAB II

TINJAUAN PUSTAKA

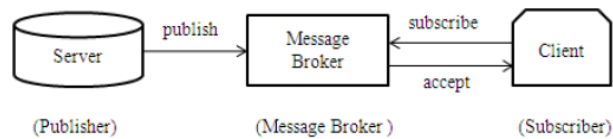
2.1 Andorid Operating System (OS)

Android adalah sistem operasi yang dikembangkan oleh Android Inc. dan dipasarkan oleh Google. Sistem operasi ini berbasis linux dan dirancang untuk perangkat layar sentuh seperti *smartphone* dan tablet. Sistem operasi android bersifat terbuka (*open source*) yang memudahkan pengembang aplikasi untuk membuat aplikasi nya sendiri. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi yang memperluas fungsionalitas dari perangkat, umumnya ditulis dalam bahasa pemograman Java.

Untuk dapat membuat aplikasi baru pada Android, umumnya *tools* yang digunakan adalah *Android Software Development Kit* (SDK). Di dalam Android SDK terdapat seperangkat *tools* seperti *debugger*, *library*, emulator, dokumentasi, contoh kode, dan tutorial. Pengembangan Android SDK sejalan dengan pengembangan *platform* Android secara keseluruhan. SDK ini mendukung *platform* Android dengan versi yang lebih lama jika *developer* ingin mengembangkan aplikasinya pada perangkat yang lama. Aplikasi Android dikemas dalam format *.apk* yang berisi *file .dex* (*Dalvik executables*), *file resources*, dan lain-lain.

2.2 Message Queuing Telemetry Transport (MQTT)

Message Queuing Telemetry Transport (MQTT) merupakan protokol komunikasi dengan sistem *publish* dan *subscribe* yang didesain dengan sistem pengiriman yang sangat ringan (*lightweight*) dan sederhana sehingga sangat ideal untuk digunakan bagi *smartphone*. Selain itu, MQTT memiliki konsumsi daya yang rendah. Implementasi MQTT pada *mobile application* sangat dimungkinkan untuk koneksi *Internet of Things* (IoT). Pola transmisi data menggunakan *publish* dan *subscribe* membutuhkan sebuah Broker. Broker bertanggung jawab untuk mendistribusikan data kepada *client* yang tertarik, sesuai dengan *topic* yang digunakan. Berikut ilustrasi kerja protokol MQTT:



Gambar 1 Ilustrasi kerja protokol MQTT

Pada aplikasi uBeacon, MQTT digunakan untuk melakukan pengiriman dan penerimaan paket data melalui koneksi internet. Paket data yang dikirim antara aplikasi uBeacon dengan server berupa file JSONObject dan JSONArray yang disusun sesuai ketentuan pada server. Data yang diterima dari MQTT seluruhnya digunakan untuk menghasilkan tampilan informasi uBeacon yang sesuai.

2.3 Beacon

Beacon adalah sebuah perangkat bluetooth yang digunakan untuk melakukan transmisi data dalam jangkauan maksimal 70-80 m. Beacon mengimplementasikan teknologi Bluetooth Low Energy yang dengan konsumsi daya yang rendah. Dalam aplikasinya, beacon dapat digunakan sebagai perangkat untuk melakukan pengiriman informasi yang bersifat satu arah kepada devais penerima.

Beacon dapat mengirimkan paket data dalam jangkauan tertentu yang terdiri dari 4 atribut. Paket data tersebut antara lain terdiri dari UUID, Major, Minor, dan Tx Power. Setiap atribut memiliki peran masing-masing yang membedakan antara perangkat beacon yang satu dengan lainnya. Adapun, atribut major dan minor dari setiap beacon dapat diatur sesuai dengan pengaturan penggunaan yang dibutuhkan.

Dalam perancangan sistem kampus pintar, beacon merupakan perangkat yang tepat karena memiliki beberapa keuntungan seperti instalasi yang mudah, paket transmisi data yang ringan, jangkauan yang dapat di atur (*configurable*) untuk implementasi di dalam ruang kelas, akurasi yang tinggi, serta cocok untuk penggunaan pada *outdoor-positioning system* dimana beacon dapat menjangkau area hingga 80m. Jika dibandingkan dengan perangkat teknologi berbasis lokasi lainnya, keunggulan beacon ditunjukkan oleh tabel berikut, [2]:

Tabel 1 Perbandingan antar perangkat berbasis lokasi

Kriteria	Beacon	GPS	Wifi	RFID
Jangkauan	1-80 m (configurable)	(Tidak Terbatas)	20-50 m	1-100 cm
Akurasi(1-5)	4	1	3	5
Instalasi	Mudah	-	Rumit	Mudah

2.4 Estimote SDK

Estimote SDK merupakan sebuah kit pengembangan standar (*standard development kit*) yang memungkinkan *smartphone* untuk mengukur jarak, mengukur lokasi dan objek, serta mengenal atribut beacon. Estimote SDK dapat mempermudah pengguna dalam mengembangkan interaksi antara aplikasi *smartphone* dengan sebuah sistem beacon.

Pada aplikasi ini digunakan library Estimote karena Estimote Beacon merupakan salah satu merk Beacon yang paling berkembang dan telah menyediakan berbagai fitur yang dibutuhkan oleh aplikasi uBeacon. Penggunaan Estimote SDK digunakan dalam aplikasi pada fitur *beacon monitoring* untuk melakukan pemindaian beacon pada aktivitas di belakang layar (*background activity*).

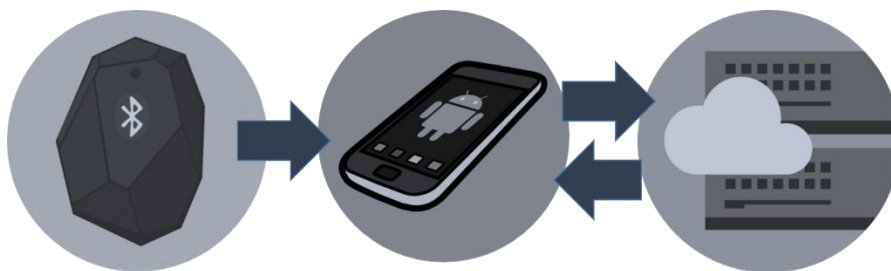
BAB III

DESAIN DAN IMPLEMENTASI

3.1 Aplikasi Sistem Informasi untuk Kampus Pintar

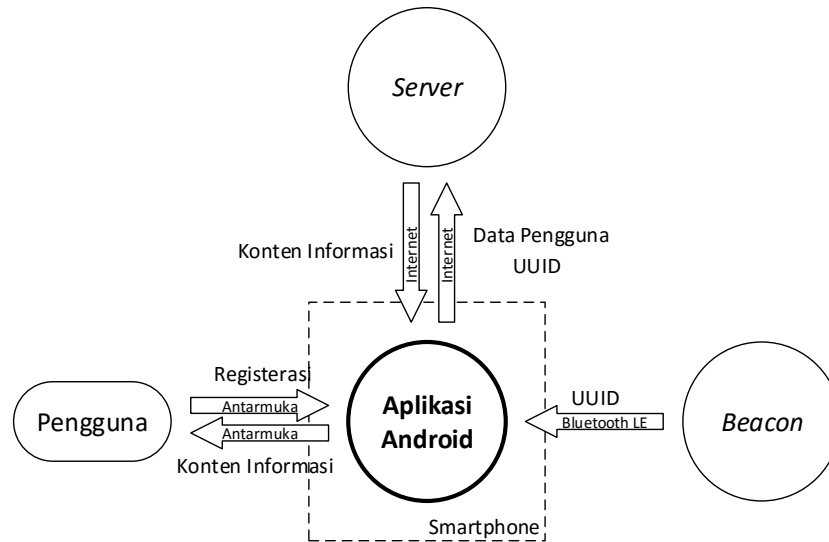
Aplikasi uBeacon memiliki arti *University Beacon*. Aplikasi ini dirancang sebagai media bagi pengguna untuk dapat berinteraksi dengan sistem informasi kampus melalui beacon dan server IBM Bluemix. Aplikasi ini dapat memberikan informasi kepada pengguna tentang berbagai acara di kampus, berita kampus, peta kampus, serta melakukan presensi kelas, khususnya bagi mahasiswa. Dengan keberadaan aplikasi ini, pengguna dapat menerima informasi yang terbaru secara aktif langsung dari *smartphone* pengguna.

