

**RANCANG BANGUN APLIKASI *FINDING* DOSEN UNTUK
MENCARI POSISI DOSEN FILKOM DI DALAM RUANGAN
MENGUNAKAN METODE
*COLOR RADIOMAP INTERPOLATION***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Miftahul Rizki Purwonegoro

NIM: 145150201111112



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

RANCANG BANGUN APLIKASI *FINDING* DOSEN UNTUK Mencari Posisi Dosen
FILKOM DI DALAM RUANGAN MENGGUNAKAN METODE
COLOR RADIOMAP INTERPOLATION

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

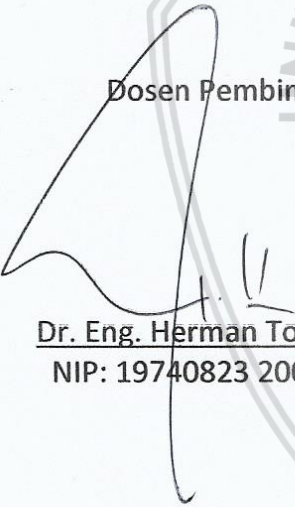
Disusun Oleh :
Miftahul Rizki Purwonegoro
NIM: 145150201111112

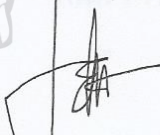
Skripsi ini telah diuji dan dinyatakan lulus pada
2 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

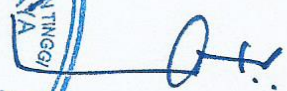

Dr. Eng. Herman Tolle, S.T, M.T
NIP: 19740823 200012 1 001


Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.
NIK: 2016078912041001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 21 Agustus 2018



Miftahul Rizki Purwonegoro
NIM: 14515020111112



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat, taufik, hidayah dan inayah-Nya penulis dapat menyelesaikan penelitian dan laporan Skripsi sebagai salah satu syarat memperoleh gelar sarjana yang berjudul **Rancang Bangun Aplikasi Finding Dosen Untuk Mencari Posisi Dosen Filkom di Dalam Ruang Menggunakan Metode *Color Radiomap Interpolation***.

Banyak hambatan dalam pelaksanaan penelitian hingga penulisan Laporan Skripsi yang penulis hadapi. Banyak pula yang memberi semangat, dorongan, bantuan serta bimbingannya sehingga kendala dan hambatan yang penulis hadapi dapat teratasi. Ucapan terima kasih penulis sampaikan kepada :

1. Bapak Adi Purwanto, Ibu Komariah Nurlaili, S.Pd selaku orang tua penulis yang tiada henti memberikan dorongan berupa do'a dan materi selama penulis melakukan penelitian ini.
2. Bapak Dr. Eng. Herman Tolle, S.T, M.T selaku Pembimbing I yang telah menyisihkan waktunya, banyak membantu, memberikan arahan dan bimbingan selama pengerjaan skripsi.
3. Bapak Mahardeka Tri Ananta, S.Kom., M.T., M.Sc. selaku Pembimbing II yang banyak membantu, memberikan saran, arahan dan bimbingan selama pengerjaan skripsi.
4. Bapak Wibisono Sukmo Wardhono, S.T, M.T dan Ibu Hanifah Muslimah Az-Zahra, S.Sn., M.Ds. selaku Dewan penguji atas segala dukungan, saran dan masukan dalam penyempurnaan dokumen skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika.
7. Bapak Agus Wahyu Widodo, S.T, M.Cs selaku Ketua Program Studi Teknik Informatika.
8. Bapak Faiz Al Huda, S.Kom., M.Kom dan Ivan Yulfrian, S.Kom yang telah membantu memberikan arahan dan masukan pada saat penulis melakukan penelitian.
9. Khairul Anwar, Yosua Wisnu P, Dicky Yoga, Arjun, Tarra dan teman-temanku satu kos 86 yang selalu mengisi hari-hari penulis untuk menemani dan menyemangati dalam menyelesaikan penulisan skripsi ini.
10. Ulfa Gita Anjeli yang selalu menyemangati setiap pagi dalam menyelesaikan skripsi.
11. Mochammad Adhy, Danang Aditya, Iqbal Pratama yang telah memberikan bantuan pada saat penulis melakukan penelitian.

12. Teman-teman BIOS, CDLE, Teknik Informatika 2014 yang selalu memotivasi penulis untuk segera menyelesaikan skripsi ini.

Akhir kata penulis menyadari bahwa penyusunan skripsi ini masih terdapat banyak kekurangan yang perlu disempurnakan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat penulis harapkan. Semoga skripsi ini bermanfaat bagi semua pihak yang membacanya.

Malang, 21 Agustus 2018

Penulis
mrizkip@hotmail.com



ABSTRAK

Miftahul Rizki Purwonegoro, Rancang Bangun Aplikasi Finding Dosen Untuk Mencari Posisi Dosen Filkom di Dalam Ruangan Menggunakan Metode *Color Radiomap Interpolation*.

