



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERANGKAT
BERGERAK PERIKLANAN BERBASIS LOKASI
DENGAN INDOOR LOCALIZATION UNTUK
SARANA PROMOSI PADA PUSAT
PERBELANJAAN

RATIH AYU INDRASWARI
NRP 5112 100 122

Dosen Pembimbing I
Dr. tech. Ir. R. V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Fajar Baskoro, S.Kom, M.T.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERANGKAT
BERGERAK PERIKLANAN BERBASIS LOKASI
DENGAN INDOOR LOCALIZATION UNTUK
SARANA PROMOSI PADA PUSAT
PERBELANJAAN

RATIH AYU INDRASWARI
NRP 5112 100 122

Dosen Pembimbing I
Dr.tech. Ir. R. V. Hari Ginardi, M.Sc.

Dosen Pembimbing II
Fajar Baskoro, S.Kom, M.T.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



FINAL PROJECT - KI141502

DESIGN AND IMPLEMENTATION OF LOCATION
BASED ADVERTISING MOBILE APPLICATION
WITH INDOOR LOCALIZATION AS A
PROMOTION MEDIA IN SHOPPING CENTER

RATIH AYU INDRASWARI
NRP 5112 100 122

Supervisor I
Dr.tech. Ir. R. V. Hari Ginardi, M.Sc.

Supervisor II
Fajar Baskoro, S.Kom, M.T.

DEPARTMENT OF INFORMATICS
Faculty of Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PERANGKAT BERGERAK PERIKLANAN BERBASIS LOKASI DENGAN *INDOOR LOCALIZATION* UNTUK SARANA PROMOSI PADA PUSAT PERBELANJAAN

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Rumpun Mata Kuliah Algoritma Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

RATIH AYU INDRASWARI

NRP. 5112 100 122

Disetujui oleh Pembimbing Tugas Akhir

1. Dr.tech. Ir. R. V. Hari ~~Gunardi~~, M.Tc. (Pembimbing 1)
NIP: 19650518 199203 1003
2. Fajar Baskoro, S.Kom, M.T. (Pembimbing 2)
NIP: 19740403 199903 1 002

**SURABAYA
JUNI, 2016**

(Halaman ini sengaja dikosongkan)

RANCANG BANGUN APLIKASI PERANGKAT BERGERAK PERIKLANAN BERBASIS LOKASI DENGAN *INDOOR LOCALIZATION* UNTUK SARANA PROMOSI PADA PUSAT PERBELANJAAN

Nama Mahasiswa : Ratih Ayu Indraswari
NRP : 5112 100 122
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R. V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Fajar Baskoro, S.Kom, M.T.

Abstrak

Periklanan atau advertising sangat erat hubungannya dengan bisnis atau usaha. Disisi lain, sebagai konsumen tentu sangat membutuhkan informasi mengenai sebuah produk atau jasa agar kebutuhan dapat terpenuhi dengan efektif. Teknologi khususnya teknologi perangkat bergerak berkembang cukup pesat. Perangkat bergerak sudah semakin menyatu dengan kehidupan personal manusia, sehingga secara langsung maupun tidak langsung perangkat bergerak yang dibawa oleh seseorang dapat menjadi sebuah pemancar yang memberikan informasi keberadaan mereka. Oleh sebab itu, dikembangkan sebuah sistem advertising yang dapat memberikan informasi promosi, iklan, dan lain-lain secara langsung kepada konsumen sesuai dengan lokasi saat itu. Tujuan dikembangkannya sistem ini adalah untuk memberikan solusi advertising yang tepat sasaran dan memudahkan konsumen untuk mengetahui informasi mengenai promosi yang ada di sekitarnya.

Pada tugas akhir ini sistem dibuat agar konsumen (dalam hal ini pengunjung pusat perbelanjaan) dapat mengetahui promosi yang sedang berlaku pada booth yang sedang mereka kunjungi. Hal ini memanfaatkan proses pendeteksian lokasi, sistem didukung oleh konsep Indoor Localization yang memungkinkan untuk

menentukan lokasi pengguna yang berada didalam ruangan atau gedung.

Pada pengimplementasian konsep Indoor Localization, dilakukan dengan mengklasifikasikan data sinyal yang didapat pada waktu tertentu. Klasifikasi dilakukan menggunakan metode Support Vector Machine setelah melakukan pengumpulan data sampling. Pengumpulan data sampling dilakukan terhadap kekuatan sinyal dan Basic Service Set Identifier (BSSID) yang ditangkap oleh smartphone pada beberapa booth yang ada di East Coast Center. Dari data-data tersebut dilakukan training dan prediksi untuk menentukan lokasi.

Uji coba dilakukan dengan melakukan pengujian terhadap fungsionalitasnya melalui skenario yang mencerminkan fitur-fitur pada aplikasi. Selain itu, pada implementasi Indoor Localization uji coba dilakukan dengan mendeteksi lokasi pada beberapa booth untuk mendapatkan tingkat akurasi hasil deteksi lokasi dengan persentase benar sebesar 73%.

Kata kunci: Location Based Advertising, Indoor Localization, Support Vector Machine, WiFi

DESIGN AND IMPLEMENTATION OF LOCATION BASED ADVERTISING MOBILE APPLICATION WITH INDOOR LOCALIZATION AS A PROMOTION MEDIA IN SHOPPING CENTER

Student's Name : Ratih Ayu Indraswari
Student's ID : 5112 100 122
Department : Informatics Department FTIF-ITS
First Advisor : Dr.tech. Ir. R. V. Hari Ginardi, M.Sc.
Second Advisor : Fajar Baskoro, S.Kom, M.T.

Abstract

Advertising is highly correlated with businesses. In customer's point of view, they require information of a product or service in order to satisfy their needs effectively. The growth of technology, mobile device specifically, is shifting rapidly. Mobil device became integrated even more with human's personal life. Therefore, mobile device which is brought by people could actually become a transmitter of their location either directly or indirectly. Given these points, an advertising system that is able to directly provide information such as promotion and advertising based on customer's location at that time is developed. The purpose of the system development is to accommodate accurate advertising solution and also to simplify customer in discovering information of promotion nearby. In this research, systems are made in order that customer, which is the shopping center visitor, will be able to have information on promotions that is available at a certain booth that they stop by. It takes advantage of location detection process; the system is supported by Indoor Localization concept which allows determining user's location inside a room or building.

On the implementation of Indoor Localization concept, it is performed by classifying the signal data that is obtained at a certain time. Classification is done by using Support Vector Machine and Binary Tree methods after collecting the data

sampling. The collection of data sampling is done towards the strength of signal and Basic Service Set Identifier (BSSID) that captured by smartphone on several booths at East Coast Center. Training and prediction to determine location are performed by those data.

Pilot testing is accomplished by examining the functionality through scenarios that reflect the features of the application. In addition, on the Indoor Localization implementation, the pilot testing is completed by detecting the location on several booths to achieve the degree of accuracy as a result from location detection with 73% of correct prediction percentage.

Keywords: Location Based Advertising, Indoor Localization, Support Vector Machine, Wi-Fi

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Bismillahirrohmanirohim.

Alhamdulillahirabil'amin, segala puji hanya milik Allah SubhanahuWata'alla, atas segala rahmat dan karunia-Nya yang tak terhingga sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Rancang Bangun Aplikasi Perangkat Bergerak Periklanan Berbasis Lokasi dengan *Indoor Localization* untuk Sarana Promosi Pada Pusat Perbelanjaan**” dengan baik dan tepat waktu.

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Kedua orang tua penulis yang tiada henti-hentinya mencurahkan kasih sayang, perhatian, doa, dukungan, semangat, dan kebutuhan penulis.
3. Pandu dan Bimo, adik penulis yang menjadi teman penulis sampai saat ini.
4. Bapak Hari dan Bapak Fajar selaku dosen pembimbing yang telah memberikan bimbingan, motivasi, dan meluangkan waktu dalam pengerjaan Tugas Akhir ini.
5. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
6. Sahabat penulis yang selalu ada, menemani, membantu, dan memberi semangat selama pengerjaan Tugas Akhir: Yeremi, Leli, Hendy, Ephe, dan Jojo.

7. Sahabat semenjak SMA yang memberi motivasi kepada penulis untuk menyusul mereka yang sudah lulus terlebih dahulu: Bela, Lia, Totoy, Emmal, Saly, dan Lalak.
8. Teman-teman yang ikut serta membantu, memberi inspirasi, dan membimbing pengerjaan Tugas Akhir penulis: Djuned, Anggara, Azizha, Dinar, Riko, Alifa, Said, Thiar, Wawan, dan Rona.
9. Keluarga LP Bermartabat yang selalu menemani penulis selama kuliah dan berjuang mengerjakan Tugas Akhir.
10. Keluarga angkatan TC 2012 yang sudah menemani keseharian penulis dari membuka mata sampai menutup mata di kampus perjuangan.
11. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu -persatu.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2016

Ratih Ayu Indraswari

DAFTAR ISI

LEMBAR PENGESAHAN	vii
Abstrak	ix
Abstract.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER.....	xxiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat	3
1.5 Metodologi	4
1.6 Sistematika Penyusunan Laporan	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Support Vector Machine	7
2.2 Location Based Services	8
2.2.1 Indoor Localization System	8
2.2.2 Location Based Advertising.....	9
2.3 Wireless Fidelity.....	9
2.4 CodeIgniter.....	10
2.5 Android	10
BAB III PERANCANGAN PERANGKAT LUNAK.....	11
3.1 Analisis.....	11
3.1.1 Analisis Permasalahan	11
3.1.2 Deskripsi Umum Perangkat Lunak.....	12
3.1.3 Arsitektur Sistem.....	14

3.1.4	Kebutuhan Fungsional Aplikasi Perangkat Bergerak..	15
3.1.5	Kebutuhan Fungsional Aplikasi Web	20
3.2	Perancangan	29
3.2.1	Perancangan Konsep Indoor Localization	29
3.2.2	Perancangan Basis Data.....	32
3.2.3	Perancangan Antarmuka Aplikasi Perangkat Bergerak	36
3.2.4	Perancangan Antarmuka Aplikasi Web	39
3.2.5	Perancangan Proses Aplikasi Perangkat Bergerak ..	43
3.2.6	Perancangan Proses Aplikasi Web	43
BAB IV IMPLEMENTASI PERANGKAT LUNAK.....		45
4.1	Lingkungan Implementasi.....	45
4.1.1	Lingkungan Implementasi Perangkat Keras	45
4.1.2	Lingkungan Implementasi Perangkat Lunak	46
4.2	Implementasi Konsep Indoor Localization	46
4.2.1	Implementasi Pengambilan Data.....	46
4.2.2	Implementasi Pre-processing	48
4.2.3	Implementasi Pembentukan Binary Tree.....	50
4.2.4	Implementasi Tahap Pendeteksian Lokasi.....	51
4.3	Implementasi Basis Data.....	53
4.3.1	Implementasi Tabel Admin.....	53
4.3.2	Implementasi Tabel AP	54
4.3.3	Implementasi Tabel Booth.....	54
4.3.4	Implementasi Tabel Scan.....	55
4.3.5	Implementasi Tabel Promo	55
4.3.6	Implementasi Tabel RSS	56
4.3.7	Implementasi Tabel Nearby	57
4.4	Implementasi Antarmuka Aplikasi Perangkat Bergerak	57
4.4.1	Implementasi Antarmuka Halaman Dashboard	57
4.4.2	Implementasi Antarmuka Halaman Nearby Booth .	58

4.4.3	Implementasi Antarmuka Halaman Promo	59
4.4.4	Implementasi Antarmuka Halaman Detail Booth....	60
4.5	Implementasi Proses Aplikasi Perangkat Bergerak	60
4.5.1	Proses Mendeteksi dan Menampilkan Lokasi	60
4.5.2	Proses Menampilkan Detail Booth	62
4.5.3	Proses Menampilkan Booth Terdekat	63
4.5.4	Proses Menampilkan Daftar Promo	65
4.6	Implementasi Antarmuka Aplikasi Web	66
4.6.1	Implementasi Antarmuka Halaman Login	66
4.6.2	Implementasi Antarmuka Halaman Melihat Data ...	67
4.6.3	Implementasi Antarmuka Halaman Menambah Data	67
4.6.4	Implementasi Antarmuka Halaman Mengubah Data...	68
4.7	Implementasi Proses Aplikasi Web	68
4.7.1	Implementasi Proses Melihat Data	69
4.7.2	Implementasi Proses Proses Menambah Data	69
4.7.3	Implementasi Proses Mengubah Data.....	71
4.7.4	Implementasi Proses Menghapus Data	73
BAB V	UJI COBA DAN EVALUASI.....	74
5.1	Lingkungan Uji Coba.....	75
5.2	Dasar Pengujian.....	75
5.3	Skenario dan Hasil Uji Coba	75
5.3.1	Pengujian Pada Konsep Indoor Localization	76
5.3.2	Pengujian Fungsionalitas Aplikasi Perangkat Bergerak 81	
5.3.3	Pengujian Fungsionalitas Aplikasi Web	85
5.4	Evaluasi.....	92
5.4.1	Evaluasi Pengujian Konsep Indoor Localization.....	92
5.4.2	Evaluasi Pengujian Fungsionalitas	94

BAB VI KESIMPULAN DAN SARAN	96
6.1 Kesimpulan	97
6.2 Saran	98
DAFTAR PUSTAKA.....	99
LAMPIRAN	101
BIODATA PENULIS.....	103

DAFTAR GAMBAR

Gambar 2.1. Hyper-plane pada SVM	7
Gambar 2.2. Topologi WiFi.....	10
Gambar 3.1. Diagram Alur Sistem.....	13
Gambar 3.2. Arsitektur Sistem.....	14
Gambar 3.3. Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak.....	15
Gambar 3.4. Diagram Aktivitas Kasus Penggunaan UC-A01	17
Gambar 3.5. Diagram Aktivitas Kasus Penggunaan UC-A02	18
Gambar 3.6. Diagram Aktivitas Kasus Penggunaan UC-A03	19
Gambar 3.7. Diagram Aktivitas Kasus Penggunaan UC-A04	20
Gambar 3.8. Diagram Kasus Penggunaan Aplikasi Web	21
Gambar 3.9. Diagram Aktivitas Kasus Penggunaan UC-B01	25
Gambar 3.10. Diagram Aktivitas Kasus Penggunaan UC-B02...	28
Gambar 3.11. Proses Prediksi Lokasi	30
Gambar 3.12. Binary Tree SVM [8].....	31
Gambar 3.13. Conceptual Data Model	32
Gambar 3.14. Physical Data Model.....	33
Gambar 3.15. Rancangan Antarmuka Halaman Dashboard	37
Gambar 3.16. Rancangan Antarmuka Halaman Nearby.....	38
Gambar 3.17. Rancangan Antarmuka Halaman Promo.....	39
Gambar 3.18. Rancangan Antarmuka Halaman Login.....	40
Gambar 3.19. Rancangan Antarmuka Halaman Melihat Data	40
Gambar 3.20. Rancangan Antarmuka Halaman Menambah Data	41
Gambar 3.21. Rancangan Antarmuka Halaman Mengubah Data	42
Gambar 3.22. Rancangan Antarmuka Halaman Menghapus Data	42
Gambar 4.1. Proses Pengumpulan Data Training	47
Gambar 4.2. Data Sinyal WiFi Dalam Basis data	47
Gambar 4.3. Contoh format array LibSVM	49
Gambar 4.4. Hasil Binary Tree untuk Data Booth	51

Gambar 4.5. Diagram Alur Implementasi SVM dan Binary Tree	52
Gambar 4.6. Implementasi Antarmuka Halaman Dashboard dan Sidebar Menu	58
Gambar 4.7. Implementasi Antarmuka Halaman Nearby Booth	59
Gambar 4.8. Implementasi Antarmuka Halaman Promo	59
Gambar 4.9. Implementasi Antarmuka Halaman Detail Booth ..	60
Gambar 4.10. Denah East Coast Center	64
Gambar 4.11. Implementasi Antarmuka Login.....	67
Gambar 4.12. Implementasi Antarmuka Melihat Data.....	67
Gambar 4.13. Implementasi Antarmuka Menambah Data	68
Gambar 4.14. Implementasi Antarmuka Mengubah Data	68
Gambar 5.1. Pengujian Mendeteksi dan Menampilkan Lokasi ..	82
Gambar 5.2. Pengujian Menampilkan Booth Terdekat	83
Gambar 5.3. Pengujian Menampilkan Detail Booth	84
Gambar 5.4. Pengujian Menampilkan Daftar Promo	85
Gambar 5.5. (a,b) Pengujian Menambah Promo	86
Gambar 5.6. (a,b) Pengujian Mengubah Promo.....	88
Gambar 5.7. Pengujian Menghapus Promo	89
Gambar 5.8. (a,b) Pengujian Menambah Booth.....	90
Gambar 5.9. (a,b) Pengujian Mengubah Booth.....	91
Gambar 5.10. Data kekuatan sinyal pada booth Chatime.....	93
Gambar 5.11. Data kekuatan sinyal pada booth Mokko.....	93
Gambar 5.12. Ilustrasi Pemancaran Sinyal dari Access Point	94

DAFTAR TABEL

Tabel 3.1. Deskripsi Kasus Penggunaan Aplikasi Perangkat Bergerak.....	16
Tabel 3.2. Rincian Alur Kasus Penggunaan UC-A01	16
Tabel 3.3. Rincian Alur Kasus Penggunaan UC-A02	17
Tabel 3.4. Rincian Alur Kasus Penggunaan UC-A03	19
Tabel 3.5. Rincian Alur Kasus Penggunaan UC-A04	20
Tabel 3.6. Deskripsi Kasus Penggunaan Aplikasi Web.....	21
Tabel 3.7. Rincian Alur Kasus Penggunaan UC-B01.....	22
Tabel 3.8. Rincian Alur Kasus Penggunaan UC-B02.....	26
Tabel 3.9. Atribut Tabel AP.....	33
Tabel 3.10. Atribut Tabel RSS.....	34
Tabel 3.11. Atribut Tabel Position	34
Tabel 3.12. Atribut Tabel Booth	35
Tabel 3.13. Atribut Tabel Nearby	35
Tabel 3.14. Atribut Tabel Promo	35
Tabel 3.15. Atribut Tabel Admin	36
Tabel 5.1. Spesifikasi Lingkungan Pengujian Perangkat Lunak. 75	
Tabel 5.2. Detail Waktu Pengujian.....	76
Tabel 5.3. Hasil Pengujian 1	76
Tabel 5.4. Hasil Pengujian 2	77
Tabel 5.5. Hasil Pengujian 3	78
Tabel 5.6. Hasil Pengujian 4	79
Tabel 5.7. Hasil Pengujian 5	79
Tabel 5.8. Persentase Benar Setiap Booth dari Semua Percobaan	80
Tabel 5.9. Skenario Pengujian Mendeteksi dan Menampilkan Lokasi	81
Tabel 5.10. Skenario Pengujian Menampilkan Booth Terdekat..	82
Tabel 5.11. Skenario Pengujian Mendeteksi Detail Booth	83
Tabel 5.12. Skenario Pengujian Menampilkan Daftar Promo.....	84
Tabel 5.13. Skenario Pengujian Menambah Promo	85

Tabel 5.14. Skenario Pengujian Mengubah Promo	87
Tabel 5.15. Skenario Pengujian Menghapus Promo	88
Tabel 5.16. Skenario Pengujian Menambah Booth.....	89
Tabel 5.17. Skenario Pengujian Mengubah Booth.....	90

DAFTAR KODE SUMBER

Kode Sumber 4.1. Implementasi Query Pemilihan Sinyal WiFi.	48
Kode Sumber 4.2. Implementasi Fungsi Pre-processing data.....	49
Kode Sumber 4.3. Implementasi Fungsi Pembentukan Binary Tree	51
Kode Sumber 4.4. Implementasi Fungsi Pelatihan dan Pendeteksian Lokasi	53
Kode Sumber 4.5. Implementasi Struktur Tabel Admin	54
Kode Sumber 4.6. Implementasi Struktur Tabel AP	54
Kode Sumber 4.7. Implementasi Struktur Tabel Booth.....	55
Kode Sumber 4.8. Implementasi Struktur Tabel Scan.....	55
Kode Sumber 4.9. Implementasi Struktur Tabel Promo.....	56
Kode Sumber 4.10. Implementasi Struktur Tabel RSS	56
Kode Sumber 4.11. Implementasi Struktur Tabel Nearby	57
Kode Sumber 4.12. Implementasi Fungsi Mendeteksi Lokasi....	61
Kode Sumber 4.13. Implementasi Fungsi Menampilkan Lokasi	62
Kode Sumber 4.14. Implementasi Fungsi Request Detail Booth	63
Kode Sumber 4.15. Implementasi Fungsi Menampilkan Booth Terdekat.....	65
Kode Sumber 4.16. Implementasi Fungsi Request Daftar Promo	66
Kode Sumber 4.17. Implementasi Fungsi Melihat Data.....	69
Kode Sumber 4.18. Implementasi Fungsi Menambah Data	71
Kode Sumber 4.19. Implementasi Fungsi Mengubah Data	73
Kode Sumber 4.20. Implementasi Fungsi Menghapus Data.....	73

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Periklanan atau *advertising* merupakan hal yang wajib dilakukan oleh para pemilik bisnis bukan hanya untuk mempromosikan dan memasarkan produknya, tetapi juga sebagai alat persuasi dan menciptakan kesan terhadap konsumen. Periklanan telah merambah ke berbagai media seperti televisi, radio, media cetak, pamflet, sosial media, *website*, dan lain-lain. Saat ini iklan telah banyak disisipi promosi sebagai daya tarik terhadap konsumen. Namun biaya yang harus dikeluarkan untuk periklanan tidak sedikit, dan seringkali tidak tepat sasaran. Dari sisi konsumen, iklan dan promosi menjadi media untuk mencari produk yang terbaik dan terjangkau sesuai dengan keinginan dan kebutuhan.