Sistem uBeacon di lingkungan kampus terdiri dari tiga buah komponen utama, yaitu beacon, *smartphone*, dan server. Ketiga komponen ini saling berinteraksi untuk membentuk sebuah sistem kampus pintar. Beacon akan ditempatkan pada lokasi *outdoor-positioning*, *indoor-positioning* di dalam gedung, dan di dalam setiap kelas perkuliahan sebagai pemicu kejadian (*events trigger*) pada *smartphone* untuk melakukan suatu aksi tertentu. Beacon akan melakukan komunikasi satu arah dengan mentransmisikan signal bluetooth, sedangkan *smartphone* akan berperan dalam menerima signal tersebut dan melakukan komunikasi dua arah dengan server (*cloud computing server*). Berikut skema dari rancangan umum sistem:



Gambar 2 Rancangan Umum Sistem

Pada sistem ini akan terjadi aliran informasi setiap kali pengguna smartphone mendeteksi keberadaan beacon di lingkungan kampus. Gambar berikut menunjukkan skema aliran informasi yang terjadi pada sistem:



Gambar 3 Skema aliran informasi

Skenario sistem yang terjadi adalah sebagai berikut:

1. Beacon akan melakukan pemancaran signal Bluetooth Low Energy sepanjang waktu selama baterai dalam kondisi aktif. Beacon akan memancarkan empat buah atribut *identifier* yang terdiri dari UUID, Major, Minor, dan Tx Power.
2. Aplikasi uBeacon melakukan *beacon monitoring* dengan interval tertentu pada aktivitas latar belakang(*background activity*) untuk mendapatkan keempat atribut identifier tersebut.
3. Ketika beacon terdeteksi, aplikasi akan membentuk suatu paket JSON file yang tersusun oleh atribut identifier beacon yang terdeteksi dan beberapa atribut data diri pengguna. Selanjutnya, JSON file ini akan di *publish* sebagai *payload* melalui koneksi MQTT kepada server.
4. Server akan melakukan pengolahan data sesuai dengan informasi yang diatur dalam basis data server. Selanjutnya, server akan mempersiapkan informasi yang akan dikembalikan kepada aplikasi uBeacon.
5. Aplikasi uBeacon melakukan *subscribe* via MQTT untuk mendapatkan

informasi balikan dari server yang berupa JSON file.

6. Aplikasi uBeacon melakukan penguraian data JSON file yang diterima dan akan ditempatkan pada variabel-variabel tertentu.
7. Kumpulan variabel tersebut akan ditempatkan menjadi sebuah kesatuan Java Object pada *constructor* di kelas *Events.java*.
8. Aliran informasi berhasil dilakukan dan disimpan pada aplikasi untuk selanjutnya ditampilkan ketika informasi tersebut dipanggil.

3.2 Spesifikasi

Aplikasi uBeacon harus dapat memenuhi spesifikasi produk sebagai berikut:

1. Aplikasi dapat dijalankan pada *smartphone* berbasis Android dengan versi minimum 4.2 *Jelly Bean* ke atas yang telah mendukung fitur Bluetooth 4.0 (Bluetooth Low Energy), bluetooth dalam keadaan aktif, serta koneksi internet yang aktif untuk melakukan interaksi dengan server IBM Bluemix.

Selain itu, aplikasi uBeacon juga memiliki spesifikasi fungsi sebagai berikut:

1. Aplikasi dapat menyediakan informasi tentang berita kampus, berita kegiatan unit kegiatan mahasiswa, seminar, dan berbagai acara lain nya.
2. Aplikasi dapat digunakan untuk memberitahukan posisi pengguna di dalam kampus.
3. Aplikasi dapat digunakan untuk melakukan konfirmasi presensi kelas kuliah.
4. Aplikasi mudah untuk digunakan (*user friendly*).

3.3 Desain

Aplikasi uBeacon dibuat untuk *smartphone* berbasis Android. Aplikasi ini dirancang dengan menggunakan *software* Android Studio dengan menggunakan bahasa pemrograman Java. Aplikasi digunakan sebagai media bagi pengguna untuk berinteraksi dengan sistem informasi di lingkungan kampus.

Dalam sistem ini, aplikasi uBeacon akan berinteraksi dengan perangkat beacon yang ditempatkan pada lokasi di lingkungan kampus. Sistem ini dirancang dengan menempatkan beacon pada lingkungan kampus Institut Teknologi Bandung (ITB)

dengan jumlah 19 buah beacon pada *outdoor-positioning system* yang dapat menjangkau seluruh area kampus, dan 280 buah beacon pada *indoor-positioning system* yang diletakkan di dalam kelas perkuliahan dan gedung di kampus. Seluruh beacon di desain untuk menyediakan informasi yang berbeda-beda pada server.

Perangkat *smartphone* yang menjalankan aplikasi uBeacon akan berinteraksi dengan beacon yang mentransmisikan sinyal BLE yang akan dideteksi oleh *smartphone*. Selanjutnya, *smartphone* akan melakukan interaksi dengan server IBM Bluemix melalui komunikasi MQTT. Dengan terjadinya interaksi tersebut, pengguna dapat menerima informasi yang sesuai dengan keberadaan nya. Hal tersebut akan tercapai jika pengguna memiliki akses internet dan koneksi bluetooth (Bluetooth Low Energy) dalam keadaan aktif.

Selain itu, aplikasi ini dirancang untuk dapat memberitahukan pengguna akan keberadaan nya di dalam kampus. Pengguna akan ditunjukkan sebuah gambar yang menunjukkan posisi nya berdasarkan beacon yang terdeteksi

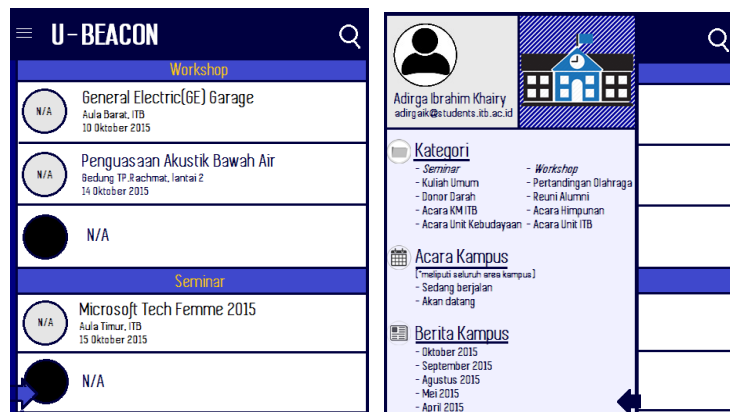
Pengguna yang tergolong sebagai mahasiswa juga memiliki fitur untuk melakukan presensi kelas kuliah dengan menggunakan aplikasi uBeacon yang telah terintegrasi dengan sistem basis data kampus. Dengan begitu, pihak kampus dapat menjamin keberadaan mahasiswa nya di dalam kelas, serta dapat meminimalisir penggunaan kertas sebagai lembar presensi mahasiswa.