Dosen Pembimbing: Dr. Eng. Herman Tolle, S.T, M.T dan Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.

Keberadaan mahasiswa dan dosen merupakan komponen utama kegiatan belajar dalam sebuah institusi pendidikan. Banyak mahasiswa yang mengalami kendala untuk mencari posisi dosen yang berada di kampus. Selain aktivitas mengajar, dosen juga memiliki tugas lain dalam menjalankan Tri Dharma Perguruan Tinggi, hal ini yang menjadi kendala mahasiswa untuk mencari keberadaan dosen yang tidak selalu di ruangnya. Penentuan posisi seseorang pada umumnya menggunakan teknologi GPS (*Global Positioning System*), namun GPS memiliki kelemahan yaitu tidak dapat menentukan posisi seseorang yang berada di dalam ruangan. Untuk menangani masalah tersebut, penelitian ini dilakukan untuk membangun aplikasi yang dapat mencari posisi dosen yang berada di dalam ruangan menggunakan metode *Color Radiomap Interpolation*. Pengembangan dilakukan dengan menggunakan nilai *Received Signal Strength Indicator* (RSSI) dari 3 buah *Access Point Reference*. Nilai RSSI ini dikonversikan menjadi peta warna yang nantinya digunakan untuk menentukan posisi dosen. Aplikasi ini dikembangkan dalam dua sisi dimana sisi dosen dapat merubah statusnya menjadi aktif untuk dapat ditemukan posisinya, dan sisi mahasiswa yang dapat mencari dosen dan melihat posisinya. Penentuan posisi dilakukan dengan membandingkan data latih dalam peta warna dengan data uji ketika perubahan status terjadi dengan mencari nilai terkecil dari persamaan *Mean Squared Error*. Hasil penelitian menunjukkan semua kebutuhan dapat tervalidasi, dengan tingkat akurasi mencapai 90%. Hal ini menandakan hasil dari penelitian memiliki tingkat akurasi yang baik.

Kata kunci: pencarian lokasi, *GPS*, *Indoor Positioning System*, *Color Radiomap Interpolation*

ABSTRACT

Miftahul Rizki Purwonegoro, Development of Finding Lecture Application to Find the Position of Filkom Lecture Inside Room Using Color Radiomap Interpolation Method.

Supervisors: Dr. Eng. Herman Tolle, S.T, M.T dan Mahardeka Tri Ananta, S.Kom., M.T., M.Sc.

The existence of student and lecture is a main component from study in educational institution. Many student who has an obstacle experiences to looking for lectures position in campus. In addition to lecturing, lecture is also have another duties to do *Tri Dharma Perguruan Tinggi*, it becomes an obstacle for student to looking for lectures position where not always located in their room. In general, to determine someone position using GPS (*Global Positioning System*), however GPS have also a weakness such as can't determine someone inside the building. Therefore, this study is to build an application that can find lectures position inside the building using *Color Radiomap Interpolation* method. Received Signal Strength Indicator (RSSI) from 3 Access Point is used to develop this application. The value from RSSI is converted turn into a color map that used to determine lecture position. This application is developed in two sides where for the lecture is used to change the status to can be found its position, for the students is also can find the lecture and see the position. Determining position with compare training data in the color map by test data when the status from lecturer is changed with collecting the smallest data from the equation of *Mean Squared Error*. The result is show that all of the requirement can be validated, with with accuracy 90%. This indicated the result from all of the study have a good level of accuracy.

Keywords: *location searching, GPS, Indoor Positioning System, Color Radiomap Interpolation*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian.....	2
1.4 Manfaat Penelitian	3
1.5 Batasan Penelitian	3
1.6 Sistematika Pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Kajian Pustaka.....	5
2.2 Landasan Teori.....	6
2.2.1 Global Positioning System (GPS).....	6
2.2.2 Wireless Fidelity (Wi-Fi)	7
2.2.2.1 Karakteristik Wifi	7
2.2.3 Trilateration	8
2.2.4 Teknik Positioning berdasarkan kekuatan sinyal wifi	9
2.2.5 Metode Color Radiomap Interpolation	9
2.2.6 Peta Gedung FILKOM Universitas Brawijaya	14
BAB 3 METODOLOGI	19
3.1 Studi Literatur.....	20
3.2 Analisis Kebutuhan	20
3.3 Pengambilan Data dan Pengolahan Data.....	20