Saat ini, Indonesia merupakan negara keenam pengguna *smartphone* terbanyak di dunia yaitu sebanyak 47 juta pengguna¹. Seiring dengan perkembangan tersebut, keberadaan *smartphone* sudah semakin menyatu dengan kehidupan masyarakat, mulai dari penggunaan media sosial hingga pemanfaatan aplikasi yang dapat mempermudah aktivitas sehari-hari. Bentuk lain dari perkembangan yang terjadi adalah kemudahan memperoleh data-data posisi spasial seseorang melalui *Location Based Services* (LBS). LBS yang dikhususkan pada bidang *service* periklanan dan promosi disebut *Location Based Advertising* (LBA). Hal ini dimungkinkan terjadi karena secara langsung maupun tidak langsung *smartphone* yang dibawa oleh seseorang dapat menjadi sebuah pemancar yang dapat memberikan informasi keberadaan orang tersebut [1]. Untuk mendapatkan lokasi geografis dari sebuah *smartphone*, ada beberapa teknologi yang digunakan oleh

¹ "techno.okezone.com," 2015 September 2015. [Online]. Available: techno.okezone.com/read/2015/09/19/57/1217340/2015-pengguna-smartphene-di-indonesia-capai-55-juta. [Accessed 12 12 2015]

LBS, diantaranya *Global Positioning System*, *Indoor Positioning System* dan *Indoor Localization*.

Dalam tugas akhir ini, aplikasi akan mengimplementasikan konsep dengan *Indoor Localization* yang merupakan teknologi yang digunakan untuk menentukan lokasi sebuah objek yang berada di dalam gedung. Lokasi objek dalam *Indoor Localization* direpresentasikan dengan nama ruangan. Banyak teknologi yang dapat digunakan untuk mendukung *Indoor Localization*, seperti sinyal pada jaringan GSM, sinyal WiFi, Bluetooth, RFID [2]. Sinyal WiFi dipilih karena pada tempat-tempat umum telah terdapat banyak *access point*, sehingga tidak perlu menambah perangkat dan dapat menekan biaya. Dengan memanfaatkan sinyal WiFi yang didapat dan mengklasifikasikannya dengan mengimplementasikan *library* algoritma *Support Vector Machine* (SVM), posisi pengunjung dapat diketahui. Pengunjung dapat melihat diskon, *event*, atau promo yang sedang berlangsung pada *booth* yang dikunjungi.

Tujuan dari dikembangkannya aplikasi ini adalah sebagai sarana promosi yang murah dan efektif bagi pemilik bisnis, dalam studi kasus ini adalah pemilik *booth* dan juga media yang informative dan tepat sasaran bagi konsumen yang dalam studi kasus ini adalah pengunjung East Coast Center, serta mengimplementasikan *Indoor Localization* yang melakukan pendeteksian lokasi seseorang di dalam suatu ruangan atau gedung.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengumpulkan data sampel kekuatan sinyal WiFi pada setiap *booth*?
2. Bagaimana melakukan penentuan lokasi dalam ruangan dengan memanfaatkan data sinyal WiFi dan klasifikasi SVM?
3. Bagaimana membangun *prototype* aplikasi perangkat bergerak berbasis lokasi dengan *Indoor Localization*?

4. Bagaimana mengintegrasikan data hasil *Indoor Localization* dengan promo yang diinputkan oleh pemilik *booth* untuk ditampilkan ke *smartphone* pengunjung?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam Tugas Akhir ini memiliki beberapa batasan, diantaranya sebagai berikut:

1. Data-data mengenai sinyal WiFi didapat dengan memanfaatkan sensor penangkap sinyal WiFi yang ada pada *smartphone* berbasis Android.
2. Implementasi algoritma SVM menggunakan *library* LibSVM dengan Bahasa pemrograman PHP.
3. Lokasi yang akan diuji hanya dua lantai East Coast Center yaitu *Ground* dan *1st Floor*.

1.4 Tujuan dan Manfaat

Tujuan dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Membuat aplikasi berbasis *mobile* yang dapat menangkap dan mengolah informasi sinyal WiFi.
2. Mengimplementasikan algoritma SVM dan informasi sinyal WiFi yang dapat digunakan untuk *Indoor Localization*.
3. Merancang dan membangun *prototype* aplikasi perangkat bergerak berbasis lokasi dengan *Indoor Localization*.
4. Merancang dan membangun aplikasi berbasis perangkat bergerak yang dapat menampilkan data hasil *Indoor Localization* dan data yang diinputkan oleh pemilik *booth*.

Manfaat yang diharapkan dari dikembangkannya aplikasi ini adalah sebagai media periklanan dan promo untuk pemilik

booth. Sedangkan pengunjung dapat menerima informasi mengenai promo dengan mudah sesuai dengan lokasi konsumen.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan Tugas Akhir ini. Berikut ini adalah tahap-tahap dalam pembuatan Tugas Akhir:

a. Studi literatur

Tahap ini merupakan tahap pengumpulan informasi dan pembelajaran yang akan digunakan pada Tugas Akhir ini. Studi literatur meliputi diskusi dan pemahaman mengenai topik Tugas Akhir ini, diantaranya mengenai:

1. *Scanning* kekuatan sinyal WiFi pada *smartphone* Android.
2. Penggunaan *library* LibSVM untuk klasifikasi dengan algoritma SVM.
3. Proses komunikasi data antara aplikasi yang berbasis android dengan *web service* yang dibangun dengan CodeIgniter PHP.
4. Pembuatan *prototype* aplikasi berbasis Android.
5. Pembuatan *prototype* aplikasi berbasis *web* dengan Bahasa pemrograman PHP.

b. Desain sistem

Tahap ini merupakan analisis dan perancangan sistem dengan menggunakan studi literatur dan mempelajari konsep aplikasi yang akan dibuat. Dengan melakukan analisa kebutuhan sistem sebagai solusi atas permasalahan yang dihadapi, dirumuskan rancangan sistem yang akan dibangun dan dapat menangani permasalahan. Kemudian dilakukan perancangan sistem dari hasil analisa terhadap sistem yang digambarkan dalam bentuk diagram untuk mepermudah gambaran rancangan sistem. Rancangan-rancangan tersebut antara lain:

1. Rancangan sistem basis data.
2. Rancangan aplikasi *scanner* untuk pengumpulan data sampel.

3. Rancangan aplikasi berbasis *web* untuk pengelolaan data *booth* dan promo.
4. Rancangan aplikasi perangkat bergerak sebagai aplikasi yang digunakan pengunjung untuk mengetahui informasi promo.

c. Implementasi

Pada tahap ini dilakukan pembuatan perangkat lunak berdasarkan rancangan yang telah dibuat pada proses sebelumnya. Rincian pada tahap ini sebagai berikut:

1. Implementasi rancangan basis data.
2. Implementasi rancangan aplikasi perangkat bergerak.
3. Implementasi rancangan aplikasi *web*.
4. Implementasi penggunaan *library* LibSVM untuk klasifikasi dengan algoritma SVM.
5. Implementasi *query* pada basis data.
6. Implementasi *web service* yang berfungsi untuk menangani komunikasi aplikasi perangkat bergerak dengan *server*.

d. Uji coba dan evaluasi

Pada tahap ini terdapat dua tahap pengujian dan evaluasi yang akan dilakukan:

1. Melakukan pengujian dan evaluasi terhadap tingkat akurasi hasil pendeteksian lokasi. Pengujian ini dilakukan untuk memberikan evaluasi dan rekomendasi penggunaan *Indoor Localization* menggunakan WiFi dan SVM untuk aplikasi perangkat bergerak berbasis lokasi dengan *Indoor Localization*.
2. Melakukan pengujian terhadap aplikasi yang dibuat menggunakan kasus yang telah disiapkan. Pengujian ini dilakukan untuk menguji fungsionalitas dari aplikasi, menganalisis masalah yang mungkin muncul, dan melakukan perbaikan yang mungkin dilakukan dan memberi saran untuk penelitian selanjutnya.

e. Penyusunan laporan tugas akhir

Tahap ini digunakan untuk membuat laporan Tugas Akhir yang berisi metode, dasar teori, dan hasil yang didapatkan selama pengerjaan Tugas Akhir.

1.6 Sistematika penyusunan laporan

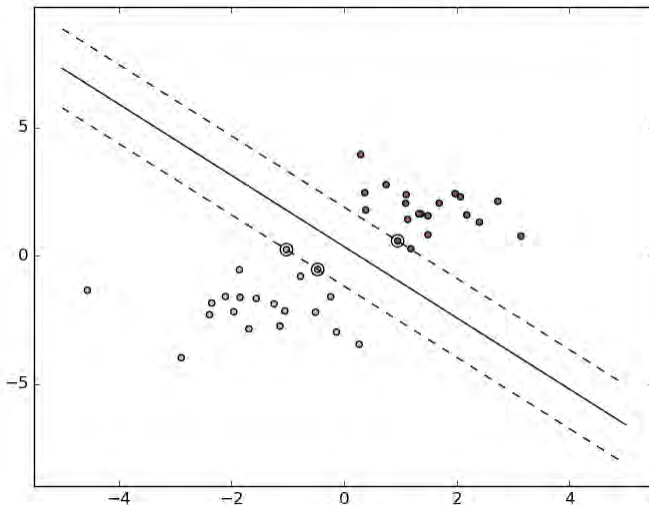
Buku Tugas Akhir ini disusun dengan sistematika laporan sebagai berikut:

1. Bab I Pendahuluan
Bab ini meliputi latar belakang masalah, rumusan permasalahan, batasan masalah, tujuan dan manfaat pembuatan Tugas Akhir, metodologi yang digunakan, dan sistematika penyusunan laporan Tugas Akhir.
2. Bab II Tinjauan Pustaka
Bab ini meliputi dasar teori dan penunjang yang berkaitan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.
3. Bab III Perancangan Perangkat Lunak
Bab ini membahas desain dari sistem yang akan dibuat meliputi arsitektur sistem, *use case* sistem, dan perancangan antarmuka sistem
4. Bab IV Implementasi Perangkat Lunak
Bab ini membahas implementasi dari desain sistem yang dilakukan pada tahap desain, meliputi *code* dan implementasi antarmuka dari perangkat lunak.
5. Bab V Uji Coba dan Evaluasi
Bab ini membahas uji coba dari perangkat lunak yang dibuat dengan melihat keluaran yang dihasilkan oleh perangkat lunak, analisis, dan evaluasi untuk mengetahui kemampuan perangkat lunak.
6. Bab VI Kesimpulan dan Saran
Bab ini berisi kesimpulan dari hasil uji coba yang dilakukan serta saran untuk pengembangan lebih lanjut perangkat lunak

BAB II TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan.

2.1 Support Vector Machine



Gambar 2.1. Hyper-plane pada SVM

Support Vector Machine (SVM) adalah sebuah teknik untuk klasifikasi data dan regresi. Merupakan *tools* untuk analisis statistik dan *machine learning* yang menunjukkan performa yang baik di banyak aplikasi dari klasifikasi dan regresi. SVM telah diaplikasikan secara luas di bidang sains, kedokteran, dan keahlian teknik dengan performa empiris yang sangat baik. *Support Vector Classification* (SVC) dan *Support Vector Regression* (SVR) telah berhasil digunakan dalam pengestimasi lokasi dan mendapatkan akurasi yang tinggi. Secara umum, algoritma SVM membuat *hyper-plane* atau himpunan *hyper-plane* pada *multi-dimensional*

space. Pemisahan yang baik adalah saat *hyper-plane* dan data latih terdekat dari sebuah kelas memiliki jarak paling besar atau biasa disebut *functional margin* ditunjukkan pada Gambar 2.1. Secara umum, semakin besar *functional margin*, maka kesalahan klasifikasi akan semakin kecil.

LibSVM adalah *library opensource* yang menunjang *Support Vector Classification, Regression, dan Distributed Estimation*. LibSVM juga mendukung *multi-class classification* dan menjadi ekstensi di berbagai bahasa, seperti Python, Java, Matlab, Ruby, hingga PHP. Terdapat beberapa langkah yang harus dilakukan untuk pengklasifikasian sinyal WiFi dengan menggunakan LibSVM [3]:

- Mengubah format data set menjadi format *software package* dari LibSVM.
- Mendapatkan model SVM dengan melatih data *training*.
- Tes dan prediksi data *testing* dengan model yang didapat.

2.2 Location Based Services

Location Based Services (LBS) atau layanan berbasis lokasi adalah informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network* untuk memanfaatkan lokasi dari *mobile device*. LBS memberikan kemungkinan komunikasi dan interaksi dua arah, sehingga pengguna bisa mendapatkan informasi referensi posisi pengguna dari penyedia layanan. Layanan berbasis lokasi dapat digambarkan sebagai pertemuan tiga teknologi yaitu: *Geographic Information System, Internet Service, dan Mobile Device*. Ada beberapa teknologi yang digunakan LBS untuk mendapatkan lokasi geografis dari sebuah perangkat *mobile*, diantaranya *Global Positioning System, Indoor Positioning System, dan Indoor Localization* [4].

2.2.1 Indoor localization system

Merupakan teknologi yang digunakan untuk menentukan lokasi sebuah objek yang berada di dalam gedung. Perbedaannya dengan *Indoor Positioning System* adalah *Indoor Localization* menentukan koordinat relatif suatu gedung [5], misal: Solaria,

ACE Hardware, Chatime. Sedangkan *Indoor Positioning System* menentukan koordinat global yang berupa *longitude* dan *latitude* sebuah lokasi.

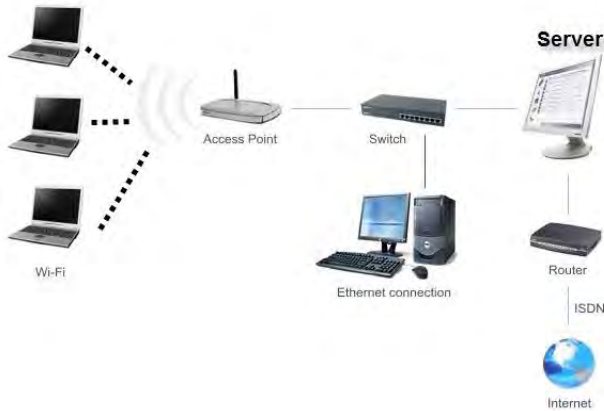
Ada dua fase yang untuk implementasi *Indoor Localization* pada tugas akhir ini, yaitu *offline training* dan *online localization*. Pada fase *offline training* dilakukan pembentukan model yang selanjutnya akan digunakan sebagai *training data set* pada pembentukan model SVM. Fase *online localization* adalah fase dimana sinyal WiFi yang diterima oleh *smartphone* diprediksi lokasinya dengan model SVM yang telah terbentuk pada fase *offline training* dengan bantuan metode *Binary Tree*.

2.2.2 Location based advertising

Location Based Advertising (LBA) adalah salah satu bentuk *Location Based Services* (LBS) yang dikhususkan pada bidang periklanan. Contoh penggunaan LBA yang konvensional adalah via SMS hingga yang populer saat ini yaitu *Cobeacon* yang menggunakan teknologi Bluetooth. LBA menawarkan kesempatan kepada para *advertiser* untuk memberikan informasi kepada masyarakat secara personal dan *real time* berdasarkan lokasinya. Cara ini disebut lebih efisien dan tepat sasaran.

2.3 Wireless Fidelity

WiFi merupakan singkatan dari *Wireless Fidelity*, yang memiliki pengertian yaitu sekumpulan standar yang digunakan untuk Jaringan Lokal Nirkabel (*Wireless Local Area Networks* – WLAN) yang didasari pada spesifikasi IEEE 802.11. Topologi WiFi dapat dilihat pada Gambar 2.2. Saat ini teknologi WiFi banyak dimanfaatkan untuk koneksi ke internet. Hal ini memungkinkan seseorang menggunakan komputer dengan kartu nirkabel (*Wireless card*) atau *personal digital assistant* (PDA) untuk terhubung dengan internet dengan menggunakan titik akses (atau dikenal dengan *hotspot*) terdekat [6]. Titik akses atau *hotspot* memiliki jangkauan sekitar 20 meter di dalam ruangan dan lebih luas lagi di dalam ruangan. Cakupan *hotspot* dapat mencakup wilayah dengan dinding.



Gambar 2.2. Topologi WiFi

2.4 CodeIgniter

Web Service adalah mekanisme terstandar yang menghubungkan sebuah aplikasi perangkat lunak dengan aplikasi perangkat lunak lain. *Web Service* menggunakan *JSON* untuk melakukan *encode* data melalui protokol *HTTP* sehingga hampir seluruh *platform* dapat menerima data yang ada dalam *web service*. Pada tugas akhir ini digunakan *web service* berbasis *PHP* dengan kerangka kerja *CodeIgniter*. *CodeIgniter* merupakan kerangka kerja *opensource* yang menggunakan arsitektur Model-View-Controller (*MVC*) dan dokumentasi yang lengkap sehingga mudah untuk dipelajari [7].

2.5 Android

Android merupakan salah satu perangkat lunak yang berjalan pada perangkat bergerak yang meliputi sistem operasi, *middleware*, dan aplikasi. *Android* menggunakan bahasa pemrograman *Java* dan *file resources* yang berupa *XML*. Kode *Java* yang terkompilasi dengan data dan *file resources* yang dibutuhkan aplikasi dan digabungkan oleh *aapt tools* menjadi paket *Android*. *File* tersebut ditandai dengan ekstensi *.apk*. *File* inilah yang didistribusikan sebagai aplikasi dan *install* pada perangkat bergerak.

BAB III

PERANCANGAN PERANGKAT LUNAK

Perancangan merupakan bagian penting dari pembuatan suatu perangkat lunak yang berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam Tugas Akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan proses, alur, dan implementasinya.

3.1 Analisis

Pada subbab berikut akan dijelaskan analisa pembuatan aplikasi perangkat bergerak dan *web*. Analisis yang dilakukan meliputi analisa permasalahan, kebutuhan umum perangkat lunak, deskripsi umum sistem, arsitektur dan kebutuhan fungsional sistem.

3.1.1 Analisis permasalahan

Advertising atau periklanan merupakan hal yang penting dalam sebuah bisnis untuk mempromosikan dan memasarkan barang atau jasa sebuah bisnis. Banyak media yang telah dimanfaatkan para pemilik bisnis untuk megiklankan produknya antara lain televisi, radio, pamflet, media cetak, dan internet tentunya dengan biaya yang tidak sedikit. Dari sisi konsumen, iklan dapat membantu konsumen untuk memperoleh informasi mengenai produk secara cepat dan mudah.

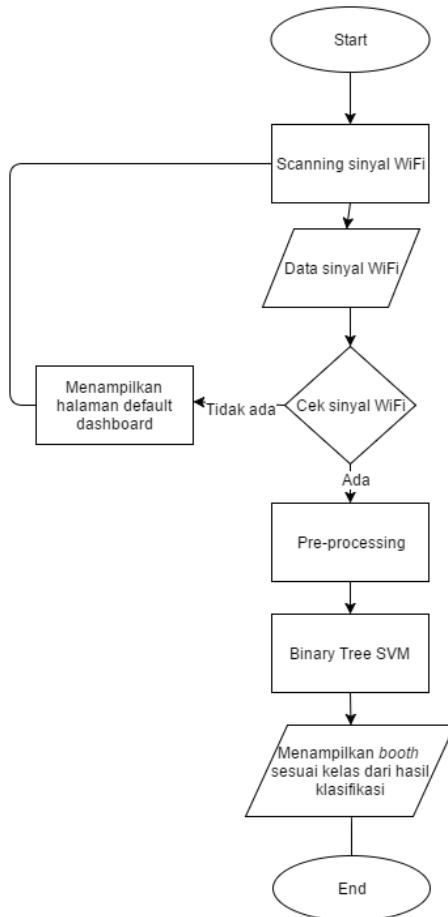
Saat ini sebagian besar masyarakat telah memiliki *smartphone* untuk melakukan komunikasi sehari-hari. Hal ini menjadi dasar penulis untuk membuat sebuah aplikasi berbasis Android untuk sarana promosi dan periklanan yang akan memprediksi lokasi pengunjung dalam suatu pusat perbelanjaan yang akan langsung memberikan informasi mengenai posisi saat ini dan promosi dari *booth* yang terdaftar di pusat perbelanjaan tersebut. Prediksi lokasi pengunjung akan menjadi dasar pemberian info promosi yang menggunakan konsep *Indoor Localization*.

3.1.2 Deskripsi umum perangkat lunak

Aplikasi yang dibangun pada Tugas Akhir ini terdiri dari dua aplikasi yaitu aplikasi yang berbasis *web* yang digunakan untuk pengelolaan data *booth* dan aplikasi perangkat bergerak berbasis sistem operasi Android yang digunakan untuk pengelolaan data promosi dan pemberian informasi promosi berdasarkan lokasi pengunjung. Aplikasi perangkat bergerak memanfaatkan teknologi WiFi pada perangkat bergerak untuk mengimplementasikan konsep *Indoor Localization* yang menjadi pendukung pada sistem ini. Aplikasi ini memiliki kemampuan untuk mendeteksi lokasi pengunjung di dalam ruangan dengan menggunakan data *Basic Service Set Identificatiom* (BSSID) dan kekuatan sinyal WiFi yang dapat ditangkap oleh perangkat bergerak saat berada di ruangan tersebut. Keluaran yang dihasilkan adalah nama *booth* atau lokasi keberadaan pengunjung. Untuk dapat mengetahui posisi, aplikasi akan melakukan *scanning* data BSSID dan kekuatan sinyal WiFi secara otomatis dari posisi pengunjung saat itu. Data hasil perekaman akan dikirim ke *server* untuk diolah. Pada *server* akan dilakukan proses *query* dari basis data dan prediksi dari model *Support Vector Machine* (SVM). Proses tersebut nantinya akan digunakan untuk mendeteksi lokasi keberadaan pengguna berdasarkan data yang telah dikirimkan aplikasi ke *server*.