Lingkungan (*environment*) dalam pemrograman dan implementasi aplikasi ini adalah:

1. Samsung Galaxy Note 2 untuk implementasi.
2. Notebook Asus K401L dengan processor Intel® Core i5 sebagai computer dalam pengembangan aplikasi.
3. Android Studio dan Android SDK untuk mengembangkan aplikasi dengan berbagai *tools* di dalamnya untuk menghasilkan file dengan format *Android Application Package* (APK).

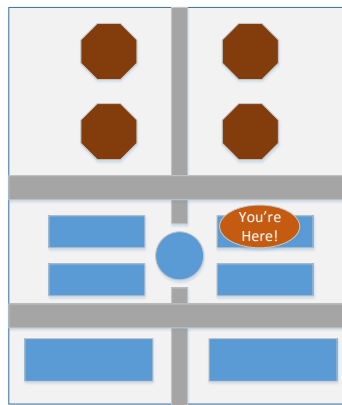
3.4 Perancangan Graphical User Interface (GUI)

Graphical User Interface dirancang agar pengguna dapat melihat dan menerima informasi yang sedang atau akan berlangsung ketika pengguna mendeteksi beacon secara nyaman. Adapun desain tampilan nya diatur seperti sebagai berikut:



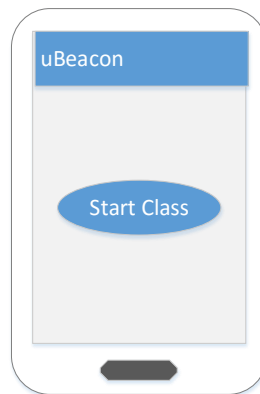
Gambar 4 Rancangan tampilan utama aplikasi uBeacon

Selanjutnya, pada fitur My Location dirancang sebuah gambar yang diterima berupa peta yang menunjukkan keberadaan pengguna. Keberadaan pengguna ditunjukkan dengan tulisan “You’re here!” seperti yang ditunjukkan gambar berikut :



Gambar 5 Rancangan tampilan *My Location*

Sedangkan, untuk fitur *Attend Class* dirancang tampilan yang seperti sebagai berikut :

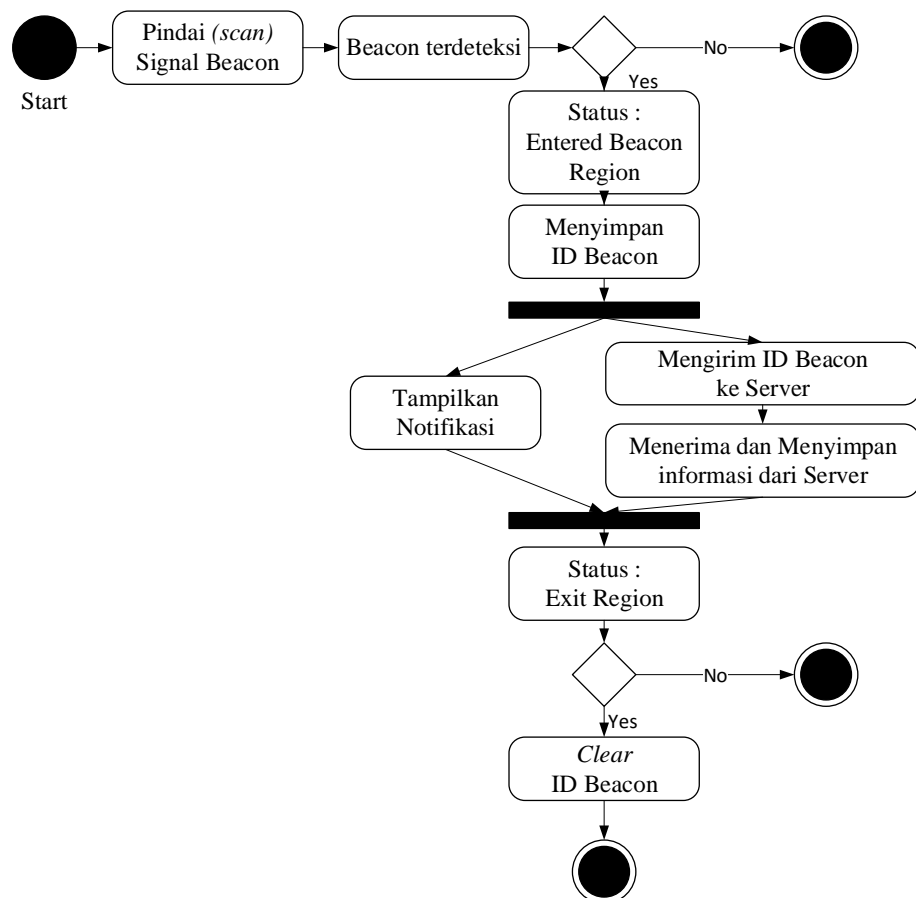


Gambar 6 Rancangan tampilan *Attend Class*

3.5 Implementasi dan Aplikasi

3.5.1 Aliran Informasi Aplikasi uBeacon

Pada proses ini dilakukan proses *Beacon Monitoring* yang terjadi pada proses *background activity* pada *smartphone* Android. Berikut proses yang terjadi pada aliran informasi uBeacon :



Gambar 7 Diagram alir(Activity Diagram) Aliran Informasi pada uBeacon

Pada proses ini, ketika keberadaan beacon terdeteksi, informasi akan didapatkan dari internet melalui koneksi MQTT pada smartphone yang melibatkan proses *publish* saat mengirim umpan kepada server dan proses *subscribe* untuk mendapatkan hasil dari server. Kedua proses tersebut melibatkan JSON file yang dikirim dan didapatkan dengan format tertentu.

JSON file untuk melakukan proses *publish* dibentuk dengan format sebagai berikut.

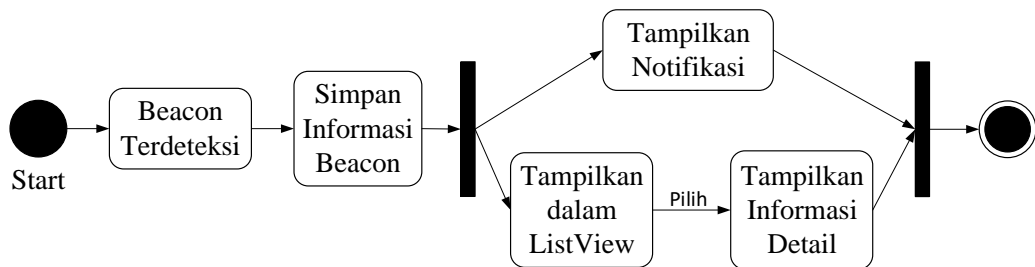
```
{ "bnm":
  [ { "descriptor": username,
    "detectedTime": 1465300800000,
    "data": {
      "proximityUUID": "cb10023f-a318-3394-4199-a8730c7c1aec",
      "major": "1",
      "minor": "284",
      "rssi": rssi,
      "accuracy": acc,
      "proximity": prox
    }
  }
]
```

Sedangkan, JSON file untuk proses *subscribe* dibentuk dengan format sebagai berikut:

```
{
  "_id": "cb10023f-a318-3394-4199-a8730c7c1aec-4-284",
  "_rev": "11-d08e34cea4be57ef8e60141ef365ad12",
  "proximityUUID": "cb10023f-a318-3394-4199-a8730c7c1aec",
  "major": "4",
  "minor": "284",
  ...
  "data": [
    {
      "category": "Pameran",
      ...
      "contact": "Vicky (Teknik Elektro 2014)"
    }
  ]
}
```