3.4 Perancangan dan Implementasi.....	21
3.5 Pengujian dan Analisis Hasil	21
3.6 Kesimpulan dan Saran	21
BAB 4 ANALISIS KEBUTUHAN	22
4.1 Gambaran Umum Sistem	22
4.2 Deskripsi Umum Perangkat Lunak.....	22
4.3 Identifikasi Aktor	22
4.4 Kebutuhan Fungsional	23
4.5 Use Case Diagram.....	24
4.6 Use Case Scenario.....	25
4.6.1 Use Case Scenario Login: SRS-FIND-01	25
4.6.2 Use Case Scenario Register: SRS-FIND-02.....	26
4.6.3 Use Case Scenario Melihat Profil: SRS-FIND-03.....	26
4.6.4 Use Case Scenario Mengubah Profil: SRS-FIND-04.....	27
4.6.5 Use Case Scenario Logout: SRS-FIND-05.....	27
4.6.6 Use Case Scenario Melihat Daftar Dosen: SRS-FIND-06.....	28
4.6.7 Use Case Scenario Mencari Dosen: SRS-FIND-07	28
4.6.8 Use Case Scenario Melihat Profil Dosen: SRS-FIND-08.....	28
4.6.9 Use Case Scenario Melihat Posisi Dosen: SRS-FIND-09.....	29
4.6.10 Use Case Scenario Mengubah Status: SRS-FIND-10	29
4.7 Sequence Diagram.....	30
4.7.1 Sequence Diagram Login (SRS-FIND-01).....	30
4.7.2 Sequence Diagram Register (SRS-FIND-02)	31
4.7.3 Sequence Diagram Melihat Profil (SRS-FIND-03)	31
4.7.4 Sequence Diagram Mengubah Profil (SRS-FIND-04)	32
4.7.5 Sequence Diagram Logout (SRS-FIND-05)	32
4.7.6 Sequence Diagram Melihat Daftar Dosen (SRS-FIND-06).....	33
4.7.7 Sequence Diagram Mencari Dosen (SRS-FIND-07).....	33
4.7.8 Sequence Diagram Melihat Profil Dosen (SRS-FIND-08)	34
4.7.9 Sequence Diagram Melihat Posisi Dosen (SRS-FIND-09).....	34
4.7.10 Sequence Diagram Mengubah Status (SRS-FIND-10).....	35
BAB 5 PENGAMBILAN DAN PENGOLAHAN DATA	36

5.1 Pengambilan Data	36
5.1.1 Peletakkan Access Point	36
5.1.2 Proses Pengambilan Data	39
5.2 Pengolahan Data	40
5.2.1 Data Sampel	41
BAB 6 PERANCANGAN DAN IMPLEMENTASI	44
6.1 Perancangan	44
6.1.1 Class Diagram	44
6.1.2 Physical Data Model	45
6.1.3 Perancangan Arsitektur	46
6.1.4 Perancangan Antar Muka	47
6.2 Implementasi	51
6.2.1 Implementasi Kode Program Back End	51
6.2.2 Implementasi Kode Program Front End Android	57
6.2.3 Implementasi Antar Muka Aplikasi	62
BAB 7 PENGUJIAN DAN ANALISIS HASIL	67
7.1 Pengujian Unit	67
7.1.1 Method Lihat Daftar Dosen	67
7.1.2 Method Lihat Posisi Dosen	68
7.1.3 Method Ubah Status Dosen	69
7.2 Hasil Pengujian Unit	70
7.2.1 Method Lihat Daftar Dosen	70
7.2.2 Method Lihat Posisi Dosen	70
7.2.3 Method Ubah Status Dosen	70
7.3 Analisis Hasil Pengujian Unit	71
7.4 Pengujian Validasi	71
7.4.1 Pengujian Kebutuhan Fungsional	71
7.5 Hasil Pengujian Validasi	72
7.5.1 Hasil Pengujian Kebutuhan Fungsional	72
7.6 Analisis Hasil Pengujian Validasi	74
7.7 Pengujian Akurasi	74
7.8 Hasil Pengujian Akurasi	74