Sebelumnya, untuk mendapatkan model SVM, dibutuhkan proses *sampling* data dengan melakukan 4 kali *scanning* WiFi di seluruh *train area* setiap ruangan. Setiap ruangan diambil 5 titik sampel secara acak per *scanning*. Setelah dilakukan proses *sampling*, seluruh data yang telah didapatkan disimpan ke dalam basis data untuk kemudian dilakukan klasifikasi dengan SVM. Cara kerja sistem secara umum ditunjukkan oleh Gambar 3.1. Fungsionalitas perangkat lunak aplikasi perangkat bergerak apabila diringkas adalah sebagai berikut:

1. Dapat menampilkan lokasi pengunjung
2. Menampilkan *booth* terdekat
3. Menampilkan detail lokasi
4. Menampilkan daftar promo



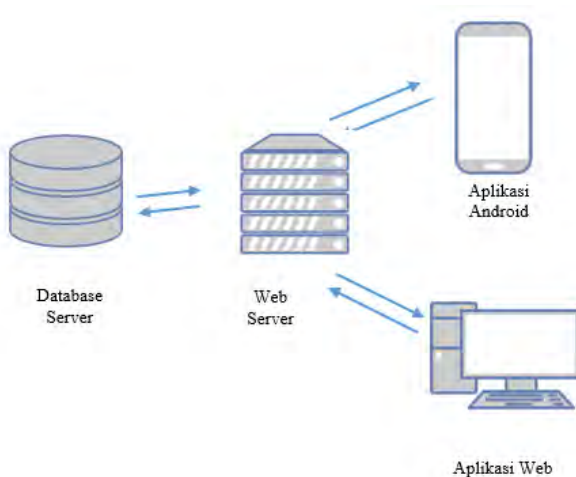
Gambar 3.1. Diagram Alur Sistem

Sedangkan untuk pengelolaan datanya menggunakan aplikasi berbasis *web* yang hanya digunakan oleh pemilik *booth* dan pihak manajemen pusat perbelanjaan sebagai administrator. Secara rinci fungsionalitas perangkat lunak aplikasi *web* adalah sebagai berikut:

1. Mengelola data promo
2. Mengelola data informasi *booth*

3.1.3 Arsitektur sistem

Berikut ini adalah arsitektur sistem dari aplikasi ditunjukkan pada Gambar 3.2:



Gambar 3.2. Arsitektur Sistem

Sistem ini terdiri dari dua aplikasi yaitu:

1. Aplikasi berbasis perangkat bergerak

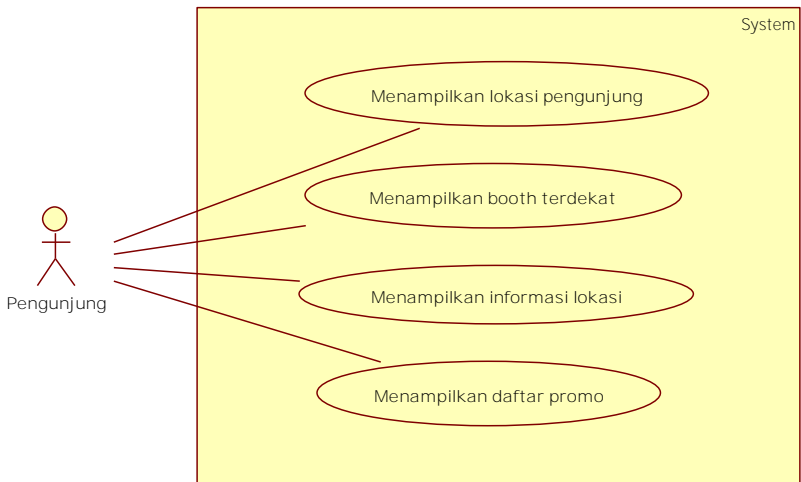
Merupakan aplikasi perangkat bergerak yang digunakan oleh pengunjung untuk mendeteksi lokasinya dan memberi informasi promo berdasarkan lokasinya. Penyimpanan data utama pada sistem ini menggunakan sistem manajemen basis data *PostgreSQL*. Sistem juga menyediakan *webservice* untuk komunikasi data aplikasi perangkat bergerak dengan basis data *server*.
2. Aplikasi berbasis *Web*

Merupakan aplikasi berbasis *web* yang digunakan untuk manajemen data dan dikelola oleh pemilik *booth* dan pihak manajemen pusat perbelanjaan sebagai administrator. Aplikasi *web* ini dibangun menggunakan kerangka kerja CodeIgniter PHP. Penyimpanan data utama pada sistem ini menggunakan sistem manajemen basis data *PostgreSQL*.

3.1.4 Kebutuhan fungsional aplikasi perangkat bergerak

Aplikasi berbasis perangkat bergerak digunakan untuk memberikan informasi promo berdasarkan lokasi pengunjung. Berikut daftar kebutuhan fungsionalitas dari aplikasi perangkat bergerak:

1. Menampilkan lokasi pengunjung
2. Menampilkan *booth* terdekat
3. Menampilkan detail lokasi
4. Menampilkan daftar promo



Gambar 3.3. Diagram Kasus Penggunaan Aplikasi Perangkat Bergerak

Kebutuhan fungsionalitas aplikasi berbasis perangkat bergerak digambarkan pada Gambar 3.3. Aplikasi perangkat bergerak digunakan oleh pengunjung. Pengunjung dapat melakukan pendeteksian lokasi, melihat *booth* terdekat, melihat informasi lokasi, dan melihat daftar promo sesuai lokasi pengunjung. Penjelasan lebih lengkap mengenai kasus penggunaan pada aplikasi berbasis perangkat bergerak berada di Tabel 3.1.

Tabel 3.1. Deskripsi Kasus Penggunaan Aplikasi Perangkat Bergerak

No	Kode	Nama	Keterangan
1	UC-A01	Menampilkan lokasi pengunjung	Merupakan fitur dimana pengguna dapat mengetahui lokasi keberadaannya di pusat perbelanjaan (Studi kasus: East Coast Center)
2	UC-A02	Menampilkan <i>booth</i> terdekat	Merupakan fitur dimana pengunjung dapat melihat daftar <i>booth</i> yang dekat dengan lokasinya
3	UC-A03	Menampilkan detail <i>booth</i>	Merupakan fitur dimana pengguna dapat melihat informasi <i>booth</i> dengan lebih detail beserta list promo yang sedang berlangsung
4	UC-A04	Menampilkan daftar promo	Merupakan fitur dimana pengguna dapat melihat daftar semua promosi yang sedang berlangsung

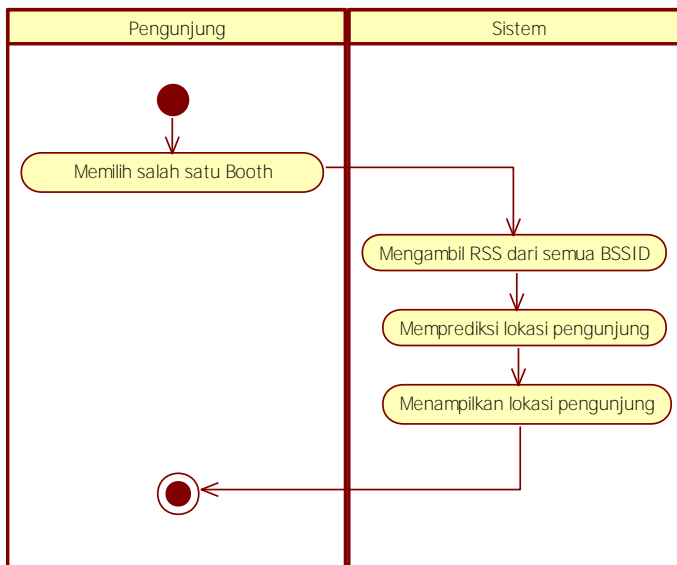
3.1.4.1 Deskripsi kasus kebutuhan UC-A01

Kasus kebutuhan kode UC-A01 merupakan kasus kebutuhan menampilkan lokasi pengunjung. Rincian alur kasus menampilkan lokasi pengguna dijelaskan pada Tabel 3.2 dan diagram aktivitas kasus dijelaskan pada Gambar 3.4.

Tabel 3.2. Rincian Alur Kasus Penggunaan UC-A01

Nama Use Case	Menampilkan lokasi pengunjung
Nomor	UC-A01
Aktor	Pengunjung
Kondisi Awal	Lokasi pengunjung belum terdeteksi

Kondisi Akhir	Lokasi pengunjung terdeteksi
Alur Normal	<ol style="list-style-type: none"> 1. Pengunjung masuk dalam sistem. 2. Sistem mengambil RSS dari semua BSSID. 3. Sistem memprediksi lokasi pengunjung. 4. Sistem menampilkan hasil kepada Pengunjung.



Gambar 3.4. Diagram Aktivitas Kasus Penggunaan UC-A01

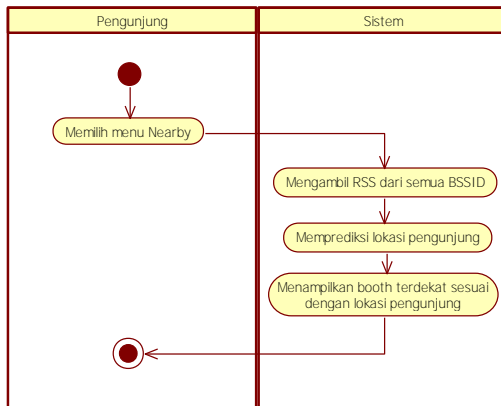
3.1.4.2 Deskripsi kasus kebutuhan UC-A02

Kasus kebutuhan kode UC-A02 merupakan kasus kebutuhan menampilkan *booth* terdekat. Rincian alur kasus menampilkan *booth* terdekat dijelaskan pada Tabel 3.3 dan diagram aktivitas kasus yang dijelaskan pada Gambar 3.5.

Tabel 3.3. Rincian Alur Kasus Penggunaan UC-A02

Nama Use Case	Menampilkan <i>booth</i> terdekat
----------------------	-----------------------------------

Nomor	UC-A02
Aktor	Pengunjung
Kondisi Awal	Daftar <i>booth</i> terdekat belum tampil
Kondisi Akhir	Daftar <i>booth</i> terdekat tampil
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu untuk melakukan melihat <i>booth</i> terdekat. 2. Sistem mengambil RSS dari semua BSSID. 3. Sistem memprediksi lokasi pengunjung. 4. Sistem menampilkan daftar <i>booth</i> terdekat dari lokasi pengunjung. 5. Kasus penggunaan menampilkan <i>booth</i> terdekat.



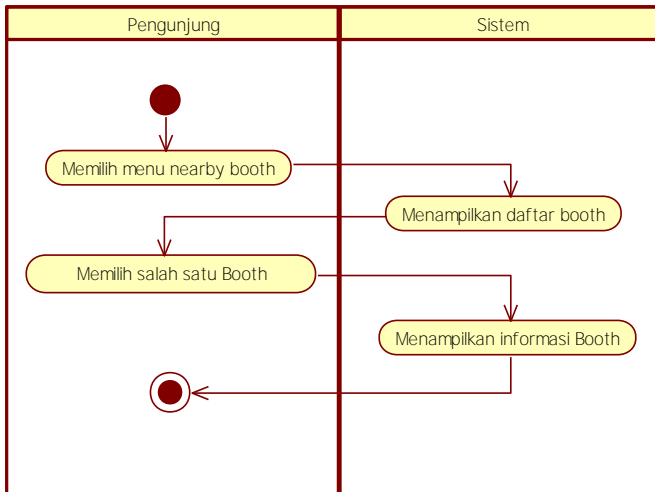
Gambar 3.5. Diagram Aktivitas Kasus Penggunaan UC-A02

3.1.4.3 Deskripsi kasus kebutuhan UC-A03

Kasus kebutuhan kode UC-A03 merupakan kasus kebutuhan menampilkan detail *booth*. Rincian alur kasus menampilkan detail *booth* dijelaskan pada Tabel 3.4 dan diagram aktivitas kasus yang dijelaskan pada Gambar 3.6.

Tabel 3.4. Rincian Alur Kasus Penggunaan UC-A03

Nama Use Case	Menampilkan Detail <i>Booth</i>
Nomor	UC-A03
Aktor	Pengunjung
Kondisi Awal	Detail <i>booth</i> belum tampil
Kondisi Akhir	Detail <i>booth</i> tampil
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu <i>Booth</i>. 2. Sistem menampilkan daftar <i>Booth</i>. 3. Pengguna memilih salah satu <i>booth</i>. 4. Sistem menampilkan detail <i>booth</i> sesuai dengan <i>booth</i> yang dipilih. 5. Kasus melihat detail <i>booth</i> berakhir.

**Gambar 3.6. Diagram Aktivitas Kasus Penggunaan UC-A03**

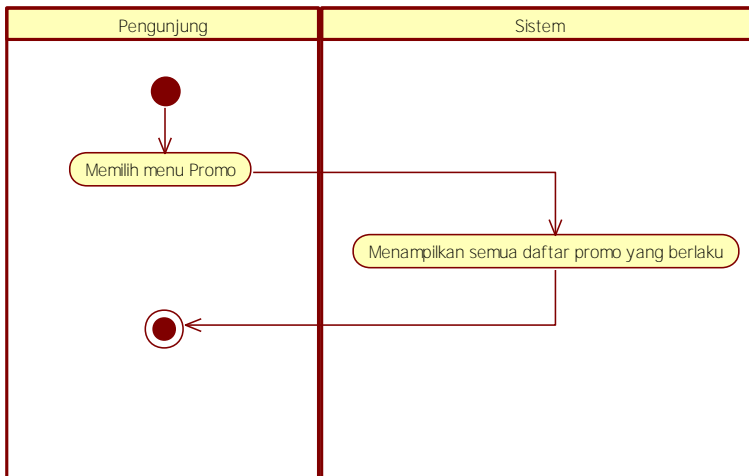
3.1.4.4 Deskripsi kasus kebutuhan UC-A04

Kasus kebutuhan kode UC-A04 merupakan kasus kebutuhan menampilkan daftar promo. Pada kasus kebutuhan ini akan ditampilkan seluruh data promo sesuai dengan *booth* pada saat *login*. Rincian alur kasus menampilkan daftar promo dijelaskan

pada Tabel 3.5 dan diagram aktivitas kasus yang dijelaskan pada Gambar 3.7

Tabel 3.5. Rincian Alur Kasus Penggunaan UC-A04

Nama Use Case	Menampilkan Daftar Promo
Nomor	UC-A04
Aktor	Pengunjung
Kondisi Awal	Daftar promo belum tampil
Kondisi Akhir	Daftar promo tampil
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu Promo. 2. Sistem menampilkan semua daftar Promo yang berlaku. 3. Kasus melihat daftar promo berakhir.



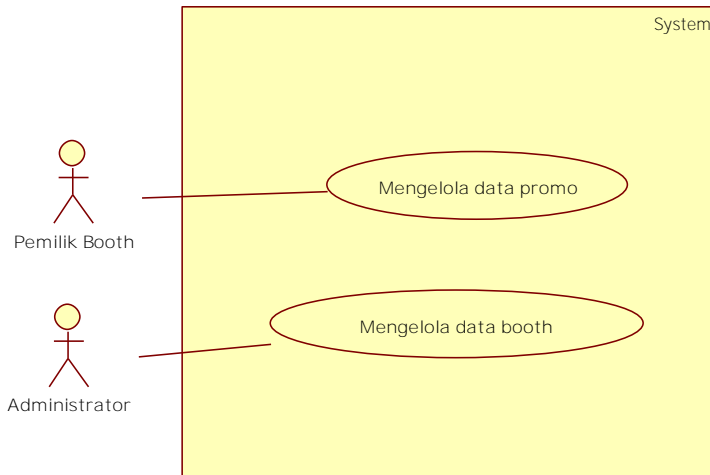
Gambar 3.7. Diagram Aktivitas Kasus Penggunaan UC-A04

3.1.5 Kebutuhan fungsional aplikasi web

Aplikasi berbasis *web* digunakan untuk menangani proses pengelolaan data *booth* dan promo. Berikut daftar kebutuhan fungsionalitas dari aplikasi *web*:

1. Mengelola data promo

2. Mengelola data *booth*
 Kebutuhan fungsionalitas aplikasi berbasis *Web*
 digambarkan pada Gambar 3.8.



Gambar 3.8. Diagram Kasus Penggunaan Aplikasi Web

Aplikasi *web* digunakan oleh Pemilik *booth* dan Administrator untuk melakukan pengelolaan data. Pemilik *booth* dapat melakukan pengelolaan data promo. Selain itu, Administrator juga dapat melakukan pengelolaan pada data *booth*. Perbedaan dari pengelolaan promo dengan pengelolaan data *booth* adalah pengelolaan data *booth* tidak memiliki fitur penghapusan data. Penjelasan lengkap mengenai kasus penggunaan pada aplikasi berbasis *web* berada di Tabel 3.6.

Tabel 3.6. Deskripsi Kasus Penggunaan Aplikasi Web

No	Kode	Nama	Keterangan
1	UC-B01	Mengelola Data Promo	Merupakan fitur dimana pengguna dapat melihat, menambah serta menghapus data promo.

2	UC-B02	Mengelola Data <i>Booth</i>	Merupakan fitur dimana pengguna dapat melihat, menambah, mengubah, serta menghapus data <i>booth</i> .
---	--------	-----------------------------	--

3.1.5.1 Deskripsi kasus kebutuhan UC-B01

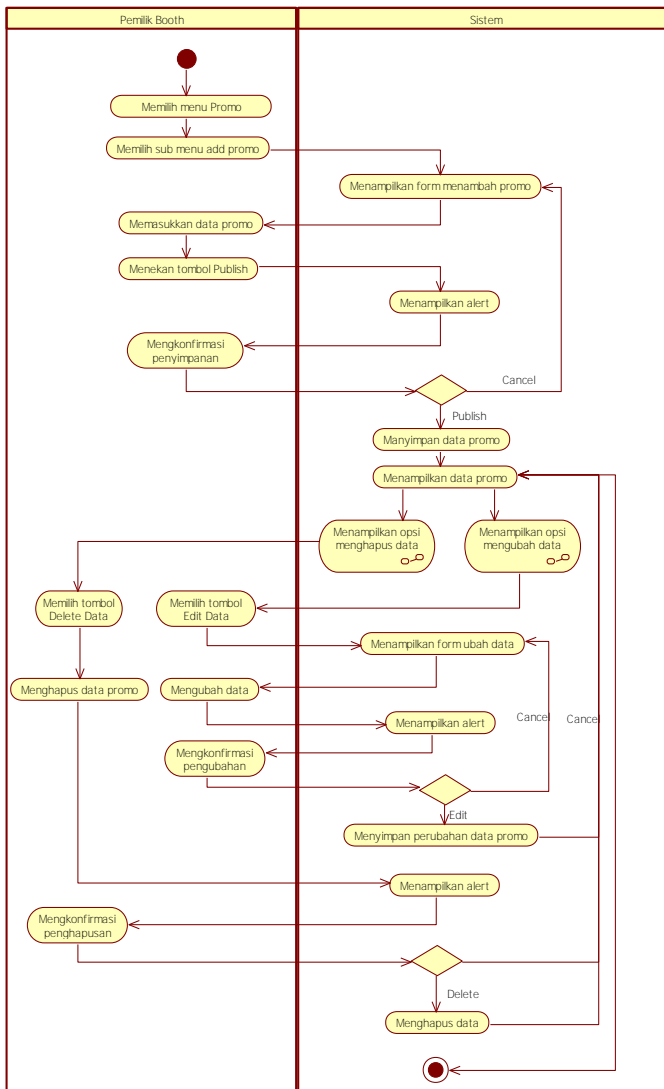
Kasus kebutuhan kode UC-B01 merupakan kasus kebutuhan mengelola data promo. Pada kasus kebutuhan mengelola data promo, ada empat proses utama yaitu menambah data promo, mengubah data promo, menghapus data promo, dan melihat seluruh data promo. Rincian alur kasus mengelola data promo dijelaskan pada Tabel 3.7 dan diagram aktivitas kasus yang dijelaskan pada Gambar 3.9.