3.5.2 Fitur *Beacon Information*

Pada fitur ini ditampilkan menu utama dari aplikasi uBeacon. Setelah informasi didapatkan pada proses *Beacon Monitoring*, informasi akan ditampilkan pada fitur ini. Tampilan utama *Beacon Information* menampilkan sekelompok informasi dalam bentuk barisan dengan menggunakan ListView. Proses yang terjadi dijelaskan dalam diagram alir berikut:

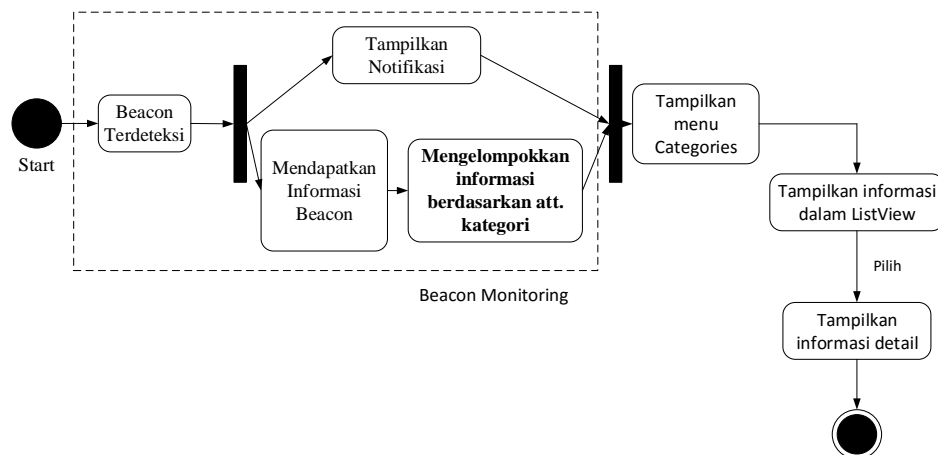


Gambar 8 Diagram alir(Activity Diagram) pengisian informasi ListView

Dalam fitur ini akan ditampilkan informasi detail ketika pengguna memilih salah satu informasi yang ditampilkan pada baris ListView.

3.5.3 Fitur *Category*

Category digunakan untuk mengelompokkan berbagai informasi yang didapatkan dari beacon berdasarkan atribut kategori informasi yang terdaftar pada bagian basis data pada server. Dalam hal ini terdapat 8 kategori yang terdiri dari *Seminar*, *Campus Info*, *Promotion*, *Unit & Himpunan Events*, *Workshop*, *Concert*, *Sports*, dan *Events*. Fitur ini berguna untuk membuat konten lebih terorganisir sesuai dengan minat (*interest*) pengguna aplikasi uBeacon.

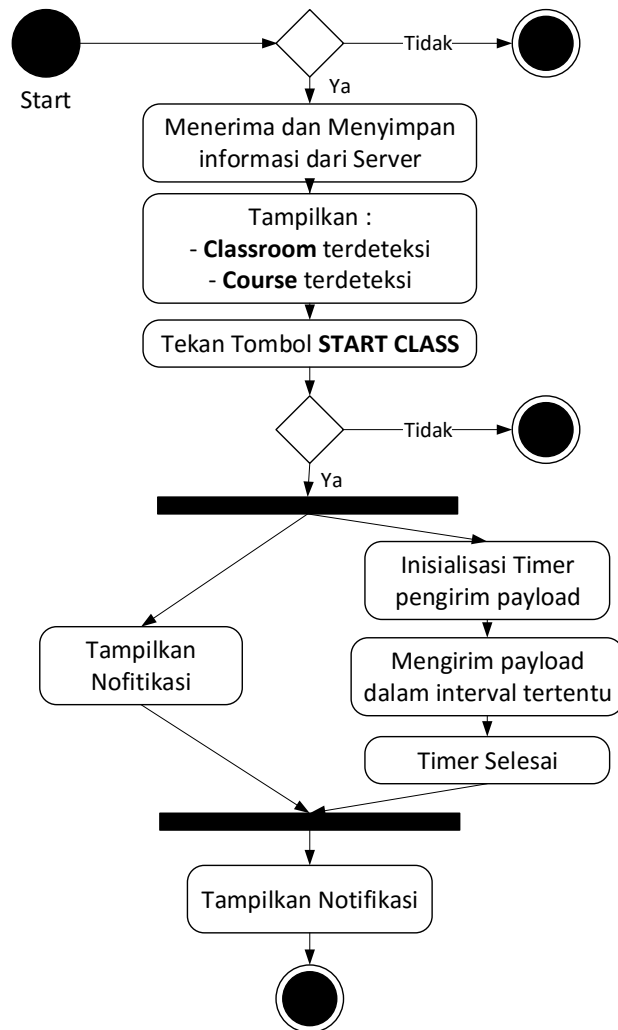


Gambar 9 Diagram alir(Activity Diagram) fitur *Categories*

3.5.3 Fitur *Attend Class*

Attend Class digunakan untuk melakukan sistem presensi kelas secara digital dan terintegrasi dengan sistem. Dalam fitur ini, pengguna hanya perlu menekan sebuah tombol *Start Class*, lalu sistem akan mengirim payload yang berisikan *identifier* beacon yang terdeteksi dalam jangka waktu(*interval*) tertentu untuk memastikan keberadaan pengguna di dalam kelas. Cara kerja fitur ini dijelaskan dalam diagram

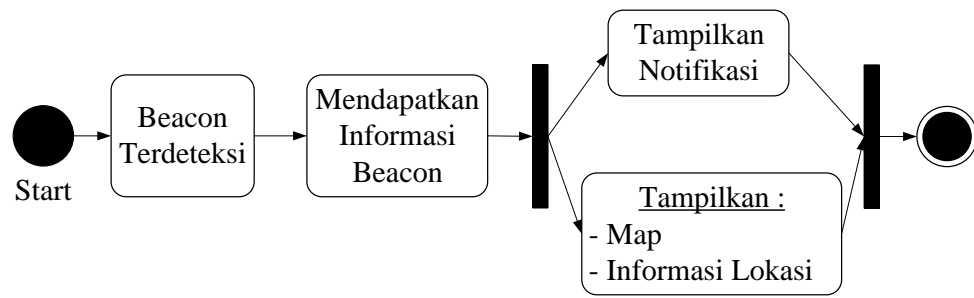
alir berikut:



Gambar 10 Diagram alir(activity diagram) fitur *Attend Class*

3.5.4 Fitur *My Location*

My Location digunakan untuk melakukan navigasi keberadaan pengguna pada lingkungan kampus. Fitur ini memanfaatkan keberadaan beacon yang diletakkan pada sudut-sudut kampus sehingga setiap beacon memiliki url gambar tertentu yang menunjukkan posisi nya di dalam kampus. Tampilan keberadaan pengguna akan ditunjukkan dengan petunjuk “You are here” pada gambar peta yang ditampilkan. Berikut diagram alir dari fitur ini:



Gambar 11 Diagram alir(activity diagram) fitur *My Location*

BAB IV

HASIL PENGUJIAN

4.1 Aspek dan Skenario Pengujian

Aspek yang ditinjau dalam perancangan aplikasi uBeacon adalah aplikasi yang dapat menyediakan informasi sesuai keberadaan pengguna dengan fitur *Beacon Information*, serta kemudahan bagi pengguna dalam menggunakan aplikasi. Aplikasi juga dirancang untuk dapat digunakan sebagai alat konfirmasi presensi kelas kuliah dan menunjukkan lokasi pengguna di dalam kampus dengan fitur *Attend Class* dan *My Location*.