7.9 Analisis Hasil Pengujian Akurasi 78

BAB 8 PENUTUP 80

8.1 Kesimpulan 80

8.2 Saran 80

DAFTAR PUSTAKA..... 81

LAMPIRAN 82



DAFTAR TABEL

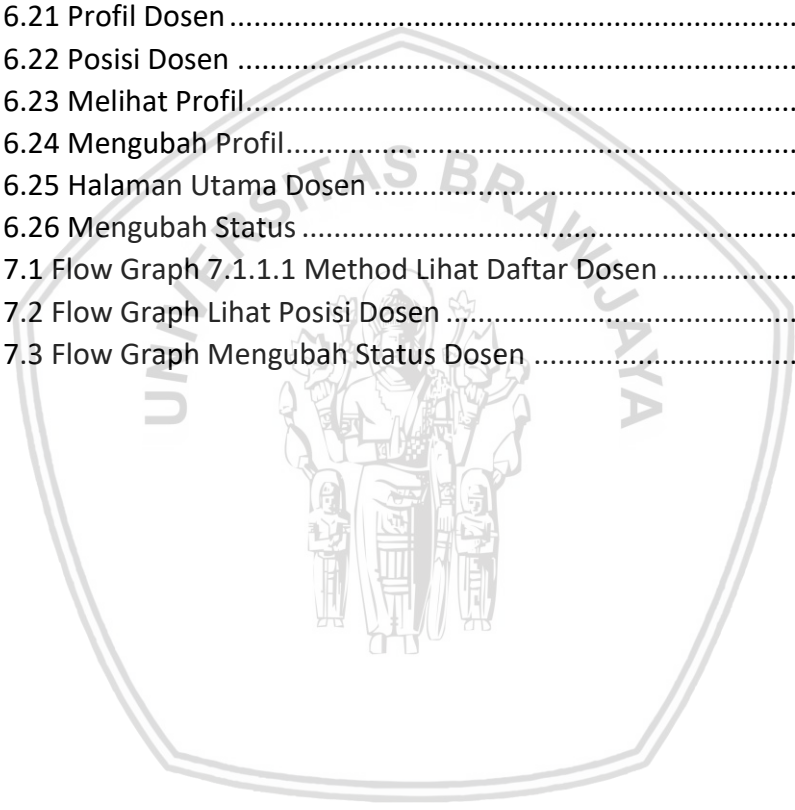
Tabel 4.1 Deskripsi Aktor	23
Tabel 4.2 Kebutuhan Fungsional	23
Tabel 4.3 Use Case Scenario Login	25
Tabel 4.4 Use Case Scenario Register	26
Tabel 4.5 Use Case Scenario Melihat Profil	26
Tabel 4.6 Use Case Scenario Mengubah Profil	27
Tabel 4.7 Use Case Scenario Logout	27
Tabel 4.8 Use Case Scenario Melihat Daftar Dosen	28
Tabel 4.9 Use Case Scenario Mencari Dosen	28
Tabel 4.10 Use Case Scenario Melihat Profil Dosen	28
Tabel 4.11 Use Case Scenario Melihat Posisi Dosen	29
Tabel 4.12 Use Case Scenario Mengubah Status	29
Tabel 7.1 Hasil Pengujian Unit Method Lihat Daftar Dosen	70
Tabel 7.2 Hasil Pengujian Unit Method Lihat Posisi Dosen	70
Tabel 7.3 Hasil Pengujian Unit Method Ubah Status Dosen	70
Tabel 7.4 Pengujian Kebutuhan Fungsional	71
Tabel 7.5 Tabel Hasil Pengujian Kebutuhan Fungsional	72
Tabel 7.6 Pengujian Akurasi	74
Tabel 7.7 Data Pengujian Akurasi Sinyal Wifi di A1.8 A	74
Tabel 7.8 Pengujian Akurasi Rata-rata dan pembulatan	75
Tabel 7.9 Pengujian Akurasi Normalisasi	75
Tabel 7.10 Pengujian Akurasi Pelevelan RGB	76
Tabel 7.11 Pengujian Akurasi Mean Squared Error	76
Tabel 7.12 Hasil Pengujian Akurasi	77

DAFTAR GAMBAR

Gambar 2.1 Trilateration 2D	8
Gambar 2.2 Data flow mekanisme pengolahan data.	10
Gambar 2.3 Pembuatan peta Color Radiomap menggunakan kombinasi 3 sinyal Akses Point	11
Gambar 2.4 Ilustrasi Grid Fusi pada interpolasi warna.....	12
Gambar 2.5 Denah Gedung A lantai 1 FILKOM Universitas Brawijaya	14
Gambar 2.6 Denah Gedung A lantai 2 FILKOM Universitas Brawijaya	14
Gambar 2.7 Denah Gedung E lantai 1 FILKOM Universitas Brawijaya	15
Gambar 2.8 Denah Gedung E lantai 2 FILKOM Universitas Brawijaya	15
Gambar 2.9 Denah Gedung F lantai 2 FILKOM Universitas Brawijaya.....	16
Gambar 2.10 Gedung F lantai 3 FILKOM Universitas Brawijaya	17
Gambar 2.11 Gedung F lantai 4 FILKOM Universitas Brawijaya	18
Gambar 4.1 Gambaran umum sistem.....	22
Gambar 4.2 Use Case Diagram.....	25
Gambar 4.3 Sequence Diagram Login.....	30
Gambar 4.4 Sequence Diagram Register	31
Gambar 4.5 Sequence Diagram Melihat Profil	31
Gambar 4.6 Sequence Diagram Mengubah Profil	32
Gambar 4.7 Sequence Diagram Logout	32
Gambar 4.8 Sequence Diagram Melihat Daftar Dosen.....	33
Gambar 4.9 Sequence Diagram Mencari Dosen	33
Gambar 4.10 Sequence Diagram Melihat Profil Dosen	34
Gambar 4.11 Sequence Diagram Melihat Posisi Dosen.....	34
Gambar 4.12 Sequence Diagram Mengubah Status.....	35
Gambar 5.1 Peta AP Gedung A Lantai 1 FILKOM Universitas Brawijaya	36
Gambar 5.2 Peta AP Gedung A Lantai 2 FILKOM Universitas Brawijaya	36
Gambar 5.3 Peta AP Gedung E Lantai 1 FILKOM Universitas Brawijaya.....	37
Gambar 5.4 Peta AP Gedung E Lantai 2 FILKOM Universitas Brawijaya.....	37
Gambar 5.5 Peta AP Gedung F Lantai 2 FILKOM Universitas Brawijaya.....	38
Gambar 5.6 Peta AP Gedung F Lantai 3 FILKOM Universitas Brawijaya.....	38
Gambar 5.7 Peta AP Gedung F Lantai 4 FILKOM Universitas Brawijaya	39
Gambar 6.1 Class Diagram Aplikasi Finding Dosen	44
Gambar 6.2 Class Diagram Backend Finding Dosen.....	45
Gambar 6.3 Physical Data Model Finding Dosen	46
Gambar 6.4 Arsitektur Komponen Aplikasi Android.....	46
Gambar 6.5 Rancangan Antar Muka Halaman Login	47
Gambar 6.6 Rancangan Antar Muka Halaman Utama Dosen.....	48
Gambar 6.7 Rancangan Antar Muka Daftar Dosen.....	48
Gambar 6.8 Rancangan Antar Muka Profil Dosen	49
Gambar 6.9 Rancangan Antar Muka Pencarian Posisi Dosen.....	49