Tabel 3.7. Rincian Alur Kasus Penggunaan UC-B01

Nama Use Case	Mengelola Data Promo
Nomor	UC-B01
Aktor	Pemilik <i>Booth</i>
Kondisi Awal	Data promo belum terisi dan form isian belum tampil
Kondisi Akhir	Informasi data promo tersimpan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu Promo. 2. Pengguna memilih sub menu add promo 3. Sistem menampilkan form menambah promo 4. Pengguna memasukkan data promo 5. Pengguna menekan tombol <i>publish</i> 6. Sistem menampilkan alert konfirmasi 7. Pengguna melakukan konfirmasi <ol style="list-style-type: none"> A1. Pengguna membatalkan penambahan 8. Sistem menyimpan data promo

	<ol style="list-style-type: none"> 9. Sistem menampilkan semua daftar Promo yang berlaku. 10. Pengguna melakukan penghapusan data promo 11. Sistem menampilkan alert konfirmasi penghapusan 12. Pengguna memberi konfirmasi penghapusan A2. Pengguna membatalkan Penghapusan 13. Sistem mengeksekusi penghapusan data 14. Sistem menampilkan semua daftar promo yang berlaku 15. Pengguna melakukan perubahan data 16. Sistem menampilkan form perubahan data 17. Pengguna mengisi form perubahan data 18. Pengguna menyimpan hasil pengisian perubahan data 19. Sistem menampilkan alert konfirmasi perubahan data 20. Pengguna mengkonfirmasi perubahan data A3. Pengguna membatalkan perubahan data 21. Sistem menyimpan perubahan data promo 22. Sistem menampilkan semua daftar promo yang berlaku 23. Kasus penggunaan mengelola data promo berakhir
Alur Alternatif	<ol style="list-style-type: none"> A1. Pengguna membatalkan penambahan A1.1. Kembali ke alur normal nomor 3

	A2. Pengguna membatalkan penghapusan A2.1. Kembali ke alur normal nomor 9 A3. Pengguna membatalkan perubahan A3.1. Kembali ke alur normal nomor 16
--	---



Gambar 3.9. Diagram Aktivitas Kasus Penggunaan UC-B01

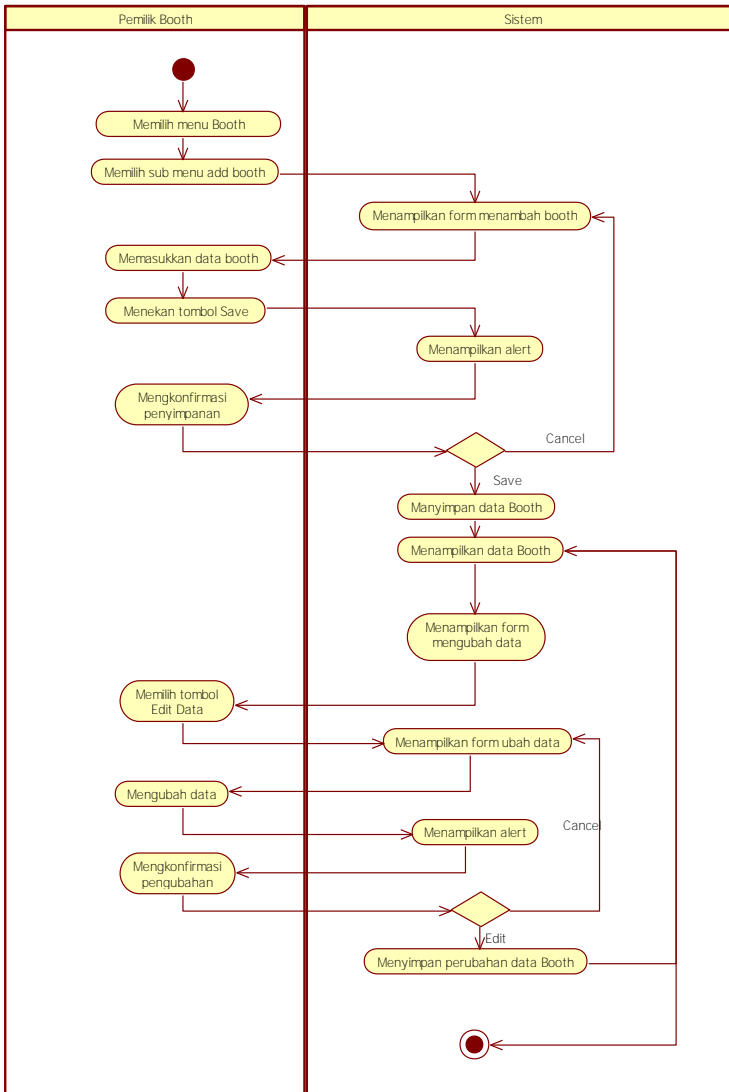
3.1.5.2 Deskripsi kasus kebutuhan UC-B02

Kasus kebutuhan kode UC-B02 merupakan kasus kebutuhan mengelola data *booth*. Pada kasus kebutuhan mengelola data *booth* ada tiga proses utama, yaitu menambah data *booth*, mengubah data *booth*, dan melihat seluruh *booth*. Rincian alur kasus mengelola data *booth* dijelaskan pada Tabel 3.8 dan diagram aktivitas kasus yang dijelaskan pada Gambar 3.10.

Tabel 3.8. Rincian Alur Kasus Penggunaan UC-B02

Nama Use Case	Mengelola Data <i>Booth</i>
Nomor	UC-B02
Aktor	Administrator
Kondisi Awal	Data <i>booth</i> belum terisi dan form isian belum tampil
Kondisi Akhir	Informasi data <i>booth</i> tersimpan
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna memilih menu Promo. 2. Pengguna memilih sub menu add promo 3. Sistem menampilkan form menambah promo 4. Pengguna memasukkan data promo 5. Pengguna menekan tombol <i>publish</i> 6. Sistem menampilkan alert konfirmasi 7. Pengguna melakukan konfirmasi <ol style="list-style-type: none"> A1. Pengguna membatalkan penambahan 8. Sistem menyimpan data promo 9. Sistem menampilkan semua daftar Promo yang berlaku 10. Pengguna melakukan pengubahan data 11. Sistem menampilkan form pengubahan data 12. Pengguna mengisi form pengubahan data 13. Pengguna menyimpan hasil pengisian pengubahan data

	<p>14. Sistem menampilkan alert konfirmasi perubahan data</p> <p>15. Pengguna mengkonfirmasi perubahan data</p> <p>A2. Pengguna membatalkan perubahan data</p> <p>16. Sistem menyimpan perubahan data promo</p> <p>17. Sistem menampilkan semua daftar promo yang berlaku</p> <p>18. Kasus penggunaan mengelola data promo berakhir</p>
Alur Alternatif	<p>A1. Pengguna membatalkan penambahan</p> <p>A1.1. Kembali ke alur normal nomor 3</p> <p>A2. Pengguna membatalkan perubahan</p> <p>A2.1. Kembali ke alur normal nomor 11</p>



Gambar 3.10. Diagram Aktivitas Kasus Penggunaan UC-B02

3.2 Perancangan

Subbab berikut membahas tentang perancangan dari aplikasi Android dan *web*. Subbab berikut membahas rancangan konsep *Location Based Advertisement*, *Indoor Localization*, rancangan antarmuka, rancangan proses dan rancangan basis data. Pembahasan lebih detail akan dibahas berikut ini.

3.2.1 Perancangan konsep indoor localization

Pada subbab perancangan konsep *Indoor Localization* akan dipaparkan mengenai bagaimana konsep *Indoor Localization* akan diimplementasikan untuk mendukung sistem yang dibangun. Hal-hal yang akan dibahas pada rancangan konsep *Indoor Localization* adalah perancangan data serta perancangan algoritma. Berikut penjelasan detail mengenai perancangan diatas.

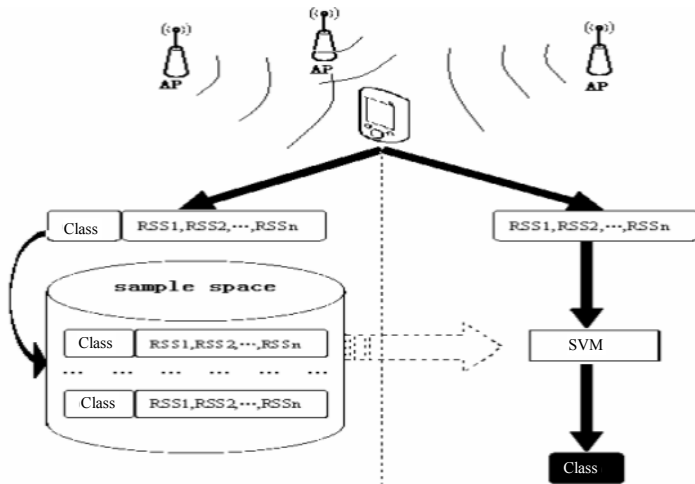
3.2.1.1 Perancangan data

Perancangan data adalah hal yang penting dalam sistem karena diperlukan data yang tepat agar sistem dapat beroperasi dengan benar. Sistem yang dibuat membutuhkan data masukan dan akan memberikan data keluaran. Data masukan berupa himpunan data yang akan digunakan sebagai data latih. Data keluaran dari sistem ini adalah hasil prediksi lokasi pengguna.

- **Data pelatihan SVM**

Data pelatihan SVM ini terdiri dari data masukan dan data keluaran. Himpunan data yang digunakan sebagai data masukan adalah data yang telah dikumpulkan dari hasil *scanning* WiFi pada beberapa *booth* di East Coast Center. Data keluaran dari tahap pelatihan adalah model SVM yang nantinya akan digunakan dalam prediksi lokasi.

Kelas dalam himpunan data adalah representasi dari id *booth* tersebut, sedangkan *feature* dalam himpunan data adalah BSSID sedangkan *signal strength* sebagai *value*. Sebelum dapat dilakukan proses pelatihan, data latih harus melewati tahap *pre-processing* terlebih dahulu untuk mengubah format data menjadi format LibSVM. Proses prediksi lokasi dari tahap pengumpulan data set dijelaskan pada Gambar 3.11.



Gambar 3.11. Proses Prediksi Lokasi

- **Data sistem pendeteksian lokasi**

Data sistem pendeteksian lokasi ini terdiri dari data masukan dan data keluaran. Data masukan pada sistem pendeteksian lokasi ini berupa data sinyal WiFi berupa BSSID dan kekuatan sinyal yang diperoleh dari dalam sebuah ruangan atau lokasi dimana pengguna berada. Sedangkan data keluaran pada proses ini berupa nama ruang dari keberadaan pengguna.

3.2.1.2 Perancangan algoritma

Perancangan proses merupakan tahap untuk membentuk alur dalam penerapan algoritma SVM yang akan digunakan sebagai metode pendeteksian lokasi.

- **Tahap pre-processing data testing**

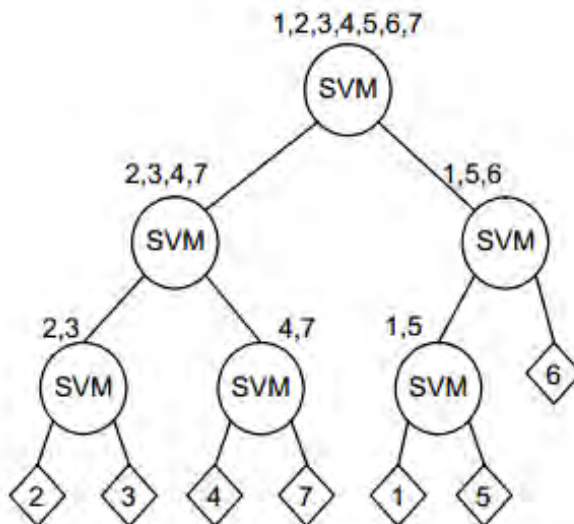
Hasil *scanning* data sinyal-sinyal WiFi yang dikirim oleh *smartphone* kemudian akan dilakukan pengecekan apakah BSSID sinyal yang masuk telah dikenali oleh sistem atau tidak dengan cara melakukan *query*. Hasil dari *query* ini kemudian akan dibentuk menjadi format *array* input LibSVM.

- **Tahap pre-processing data pelatihan**

Tahap pertama dari *pre-processing* data pelatihan adalah melakukan *query* data-data yang dibutuhkan, yaitu *BOOTHID*, *APID*, *SIGNALSTRENGTH*, dan *POSITIONID*. *APID* merupakan representasi dari *BSSID* dalam bentuk integer. Kemudian seluruh hasil dari *query* diubah menjadi format array input LibSVM.

- **Tahap pembentukan binary tree**

SVM pada awalnya dikembangkan untuk memecahkan masalah *binary-decision*, oleh karena itu pada permasalahan *multi-class classification* ini, data akan dipecah menjadi *problem binary* dengan *Binary Tree* seperti yang dijelaskan pada Gambar 3.12.



Gambar 3.12. Binary Tree SVM [8]

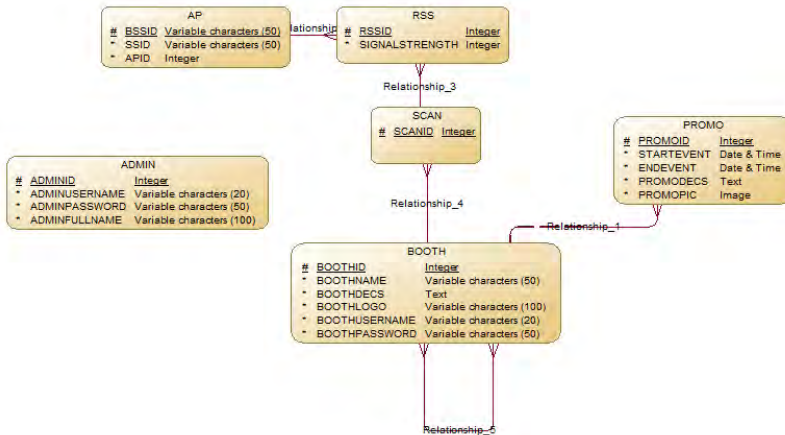
Kelas yang merepresentasikan *booth* akan dibagi menjadi *left class*, *middle class*, dan *right class* untuk agar terbentuk *Binary Tree*. Diasumsikan kelas yang berurutan adalah kelas yang berdekatan.

- **Tahap pelatihan dan pendeteksian lokasi**

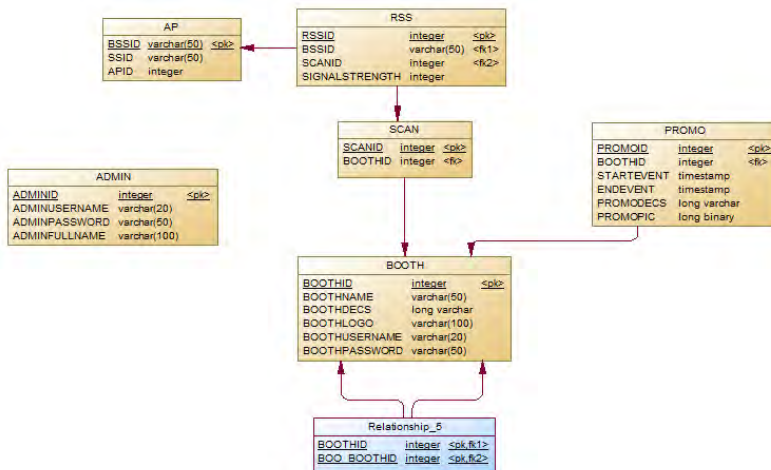
Tahap pelatihan akan selalu dilakukan setiap proses pendeteksian lokasi. Setiap kelas akan selalu dibagi menjadi dua bagian kiri dengan label -1 dan kanan dengan label 1 yang berasal dari klasifikasi kelas sebelumnya. Proses tersebut diulang hingga menghasilkan satu kelas pada *node* paling bawah yang merepresentasikan id dari suatu *booth*.

3.2.2 Perancangan basis data

Pada subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada aplikasi perangkat bergerak maupun *web*. Perancangan basis data dibedakan per tabel dan diberi penjelasan pada masing-masing tabelnya. Basis data pada sistem yang ini menggunakan sistem manajemen basis data relasional PostgreSQL. *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) dari basis data sistem ini dapat dilihat pada Gambar 3.13 dan Gambar 3.14.



Gambar 3.13. Conceptual Data Model



Gambar 3.14. Physical Data Model

3.2.2.1 Rancangan tabel AP

Tabel AP digunakan untuk menyimpan data *access point* hasil dari proses *scanning*. Tabel AP menyimpan data yang akan digunakan untuk penyimpanan data pelatihan yang berisi BSSID, SSID, dan APID yang merupakan representasi integer dari BSSID. Detail tabel AP dijelaskan pada Tabel 3.9.

Tabel 3.9. Atribut Tabel AP

Nama Kolom	Tipe Data	Keterangan
BSSID	Varchar(50)	BSSID dari <i>access point</i> yang diterima. <i>Primary Key</i> dari tabel AP
SSID	Varchar(50)	SSID dari BSSID yang diterima
APID	Integer	ID unik pada tabel AP

3.2.2.2 Rancangan tabel RSS

Tabel RSS digunakan untuk menyimpan data RSS yang diterima, karena satu BSSID bisa memiliki RSS yang berbeda setiap *scanning*. Detail tabel RSS dijelaskan pada Tabel 3.10.

Tabel 3.10. Atribut Tabel RSS

Nama Kolom	Tipe Data	Keterangan
RSSID	Integer	<i>Primary Key</i> dari tabel RSS
BSSID	Varchar(50)	BSSID dari <i>signal strength</i> yang disimpan
SCANID	Integer	<i>Foreign Key</i> dari tabel SCAN
SIGNALSTRENGTH	Integer	<i>Signal strength</i> yang diterima setiap <i>scanning</i>

3.2.2.3 Rancangan tabel Position

Tabel Position digunakan untuk menyimpan koordinat titik ruangan atau lokasi sekaligus menjadi id unik *scanning*. Detail tabel position dijelaskan pada Tabel 3.11.

Tabel 3.11. Atribut Tabel Position

Nama Kolom	Tipe Data	Keterangan
SCANID	Int	<i>Primary Key</i> dari tabel SCAN
<i>BOOTHID</i>	Int	<i>Foreign key</i> dari tabel <i>BOOTH</i>

3.2.2.4 Rancangan tabel Booth

Tabel matakuliah digunakan untuk menyimpan daftar semua *booth* yang ada di East Coast Center beserta *username* dan *password* untuk login Pemilik *booth*. Detail tabel *booth* dijelaskan pada Tabel 3.12.

Tabel 3.12. Atribut Tabel Booth

Nama Kolom	Tipe Data	Keterangan
<i>BOOTHID</i>	Int	<i>Primary Key</i> dari tabel <i>BOOTH</i>
<i>BOOTHNAME</i>	Varchar(50)	Nama <i>booth</i>
<i>BOOTHDESC</i>	Text	Deskripsi dari <i>booth</i>
<i>BOOTHLOGO</i>	Varchar (100)	Path dari logo <i>booth</i>
<i>BOOTHUSERNAME</i>	Varchar(20)	Username <i>booth</i>
<i>BOOTHPASSWORD</i>	Varchar(50)	Password <i>booth</i>

3.2.2.5 Rancangan tabel Nearby

Tabel nearby merupakan tabel hasil *many to many* dari tabel *booth*. Tabel nearby menyimpan *booth* mana saja yang berada di dekat suatu *booth*. Detail tabel nearby dijelaskan pada Tabel 3.13.

Tabel 3.13. Atribut Tabel Nearby

Nama Kolom	Tipe Data	Keterangan
<i>BOOTHID</i>	Int	<i>Foreign key</i> dari tabel <i>BOOTH</i>
<i>BOO_BOOTHID</i>	Int	Id <i>booth</i> nearby

3.2.2.6 Rancangan tabel Promo

Tabel promo digunakan untuk menyimpan data daftar semua promo yang dimiliki oleh *booth*. Detail tabel promo dijelaskan pada Tabel 3.14.

Tabel 3.14. Atribut Tabel Promo

Nama Kolom	Tipe Data	Keterangan
PROMOID	Int	<i>Primary Key</i> dari tabel <i>PROMO</i>
<i>BOOTHID</i>	Int	<i>Foreign key</i> dari tabel <i>BOOTH</i>

STARTEVENT	Timestamp	Waktu mulai promo berlaku
ENDEVENT	Timestamp	Waktu selesai promo berlaku
PROMODESC	Text	Deskripsi atau keterangan dari sebuah promo
PROMOPIC	Varchar	Path dari poster promo

3.2.2.7 Rancangan tabel Admin

Tabel admin digunakan untuk menyimpan akun admin yang terdaftar. Detail tabel admin dijelaskan pada Tabel 3.15.

Tabel 3.15. Atribut Tabel Admin

Nama Kolom	Tipe Data	Keterangan
ADMINID	Int	<i>Primary Key</i> dari tabel ADMIN
ADMINUSERNAME	Varchar(20)	Username dari admin
ADMINPASSWORD	Varchar(50)	Password dari admin
ADMINFULLNAME	Varchar(100)	Nama lengkap admin

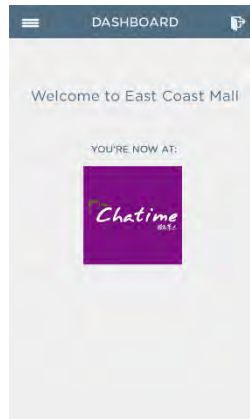
3.2.3 Perancangan antarmuka aplikasi perangkat bergerak

Pada subbab ini akan dibahas secara mendetail dari rancangan antarmuka aplikasi perangkat bergerak.

3.2.3.1 Antarmuka halaman dashboard

Perancangan antarmuka *dashboard* ditunjukkan pada Gambar 3.15. Terdapat *ImageView* yang akan menampilkan logo *booth* dimana pengguna sedang berada. Antarmuka *dashboard* merupakan antarmuka pertama saat aplikasi dibuka. Saat aplikasi dibuka, semua BSSID dan *signal strength* yang ditangkap oleh *smartphone* akan dikirim ke *server* untuk memprediksi lokasi pengguna, lalu *server* akan memberikan keluaran berupa data

booth dimana pengguna berada dan aplikasi akan menampilkan logo dari *booth* tersebut di halaman *dashboard*.



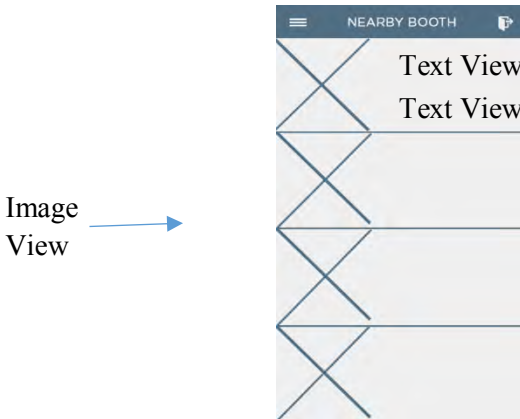
Gambar 3.15. Rancangan Antarmuka Halaman Dashboard

3.2.3.2 Antarmuka halaman nearby

Gambar 3.16 merupakan gambar rancangan antarmuka halaman Nearby. Halaman ini akan terbuka ketika pengguna memilih menu nearby pada *sidebar* menu dan akan menampilkan semua *booth* yang terdekat dengan lokasi pengguna. Halaman Nearby sangat berkaitan dengan apa yang ditampilkan pada halaman *dashboard*.

Antarmuka halaman Nearby dibuat dengan Recycler View agar dapat menampilkan daftar *booth*. Komponen-Komponen yang ada dalam Recycler View halaman Nearby adalah sebagai berikut:

1. Image View digunakan untuk menampilkan logo *booth*.
2. Text View digunakan untuk menampilkan nama *booth*.
3. Text View digunakan untuk menampilkan deskripsi singkat *booth*.



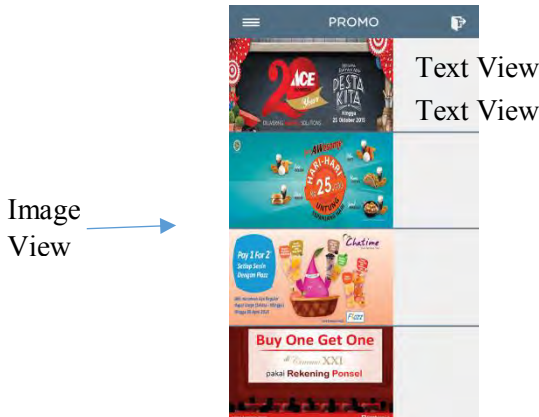
Gambar 3.16. Rancangan Antarmuka Halaman Nearby

3.2.3.3 Antarmuka halaman promo

Gambar 3.17 merupakan gambar rancangan antarmuka promo. Halaman ini baru akan terbuka ketika pengguna memilih menu Promo pada *side bar* menu. Halaman ini akan menampilkan seluruh daftar promo yang masih berlaku di semua *booth*. Pengguna dapat memilih salah satu dari daftar promo untuk melihat poster promo dengan ukuran lebih besar.