Untuk dapat mengakses seluruh fitur pada aplikasi uBeacon dibutuhkan koneksi internet dan bluetooth 4.0 yang aktif untuk dapat mengakses atribut beacon dan server IBM Bluemix yang menampung informasi nya. Pada bagian aliran informasi aplikasi uBeacon akan diuji jika terdapat notifikasi tentang perangkat beacon yang terdeteksi dan akan menunjukkan Beacon Information ketika di klik.

Selanjutnya, pada fitur *Beacon Information* akan ditampilkan sebuah baris yang terdiri dari berbagai informasi yang ditunjukkan berdasarkan nomor perangkat beacon yang terdeteksi. Jika pengguna menekan salah satu informasi, maka akan ditampilkan informasi detail dari yang bersangkutan.

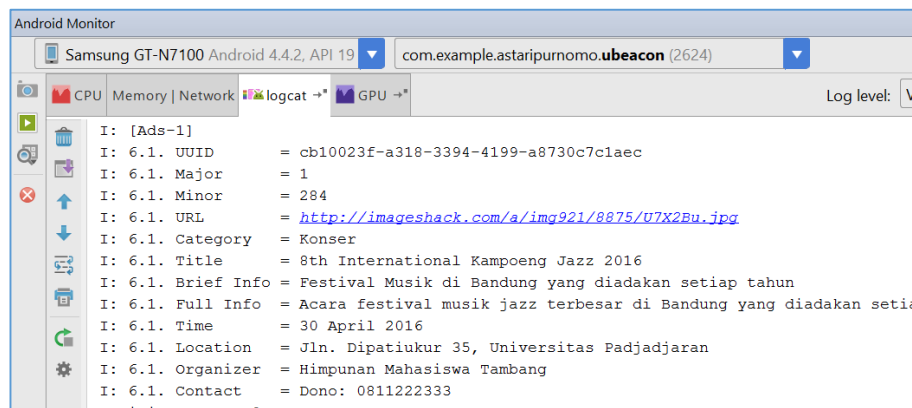
Pada fitur *Attend Class*, pengguna akan ditampilkan halaman dengan keterangan kelas dan mata kuliah yang akan diujikan sesuai dengan beacon yang terdeteksi. Jika pengguna menekan tombol *Attend Class*, pengguna telah berhasil melakukan konfirmasi presensi kelas kuliah dan hanya perlu menunggu hingga kelas selesai.

Pada fitur *My Location*, ketika pengguna mendeteksi keberadaan sebuah beacon maka pengguna dapat menekan tombol *My Location* yang akan menunjukkan sebuah gambar pada layar *smartphone* nya yang berdasarkan pada atribut beacon yang terdeteksi di kampus.

4.2 Prosedur dan Hasil Pengujian

4.2.1 Aliran Informasi Aplikasi uBeacon

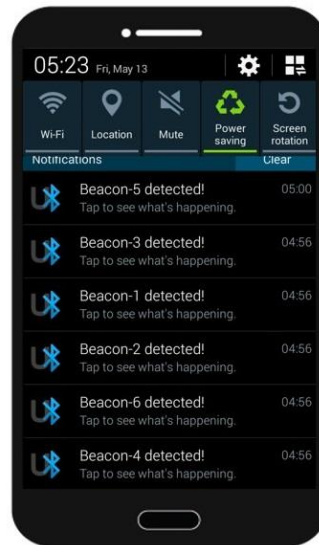
Aliran informasi pada aplikasi uBeacon terjadi pada *background activity* aplikasi. Proses ini dapat dilihat pada bagian *logcat* (log) pada software Android Studio untuk melihat aliran informasi yang terjadi. Berikut contoh tampilan aliran data yang terjadi saat *smartphone* mendeteksi keberadaan perangkat beacon :



Gambar 12 logcat pada Android Studio

Informasi yang didapatkan akan ditampung terlebih dahulu untuk ditampilkan pada halaman utama ketika aplikasi dijalankan.

Pada saat yang bersamaan ketika perangkat beacon terdeteksi, proses ini akan menampilkan notifikasi yang menandakan keberadaan beacon dan terjadi aliran informasi. Notifikasi akan muncul dengan sebuah tulisan berupa “Beacon detected!” untuk menandakan keberadaan beacon. Berikut contoh tampilan ketika terdapat enam (6) buah beacon yang terdeteksi :

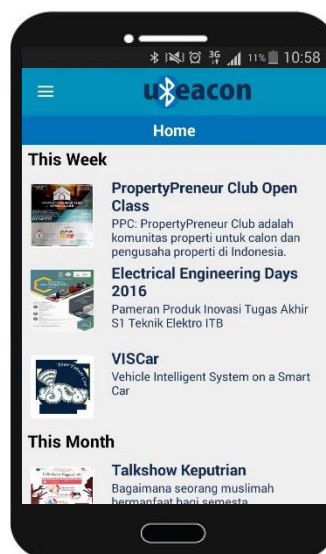


Gambar 13 Notifikasi pada *smartphone* yang mendeteksi Beacon

Jika salah satu dari notifikasi tersebut ditekan, maka akan masuk ke bagian *Beacon Information* yang akan dijelaskan pada bagian selanjutnya.

4.2.2 Fitur *Beacon Information*

Pada bagian ini akan ditampilkan aktivitas utama berupa barisan berisi informasi yang terdapat pada beacon yang terdeteksi. Informasi akan ditampilkan dalam bentuk ListView dan informasi akan dikelompokkan sesuai dengan waktu kejadian nya. Pengelompokkan informasi berdasarkan waktu dibagi dalam lima buah kelompok, antara lain *Today*, *This Week*, *This Month*, *This Year*, dan *All Time*. Berikut tampilan utama dari aplikasi uBeacon pada proses *Beacon Information*.



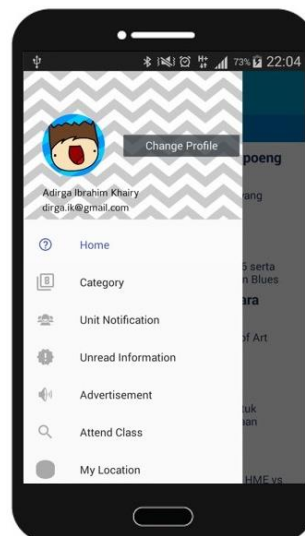
Gambar 14 Tampilan utama aplikasi uBeacon

Jika pengguna menekan salah satu informasi yang terdapat pada baris tersebut, pengguna akan ditampilkan detil informasi tentang salah satu informasi tersebut. Detil informasi yang ditampilkan terdiri dari *Organizer*, *Category*, *Time*, *Location*, *Description*, *Contact*, serta sebuah gambar poster dari acara yang bersangkutan. Contoh nya adalah ketika pengguna menekan informasi pada baris-2, yaitu Electrical Engineering Days 2016, maka akan ditampilkan tampilan sebagai berikut:



Gambar 15 Tampilan detil Beacon Information

Selanjutnya, pengguna dapat mengakses berbagai fitur lain dengan cara membuka menu *navigation drawer* yang terdapat pada halaman utama yang terdiri dari *Home*, *Category*, *Unit Notification*, *Attend Class*, dan *My Location*.