Gambar 6.10 Rancangan Antar Muka Halaman Profil	50
Gambar 6.11 Rancangan Antar Muka Mengubah Status	50
Gambar 6.12 Kode Program Melihat Posisi Dosen Back End	51
Gambar 6.13 Kode Program Melihat Daftar Dosen Back End	52
Gambar 6.14 Kode Program Mengubah Status Back End.....	57
Gambar 6.15 Kode Program Melihat Daftar Dosen Android.....	58
Gambar 6.16 Kode Program Melihat Posisi Dosen Android	59
Gambar 6.17 Kode Program Merubah Status Android.....	61
Gambar 6.18 Login	62
Gambar 6.19 Register.....	63
Gambar 6.20 Daftar Dosen	63
Gambar 6.21 Profil Dosen	64
Gambar 6.22 Posisi Dosen	64
Gambar 6.23 Melihat Profil.....	65
Gambar 6.24 Mengubah Profil.....	65
Gambar 6.25 Halaman Utama Dosen	66
Gambar 6.26 Mengubah Status	66
Gambar 7.1 Flow Graph 7.1.1.1 Method Lihat Daftar Dosen	67
Gambar 7.2 Flow Graph Lihat Posisi Dosen	68
Gambar 7.3 Flow Graph Mengubah Status Dosen	69



DAFTAR LAMPIRAN

LAMPIRAN A KUISIONER	82
LAMPIRAN B HASIL KUISIONER	83
LAMPIRAN C HASIL KUISIONER (LANJUTAN).....	84
LAMPIRAN D HASIL KUISIONER (LANJUTAN)	85
LAMPIRAN E DATA TRAINING PETA WARNA	86
E.1 Data Training Gedung A Lantai 1	86
E.2 Data Training Gedung A Lantai 2	87
E.3 Data Training Gedung E Lantai 1	89
E.4 Data Training Gedung E Lantai 2	90
E.5 Data Training Gedung F Lantai 2	92
E.6 Data Training Gedung F Lantai 3	93
E.7 Data Training Gedung F Lantai 4	95
LAMPIRAN F PETA WARNA	97
F.1 Peta Warna Gedung A Lantai 1	97
F.2 Peta Warna Gedung A Lantai 2.....	98
F.3 Peta Warna Gedung E Lantai 1	99
F.4 Peta Warna Gedung E Lantai 2	100
F.5 Peta Warna Gedung F Lantai 2	101
F.6 Peta Warna Gedung F Lantai 3	102
F.7 Peta Warna Gedung F Lantai 4	103

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Keberadaan mahasiswa dan dosen merupakan komponen utama kegiatan belajar dalam sebuah institusi pendidikan. Menurut hasil survey yang dilakukan peneliti, 88,9% dari 45 mahasiswa mengalami kendala untuk mencari posisi dosen yang berada di kampus. Selain aktivitas mengajar, dosen juga memiliki tugas lain dalam menjalankan Tri Dharma Perguruan Tinggi, hal ini yang menjadi kendala mahasiswa mencari keberadaan dosen untuk melakukan bimbingan ataupun mencari tanda tangan baik sifatnya mendesak ataupun tidak, terkadang dosen hadir di kampus tetapi tidak berada di ruangnya. Mengetahui lokasi seseorang dari perangkat *mobile* di dalam ruangan, adalah sebuah kebutuhan untuk layanan berbasis lokasi. Pada umumnya aplikasi yang digunakan untuk menentukan suatu lokasi adalah GPS (*Global Positioning System*) dimana GPS memiliki kelebihan untuk memetakan seluruh permukaan bumi, bahkan dapat memberikan layanan pemetaan 3D. Namun GPS memiliki kelemahan jika digunakan untuk menentukan posisi seseorang yang berada di dalam ruangan atau gedung.