Antarmuka halaman promo dibuat dengan Recycler View sehingga dapat menampilkan daftar promo agar mudah dibaca. Komponen yang ada dalam RecyclerView halaman promo adalah sebagai berikut:

1. ImageView digunakan untuk menampilkan gambar poster promo.
2. TextView digunakan untuk menampilkan tanggal dimulainya promo.
3. TextView digunakan untuk menampilkan tanggal berakhirnya promo.
4. Text View digunakan untuk menampilkan deskripsi singkat dari promo.



Gambar 3.17. Rancangan Antarmuka Halaman Promo

3.2.4 Perancangan antarmuka aplikasi web

Pada subbab ini akan dibahas secara mendetail dari rancangan antarmuka aplikasi *web*.

3.2.4.1 Antarmuka halaman login

Gambar 3.18 merupakan gambar rancangan antarmuka pertama yang akan muncul ketika aplikasi *web* diakses. Pada halaman login ini, pemilik *booth* dan administrator sebagai pengguna diminta untuk mengisi data untuk diautentikasi agar dapat masuk ke halaman sesuai dengan hak aksesnya. Pemilik *booth* hanya memiliki akses untuk pengelolaan data promo saja, sedangkan administrator dapat mengelola data *booth* serta data-data administrator itu sendiri.

Berikut penjelasan komponen-komponen yang terdapat pada gambar:

1. *Form* berisi username dan password serta hak akses.
2. *Button login* untuk melanjutkan ke halaman berikutnya apabila autentikasi berhasil.
3. *Alert* apabila gagal dalam proses autentikasi.

EAST COAST CENTER

Username:

Password:

Access:

Administrator

Booth

Gambar 3.18. Rancangan Antarmuka Halaman Login

3.2.4.2 Antarmuka halaman melihat data

Gambar 3.19 merupakan gambar rancangan antarmuka melihat data. Proses melihat data digunakan pada saat melihat data *booth*, melihat data administrator, dan melihat data promo.

Berikut penjelasan masing-masing komponen yang terdapat pada gambar:

1. *Button add* digunakan untuk navigasi ke halaman menambah data.
2. Tabel untuk menampilkan data-data.

Tue, 15 March 2016

Promo ▾ **Promo Management**

▼ Promo Id	▼ Promo Description	▼ Promo Pic	▼ Start Event	▼ End Event	▼ Edit	▼ Delete
1	Promo 1 Desc	http://urltopath/booth/promo1.png	12-01-2016	16-01-2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Promo 2 Desc	http://urltopath/booth/promo2.png	29-01-2016	02-02-2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gambar 3.19. Rancangan Antarmuka Halaman Melihat Data

3.2.4.3 Antarmuka halaman menambah data

Gambar 3.20 merupakan gambar rancangan antarmuka menambah data. Antarmuka menambah data berupa form, digunakan dalam proses menambah data *booth*, data promo, dan data administrator.

Berikut penjelasan masing-masing komponen yang terdapat pada gambar:

1. *Form* digunakan untuk mengisi data baru.
2. *Button save* untuk menyimpan data yang telah diisi.

Gambar 3.20. Rancangan Antarmuka Halaman Menambah Data

3.2.4.4 Antarmuka halaman mengubah data

Gambar 3.21 merupakan gambar rancangan antarmuka mengubah data. Antarmuka mengubah data digunakan dalam proses mengubah data promo, data *booth*, dan data administrator.

Berikut penjelasan masing-masing komponen yang terdapat pada gambar:

1. *Form* yang telah terisi data sebelumnya, isi form dapat diubah sesuai kebutuhan.
2. *Button edit* untuk menyimpan perubahan yang dilakukan pada data.
3. *Alert* untuk memperingatkan apakah benar data akan diubah.

Tue, 15 March 2016

Promo Management

Promo 1's Description

4/22/2012

4/22/2012

Upload Promo Pic

EDIT

Gambar 3.21. Rancangan Antarmuka Halaman Mengubah Data

3.2.4.5 Antarmuka Halaman Menghapus Data

Gambar 3.22 merupakan gambar rancangan antarmuka menghapus data. Berikut penjelasan masing-masing komponen pada gambar:

1. Tabel berisi seluruh *record* data.
2. *Button delete* pada setiap *record* data untuk mengeksekusi fungsi menghapus data.
3. *Alert* peringatan penghapusan data.

Tue, 15 March 2016

Promo Management

ADD PROMO

Promo Id	Promo Description	Are you sure to delete this Promo?	Start Event	End Event	Edit	Delete
1	Promo 1 Desc	http://antopath/booth/promo1.png	12-01-2016	16-01-2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Promo 2 Desc	http://antopath/booth/promo2.png	29-01-2016	02-02-2016	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gambar 3.22. Rancangan Antarmuka Halaman Menghapus Data

3.2.5 Perancangan proses aplikasi perangkat bergerak

Pada subbab ini akan dijelaskan rancangan proses aplikasi dari perangkat lunak untuk memenuhi kebutuhan fungsional.

3.2.5.1 Proses mendeteksi dan menampilkan lokasi

Proses mendeteksi dan menampilkan lokasi dilakukan ketika aplikasi pertamakali dijalankan. Pertama *smartphone* akan menangkap semua sinyal wifi beserta kekuatan sinyalnya kemudian dikirimkan ke *server* dengan *httprequest*. Proses pendeteksian dilakukan di sisi *server* dengan perhitungan dengan *svm* beserta *pre-processing*nya. Hasil dari proses perhitungan kemudian akan menjadi *response* untuk ditampilkan pada halaman dashboard.

3.2.5.2 Proses menampilkan detail booth

Proses menampilkan detail *booth* dilakukan ketika pengguna memilih salah satu *booth* pada menu *nearby booth* atau *booth* yang muncul pada halaman dashboard. Pada proses ini akan terjadi *request* detail data *booth* berdasarkan id *booth*, setelah data id *booth* dikirim ke *server*, dilakukan *query* untuk mendapatkan *response* detail *booth*.

3.2.5.3 Proses menampilkan booth terdekat

Proses menampilkan *booth* terdekat dilakukan ketika pengguna memilih menu *nearby booth*, pada saat memilih menu dilakukan *httprequest* sesuai dengan data id *booth* dari proses pendeteksian lokasi. Hasil *response* dari server berupa *booth* yang terdekat dari lokasi *booth* saat itu.

3.2.5.4 Proses menampilkan daftar promo

Proses menampilkan data promo terjadi ketika pengguna memilih menu promo, ketika memilih *httprequest* akan dijalankan dan *server* merespon dengan mengirimkan *response* berupa seluruh data promo yang sedang berlaku.

3.2.6 Perancangan proses aplikasi web

Pada subbab ini akan dibahas secara mendetail dari rancangan proses aplikasi *web* untuk memenuhi kebutuhan fungsionalnya.

3.2.6.1 Proses melihat data

Proses melihat data ini dapat dilakukan pada data *booth* dan promo. Proses ini dilakukan oleh administrator dan pemilik *booth*, namun pemilik *booth* hanya dapat melihat data promo saja. Setelah memilih, aplikasi akan mengirimkan data yang terpilih dan mendapatkan hasil *query* seluruh data yang terpilih yang dilakukan pada *server*.

3.2.6.2 Proses menambah data

Proses penambahan data ini dapat dilakukan pada data *booth* dan promo. Proses ini dilakukan oleh administrator dan pemilik *booth*, namun pemilik *booth* hanya dapat melakukan penambahan data pada data promo saja. Proses penambahan data dilakukan dengan mengisi *form* yang telah disediakan. Setelah proses pengisian selesai, data akan dikirim oleh aplikasi untuk disimpan dalam basis data *server*.

3.2.6.3 Proses mengubah data

Proses pengubahan data ini dapat dilakukan pada data *booth* dan promo. Proses ini dilakukan oleh administrator dan pemilik *booth*, namun pemilik *booth* hanya dapat melakukan ubah data pada data promo saja. Proses pengubahan data dilakukan dengan memilih data mana yang akan diubah kemudian sistem akan menampilkan *form* yang berisi detail data terpilih yang dapat diubah kontennya. Setelah proses pengubahan selesai, data akan dikirim oleh aplikasi untuk dilakukan perubahan dalam basis data *server*.

3.2.6.4 Proses menghapus data

Proses penghapusan data ini dapat dilakukan pada data *booth* dan promo. Proses ini dilakukan oleh administrator dan pemilik *booth*, namun pemilik *booth* hanya dapat menghapus data promo saja. Proses penghapusan data dilakukan dengan memilih data mana yang akan dihapus. Setelah memilih, aplikasi akan menampilkan detail data terpilih dan meminta konfirmasi penghapusan. Setelah konfirmasi penghapusan diterima, aplikasi akan mengeksekusi proses penghapusan data terpilih pada basis data *server*.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Pada bab ini dijelaskan mengenai implementasi dari perancangan perangkat lunak. Dalam Bab ini akan dibahas mengenai algoritma, potongan kode, lingkungan pembangunan perangkat lunak, implementasi antarmuka pengguna, dan implementasi proses-proses yang terjadi pada masing-masing kasus penggunaan pada perangkat lunak. Implementasi sistem mengacu pada perancangan yang ditulis pada Bab 3. Namun, tidak menutup kemungkinan adanya perubahan-perubahan dari rancangan tersebut apabila memang diperlukan.

4.1 Lingkungan Implementasi

Dalam merancang dan mengimplementasikan perangkat lunak ini, digunakan beberapa perangkat pendukung sebagai berikut.

4.1.1 Lingkungan implementasi perangkat keras

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan perangkat lunak ini adalah sebagai berikut:

- Laptop Samsung ATIV Book 9 Lite
 - Windows 10 64-bit
 - Quad-Core Prosesor up to 1.4 GHz
 - RAM 4.00 GB
- Perangkat Komputer Lenovo
 - Windows 8 Pro 64-bit
 - Prosesor Intel® i3 Core™ i3-3240 @ 3.4GHz
 - RAM 4.00 GB
- Perangkat *smartphone* Sony Xperia M4 Aqua E2353
 - Sistem Operasi : Android v5.0 (Lollipop)
 - Processor Qualcomm Snapdragon 615 Octa Core 64 bit
 - Memori internal : 8.00 GB
 - RAM : 2.00 GB
 - WLAN : WiFi 802.11 a/b/g/n, *dual-band*, *hotspot*, DLNA

4.1.2 Lingkungan implementasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

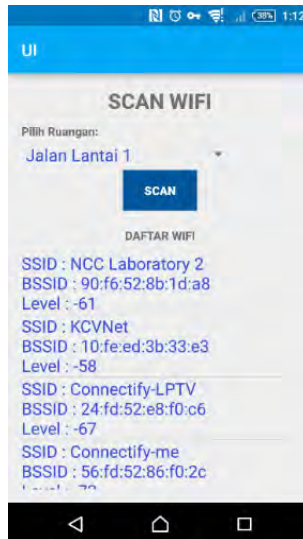
- Microsoft Windows 10 sebagai sistem operasi,
- Android Studio 2.1 untuk mengimplementasikan aplikasi perangkat bergerak,
- Sublime sebagai *text editor* untuk implementasi *web service* dan aplikasi *web*,
- PostgreSQL sebagai basis data *server*,
- Cloud VPS sebagai *web service server*,
- Google Chrome sebagai *browser* dalam tahap uji coba *server*,
- Power Designer untuk merancang basis data,
- StarUML untuk merancang diagram perencanaan perangkat lunak.

4.2 Implementasi Konsep Indoor Localization

Pada subbab ini akan dibahas mengenai implementasi konsep *Indoor Localization* berdasarkan rancangan yang telah dibahas pada bab 3. Hal-hal yang akan dibahas dalam subbab ini antara lain implementasi pengambilan data, fase pembentukan *Binary Tree*, dan fase pendeteksian lokasi dengan SVM.

4.2.1 Implementasi pengambilan data

Sebelum dilakukan proses pendeteksian lokasi, perlu disiapkan data sampel yang digunakan sebagai data *training* dalam SVM. Data dikumpulkan dengan melakukan empat kali proses *scanning* WiFi pada lima titik di seluruh *booth* pada East Coast Center dalam empat hari yang berbeda pada setiap rentang waktu 11.00 – 15.00 dengan menggunakan aplikasi scanner WiFi yang terdapat pada Gambar 4.1. Data tersebut disimpan kedalam basis data yang telah dibuat, ditunjukkan oleh Gambar 4.2.



Gambar 4.1. Proses Pengumpulan Data Training

RSSID (integer)	SIGNAL STRENGTH (integer)	BSSID (character varying(50))	SSID (integer)
1	-56	64:66:0b:1c:a:52:a8	3
2	-55	64:66:0b:1c:a:52:a8	3
3	-67	10:74:0a:12:7b:24	8
4	-67	10:74:0a:12:7b:24	3
5	-67	64:70:02:8b:01:90	3
6	-67	64:70:02:8b:01:90	3
7	-67	64:70:02:8b:01:90	3
8	-67	64:70:02:8b:01:90	3
9	-70	23:01:13:06:e0:84	3
10	-70	23:01:13:06:e0:84	3
11	-55	64:66:0b:1c:a:52:a8	4
12	-56	64:66:0b:1c:a:52:a8	4
13	-69	10:74:0a:12:7b:24	4
14	-69	10:74:0a:12:7b:24	4
15	-60	64:70:02:8b:01:90	4
16	-67	64:70:02:8b:01:90	4
17	-60	64:70:02:8b:01:90	4
18	-64	64:66:0b:1c:a:52:a8	5
19	-64	64:66:0b:1c:a:52:a8	5
20	-79	10:74:0a:12:7b:24	5
21	-79	10:74:0a:12:7b:24	5
22	-79	64:70:02:8b:01:90	5
23	-79	64:70:02:8b:01:90	5

Gambar 4.2. Data Sinyal WiFi Dalam Basis data

Pada saat penyimpanan data, secara tidak sengaja sinyal-sinyal WiFi dari *tethering* akan ikut tersimpan dan akan mengganggu proses pendeteksian lokasi. Oleh karena itu, sebelum data *training* digunakan, akan dilakukan proses penghapusan data-data WiFi *tethering* dengan cara melakukan *query* seperti yang ditunjukkan pada Kode Sumber 4.1. Karena proses pengumpulan data dilakukan sebanyak empat kali pada setiap *booth* di lima titik berbeda, maka dapat disimpulkan apabila sebuah BSSID pada suatu *booth* jumlahnya kurang dari empat, maka BSSID tersebut dianggap sebagai WiFi *tethering* dan dihapus dari basis data.

```
select r."BSSID", count(r."BSSID") as jumlah
from (public."RSS" r INNER JOIN public."SCAN"
s ON r."SCANID" = s."SCANID" AND s."BOOTHID" =
$boothid)
GROUP BY r."BSSID"
HAVING COUNT (r."BSSID") < 4;
```

Kode Sumber 4.1. Implementasi *Query* Pemilihan Sinyal WiFi

4.2.2 Implementasi pre-processing

Tahap *pre-processing* data merupakan tahap awal dari proses prediksi lokasi. *Pre-processing* diperlukan agar data set yang telah didapatkan dari proses *scanning* memiliki format yang sama dengan input yang diharapkan oleh library. Format input library LibSVM berbentuk *array* seperti pada Gambar 4.3. Index atau *key* ke 0 merupakan representasi dari *class* dari array tersebut, index selanjutnya sebagai *feature*, dan *value* sebagai *value* dari *feature*.

Proses *pre-processing* mengubah kolom APID menjadi *feature*, kolom SIGNALSTRENGTH menjadi *value*, dan kolom *BOOTHID* menjadi *value* dari index ke 0 yang menunjukkan kelas dari data tersebut. Setelah *pre-processing* selesai, yang dilakukan selanjutnya adalah melakukan *training* dan *predict* data yang dihasilkan oleh tahap *pre-processing* dengan klasifikasi SVM dan metode *Binary Tree*. Kode Sumber 4.2 adalah implementasi kode dari tahap *pre-processing*.


```

Array (
  [0] => 3      → Class
  [50] => -67
  [51] => -67
  [52] => -67
  [18] => -67
  [53] => -67
  [43] => -56
  [44] => -55
)

```

Feature => Value

Gambar 4.3. Contoh format array LibSVM

```

public function data(){
    $bssid = $this->input->post('bssid');
    $rss = $this->input->post('rss');
    $listbssid = explode(";", $bssid);

    $this->load->model('M_ap');
    $listapid = array();

    foreach ($listbssid as $key) {
        $apid = $this->M_ap->check($key);
        array_push($listapid, $apid);
    }

    $listss = explode(";", $rss);
    $dataset = array();
    foreach ($listapid as $key => $value) {
        if ($value != "") {
            $dataset[$value] = $listss[$key];
        }
    }
    $this->startClassification($dataset);
}

```

Kode Sumber 4.2. Implementasi Fungsi *Pre-processing* data

4.2.3 Implementasi pembentukan binary tree

Binary Tree dibuat untuk merepresentasikan *binary classification* di dalam *multi-class*. Semua data dibagi menjadi dua bagian sampai bagian terkecil yang disebut *leaf*. Implementasi kode ditunjukkan pada Kode Sumber 4.3. Hasil dari pembentukan *Binary Tree* ditunjukkan oleh Gambar 4.4.

```
function startClassification($dataset){
    $this->load->model('M_klasifikasi');
    $coba = $this->M_klasifikasi->data();
    $class = array();
    foreach ($coba as $key => $row) {
        if(in_array($row->BOOTHID, $class)){
            continue;
        }
        else{
            array_push($class, $row->BOOTHID);
        }
    }
    $l = min($class);
    $r = max($class);
    $m = $class[floor(sizeof($class)/2)];

    $this->findClass($l, $m, $r, $dataset);
}

function findClass($l, $m, $r, $dataset){
    $c = $this->doSVM($l, $m, $r, $dataset);
    $res;
    echo $c.'  
';

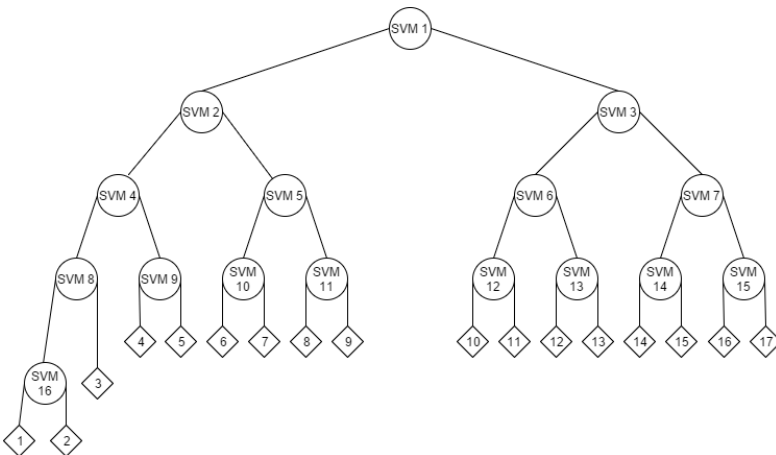
    if ($c == -1) {
        if ($l + 1 == $r) {
            $res = $l;
            echo $res;
        }
        else{
            $res = $this->findClass($l,
            floor(($l+$m)/2), $m, $dataset);
        }
    }
    else{
        if ($l+1 == $r || $l+1 == $m) {
```

```

    $res = $r;
    echo $res;
  }
  else{
    $res = $this->findClass($m+1,
    floor(($m+1+$r)/2), $r, $dataset);
  }
}
return $res;
}

```

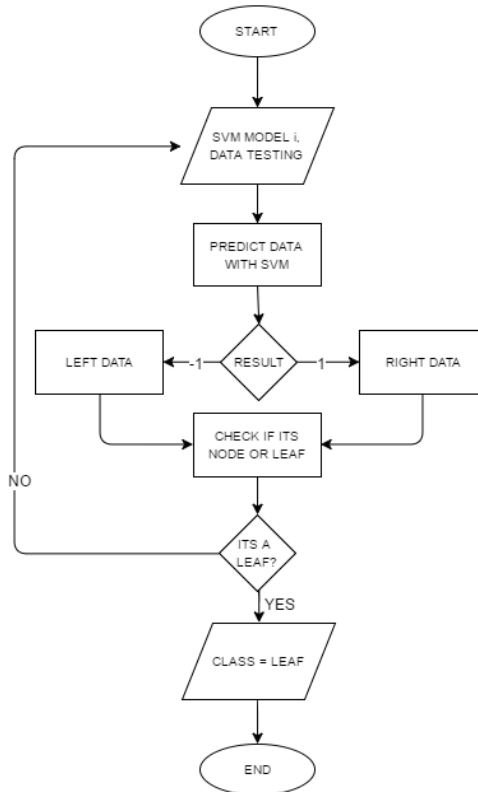
Kode Sumber 4.3. Implementasi Fungsi Pembentukan *Binary Tree*



Gambar 4.4. Hasil *Binary Tree* untuk Data *Booth*

4.2.4 Implementasi tahap pendeteksian lokasi

Tahap ini akan selalu dilakukan di setiap *node* terpilih pada *Binary Tree*. Kelas yang akan diproses hanya kelas yang ada di dalam *node* tersebut. Di dalam tahap ini akan dilakukan prediksi kelas untuk kemudian digunakan untuk memilih *node* kiri atau kanan sesuai dengan nilai hasil prediksi. Dilakukan berulang-ulang hingga akan tersisa satu kelas yang merupakan hasil akhir dari proses. Implementasi seperti yang ditunjukkan pada Kode Sumber 4.4. Alur SVM dengan *Binary Tree* ditunjukkan pada Gambar 4.5.