Gambar 16 Tampilan pada *Navigation Drawer*

4.2.3 Fitur *Categories*

Pada bagian ini akan dilakukan *filter* informasi sesuai dengan atribut *category* yang terdapat pada detail informasi nya. Pengguna dapat memilih berbagai kategori yang terdiri dari *Seminar*, *Campus Info*, *Promotion*, *Unit dan Himpunan Event*, *Workshop*, dan *Concert*.



Gambar 17 Tampilan pada menu kategori

Jika pengguna menekan salah satu *icon*, pengguna akan mengakses halaman utama kembali dengan tampilan ListView yang menunjukkan informasi yang tersaring yang termasuk dalam kategori tertentu saja. Berikut contoh tampilan dari fitur *Categories*:



Gambar 18 Tampilan menu *Categories*

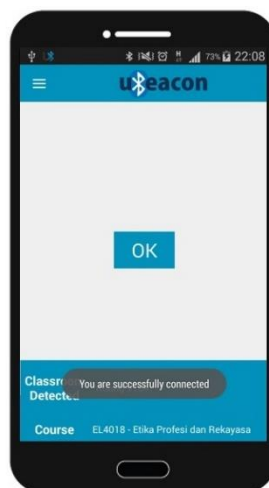
4.2.4 Fitur *Attend Class*

Selanjutnya jika pengguna memasuki ruang kelas kuliah, pengguna akan mendeteksi keberadaan beacon di dalam kelas dan mengakses informasi pada beacon tersebut kepada server IBM Bluemix. Aplikasi akan mendapatkan nomor ruang kelas beserta mata kuliah yang sedang diikuti untuk memastikan kebenaran informasi kepada pengguna. Berikut tampilan yang akan muncul ketika pengguna memasuki kelas dan memilih menu *Attend Class* :



Gambar 19 Tampilan pada *Attend Class*

Jika keterangan kelas sudah sesuai, pengguna perlu melakukan penekanan tombol *Start Class* untuk mendaftarkan diri pada presensi kelas. Berikut tampilan sesudah tombol ditekan:

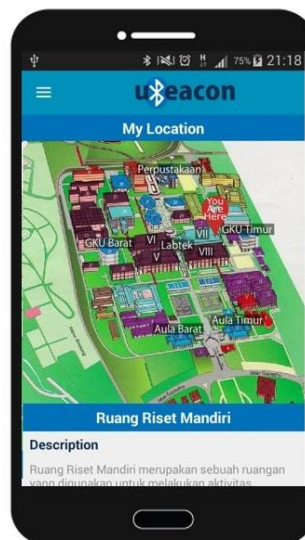


Gambar 20 Tampilan akhir *Attend Class*

Pengguna akan terus mengirimkan sebuah payload dalam jangka waktu tertentu setiap 300 sekon (5 menit) untuk memastikan keberadaan pengguna di dalam kelas.

4.2.5 Fitur *My Location*

Selanjutnya, fitur yang terakhir dapat digunakan ketika pengguna telah mendeteksi keberadaan sebuah beacon di sekitarnya. Pengguna akan mendapatkan atribut dari beacon yang terdeteksi dan menerima informasi dari server IBM Bluemix yang berisikan sebuah gambar peta kampus dengan sebuah *pinpoint* yang menunjukkan keberadaan pengguna dengan indicator tulisan “You’re Here”. Pengguna juga akan mendapatkan deskripsi tentang lokasi yang bersangkutan untuk mempermudah pengguna mengetahui sebuah lokasi baru, khususnya bagi mahasiswa baru. Berikut gambar yang akan ditampilkan pada layar *smartphone* pengguna:



Gambar 21 Tampilan pada fitur My Location

Fitur ini menyajikan gambar peta yang *user friendly* agar dapat mempermudah pengguna dalam mengenal lokasi di lingkungan kampus.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapatkan dari tugas akhir ini adalah:

1. Aplikasi uBeacon dapat berinteraksi dengan perangkat beacon dan server IBM Bluemix.
2. Aplikasi dapat menunjukkan informasi yang terdapat pada setiap beacon yang diatur dalam server IBM Bluemix.
3. Graphical User Interface dapat bekerja dengan baik dan bersifat *user friendly*.
4. Fitur *Attend Class* dan *My Location* dapat berfungsi dengan baik.

5.2 Saran

Saran yang dapat disampaikan untuk pengembangan tugas akhir ini:

1. Sistem ini dapat dikembangkan dan diimplementasikan secara nyata dengan mengintegrasikan sistem dengan server di kampus.
2. Untuk dapat meminimalisir pengeluaran dana jangka panjang, pembangunan infrastruktur server khusus di ITB (tanpa IBM Bluemix) dapat dilakukan tanpa menghapus fitur yang dibutuhkan.
3. Aplikasi dapat dikembangkan lebih luas pada *smartphone* dengan sistem operasi yang berbeda, seperti pada Apple dan berbagai OS yang lainnya.

DAFTAR PUSTAKA

- [1] IbeacoInsider's Web, <http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/>
- [2] Indoor Location Technologies Compared, <http://lighthouse.io/indoor-location-technologies-compared/>
- [3] Dirk Van Merode, "Flexible Technologies for Smart Campus," *2016 13th International Conference on REV*, vol. ED-11, no. 1, pp. 64-68 , Feb. 2016.
- [4] Beaconsandwich Beacon, <http://www.beaconsandwich.com/what-is-ibeacon.html>
- [5] HiveMQ MQTT Broker, <http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>
- [6] Michael Wang, "Managing Large Scale, Ultra-Dense Beacon Deployments in Smart Campuses," *The First International Workshop on Smart Cities and Urban Informatics 2015*, vol. ED-11, no. 1, pp. 606-611 , 2015.