Dalam pembangunan konteks aplikasi modern, pentingnya penentuan lokasi dalam ruangan (*Indoor Positioning System*) telah menjadi hal yang tidak bisa ditutupi atau dibantahkan. Hal ini perlu disebutkan bahwa persyaratan untuk mencari secara akurat dan mengambil data posisional telah berkembang dengan munculnya kendaraan robot canggih. Meskipun inovasi yang inkremental, sistem penentuan posisi dalam ruangan kontemporer (*Indoor Positioning System / IPS*) menderita beberapa kelemahan sistematis. Selain tingginya biaya pelaksanaan dan persyaratan *hardware* yang kompleks, hari ini IPS gagal untuk memberikan level meter yang presisi (Rashid, 2016).

Penelitian yang pertama dilakukan oleh Rashid (2016), mengusulkan sebuah penentuan lokasi dalam ruangan yang unik yang menggunakan jaringan Wi-Fi dengan *repeater*. Peneliti menggunakan dua metode populer untuk menentukan lokasi dari *Wi-Fi positioning system* adalah *Time of Arrival (TOA)* dan *Angle of Arrival (AOA)*. TOA membutuhkan tiga Akses Poin (AP) untuk mendapatkan kunci lokasi dan menggunakan metode Triangulasi yang kadang-kadang memberikan keluar perkiraan wilayah yang melingkar daripada set koordinat yang tepat. Untuk AOA, akurasi posisi yang sangat dinegasikan dengan meningkatkan ruang, AOA bekerja baik di daerah kecil (Rashid, 2016).

Widyaman (2012) telah mengembangkan sebuah sistem yang dapat mengakses nilai *Received Signal Strength Indication (RSSI)*, membandingkan dan memvalidasi dengan daftar data historis dan titik-titik lokasi secara visual pada peta menggunakan algoritma definitif. Dengan menggunakan nilai *Received Signal Strength Indication (RSSI)* sebagai metode untuk sistem penentu lokasi dalam gedung adalah tidak memerlukan sinkronisasi antara pemancar (AP) dan penerima sehingga mudah untuk diimplementasikan di banyak tempat (Widyaman, 2012).

Penentuan posisi seseorang di dalam ruangan juga pernah dilakukan oleh Mochamad Susantok dari Politeknik Caltex Riau, beliau meneliti tentang "*Wifi*

Positioning System (WPS) Menggunakan Algoritma Neural Network Backpropagation di Area Politeknik Caltex Riau". Penelitian ini berisikan implementasi *client-server* estimasi posisi menggunakan kekuatan sinyal *wifi* dengan algoritma *neural network* dan metode *fingerprint localization*, yaitu posisi ditentukan oleh *Reference Point* (RP) yang berisi data RSS dari 3 AP dalam satu area cakupan (Susantok et al., 2013).

Penelitian yang dilakukan oleh Arai dan Tolle (2013), dengan judul "*Color Radiomap Interpolation for Efficient Fingerprint WiFi-based Indoor Location Estimation*" meneliti tentang estimasi posisi dengan memetakan kekuatan sinyal yang dipancarkan oleh AP dan mengkonversi nilai RSSI menjadi spectrum warna dengan ukuran grid 1.25 meter x 2.5 meter, penelitian ini dilakukan pada suatu ruangan koridor dan menggunakan 3 AP sebagai *Reference Point*, proses berjalannya sistem berturut-turut adalah pertama membuat peta *offline fingerprint* dengan cara mengkonversi nilai RSS yang ditangkap kedalam warna pada setiap grid yang telah ditentukan, kemudian dilakukan proses interpolasi untuk mendapatkan peta yang lebih detail dan akurat, kedua mengkonversi nilai yang diterima pengguna kedalam warna dengan metode yang sama dengan pembuatan peta *offline*, ketiga melakukan masking warna hasil konversi RSS pengguna dengan peta *offline* yang telah dibuat sebelumnya, kelebihan metode ini adalah aplikasi tidak perlu menggunakan database dan memiliki akurasi hingga 1.108 meter (Arai, 2013). Metode *Color Radiomap Interpolation* telah diimplementasikan pada penelitian yang dilakukan oleh Ananda (2016) dengan membuat aplikasi *mobile* estimasi posisi yang mendapatkan akurasi sebesar 85% (Ananda, 2016).