Gambar 4.5. Diagram Alur Implementasi SVM dan *Binary Tree*

```

function doSVM($l, $m, $r, $dataset){
    $this->load->model('M_klasifikasi');
    $coba = $this->M_klasifikasi->data();
    $count = sizeof($coba);
    $inner = array();
    $outer = array();
    $i = 1;
    $state = 0;

    foreach ($coba as $key => $row) {
        if($l <= $row->BOOTHID && $row->BOOTHID <=
    $r){

```

```

        if($row->SCANID != $state){
            array_push($outer, $inner);
            if ($row->BOOTHID <= $m) {
                $inner = array(-1);
            }
            else $inner = array(1);
            $state = $row->SCANID;
        }
        $inner[$row->APID] = $row->SIGNALSTRENGTH;
    }
    else continue;
}
unset($outer[0]);

$svm = new SVM();
$model = $svm->train($outer);
$result = $model->predict($dataset);
return $result;
}

```

Kode Sumber 4.4. Implementasi Fungsi Pelatihan dan Pendeteksian Lokasi

4.3 Implementasi Basis Data

Pada subbab ini akan dibahas implementasi dari rancangan basis data yang telah dibahas pada Bab 3. Pada subbab ini akan dibahas implementasi struktur basis data. Implementasi basis data menggunakan sintaks SQL dan dibedakan menjadi beberapa tabel.

4.3.1 Implementasi tabel Admin

Tabel Admin digunakan untuk menyimpan data admin akan digunakan administrator untuk mendapatkan autentikasi untuk mengakses sistem. Implementasi tabel admin dapat dilihat pada Kode Sumber 4.5

```

CREATE TABLE public."ADMIN"
(
    "ADMINNAME" character varying(20) NOT NULL,
    "ADMINPASSWORD" character varying(50),
    "ADMINID" integer NOT NULL DEFAULT
nextval(' "ADMIN_ADMINID_seq" '::regclass),

```

```

"ADMINFULLNAME" character varying(100),
CONSTRAINT "PK_ADMIN" PRIMARY KEY ("ADMINID")
)
WITH (
  OIDS=FALSE
);

```

Kode Sumber 4.5. Implementasi Struktur Tabel Admin

4.3.2 Implementasi tabel AP

Tabel AP digunakan untuk menyimpan data BSSID dan SSID yang di dapat pada proses *scanning* WiFi. Data pada tabel AP digunakan untuk data set setelah dilakukan *pre-processing* sebelumnya. Implementasi tabel AP dapat dilihat pada Kode Sumber 4.6.

```

CREATE TABLE public."AP"
(
  "BSSID" character varying(50) NOT NULL,
  "SSID" character varying(50),
  "APID" integer NOT NULL DEFAULT
nextval('"AP_APID_seq"'::regclass),
  CONSTRAINT "PK_AP" PRIMARY KEY ("BSSID")
)
WITH (
  OIDS=FALSE
);

```

Kode Sumber 4.6. Implementasi Struktur Tabel AP

4.3.3 Implementasi tabel Booth

Tabel *Booth* digunakan untuk menyimpan daftar lokasi (*booth*) yang terdapat pada East Coast Center. Tabel *Booth* menyimpan data nama dan keterangan lain pada masing-masing *Booth*. Implementasi tabel *Booth* dapat dilihat pada Kode Sumber 4.7.

```

CREATE TABLE public."BOOTH"
(
  "BOOTHID" integer NOT NULL DEFAULT
nextval('"BOOTH_BOOTHID_seq"'::regclass),
  "BOOTHNAME" character varying(50),
  "BOOTHDESC" text,

```

```

"BOOTHLOGO" character varying(100),
"BOOTHUSERNAME" character varying(50),
"BOOTHPASSWORD" character varying(50),
CONSTRAINT "PK_BOOTH" PRIMARY KEY ("BOOTHID")
)
WITH (
  OIDS=FALSE
);

```

Kode Sumber 4.7. Implementasi Struktur Tabel *Booth*

4.3.4 Implementasi tabel Scan

Tabel Scan digunakan untuk menyimpan id scan pada saat *scanning*. Implementasi tabel Scan dapat dilihat pada Kode Sumber 4.8.

```

CREATE TABLE public."SCAN"
(
  "SCANID" integer NOT NULL DEFAULT
nextval('"SCAN_SCANID_seq"'::regclass),
  "BOOTHID" integer,
  CONSTRAINT "PK_SCAN" PRIMARY KEY ("SCANID"),
  CONSTRAINT "FK_BOOTH" FOREIGN KEY ("BOOTHID")
REFERENCES public."BOOTH" ("BOOTHID") MATCH
SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
);

```

Kode Sumber 4.8. Implementasi Struktur Tabel Scan

4.3.5 Implementasi tabel Promo

Tabel Promo digunakan untuk menyimpan data promo yang akan dimiliki oleh *booth*. Implementasi tabel Promo dapat dilihat pada Kode Sumber 4.9.

```

CREATE TABLE public."PROMO"
(
  "PROMOID" integer NOT NULL DEFAULT
nextval('"PROMO_PROMOID_seq"'::regclass),
  "STARTEVENT" timestamp without time zone,
  "ENDEVENT" timestamp without time zone,

```

```

"PROMODESC" text,
"PROMOPIC" character varying(100),
"BOOTHID" integer,
CONSTRAINT "PK_PROMO" PRIMARY KEY ("PROMOID"),
CONSTRAINT "FK_PROMO_BOOTH" FOREIGN KEY
("BOOTHID")
REFERENCES public."BOOTH" ("BOOTHID") MATCH
SIMPLE
ON UPDATE CASCADE ON DELETE NO ACTION
)
WITH (
OIDS=FALSE
);

```

Kode Sumber 4.9. Implementasi Struktur Tabel Promo

4.3.6 Implementasi tabel RSS

Tabel RSS digunakan untuk menyimpan data *signal strength* dari setiap BSSID yang di dapat. Data pada tabel RSS nanti akan dilakukan *pre-processing* untuk menjadi data set. Implementasi tabel RSS dapat dilihat pada Kode Sumber 4.10.

```

CREATE TABLE public."RSS"
(
"RSSID" integer NOT NULL DEFAULT
nextval(' "RSS_RSSID_seq" '::regclass),
"SIGNALSTRENGTH" integer,
"BSSID" character varying(50),
"SCANID" integer,
CONSTRAINT "PK_RSS" PRIMARY KEY ("RSSID"),
CONSTRAINT "FK_AP_RSS" FOREIGN KEY ("BSSID")
REFERENCES public."AP" ("BSSID") MATCH SIMPLE
ON UPDATE CASCADE ON DELETE NO ACTION,
CONSTRAINT "FK_SCAN" FOREIGN KEY ("SCANID")
REFERENCES public."SCAN" ("SCANID") MATCH
SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
OIDS=FALSE );

```

Kode Sumber 4.10. Implementasi Struktur Tabel RSS

4.3.7 Implementasi tabel Nearby

Tabel Nearby digunakan untuk menyimpan data *booth* terdekat. tabel Nearby merupakan tabel hasil *many-to-many* dari tabel *Booth* dengan dirinya sendiri. Berisi id *booth* dan id dari Nearby *Booth*. Implementasi tabel Nearby dapat dilihat pada Kode Sumber 4.11.

```
CREATE TABLE public."NEARBYBOOTH"
(
  "BOOTHID" integer,
  "NEARBYBOOTHID" integer,
  "ID" integer NOT NULL DEFAULT
nextval(' "NEARBYBOOTH_ID_seq" '::regclass),
  CONSTRAINT "NEARBYBOOTH_pkey" PRIMARY KEY ("ID"),
  CONSTRAINT "FK_BOOTH" FOREIGN KEY ("BOOTHID")
  REFERENCES public."BOOTH" ("BOOTHID") MATCH
SIMPLE
  ON UPDATE CASCADE ON DELETE NO ACTION,
  CONSTRAINT "FK_NEARBY" FOREIGN KEY
("NEARBYBOOTHID")
  REFERENCES public."BOOTH" ("BOOTHID") MATCH
SIMPLE
  ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE );
```

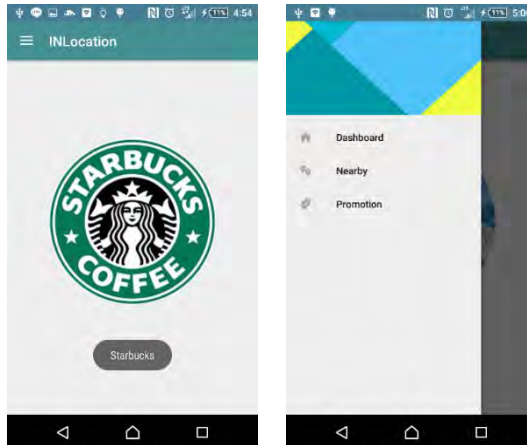
Kode Sumber 4.11. Implementasi Struktur Tabel Nearby

4.4 Implementasi Antarmuka Aplikasi Perangkat Bergerak

Pada subbab ini akan dibahas secara mendetail dari implementasi dari antarmuka perangkat bergerak beserta penjelasannya yang disesuaikan dengan rancangan yang telah dijelaskan pada Bab 3.

4.4.1 Implementasi antarmuka halaman dashboard

Pada antarmuka halaman dashboard hanya terdapat logo dari *booth* hasil dari pendeteksian lokasi. Terdapat pula *sidebar* untuk navigasi ke halaman-halaman berikutnya.



Gambar 4.6. Implementasi Antarmuka Halaman Dashboard dan Sidebar Menu

Pada saat pertama kali masuk ke dashboard, aplikasi akan melakukan *scanning* sinyal WiFi yang kemudian akan dikirim dan diolah oleh *server*. Hasil *response* dari server berupa *image path* dari logo *booth* yang terdeteksi lokasinya. *Image path* disimpan dalam *ImageView* seperti pada Gambar 4.6.

4.4.2 Implementasi antarmuka halaman nearby booth

Antarmuka halaman *nearby booth* merupakan antarmuka yang akan terbuka jika pengguna memilih menu *Nearby booth* pada *sidebar*. Implementasi antarmuka dapat dilihat pada Gambar 4.7. Pada halaman ini, *booth* yang ditampilkan adalah *booth* terdekat dari *booth* yang posisinya ditampilkan pada halaman Dashboard. Disimpan dalam *Recycler View*, terdapat *ImageView* untuk logo *booth*, *TextView* untuk nama *booth* dan deskripsi *booth*.



Gambar 4.7. Implementasi Antarmuka Halaman Nearby *Booth*

4.4.3 Implementasi antarmuka halaman promo

Antarmuka halaman promo merupakan antarmuka yang menampilkan seluruh promo yang sedang berlangsung. Implementasi antarmuka dapat dilihat pada Gambar 4.8.

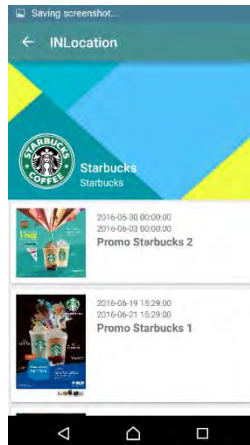


Gambar 4.8. Implementasi Antarmuka Halaman Promo

Semua list promo disimpan dalam *Recycler View* yang di dalamnya terdapat *ImageView* dan *TextView* untuk menampilkan poster promo, tanggal mulai, tanggal selesai, dan deskripsi promo sebagai data-data informasi promo.

4.4.4 Implementasi antarmuka halaman detail booth

Antarmuka ini akan menampilkan detail *booth* apabila pengguna menekan salah satu *booth* yang ada dalam *Nearby Booth* atau *booth* yang posisinya sedang didapatkan. Implementasi antarmuka dapat dilihat pada Gambar 4.9. Terdapat detail *booth* pada bagian atas dan *RecyclerView* yang merupakan daftar dari promo yang sedang berlangsung pada *booth* tersebut.



Gambar 4.9. Implementasi Antarmuka Halaman Detail *Booth*

4.5 Implementasi Proses Aplikasi Perangkat Bergerak

Pada subbab ini akan dibahas secara mendetail dari implementasi proses aplikasi perangkat bergerak yang disesuaikan dengan rancangan yang telah dijelaskan pada bab 3.

4.5.1 Proses mendeteksi dan menampilkan lokasi

Kode Sumber 4.12 adalah implementasi dari proses mendeteksi dan menampilkan lokasi. Proses ini menggunakan *asynchronous task* untuk melakukan *request* kepada *server*. Proses

ini diawali dengan melakukan *scanning* WiFi, selanjutnya sinyal WiFi yang diterima disimpan dalam bentuk *string* untuk kemudian dikirim ke *server* sebagai data yang digunakan untuk mendeteksi lokasi.

```
public void getPosition(){
    String BSSID = "";
    String SS = "";
    for (int i = 0; i < wifiScanList.size(); i++) {
        BSSID += (wifiScanList.get(i).BSSID + ";");
        SS +=
        (String.valueOf(wifiScanList.get(i).level) + ";");
    }
    Map<String, String> data = new HashMap<>();
    data.put("bssid", BSSID);
    data.put("ss", SS);
    sendWifi(data, null,
    "http://13.76.92.195/ta/Klasifikasi/data");
}
```

Kode Sumber 4.12. Implementasi Fungsi Mendeteksi Lokasi

Hasil dari pendeteksian lokasi adalah id dari *booth*. Proses menampilkan lokasi merupakan hasil *query* dari id *booth* yang didapat seperti yang diperlihatkan pada Kode Sumber 4.13.

```
private void sendWifi(Map<String, String> arguments,
final Map<String, File> files, final String url){
    RequestAsyncTask addPositionProcess = new
    RequestAsyncTask(arguments, files) {
        @Override
        protected void
        setAfterThread(ResponseManager responseManager) {
            if
            (responseManager.getCode().equals(ResponseManager.SU
            CCESS)){
                textPosition =
                responseManager.getData();

                booth = ((Booth)
                responseManager.getOne(Booth.class));
                Log.v("response klasifikasi",
                "error: " + booth.getBOOTHID());
            }
        }
    };
}
```

```

        Picasso.with(activity)
            .load(booth.getBOOTHLOGO())
            .error(R.drawable.ic_menu_ma
nage)
            .placeholder(R.drawable.ic_m
enu_gallery)
            .into(imgLogo);

tvName.setText(booth.getBOOTHNAME());

        imgLogo.setVisibility(View.VISIBLE);
        Log.d("cekbooth", "setAfterThread: "
+ booth.getBOOTHID());
        gd.setIdBooth(booth.getBOOTHID());

        Toast.makeText(activity,
booth.getBOOTHNAME(), Toast.LENGTH_SHORT).show();
    }
    else if
(responseManager.getCode().equals(ResponseManager.FA
IL)){
        Toast.makeText(activity, "Gagal
Mendapatkan Posisi", Toast.LENGTH_SHORT).show();
    }
}
};
addPositionProcess.execute(url);
}

```

Kode Sumber 4.13. Implementasi Fungsi Menampilkan Lokasi

4.5.2 Proses menampilkan detail booth

Proses menampilkan detail *booth* dijalankan ketika pengguna memilih salah satu *booth*. Kode Sumber 4.14 merupakan implementasi dari proses menampilkan detail *booth*. *BOOTHID* yang didapatkan saat pengguna memilih salah satu *booth* menjadi parameter *request* kepada *server*, kemudian *server* melakukan *query* untuk mengirimkan data detail *booth* dalam bentuk *HttpResponse*.

```

public View onCreateView(LayoutInflater inflater,
ViewGroup container, Bundle savedInstanceState) {
    v =

```

```

inflater.inflate(R.layout.fragment_detail_booth,
container, false);

    imgLogo = (ImageView)
v.findViewById(R.id.imgLogo);
    tvName = (TextView) v.findViewById(R.id.tvName);
    tvDesc = (TextView) v.findViewById(R.id.tvDesc);
    rvBooth = (RecyclerView)
v.findViewById(R.id.rvBooth);

    rvBooth.setLayoutManager(layoutManager);

    adapter = new PromoAdapter(promoList, new
PromoAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(Promo promo) {
            Toast.makeText(detailBoothActivity,
promo.getPROMOID(), Toast.LENGTH_SHORT).show();

            Intent intent = new
Intent(detailBoothActivity,
DetailPromoActivity.class);
            intent.putExtra("promo", promo);
            startActivity(intent);
        }
    });
    rvBooth.setAdapter(adapter);
    RequestBoothPromo requestBoothPromo = new
RequestBoothPromo();

requestBoothPromo.execute("http://13.76.92.195/ta/Bo
oth/showDetailBooth/"+ booth.getBOOTHID());
    return v;
}

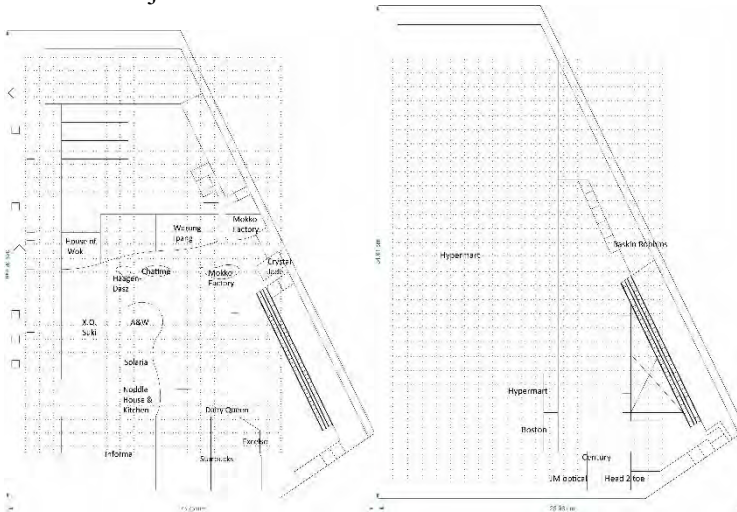
```

Kode Sumber 4.14. Implementasi Fungsi Request Detail Booth

4.5.3 Proses menampilkan booth terdekat

Berikut ini adalah implementasi dari proses menampilkan *booth* terdekat. Proses ini berkaitan erat dengan proses pendeteksian lokasi, karena pada proses ini akan ditampilkan *booth* yang terdekat dengan lokasi pengunjung berdasarkan *Adjacency*

Matrix yang dibuat sesuai dengan denah. Gambar denah East Coast Center ditunjukkan oleh Gambar 4.10.



Gambar 4.10. Denah East Coast Center

Setelah itu proses menampilkan *booth* terdekat dilakukan dengan memberikan informasi *booth* yang terdeteksi sebagai lokasi pengunjung saat itu lalu dilakukan *query* untuk menampilkan data-data *booth* yang terdekat seperti pada Kode Sumber 4.15.

```
@Override
public View onCreateView(LayoutInflater inflater,
    ViewGroup container,
                           Bundle savedInstanceState)
{
    v = inflater.inflate(R.layout.fragment_booth,
        container, false);

    rvBooth = (RecyclerView)
v.findViewById(R.id.rvBooth);
    rvBooth.setLayoutManager(layoutManager);
    adapter = new BoothAdapter(boothList, new
    BoothAdapter.OnItemClickListener()) {
```



```

@Override
public void onItemClick(Booth booth) {
    Toast.makeText(boothActivity,
        booth.getBOOTHNAME(), Toast.LENGTH_SHORT).show();

    Intent intent = new
Intent(boothActivity, DetailBoothActivity.class);
    intent.putExtra("booth", booth);
    startActivity(intent);
    }
});
rvBooth.setAdapter(adapter);

Log.v("boothid", String.valueOf(idBooth));

if(idBooth != 1){
    RequestBooth rb = new RequestBooth();
    rb.execute("http://13.76.92.195/ta/Booth/getNearby/"
+ idBooth);
    }
else {
    RequestBooth rb = new RequestBooth();
    rb.execute("http://13.76.92.195/ta/Booth/get");
    }

return v;
}

```

Kode Sumber 4.15. Implementasi Fungsi Menampilkan Booth Terdekat

4.5.4 Proses menampilkan daftar promo

Kode Sumber 4.16 adalah implementasi dari proses menampilkan daftar promo. Proses ini menampilkan promo-promo yang masih berlaku. *Store procedure* digunakan untuk menghapus promo yang sudah melewati tanggal dan jam berlakunya. *Request* data dilakukan pada saat pengguna memilih menu Promo dengan metode GET. Server akan mengirimkan *response* berupa data-data promo.

```

public View onCreateView(LayoutInflater inflater,
    ViewGroup container,
                           Bundle savedInstanceState)
{
    v = inflater.inflate(R.layout.fragment_promo,
        container, false);

    rvPromo = (RecyclerView)
v.findViewById(R.id.rvPromo);
    rvPromo.setLayoutManager(layoutManager);
    adapter = new PromoAdapter(promoList, new
PromoAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(Promo promo) {
            Intent intent = new
Intent(promoActivity, DetailPromoActivity.class);
            intent.putExtra("promo", promo);
            startActivity(intent);
        }
    });
    rvPromo.setAdapter(adapter);
    RequestPromo rp = new RequestPromo();
    rp.execute("http://10.151.43.250/TA/Promo/get");
    return v;
}

```

Kode Sumber 4.16. Implementasi Fungsi Request Daftar Promo

4.6 Implementasi Antarmuka Aplikasi Web

Pada subbab ini akan dibahas secara mendetail dari implementasi antarmuka aplikasi *web* yang disesuaikan dengan rancangan yang telah dijelaskan pada bab 3.

4.6.1 Implementasi antarmuka halaman login

Antarmuka login merupakan antarmuka yang pertamakali muncul saat halaman aplikasi web dibuka. Implementasi antarmuka dapat dilihat pada Gambar 4.11.

Gambar 4.11. Implementasi Antarmuka Login

Pada saat pertama kali masuk ke halaman login, pengguna akan diminta untuk memasukkan username, password, dan hak akses untuk autentikasi.

4.6.2 Implementasi antarmuka halaman melihat data

Pada antarmuka halaman melihat data, data ditampilkan dalam bentuk tabel, fungsi *edit* dan *delete* ada dalam setiap *record*. Implementasi antarmuka dapat dilihat pada Gambar 4.12.