LAMPIRAN

Source Code untuk melakukan Beacon Monitoring

```
//Import Library yang dibutuhkan
import android.app.Application;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.app.Notification;
import android.content.SharedPreferences;
import android.util.Log;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;
import java.util.UUID;
import java.util.List;
import com.estimote.sdk.BeaconManager;
import com.estimote.sdk.Region;
import com.estimote.sdk.Beacon;
import com.estimote.sdk.Utls;
import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;
import org.eclipse.paho.client.mqttv3.MqttCallback;
import org.eclipse.paho.client.mqttv3.MqttClient;
import org.eclipse.paho.client.mqttv3.MqttException;
import org.eclipse.paho.client.mqttv3.MqttMessage;
import org.eclipse.paho.client.mqttv3.MqttTopic;
import org.eclipse.paho.client.mqttv3.persist.MqttDefaultFilePersistence;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

public class MyApplication extends Application implements MqttCallback {

    public static MqttClient client;

    //Deklarasi Beacon Region
    private BeaconManager beaconManager1, beaconManager2, beaconManager3,
        beaconManager4, beaconManager5, beaconManager6;

    //Deklarasi variabel akun user
    private String nameHeader, emailHeader, username, password, unit, nim,
        nim_notif, infourl, title,time,notifications,location,contact ;

    //Deklarasi variabel iterasi
    private int index11 = 1, j=1;

    //onCreate Cycle
    @Override
    public void onCreate() {
        super.onCreate();

        //Define Beacon Region Object, didefinisikan dalam 6 beacon
        //Dibedakan dengan nilai Major nya
        beaconManager1 = new BeaconManager(getApplicationContext());
        beaconManager2 = new BeaconManager(getApplicationContext());
        beaconManager3 = new BeaconManager(getApplicationContext());
        beaconManager4 = new BeaconManager(getApplicationContext());
        beaconManager5 = new BeaconManager(getApplicationContext());
        beaconManager6 = new BeaconManager(getApplicationContext());
```

```

//Define nilai scan = 60s, wait = 180s setiap beacon
beaconManager1.setBackgroundScanPeriod(60000,180000);
beaconManager2.setBackgroundScanPeriod(60000,180000);
beaconManager3.setBackgroundScanPeriod(60000,180000);
beaconManager4.setBackgroundScanPeriod(60000,180000);
beaconManager5.setBackgroundScanPeriod(60000,180000);
beaconManager6.setBackgroundScanPeriod(60000,180000);

//Mendapatkan user profile setelah login
//Untuk fitur personalisasi
SharedPreferences pref = getSharedPreferences(Data.SPREF_NAME,
MODE_PRIVATE);

nameHeader = pref.getString("Name", "");
emailHeader = pref.getString("Email", "");
username = pref.getString("Username", "");
password = pref.getString("Password", "");
unit = pref.getString("Unit", "");
nim = pref.getString("NIM", "");
if (nim == null || nim.length() <= 0) {
    nim_notif = "";
}
else {
    nim_notif = nim.substring(0, 5);
}

//Inisialisasi hubungan user dengan koneksi MQTT
try {
    String cacheDir = getCacheDir().getAbsolutePath();

    client = new MqttClient("tcp://test.mosquitto.org:1883",
"ubeacon/user/request",
    new MqttDefaultFilePersistence(cacheDir));

    //connect to MQTT
    client.connect();
    //callback untuk inisialisasi publish dan subscribe
    client.setCallback(MyApplication.this);
    //subscribe MQTT
    client.subscribe("ubeacon/user/response/" + username);
} catch (MqttException e) {
    throw new RuntimeException(e);
}

//Beacon Major-1
beaconManager1.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {
        beaconManager1.startMonitoring(new Region(
            "monitored region",
            UUID.fromString("CB10023F-A318-3394-4199-
A8730C7C1AEC"), 1, 284));
        //UUID.fromString(UUID, Major, Minor);
    }
});

beaconManager1.setMonitoringListener(new
BeaconManager.MonitoringListener() {
    @Override
    public void onEnteredRegion(Region region, List<Beacon> list)
{
        //Assign variabel Major_onEntered = "1"
        Data.Major_onEntered = "1";
    }
}

```

```

        //Memberikan notifikasi bahwa beacon terdeteksi
        //Memanggil fungsi showNotification
        showNotification(
            "Beacon detected!",
            "Tap to see what's happening.",
            SplashWelcome.class);

        //Melakukan publish payload ke server melalui MQTT
        publishPayload("1");
    }

    @Override
    public void onExitedRegion(Region region) {
        //Assign nilai Major_onEntered = "99"
        //ketika meninggalkan beacon region
        Data.Major_onEntered = "99";
    }
});

//Beacon Major-2
beaconManager2.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {
        beaconManager2.startMonitoring(new Region(
            "monitored region",
            UUID.fromString("CB10023F-A318-3394-4199-
A8730C7C1AEC"), 2, 284));
        //UUID.fromString(UUID, Major, Minor);
    }
});

beaconManager2.setMonitoringListener(new
BeaconManager.MonitoringListener() {
    @Override
    public void onEnteredRegion(Region region, List<Beacon> list)
    {
        //Assign variabel Major_onEntered = "2"
        Data.Major_onEntered = "2";

        //Memberikan notifikasi bahwa beacon terdeteksi
        showNotification(
            "Beacon detected!",
            "Tap to see what's happening.",
            SplashWelcome.class);

        //publish Payload 2 kepada server
        publishPayload("2");
    }

    @Override
    public void onExitedRegion(Region region) {
        //Assign nilai 99 ketika meninggalkan region beacon
        Data.Major_onEntered = "99";
    }
});
//-----//
//Beacon Major-3
beaconManager3.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {
        beaconManager3.startMonitoring(new Region(
            "monitored region",
            UUID.fromString

```

```

        ("CB10023F-A318-3394-4199-
A8730C7C1AEC"),3,284));
        //UUID.fromString(UUID, Major, Minor);
    }
    });

    beaconManager3.setMonitoringListener(new
    BeaconManager.MonitoringListener() {
        @Override
        public void onEnteredRegion(Region region, List<Beacon> list)
        {
            //Assign nilai 3 saat masuk region 3
            Data.Major_onEntered = "3";

            //Menampilkan notifikasi
            showNotification(
                "Beacon detected!",
                "Tap to see what's happening.",
                SplashWelcome.class);

            //publish payload 3
            publishPayload("3");
        }

        @Override
        public void onExitedRegion(Region region) {
            //assign nilai 99 ketika meninggalkan lokasi
            Data.Major_onEntered = "99";
        }
    });

    //Beacon Major-4
    beaconManager4.connect(new BeaconManager.ServiceReadyCallback() {
        @Override
        public void onServiceReady() {
            beaconManager4.startMonitoring(new Region(
                "monitored region",
                UUID.fromString
                ("CB10023F-A318-3394-4199-
A8730C7C1AEC"),4,284));
            //UUID.fromString(UUID, Major, Minor);
        }
    });

    beaconManager4.setMonitoringListener(new
    BeaconManager.MonitoringListener() {
        @Override
        public void onEnteredRegion(Region region, List<Beacon> list)
        {
            //Assign Major_onEntered = 4
            Data.Major_onEntered = "4";

            //Menampilkan notifikasi keberadaan beacon
            showNotification(
                "Beacon detected!",
                "Tap to see what's happening.",
                SplashWelcome.class);

            //publish payload 4
            publishPayload("4");
        }

        @Override
        public void onExitedRegion(Region region) {

```

```

        //Assign nilai Major_onEntered 99
        //ketika meninggalkan lokasi beacon
        Data.Major_onEntered = "99";
    }
});

//Beacon Major-5
beaconManager5.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {
        beaconManager5.startMonitoring(new Region(
            "monitored region",
            UUID.fromString
                ("CB10023F-A318-3394-4199-
A8730C7C1AEC"), 5, 284));
        //UUID.fromString(UUID, Major, Minor);
    }
});

beaconManager5.setMonitoringListener(new
BeaconManager.MonitoringListener() {
    @Override
    public void onEnteredRegion(Region region, List<Beacon> list)
    {
        //Assign Major_onEntered 5
        Data.Major_onEntered = "5";

        //Assign bahwa beacon kelas terdeteksi
        Data.Major_Class_onEntered = "5";
        Data.BeaconClass_detected = "yes";

        showNotification(
            "Classroom Beacon detected!",
            "Tap to see what's happening.",
            SplashWelcome.class);
        publishPayload("5");
    }

    @Override
    public void onExitedRegion(Region region) {
        Data.Major_onEntered = "99";

        //Jika meninggalkan kelas, beacon terdeteksi adalah 99
        //Ignore payload
        Data.Major_Class_onEntered = "99";
    }
});

//Beacon Major-6
beaconManager6.connect(new BeaconManager.ServiceReadyCallback() {
    @Override
    public void onServiceReady() {
        beaconManager6.startMonitoring(new Region(
            "monitored region",
            UUID.fromString
                ("CB10023F-A318-3394-4199-
A8730C7C1AEC"), 6, 284));
        //UUID.fromString(UUID, Major, Minor);
    }
});

beaconManager6.setMonitoringListener(new
BeaconManager.MonitoringListener() {