Oleh karena permasalahan dan beberapa solusi yang telah dipaparkan di atas, maka diperlukan aplikasi *Finding Dosen* untuk mencari Dosen FILKOM Universitas Brawijaya menggunakan metode *Color Radiomap Interpolation* untuk mendapatkan tingkat akurasi 90% dalam *Indoor Positioning System* serta memudahkan mahasiswa untuk menggunakan aplikasi karena aplikasi dibuat pada perangkat *mobile*.

1.2 Rumusan Masalah

Berdasarkan paparan latar belakang tersebut, maka rumusan masalah dari penelitian ini sebagai berikut:

1. Bagaimana rancangan dan implementasi aplikasi pencari posisi Dosen FILKOM Universitas Brawijaya yang menerapkan metode *Color Radiomap Interpolation*?
2. Seberapa tinggi tingkat akurasi dalam penggunaan metode *Color Radiomap Interpolation* pada aplikasi *Finding Dosen*?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini sebagai berikut:

1. Merancang dan mengimplementasikan aplikasi pencari posisi Dosen FILKOM Universitas Brawijaya menggunakan metode *Color Radiomap Interpolation*.
2. Mengukur tingkat akurasi metode *Color Radiomap Interpolation* pada aplikasi *Finding Dosen*.

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini sebagai berikut:

1. Manfaat untuk peneliti adalah dapat mengembangkan lebih jauh tentang penentuan lokasi seseorang di dalam ruangan menggunakan metode *Color Raiomap Interpolation*.
2. Memudahkan pengguna untuk mencari posisi dosen yang berada di dalam ruangan secara tepat.

1.5 Batasan Penelitian

Agar penelitian yang dirumuskan lebih terfokus, maka penelitian ini dibatasi dalam hal:

1. Ruang lingkup penelitian pada lingkungan FILKOM Universitas Brawijaya khususnya pada Gedung A, Gedung E, dan Gedung F lantai 2-4.
2. Minimal terdapat 3 buah *access point* pada setiap lantai pada gedung.
3. Platform yang digunakan untuk aplikasi ini adalah *Android Mobile*.
4. Dosen harus terhubung dengan *Wifi* atau *Access Point* dengan SSID FILKOM.X dan mengaktifkan statusnya untuk dapat ditemukan posisinya di dalam ruangan.

1.6 Sistematika Pembahasan

Sistematika pembahasan secara garis besar meliputi beberapa bab yaitu sebagai berikut:

Bab 1 Pendahuluan

Bab ini berisi tentang latar belakang, identifikasi masalah, rumusan masalah, tujuan, manfaat, sistematika pembahasan dan jadwal pelaksanaan.

Bab 2 Dasar Teori

Bab ini berisi tentang teori-teori dan referensi yang mendukung dalam perancangan, implementasi, dan pengujian dalam metode yang digunakan.

Bab 3 Metodologi Penelitian

Bab ini membahas tentang metode yang digunakan dalam penelitian yang meliputi dasar teori, metode perancangan, metode pengujian serta pengambilan kesimpulan dan saran.

Bab 4 Anaisis Kebutuhan

Bab ini memuat tentang analisis kebutuhan dalam pembangunan aplikasi *Finding Dosen*.

Bab 5 Pengambilan Data dan Pengolahan Data

Bab ini memuat tentang pengambilan data yang akan digunakan dalam aplikasi Finding Dosen dan pengolahannya dalam menjadikan sebuah peta *color radiomap*.

Bab 6 Perancangan dan Implementasi

Bab ini memuat tentang perancangan untuk membuat aplikasi Finding Dosen serta mengimplementasikannya ke dalam platform *android mobile*.

Bab 7 Pengujian dan Analisis Hasil

Bab ini memuat tentang cara-cara pengujian dan hasil pengujian serta analisis dari hasil yang didapatkan dari pengujian.

Bab 8 Penutup

Bab ini berisi tentang kesimpulan dari penelitian yang telah dilakukan dan saran yang ditujukan untuk pembaca penelitian ini.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Melalui kajian pustaka, peneliti akan membahas beberapa hasil penelitian lain yang relevan. Pembahasan ini bertujuan untuk mendukung penelitian yang akan dilakukan. Beberapa penelitian tersebut adalah yang berkaitan dengan *Wifi Positioning System (WPS)*. *Composite Indoor Localization and Positioning System Based on Wi-Fi Repeater Technology* dan *Color Radiomap Interpolation for Efficient Fingerprint WiFi-based Indoor Location Estimation*.