BOOTH ID	BOOTH NAME	BOOTH DESCRIPTION	BOOTH LOGO	BOOTH USER NAME	BOOTH PASSWORD	EDIT
10	AGI	AGI	http://13.76.92.29:80/ta/booth/aw.png	aw	171752248133034258851603811161	
8	Mokko Factory	Mokko Factory	http://13.76.92.29:80/ta/booth/mokko.jpg	mokko	212076997878c5aa02882af8f8ab4	
9	Crystal Jade	Crystal Jade	http://13.76.92.29:80/ta/booth/cj.png	cj	28198038017680adab9f6d59848f6	
12	Noodle House & Kitchen	Noodle House & Kitchen	http://13.76.92.29:80/ta/booth/nhk.jpg	swania	12160ee112195131f0cc8a2f2a486	
13	Informa	Informa	http://13.76.92.29:80/ta/booth/informasi.png	informa	44c1d3459818a7c2c01289476a21e6	
10	Starbucks	Starbucks	http://13.76.92.29:80/ta/booth/starbucks.png	starbucks	7120ed4911f1f9632e6ad800d01c7e	
17	Dankin Robbin	Dankin Robbin	http://13.76.92.29:80/ta/booth/dr.png	dr	602796297422776c160e98262329591	
16	Danelo	Danelo	http://13.76.92.29:80/ta/booth/danelo.png	danelo	50dd112d607f9d535a3a0e6a1070	

Gambar 4.12. Implementasi Antarmuka Melihat Data

4.6.3 Implementasi antarmuka halaman menambah data

Antarmuka menambah data berupa *form-form* sesuai dengan data yang akan ditambahkan. Pada akhir *form* terdapat button save untuk menyimpan data-data yang telah diisi. Implementasi antarmuka dapat dilihat pada Gambar 4.13.

The screenshot shows a web application interface for 'Booth Management'. The page title is 'Booth Management' and the user is logged in as 'Administrator'. The main content area is titled 'New Booth' and contains a form with the following fields:

- Booth Name:** A text input field with a small 'Booth Name' label below it.
- Booth Username:** A text input field with a small 'Booth Username' label below it.
- Booth Password:** A text input field with a small 'Booth Password' label below it.
- Booth Description:** A text input field with a small 'Booth Description' label below it.
- Booth Logo:** A file upload field with a 'Choose File' button and the text '(No file chosen)'.

A 'Save' button is located at the bottom right of the form.

Gambar 4.13. Implementasi Antarmuka Menambah Data

4.6.4 Implementasi antarmuka halaman mengubah data

Antarmuka halaman mengubah data hampir sama dengan antarmuka halaman menambah data, yang membedakan hanya data-data sebelumnya dapat dilihat di dalam form. Pada akhir form terdapat tombol edit untuk menyimpan perubahan yang telah dilakukan. Implementasi antarmuka dapat dilihat pada Gambar 4.14.

The screenshot shows a web application interface for 'Booth Management'. The page title is 'Booth Management' and the user is logged in as 'Administrator'. The main content area is titled 'New Booth' and contains a form with the following fields:

- Booth Name:** A text input field with a small 'Booth Name' label below it.
- Booth Username:** A text input field with a small 'Booth Username' label below it.
- Booth Password:** A text input field with a small 'Booth Password' label below it.
- Booth Description:** A text input field with a small 'Booth Description' label below it.
- Booth Logo:** A file upload field with a 'Choose File' button and the text '(No file chosen)'.

A 'Save' button is located at the bottom right of the form.

Gambar 4.14. Implementasi Antarmuka Mengubah Data

4.7 Implementasi Proses Aplikasi Web

Pada subbab ini akan dibahas secara mendetail dari implementasi proses aplikasi perangkat *web* yang disesuaikan dengan rancangan yang telah dijelaskan pada Bab 3.

4.7.1 Implementasi proses melihat data

Kode Sumber 4.17 adalah implementasi dari proses melihat data promo dan *booth*. Data *booth* hanya dapat diakses oleh administrator, sedangkan pemilik *booth* hanya dapat melihat data promo.

```

public function showAllBooth(){
    $this->load->model('M_booth');

    $this->header();
    $data['listBooth'] = $this->M_booth->getAll();
    $this->load-
>view('manajemen_booth/view_all_booth', $data);
    $this->load->view('design/footer');
}

public function getAll()
{
    $query = $this->db->get_where('BOOTH');

    if ($query->num_rows() > 0)
    {
        foreach ($query->result() as $row)
        {
            $data[] = $row;
        }
        return $data;
    }
    else
    {
        return false;
    }
}
}

```

Kode Sumber 4.17. Implementasi Fungsi Melihat Data

4.7.2 Implementasi proses proses menambah data

Kode Sumber 4.18 merupakan implementasi dari proses menambah data promo dan *booth*. Pada proses ini terdapat proses mengunggah gambar dan data-data lainnya ke *server*.

```

public function add(){
    $boothname = $this->input->post('nama');

```

```

$boothdesc = $this->input->post('desc');
$boothusername = $this->input->post('username');
$boothpassword = $this->input->post('pass');
$logo = $this->input->post('logo');

$data = array(
    'BOOTHNAME' => $boothname,
    'BOOTHDESC' => $boothdesc,
    'BOOTHLOGO' => NULL,
    'BOOTHUSERNAME' => $boothusername,
    'BOOTHPASSWORD' => md5($boothpassword)
);

$target_path = NULL;
if($_FILES['logo']['name'] != NULL){
    $path_parts =
pathinfo($_FILES["logo"]["name"]);
    $extension = $path_parts['extension'];
    $extension = ".$extension";
    $target_path = "booth/";
    $target_path =
$target_path.$boothusername.$extension;
    $data['BOOTHLOGO'] =
base_url().$target_path;
}

$this->load->model('M_booth');

$added = $this->M_booth->insert('BOOTH', $data);
if($added){
    if($target_path != NULL){
move_uploaded_file( $_FILES['logo']['tmp_name'],
$target_path);
    }
}

redirect ('Booth/showAllBooth');
}

function insert($table, $data){
    $query = $this->db->insert($table, $data);

    if ($query) {

```

```

        return TRUE;
    }
    else{
        return FALSE;
    }
}

```

Kode Sumber 4.18. Implementasi Fungsi Menambah Data

4.7.3 Implementasi proses mengubah data

Kode Sumber 4.19 merupakan implementasi dari proses mengubah data promo dan *booth*. Apabila terjadi kesalahan pada saat melakukan input, atau terjadi perubahan data sewaktu-waktu, data dapat diubah dan disimpan kembali.

```

public function edit($boothid){
    $this->load->model('M_booth');

    $this->header();
    $data['dataBooth'] = $this->M_booth-
>edit($boothid);
    $this->load->view('manajemen_booth/edit_booth',
    $data);
    $this->load->view('design/footer');
}

public function update(){
    $boothid = $this->input->post('id');
    $boothname = $this->input->post('nama');
    $boothdesc = $this->input->post('desc');
    $boothusername = $this->input->post('username');
    $boothpassword = $this->input->post('pass');
    $logo = $this->input->post('logo');

    $data = array(
        'BOOTHNAME' => $boothname,
        'BOOTHDESC' => $boothdesc,
        'BOOTHLOGO' => NULL,
        'BOOTHUSERNAME' => $boothusername,
        'BOOTHPASSWORD' => md5($boothpassword)
    );

    $target_path = NULL;
    if($_FILES['logo']['name'] != NULL){

```

```

    $path_parts =
pathinfo($_FILES["logo"]["name"]);
    $extension = $path_parts['extension'];
    $extension = ".$extension";
    $target_path = "booth/";
    $target_path =
$target_path.$boothusername.$extension;
    $data['BOOTHLOGO'] =
base_url().$target_path;
}

    $this->load->model('M_booth');

    $added = $this->M_booth->update('BOOTH', $data,
$boothid);
    if($added){
        if($target_path != NULL){

move_uploaded_file( $_FILES['logo']['tmp_name'],
$target_path);
        }
    }

    redirect ('Booth/showAllBooth');
}
public function edit($boothid){
    $query = $this->db->get_where('BOOTH',
array('BOOTHID' => $boothid));
    if ($query->num_rows() > 0) {
        return $query->result_array();
    }
    else{
        return FALSE;
    }
}

public function update($table, $data, $boothid){

    $this->db->where('BOOTHID', $boothid);
    $query = $this->db->update($table, $data);

    if ($query) {
        return TRUE;
    }
}

```



```

else{
    return FALSE;
}
}

```

Kode Sumber 4.19. Implementasi Fungsi Mengubah Data

4.7.4 Implementasi proses menghapus data

Kode Sumber 4.20 merupakan implementasi dari proses menghapus data promo. Proses menghapus hanya bisa dilakukan pada data promo saja, data *booth* tidak mengamodasi fungsi penghapusan.

```

public function delete($promoid)
{
    $this->load->model('M_promo');
    $this->M_promo->delete($promoid);
    echo '<script language="javascript">';
    echo 'alert("Promo deleted
successfully!");';
    echo 'window.location.href = "' .
site_url('Promo/showAllPromo') . '";';
    echo '</script>';
}

public function delete($promoid){
    $this->db->delete('PROMO',
array('PROMOID' => $promoid));
}

```

Kode Sumber 4.20. Implementasi Fungsi Menghapus Data

(Halaman ini sengaja dikosongkan)

BAB V UJI COBA DAN EVALUASI

Pada bab ini dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan terhadap sistem yang dibuat. Pembahasan yang dipaparkan meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

5.1 Lingkungan Uji Coba

Lingkungan pengujian merupakan perangkat-perangkat yang dilibatkan dalam proses pengujian. Lingkungan pengujian ini menggunakan perangkat keras berupa perangkat bergerak berbasis Android yang memiliki fasilitas perangkat WiFi dan terhubung dengan internet. Spesifikasi lingkungan pengujian dijelaskan pada Tabel 5.1.

Tabel 5.1. Spesifikasi Lingkungan Pengujian Perangkat Lunak

Perangkat Bergerak Android	Processor	Qualcomm Snapdragon 615 Octa Core 64 bit
	RAM	2 GB
	Sistem Operasi	Android Lollipop 5.0
	WLAN	WiFi 802.11 a/b/g/n, <i>dual-band, hotspot, DLNA</i>

5.2 Dasar Pengujian

Pengujian yang dilakukan berupa pengujian fungsionalitas pada aplikasi perangkat bergerak maupun aplikasi *web* dan pengujian pada implementasi konsep *Indoor Localization*. Pengujian fungsionalitas dilakukan untuk menguji apakah semua fungsionalitas yang telah dirancang berjalan sesuai yang diinginkan. Untuk pengujian konsep *Indoor Localization* dilakukan pengujian akurasi dengan percobaan.

5.3 Skenario dan Hasil Uji Coba

Pada subbab ini dijelaskan skenario uji coba yang dilakukan dan hasil yang didapatkan.

5.3.1 Pengujian pada konsep indoor localization

Pengujian konsep *Indoor Localization* dilakukan dengan menguji terhadap uji coba keakuratan pendeteksian lokasi oleh sistem ketika diujicobakan pada keadaan yang sebenarnya.

Pada pengujian ini, pembagian *test area* berdasarkan *booth* yang sudah ada. Pada setiap *test area* dilakukan lima kali percobaan pendeteksian lokasi. Dimana dalam setiap percobaan pendeteksian lokasi, persentase kebenaran diambil dengan membagi jumlah ruangan yang terdeteksi benar dengan jumlah seluruh *test area*. Lokasi yang terdeteksi benar ditunjukkan dengan nilai 1 dan salah atau tidak terdeteksi dengan nilai 0. Pengujian dilakukan pada tanggal 8-9 Juni 2016 di East Coast Center pada waktu yang berbeda-beda. Detail waktu pengujian ditunjukkan oleh Tabel 5.2.

Tabel 5.2. Detail Waktu Pengujian

Pengujian ke-	Tanggal	Waktu
1	8 Juni 2016	11.00-13.00
2	8 Juni 2016	15.00-18.00
3	8 Juni 2016	20.00-22.00
4	9 Juni 2016	11.00-13.00
5	9 Juni 2016	15.00-18.00

Tabel 5.3. Hasil Pengujian 1

Nama <i>Booth</i> Sebenarnya	Nama <i>Booth</i> yang Muncul	Benar/Salah
House of Wok	House of Wok	1
X. O. Suki	X. O. Suki	1
Haagen Dasz	Chatime	0
Chatime	Chatime	1
Warung Ipang	Warung Ipang	1
Mokko Factory	Chatime	0
A&W	A&W	1

Solaria	Solaria	1
Informa	Informa	1
Starbucks	Starbucks	1
Dairy Queen	Dairy Queen	1
Excelso	Dairy Queen	0
Baskin Robbins	Baskin Robbins	1
Hypermart	Hypermart	1
JM Optical	JM Optical	1
Century	Century	1
Head2Toe	Century	0
Jumlah Posisi Terdeteksi Benar		13
Persentase Benar		76.47058824

Tabel 5.4. Hasil Pengujian 2

Nama Booth Sebenarnya	Nama Booth yang Muncul	Benar/Salah
House of Wok	House of Wok	1
X. O. Suki	X. O. Suki	1
Haagen Dasz	Chatime	0
Chatime	Chatime	1
Warung Ipang	Chatime	0
Mokko Factory	Chatime	0
A&W	A&W	1
Solaria	Solaria	1
Informa	Informa	1
Starbucks	Starbucks	1
Dairy Queen	Dairy Queen	1
Excelso	Dairy Queen	0
Baskin Robbins	Baskin Robbins	1
Hypermart	Hypermart	1

JM Optical	JM Optical	1
Century	Century	1
Head2Toe	Century	0
Jumlah Posisi Terdeteksi Benar		12
Persentase Benar		70.58824

Tabel 5.5. Hasil Pengujian 3

Nama Booth Sebenarnya	Nama Booth yang Muncul	Benar/Salah
House of Wok	House of Wok	1
X. O. Suki	X. O. Suki	1
Haagen Dasz	Chatime	0
Chatime	Chatime	1
Warung Ipang	Chatime	0
Mokko Factory	Chatime	0
A&W	Chatime	0
Solaria	Solaria	1
Informa	Informa	1
Starbucks	Starbucks	1
Dairy Queen	Dairy Queen	1
Excelso	Dairy Queen	0
Baskin Robbins	Hypermart	0
Hypermart	Hypermart	1
JM Optical	JM Optical	1
Century	Century	1
Head2Toe	Century	0
Jumlah Posisi Terdeteksi Benar		10
Persentase Benar		58.82353

Tabel 5.6. Hasil Pengujian 4

Nama Booth Sebenarnya	Nama Booth yang Muncul	Benar/Salah
House of Wok	House of Wok	1
X. O. Suki	X. O. Suki	1
Haagen Dasz	Chatime	0
Chatime	Chatime	1
Warung Ipang	Warung Ipang	1
Mokko Factory	Chatime	0
A&W	A&W	1
Solaria	Solaria	1
Informa	Informa	1
Starbucks	Starbucks	1
Dairy Queen	Dairy Queen	1
Excelso	Dairy Queen	0
Baskin Robbins	Baskin Robbins	1
Hypermart	Hypermart	1
JM Optical	JM Optical	1
Century	Century	1
Head2Toe	Head2Toe	1
Jumlah Posisi Terdeteksi Benar		14
Persentase Benar		82.35294

Tabel 5.7. Hasil Pengujian 5

Nama Booth Sebenarnya	Nama Booth yang Muncul	Benar/Salah
House of Wok	House of Wok	1
X. O. Suki	X. O. Suki	1
Haagen Dasz	Chatime	0

Chatime	Chatime	1
Warung Ipang	Warung Ipang	1
Mokko Factory	Chatime	0
A&W	Chatime	0
Solaria	Solaria	1
Informa	Informa	1
Starbucks	Starbucks	1
Dairy Queen	Dairy Queen	1
Excelso	Dairy Queen	0
Baskin Robbins	Baskin Robbins	1
Hypermart	Hypermart	1
JM Optical	JM Optical	1
Century	Century	1
Head2Toe	Head2Toe	1
Jumlah Posisi Terdeteksi Benar		13
Persentase Benar		76.47058824

Tabel 5.8. Persentase Benar Setiap *Booth* dari Semua Percobaan

Nama <i>Booth</i>	Persentase Benar (%)
House of Wok	100
X. O. Suki	100
Haagen Dasz	0
Chatime	100
Warung Ipang	60
Mokko Factory	0
A&W	60
Solaria	100
Informa	100
Starbucks	100
Dairy Queen	100
Excelso	0

Baskin Robbins	80
Hypermart	100
JM Optical	100
Century	100
Head2Toe	40

5.3.2 Pengujian fungsionalitas aplikasi perangkat bergerak

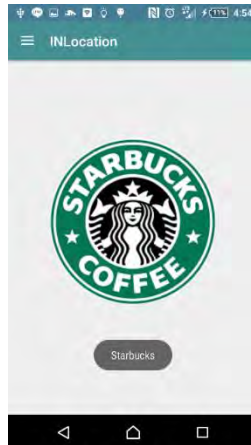
Pada subbab ini dibahas pengujian yang dilakukan pada aplikasi berbasis perangkat bergerak. Pengujian ini dilakukan oleh pengguna dengan tujuan untuk memeriksa apakah fungsionalitas pada perangkat telah terimplementasi dengan benar.

5.3.2.1 Pengujian fungsionalitas mendeteksi dan menampilkan lokasi

Berikut ini merupakan pembahasan pengujian untuk menampilkan lokasi. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.9.

Tabel 5.9. Skenario Pengujian Mendeteksi dan Menampilkan Lokasi

Nomor	P-A01
Nama	Menampilkan Lokasi Pengunjung
Use Case	UC-A01
Tujuan	Memeriksa fungsi untuk menampilkan lokasi pengguna berfungsi atau tidak.
Kondisi awal	Lokasi pengguna belum diketahui
Skenario	1. Pengguna membuka aplikasi 2. Pengguna menunggu proses setiap 10 detik sekali
Masukan	-
Keluaran yang diharapkan	Nama lokasi terdeteksi ditampilkan.
Hasil pengujian	Berhasil



Gambar 5.1. Pengujian Mendeteksi dan Menampilkan Lokasi

5.3.2.2 Pengujian fungsionalitas menampilkan booth terdekat

Berikut ini merupakan pembahasan pengujian untuk menampilkan *booth* terdekat. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.10.

Tabel 5.10. Skenario Pengujian Menampilkan *Booth* Terdekat

Nomor	P-A02
Nama	Menampilkan <i>Booth</i> Terdekat
Use Case	UC-A02
Tujuan	Memeriksa fungsi untuk menampilkan <i>booth</i> terdekat
Kondisi awal	<i>Booth</i> terdekat belum tampil
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu nearby <i>booth</i> 2. Pengguna menunggu proses menampilkan <i>booth</i> terdekat
Masukan	-
Keluaran yang diharapkan	<i>Booth</i> terdekat ditampilkan.
Hasil pengujian	Berhasil



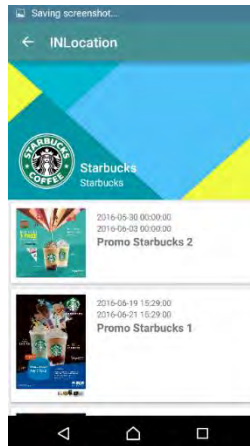
Gambar 5.2. Pengujian Menampilkan *Booth* Terdekat

5.3.2.3 Pengujian fungsionalitas menampilkan detail booth

Berikut ini merupakan pembahasan pengujian untuk menampilkan detail *booth*. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.11.

Tabel 5.11. Skenario Pengujian Mendeteksi Detail *Booth*

Nomor	P-A03
Nama	Menampilkan Detail <i>Booth</i>
Use Case	UC-A03
Tujuan	Memeriksa fungsi untuk menampilkan detail <i>booth</i> berfungsi atau tidak.
Kondisi awal	Detail <i>booth</i> belum tampil
Skenario	3. Pengguna memilih salah satu <i>booth</i> 4. Pengguna menunggu proses menampilkan detail <i>booth</i>
Masukan	-
Keluaran yang diharapkan	Detail <i>booth</i> tampil
Hasil pengujian	Berhasil



Gambar 5.3. Pengujian Menampilkan Detail Booth

5.3.2.4 Pengujian fungsionalitas menampilkan daftar promo

Berikut ini merupakan pembahasan pengujian untuk menampilkan daftar promo. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.12.

Tabel 5.12. Skenario Pengujian Menampilkan Daftar Promo

Nomor	P-A04
Nama	Menampilkan Daftar
Use Case	UC-A04
Tujuan	Memeriksa fungsi untuk menampilkan daftar promo
Kondisi awal	Daftar promo belum tampil
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu promo 2. Pengguna menunggu proses menampilkan daftar promo
Masukan	-
Keluaran yang diharapkan	Daftar promo
Hasil pengujian	Berhasil



Gambar 5.4. Pengujian Menampilkan Daftar Promo

5.3.3 Pengujian fungsionalitas aplikasi web

Pada subbab ini dibahas pengujian yang dilakukan pada aplikasi berbasis *web*. Pengujian pada aplikasi berbasis *web* dilakukan secara sistematis sekaligus melakukan dokumentasi. Pengguna aplikasi *web* adalah administrator dan pemilik *booth*.

5.3.3.1 Pengujian pengelolaan data promo

Pengujian pengelolaan data promo dilakukan dengan skenario pengguna memilih menu promo. Pengelolaan data promo dibagi menjadi beberapa proses yaitu memasukkan dan menghapus data. Skenario menambah data promo dijelaskan pada Tabel 5.13. Skenario mengubah data promo dijelaskan pada Tabel 5.14. Skenario menghapus promo dijelaskan pada Tabel 5.15.