```

```

        @Override
        public void onEnteredRegion(Region region, List<Beacon> list)
        {
            //Assign Major_onEntered 6
            Data.Major_onEntered = "6";

            //Assign bahwa beacon kelas terdeteksi
            Data.Major_Class_onEntered = "6";
            Data.BeaconClass_detected = "yes";

            //Tampilkan notifikasi bahwa beacon terdeteksi
            showNotification(
                "Classroom Beacon detected!",
                "Tap to see what's happening.",
                Navigation.class);
            publishPayload("6");
        }

        @Override
        public void onExitedRegion(Region region) {
            Data.Major_onEntered = "99";

            //Jika meninggalkan kelas, beacon terdeteksi adalah 99
            //Ignore payload
            Data.Major_Class_onEntered = "99";
        }
    });
}

//Fungsi untuk menampilkan notifikasi
public void showNotification(String title, String message, Class Move)
{
    //Fungsi random untuk melakukan acak pada reserve notifikasi
    Random random = new Random();
    int m = random.nextInt(9999 - 1000) + 1000;

    //Perintah untuk menampilkan notifikasi
    Intent notifyIntent = new Intent(this, Move);
    notifyIntent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
    PendingIntent pendingIntent = PendingIntent.getActivities(this, 0,
        new Intent[] { notifyIntent },
        PendingIntent.FLAG_UPDATE_CURRENT);
    Notification notification = new Notification.Builder(this)
        .setSmallIcon(R.drawable.icon)
        .setContentTitle(title)
        .setContentText(message)
        .setAutoCancel(true)
        .setContentIntent(pendingIntent)
        .build();
    notification.defaults |= Notification.DEFAULT_SOUND;
    NotificationManager notificationManager =
        (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
    notificationManager.notify(m, notification);
}

//Fungsi untuk melakukan publish payload via MQTT
private void publishPayload(String majorDetected){
    //Paket JSON Payload pada MQTT
    JSONObject jsonObject = new JSONObject();

    //Generate beacon payload berdasarkan beacon yang terdeteksi

```



```

        try {
            JSONArray array1 = new JSONArray();
            JSONObject obj = new JSONObject();
            JSONObject obj11 = new JSONObject();

            obj.put("descriptor", username);
            obj.put("detectedTime", System.currentTimeMillis());
            obj11.put("proximityUUID", "cb10023f-a318-3394-4199-a8730c7c1aec");
            obj11.put("major", majorDetected);
            obj11.put("minor", "284");
            obj11.put("rssi", -81);
            obj11.put("accuracy", 1.85412312);
            obj11.put("proximity", "Near");
            obj.put("data", obj11);
            array1.put(obj);
            jsonObject.put("bnm", array1);

            Log.d("Sending Data-" + index11, " ");

            //Menentukan topic dari MQTT - topic = request
            MqttTopic mqttTopic = client.getTopic("ubeacon/user/request");
            MqttMessage mqttMessage = new
MqttMessage(jsonObject.toString().getBytes());
            mqttMessage.setQos(1);
            mqttTopic.publish(mqttMessage);

            //Iterasi untuk pengecekan pada log
            index11++;

        } catch (MqttException e) {
            throw new RuntimeException(e);
        } catch (JSONException e) {
        }
    }

    @Override
    public void connectionLost(Throwable throwable) {
    }

    //Proses Subscribe
    //Untuk menerima data balikan dari server via MQTT
    @Override
    public void messageArrived(String s, MqttMessage mqttMessage) throws
Exception {
        JSONObject object = new JSONObject (String.valueOf(mqttMessage));

        try {
            int i;

            //Inisialisasi variable penampung informasi
            String e_uuid, e_major, e_minor, e_url, e_category, e_title,
e_brief, e_full, e_time,
                e_location, e_organizer, e_contact;
            //Inisialisasi variabel untuk fitur My Location
            String e_loc_Url, e_loc_Title, e_loc_Desc;

            //Inisialisasi variabel untuk kelas terdeteksi
            String e_course, e_classroom;

            Long converted_time;
            String converted_result;

            //store JSON text subscribed ke dalam variabel

```

```

e_uuid = object.getString("proximityUUID").toString();
e_major = object.getString("major").toString();
e_minor = object.getString("minor").toString();

//store JSON text subscribed untuk fitur My Location
e_loc_Url = object.getString("location_imgurl");
e_loc_Title = object.getString("location_title");
e_loc_Desc = object.getString("location_desc");

Data.loc_imgUrl = e_loc_Url;
Data.loc_myLoc = e_loc_Title;
Data.loc_myDesc = e_loc_Desc;

//store JSON text subscribed untuk fitur Attend Class
e_course = object.getString("course").toString();
e_classroom = object.getString("classroom").toString();

if (!((e_course.equals(" ")) && (e_classroom.equals(" ")))) {
    Data.ClassRoom_onEntered = e_classroom;
    Data.Course_onEntered = e_course;
}

//Store JSON Array
String tempArray = object.getString("data").toString();
JSONArray dataArray = new JSONArray(tempArray);
for (i = 0; i < dataArray.length(); i++) {
    JSONObject object2 = dataArray.getJSONObject(i);

    e_url = object2.getString("image_url").toString();
    e_category = object2.getString("category").toString();
    e_title = object2.getString("title").toString();
    e_brief = object2.getString("brief").toString();
    e_full = object2.getString("full").toString();
    //waktu didapatkan dalam format epoch time
    e_time = object2.getString("time").toString();
    //Untuk mensortir epoch time yang didapatkan
    berdasarkan waktu
    converted_time = Long.parseLong(e_time);
    Date date = new Date(converted_time);
    SimpleDateFormat sdf = new SimpleDateFormat("dd MMMM
yyyy");

    SimpleDateFormat sdf2 = new SimpleDateFormat("HH:mm");
    String formattedDate = sdf.format(date);
    String formattedDate2 = sdf2.format(date);
    converted_result = (formattedDate + ",
"+formattedDate2+ " WIB");
    e_location = object2.getString("location").toString();
    e_organizer = object2.getString("organizer").toString();
    e_contact = object2.getString("contact").toString();

    //Menampung variabel dalam sebuah java constructor
    Events events = new
Events(e_url,e_title,e_full,e_brief,e_category,e_location,
e_organizer, e_contact, e_major, converted_result,
e_uuid, e_minor);
    events.date = date;

    Data.addEventList(events);

    //Categories -- Filter Information

    //Seminar

if(e_category.equals("Seminar")|e_category.equals("Pameran")){

```

```

        Data.filtered_Seminar.add(events);
    }

    //Campus Info
    if(e_category.equals("Info
Kampus") | e_category.equals("Campus Info")) {
        Data.filtered_CampusInfo.add(events);
    }

    //Workshop
    if(e_category.equals("Workshop") | e_category.equals("Pelatihan")) {
        Data.filtered_Workshop.add(events);
    }

    //Concert
    if(e_category.equals("Konser") | e_category.equals("Concert")
        | e_category.equals("Festival")) {
        Data.filtered_Concert.add(events);
    }

    //Olahraga
    if(e_category.equals("Olahraga") | e_category.equals("sports")) {
        Data.filtered_Sports.add(events);
    }

    //Acara Kampus
    if(e_category.equals("Acara
Kampus") | e_category.equals("Campus Events")
        | e_category.equals("Pameran")) {
        Data.filtered_Events.add(events);
    }

    //Promosi
    if(e_category.equals("Promosi") | e_category.equals("Promotion")) {
        Data.filtered_Promotion.add(events);
    }

    //UKM dan Unit
    if(e_category.equals("Unit
Kemahasiswaan") | e_category.equals("Acara Himpunan")
        | e_category.equals("Kegiatan Himpunan")) {
        Data.filtered_UKM_dan_Unit.add(events);
    }
}

//Iterasi nilai log
index11++;
} catch (JSONException e) {
    e.printStackTrace();
}
}

@Override
public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {
    System.out.println("Sending Data-"+j+" Completed");
    j++;
}
}

```