Tabel 5.13. Skenario Pengujian Menambah Promo

Nomor	P-B01
Nama	Menambah data promo
Use Case	UC-B01
Tujuan	Memeriksa fungsi untuk menambah promo berfungsi atau tidak
Kondisi awal	Data promo pada <i>booth</i> belum ada

Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu menambah data promo. 2. Pengguna mengisi data promo pada form. 3. Pengguna menyimpan data dengan menekan tombol <i>save</i>.
Masukan	Data promo yang baru
Keluaran yang diharapkan	Data promo tersimpan
Hasil pengujian	Berhasil

The screenshot shows the 'Add Promo' form in the Promo Management application. The form is titled 'New Promo' and contains the following fields:

- Promo Description :** A text input field with a placeholder 'Promo Description'.
- Start Event Date :** A date and time picker showing '06/19/2016 3:23 PM'.
- End Event Date :** A date and time picker showing '06/19/2016 3:23 PM'.
- Promo Poster :** A file upload section with a 'Choose File' button and the text 'No file chosen'.

A 'Save' button is located at the bottom right of the form.

(a)

The screenshot shows the 'Promo Management' application displaying a list of added promo items. The table has the following columns:

PROMO ID	PROMO DESCRIPTION	PROMO PIC	PROMO START EVENT	PROMO END EVENT	EDIT	DELETE
8	Promo Starbucks 1	http://13.76.92.195:80/ta/promo/starbucks1.jpg	2016-05-30 11:54:00	2016-05-31 11:54:00		
9	Promo Starbucks 2	http://13.76.92.195:80/ta/promo/starbucks2.png	2016-05-30 00:00:00	2016-05-03 00:00:00		

The table also includes a header row for 'Add Promo' and a footer row with the same column headers.

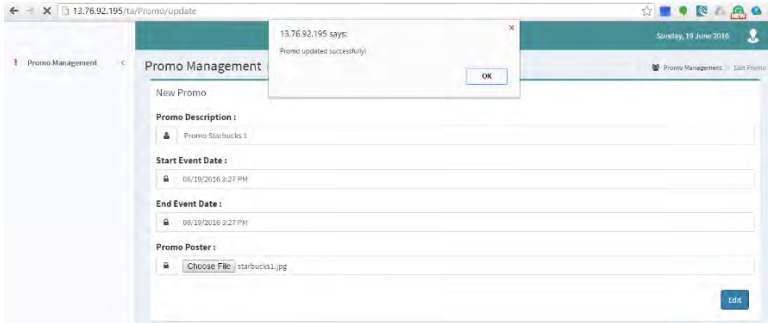
(b)

Gambar 5.5. (a,b) Pengujian Menambah Promo

Tabel 5.14. Skenario Pengujian Mengubah Promo

Nomor	P-B02
Nama	Mengubah Promo
Use Case	UC-B01
Tujuan	Memeriksa fungsi untuk mengubah promo berfungsi atau tidak
Kondisi awal	Promo telah tersimpan
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu untuk melihat daftar promo. 2. Pengguna memilih salah satu promo. 3. Pengguna mengisikan data promo yang ingin diubah pada form. 4. Pengguna menyimpan promo dengan menekan tombol <i>save</i>.
Masukan	Perubahan data promo.
Keluaran yang diharapkan	Promo terpilih telah berubah.
Hasil pengujian	Berhasil

(a)

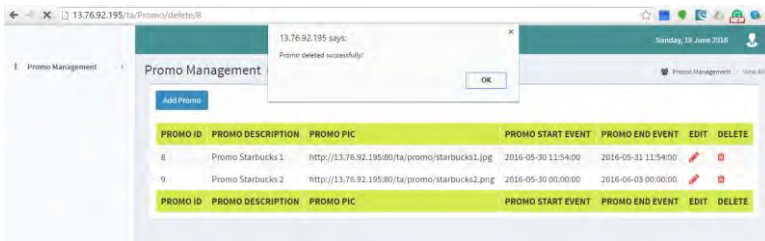


(b)

Gambar 5.6. (a,b) Pengujian Mengubah Promo

Tabel 5.15. Skenario Pengujian Menghapus Promo

Nomor	P-B03
Nama	Menghapus data promo.
Use Case	UC-B01
Tujuan	Memeriksa fungsi untuk menghapus data promo berfungsi atau tidak.
Kondisi awal	Data promo telah tersimpan
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menampilkan daftar promo. 2. Pengguna memilih salah satu data promo. 3. Pengguna memilih aksi menghapus data pada data terpilih. 4. Pengguna memberikan konfirmasi penghapusan data terpilih.
Masukan	Penghapusan data promo
Keluaran yang diharapkan	Data promo telah terhapus
Hasil pengujian	Berhasil



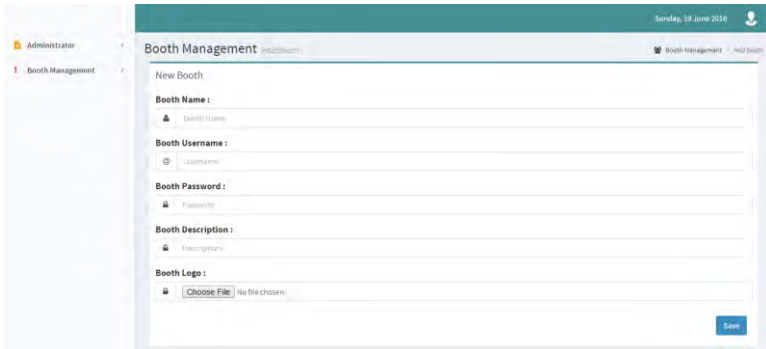
Gambar 5.7. Pengujian Menghapus Promo

5.3.3.2 Pengujian pengelolaan booth

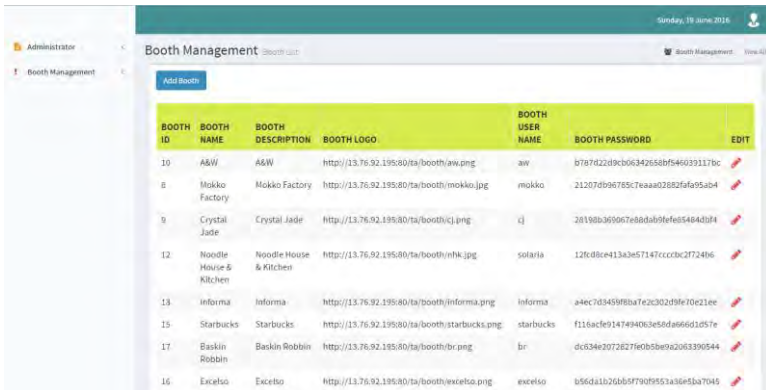
Pengujian pengelolaan *booth* dilakukan dengan skenario pengguna memilih menu pengelolaan *booth*. Pengelolaan *booth* dibagi menjadi beberapa proses yaitu memasukkan dan mengubah data. Skenario menambah *booth* dijelaskan pada Tabel 5.16 dan mengubah *booth* pada Tabel 5.17.

Tabel 5.16. Skenario Pengujian Menambah Booth

Nomor	P-B04
Nama	Menambah <i>Booth</i>
Use Case	UC-B02
Tujuan	Memeriksa fungsi untuk menambah <i>booth</i> berfungsi atau tidak
Kondisi awal	<i>Booth</i> belum ada
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih menu menambah <i>booth</i>. 2. Pengguna mengisikan <i>booth</i> pada form. 3. Pengguna menyimpan data dengan menekan tombol <i>save</i>.
Masukan	<i>Booth</i> yang baru
Keluaran yang diharapkan	<i>Booth</i> tersimpan
Hasil pengujian	Berhasil



(a)



(b)

Gambar 5.8. (a,b) Pengujian Menambah *Booth*

Tabel 5.17. Skenario Pengujian Mengubah *Booth*

Nomor	P-B05
Nama	Mengubah <i>Booth</i>
Use Case	UC-B02
Tujuan	Memeriksa fungsi untuk mengubah <i>booth</i> berfungsi atau tidak
Kondisi awal	<i>Booth</i> telah tersimpan
Skenario	5. Pengguna memilih menu untuk melihat daftar <i>booth</i> .

	6. Pengguna memilih salah satu <i>booth</i> 7. Pengguna mengisikan data <i>booth</i> yang ingin diubah pada form. 8. Pengguna menyimpan <i>booth</i> dengan menekan tombol <i>save</i> .
Masukan	Perubahan <i>booth</i>
Keluaran yang diharapkan	<i>Booth</i> terpilih telah berubah.
Hasil pengujian	Berhasil

(a)

(b)

Gambar 5.9. (a,b) Pengujian Mengubah *Booth*

5.4 Evaluasi

Berdasarkan hasil dari uji coba yang telah dilakukan pada implementasi konsep *Indoor Localization* dan uji fungsionalitas pada aplikasi, maka dapat dilakukan evaluasi sebagai berikut.

5.4.1 Evaluasi pengujian konsep indoor localization

Pengujian yang dilakukan terhadap konsep *Indoor Localization* dilakukan dengan menguji keakuratan hasil pendeteksian lokasi. Secara umum, sistem telah memberikan performa yang cukup baik dengan presentase rata-rata akurasi pendeteksian lokasi sebesar 73% diambil dari seluruh *test area* pada setiap lokasi uji coba. Meski demikian, masih terdapat lokasi yang tidak dapat terdeteksi dengan baik. Hal ini disebabkan karena sangat miripnya data sinyal WiFi yang ditangkap oleh *smartphone* saat melakukan pendeteksian lokasi.

Berdasarkan data uji coba, dapat dianalisis bahwa waktu mempengaruhi akurasi benar pada setiap percobaan. Dapat dilihat pada uji coba 1 dan uji coba 4 yang dilakukan pada siang hari menunjukkan hasil paling baik karena pengumpulan data *training* dilakukan pada rentang waktu jam 13.00-15.00. Selanjutnya uji coba 2 dan 4 yang dilakukan sore hari menunjukkan hasil yang tidak jauh berbeda. Sedangkan uji coba 3 terjadi anomali hasil akurasi yang sangat rendah apabila dibandingkan dengan uji coba lainnya. Hal ini terjadi dikarenakan waktu uji coba 3 dilakukan pada malam hari pada rentang jam 20.00-22.00 yang sangat jauh dari rentang waktu pengumpulan data *training*. Fluktuasi sinyal WiFi sangat berbeda pada kedua waktu tersebut.

Terdapat kasus dimana lokasi uji coba A yang terdeteksi sebagai lokasi uji coba B yang memiliki jarak yang sangat dekat dan tidak ada tembok pemisah dengan lokasi A. Hal ini bisa diakibatkan karena kemiripan sinyal Wi-Fi yang tertangkap saat pada saat uji coba. Hal inilah yang dapat mengakibatkan salahnya klasifikasi.

SIGNALSTRENGTH integer	BSSID character varying(50)	SCANID integer	SCANID integer	BOOTHID integer
-77	1c:74:0d:12:7b:f5	23	23	6
-73	30:b5:c2:76:f3:34	23	23	6
-75	30:b5:c2:76:f3:35	23	23	6
-77	60:e3:27:42:cc:64	23	23	6
-71	60:e3:27:42:cc:65	23	23	6
-64	64:70:02:88:01:90	23	23	6
-64	64:70:02:88:01:91	23	23	6
-69	64:70:02:88:01:92	23	23	6
-69	64:70:02:88:01:93	23	23	6
-79	e8:cc:18:a5:6d:22	23	23	6
-51	f8:01:13:d6:e0:94	23	23	6
-53	f8:01:13:d6:e0:95	23	23	6

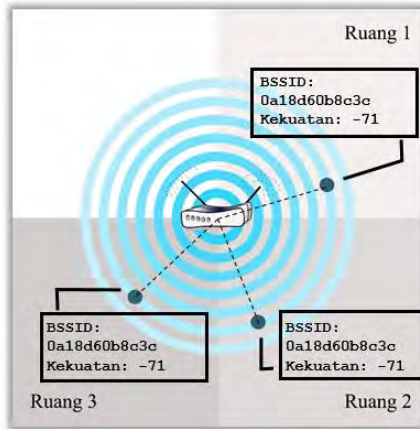
Gambar 5.10. Data kekuatan sinyal pada *booth* Chatime

SIGNALSTRENGTH integer	BSSID character varying(50)	SCANID integer	SCANID integer	BOOTHID integer
-80	30:b5:c2:76:f3:34	38	38	8
-78	30:b5:c2:76:f3:35	38	38	8
-80	30:b5:c2:f7:c0:a0	38	38	8
-80	30:b5:c2:f7:c0:a1	38	38	8
-88	64:70:02:88:01:90	38	38	8
-89	64:70:02:88:01:91	38	38	8
-89	64:70:02:88:01:92	38	38	8
-89	64:70:02:88:01:93	38	38	8
-80	e8:cc:18:a5:6d:22	38	38	8
-74	f8:01:13:d6:e0:94	38	38	8
-79	f8:01:13:d6:e0:95	38	38	8

Gambar 5.11. Data kekuatan sinyal pada *booth* Mokko

Gambar 5.10 dan Gambar 5.11 memperlihatkan data BSSID yang dimiliki kedua *booth* sangat mirip, apabila terjadi fluktuasi kekuatan sinyal, data kedua *booth* dapat menjadi sama sehingga akan terdapat dua data yang sama pada dua kelas yang berbeda dan mengakibatkan kesalahan klasifikasi.

Kasus yang kedua adalah apabila *access point* berada tepat di tengah antara dua atau tiga *booth*, karena pancaran sinyal WiFi yang memancar ke segala arah sehingga dimungkinkan adanya kekuatan sinyal WiFi yang sama dapat ditangkap pada *booth* yang berbeda. Ilustrasi dapat dilihat pada Gambar 5.12.



Gambar 5.12. Ilustrasi Pemancaran Sinyal dari Access Point

Secara umum, konsep *Indoor Localization* telah mendukung *Location Based Advertising* dengan cukup baik. Pada uji coba terhadap seluruh *test area* pada masing-masing lokasi uji coba, terdapat sepuluh lokasi yang telah memberikan performa secara penuh yakni 100%. Kemudian pada uji coba lokasi terhadap seluruh *test area* yang memberikan performa antara 80% - 99.99% terdapat satu lokasi, performa antara 60% - 79.99% terdapat dua lokasi, performa antara 40% - 59.99% terdapat satu lokasi, dan tiga lokasi dengan performa dibawah 40%.

5.4.2 Evaluasi pengujian fungsionalitas

Pengujian fungsionalitas yang telah dilakukan terhadap 6 fungsionalitas aplikasi perangkat bergerak dan *web* memberikan hasil yang sesuai dengan skenario yang telah direncanakan.

Evaluasi pengujian pada masing-masing fungsionalitas dijelaskan dengan sebagai berikut:

1. Pengujian mendeteksi dan menampilkan lokasi pengguna sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-A01 yang memberikan informasi bahwa proses mendeteksi dan menampilkan lokasi pengguna berjalan dengan benar.

2. Pengujian menampilkan *booth* terdekat sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-A02 yang memberikan informasi bahwa proses menampilkan *booth* terdekat berjalan dengan benar.
3. Pengujian menampilkan detail *booth* sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-A03 yang memberikan informasi bahwa proses menampilkan detail *booth* berjalan dengan benar.
4. Pengujian menampilkan daftar promo sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-A04 yang memberikan informasi bahwa proses menampilkan semua daftar promo berjalan dengan benar.
5. Pengujian mengelola data promo sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-B01 yang memberikan informasi bahwa proses mengelola data promo berjalan dengan benar.
6. Pengujian mengelola data *booth* sesuai dengan yang diharapkan. Kondisi ini diperlihatkan pada pengujian P-B02 yang memberikan informasi bahwa proses pengelolaan data *booth* berjalan dengan benar.

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Lampiran 1: Adjacency Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0
2	1	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0
3	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
4	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
7	0	0	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	1	1	0	1	1	1	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0
12	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1
16	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

(Halaman ini sengaja dikosongkan)

BAB VI KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dalam proses pengerjaan Tugas Akhir dari tahap pendahuluan, kajian pustaka, analisis, perancangan, implementasi dan pengujian sistem layanan pencatatan promo berbasis mobile yang didukung oleh konsep *Indoor Localization* dengan *Support Vector Machine* (SVM) ini, diperoleh kesimpulan sebagai berikut:

1. Proses *sampling* kekuatan sinyal WiFi pada setiap ruangan dilakukan dengan dilakukan lima kali *scanning* data WiFi untuk merepresentasikan fluktuasi sinyal WiFi di titik random di dalam *booth*. Sedangkan uji coba untuk mengetahui keakuratan pendeteksian lokasi dapat dilakukan dengan melakukan lima kali percobaan pada setiap *booth* untuk memastikan hasil dari pendeteksian lokasi.
2. Proses pendeteksian lokasi menggunakan algoritma SVM dan informasi sinyal WiFi yang mengimplementasi konsep *Indoor Localization* pada Tugas Akhir ini berhasil mendukung aplikasi periklanan berbasis lokasi *indoor*. Hal ini ditunjukkan dengan hasil lima kali uji coba dengan rata-rata persentase benar sebesar 73%, sementara menunjukkan pendeteksian lokasi sangat bagus apabila tidak pada *booth* yang terbuka.
3. Dari data-data yang diperoleh, dapat disimpulkan bahwa seluruh lokasi yang diuji coba pada Tugas Akhir ini telah terjangkau oleh beberapa *access point*. Hanya saja, karena East Coast Center sebagai lokasi uji coba banyak memiliki *booth* yang tidak terpisahkan oleh tembok, maka sinyal-sinyal WiFi yang didapatkan antar *booth* menjadi sangat mirip dan data tersebut sangat mengganggu pada saat tahap klasifikasi.
4. Aplikasi perangkat bergerak yang dibangun dengan Android Studio dan *web service* dengan CodeIgniter dapat menampilkan promosi sesuai dengan lokasi pengunjung

dengan dukungan konsep *Indoor Localization* menggunakan sinyal WiFi dan *Support Vector Machine*.

5. Aplikasi *web* dapat melakukan pengelolaan data promosi dan *booth*.

6.2 Saran

Berikut ini merupakan beberapa saran mengenai pengembangan lebih lanjut aplikasi berdasarkan hasil rancangan, implementasi dan uji coba yang telah dilakukan.

1. Melakukan kerjasama dengan pihak pusat perbelanjaan agar bisa mendapatkan informasi promosi yang *up-to-date*.
2. Untuk *booth* yang terbuka, disarankan untuk menggunakan sinyal selain WiFi karena jangkauan sinyalnya cukup jauh, akan lebih baik menggunakan sinyal *Bluetooth* agar sinyal tidak terlalu mirip satu dengan yang lain.
3. Melakukan perbandingan dengan menggunakan metode *multi-class SVM* yang lain, seperti *One-Against-One* atau *One-Against-All* dan mencatat kelebihan dan kekurangannya dibandingkan dengan metode *Binary Tree*.
4. Proses mendeteksi *booth* terdekat dapat memanfaatkan rumus *Euclidean Distance* untuk mendapat jarak antar *booth*, namun sebelumnya harus menentukan koordinat relatif dari setiap *booth*.
5. Memperbanyak jumlah dan variasi data sampel.
6. Menambahkan fitur peta dan navigasi untuk menuju ke suatu *booth*.

DAFTAR PUSTAKA

- [1] R. W. Purbojati, "UI - Skripsi (Membership): Perpustakaan Universitas Indonesia," 2004. [Online]. Available: <http://lib.ui.ac.id/opac/ui/detail.jsp?id=124252&lokasi=lokal>.
- [2] A. Salim and J. Irawan, "STIKOM INSTITUTIONAL REPOSITORIES," 2008. [Online]. Available: <http://sir.stikom.edu/414/1/2008-II-99.pdf>. [Accessed 1 October 2014].
- [3] Y. Feng, J. Minghua, L. Jing, Q. Xiao, H. Ming, P. Tao and H. Xinrong, "An Improved Indoor Localization of WiFiBased on *Support Vector Machine*," *International Journal of Future Generation Communication and Networking*, vol. 7, pp. 191-206, 2014.
- [4] S. Steiniger, N. Moritz and E. Alistar, *Foundation of Location Based Services*, 2005.
- [5] K. A. Nastiti, R. V. H. Ginardi and A. Munif, "Sistem Layanan Pencatatan Kehadiran Mahasiswa Berbasis Mobile dengan Mengimplementasikan Indoor Localization Menggunakan Sinyal Wi-Fi dan Decision Tree di Kampus Teknik Informatika ITS," *JURNAL TEKNIK ITS*, vol. 4, 2015.
- [6] P. U. Jakarta, "Pdf: Perpustakaan UPN Jakarta," [Online]. Available: <http://www.library.upnvj.ac.id/pdf/3tipdf/206511001/bab2.pdf>. [Accessed 1 October 2014].

- [7] "Code Igniter," EllisLab, 2016. [Online]. Available: <https://www.codeigniter.com/>. [Accessed 20 February 2016].
- [8] G. Madzarov, D. Gjorgjevikj and I. Chorbev, "A Multi-class SVM Classifier Utilizing Binary Decision Tree," *Informatika*, vol. 33, pp. 233-241, 2009.
- [9] M. F. Ghanianto, "Implementasi Indoor Localization Menggunakan Sinyal Wi-Fi dan Clustering Filtered K-Nearest Neighbors untuk Pelacakan Keberadaan Seseorang dan Evaluasi Akurasi Pelacakan di Kampus Teknik Informatika ITS," Surabaya, 2015.
- [10] C.-W. Hsu, C.-C. Chang and C.-J. Lin, "A Practical Guide to Support Vector Classification," 2016.

BIODATA PENULIS



Ratih Ayu Indraswari, lahir di Malang pada tanggal 19 September 1994. Anak pertama dari 3 bersaudara. Penulis menempuh pendidikan di TK Kartika IV-5, selanjutnya melanjutkan pendidikan di SD Kartika IV-7, SMP Negeri 3 Malang, dan SMA Negeri 3 Malang. Penulis melanjutkan pendidikan sarjana di Teknik Informatika Institut Teknologi Sepuluh Nopember Surabaya. Dalam menyelesaikan pendidikan S1 penulis

mengambil rumpun matakuliah Algoritma dan Pemrograman.

Penulis juga aktif dalam organisasi kemahasiswaan kampus diantaranya Staff Hubungan Luar pada periode 2013/2014 dan Sekretaris Umum periode 2014/2015 Himpunan Mahasiswa Teknik Computer-Informatika (HMTC). Pernah menjadi asisten dosen Sistem Basis Data dan sedang menjadi administrator Laboratorium Pemrograman.