Penelitian pertama dilakukan oleh Tahmid Rasyid, MD dan kawan-kawan tentang “Composite Indoor Localization and Positioning System Based on Wi-Fi Repeater Technology” mereka menggunakan dua metode populer untuk menentukan lokasi dari Wi-Fi positioning system adalah *Time of Arrival (TOA)* dan *Angle of Arrival (AOA)*. Untuk kedua teknik tersebut, unit pengukuran atau standarnya adalah satu set perkiraan koordinat yang dipra-petakan sebelum diaplikasikan. Terlepas dari keuntungan intrinsik mereka, baik TOA dan AOA memiliki beberapa kelemahan ketika menentukan posisi yang akurat. TOA membutuhkan tiga Akses Poin (AP) untuk mendapatkan kunci lokasi dan menggunakan metode Triangulasi yang kadang-kadang memberikan keluar perkiraan wilayah yang melingkar daripada set koordinat yang tepat. Untuk AOA, akurasi posisi yang sangat dinegasikan dengan meningkatkan ruang, AOA bekerja baik di daerah kecil (Rashid, 2016).

Menggunakan informasi yang didapatkan dari dua metode di atas yaitu TOA dan AOA, sistem yang mereka buat berhasil memberikan akurasi sekitar 1,5 meter yang dipercaya sudah luar biasa untuk ruang lingkup terbatas dalam penelitian. Selain pendekatan komposit, sistem tersebut juga menggunakan Akses Poin yang juga bertindak sebagai *repeater* untuk meminimalkan masalah sinkronisasi yang dihadapi oleh TOA. Mereka juga telah menyimpan ketentuan untuk menghilangkan pembacaan anomali yang dihasilkan oleh AOA dengan membandingkan set data sekarang dengan sebelumnya.

Penentuan posisi seseorang di dalam ruangan juga pernah dilakukan oleh Mochamad Susantok dari politeknik Caltex Riau, beliau meneliti tentang “*Wifi Positioning System (WPS) Menggunakan Algoritma Neural Network Backpropagation di Area Politeknik Caltex Riau*”. Penelitian ini berisikan implementasi client-server estimasi posisi menggunakan kekuatan sinyal wifi dengan algoritma neural network dan metode *fingerprint localization*, yaitu posisi ditentukan oleh *Reference Point (RP)* yang berisi data RSS dari 3 AP dalam satu area cakupan. Hasil dari implementasi mendapatkan tingkat akurasi posisi sebesar 44% dengan penyimpangan terjauh 2 meter dan 64% dengan penyimpangan terjauh 4 meter. Kedua hasil tersebut untuk kondisi pengguna yang diam, namun jika pengguna perjalan akurasi menjadi 50% untuk penyimpangan terjauh 2 meter dan 83,3% untuk penyimpangan terjauh 4 meter (Susantok et al., 2013).

Penelitian ketiga dilakukan oleh Arai dan Tolle dengan judul “*Color Radiomap Interpolation for Efficient Fingerprint WiFi-based Indoor Location Estimation*”

penelitian ini meneliti tentang estimasi posisi dengan memetakan kekuatan sinyal yang dipancarkan oleh AP dan mengkonversi nilai RSSI menjadi *spectrum* warna dengan ukuran grid 1.25 meter x 2.5 meter, penelitian ini dilakukan pada suatu ruangan koridor dan menggunakan 3 AP sebagai *Reference Point*, proses berjalannya sistem berturut-turut adalah pertama membuat peta *offline fingerprint* dengan cara mengkonversi nilai RSS yang ditangkap kedalam warna pada setiap grid yang telah ditentukan, kemudian dilakukan proses interpolasi untuk mendapatkan peta yang lebih detail dan akurat, kedua mengkonversi nilai yang diterima pengguna kedalam warna dengan metode yang sama dengan pembuatan peta *offline*, ketika melakukan *masking* warna hasil konversi RSS pengguna dengan peta *offline* yang telah dibuat sebelumnya, kelebihan metode ini adalah aplikasi tidak perlu menggunakan *database* dan memiliki akurasi hingga 1.108 meter (Arai, 2013).

2.2 Landasan Teori

2.2.1 Global Positioning System (GPS)

Global Positioning System (GPS) adalah alat navigasi elektronik yang menerima informasi dari 4-12 satelit sehingga GPS bisa memperhitungkan posisi di mana pengguna berada di Bumi. Satelit GPS tidak mentransmisikan informasi posisi melainkan posisi satelit dan jarak penerima GPS dari satelit. Informasi ini diolah oleh alat penerima GPS pengguna dan hasilnya ditampilkan kepada pengguna. Dengan kata lain *Global Positioning System (GPS)* adalah sistem untuk menentukan letak di permukaan bumi dengan bantuan penyelarasan (*synchronization*) sinyal satelit. Sinyal ini diterima oleh alat penerima di permukaan Bumi, dan dapat digunakan untuk menentukan letak, kecepatan, arah, dan waktu (Rohman, 2012).

GPS sebenarnya adalah proyek departemen pertahanan Amerika Serikat (AS) yang memberinya nama resmi NAVSTAR (*NAVigation Satellite Timing and Ranging*). Bagian utamanya dari sistem GPS adalah 24 satelit yang mengorbit Bumi di ketinggian 20.200 Kilometer. Orbit satelit dirancang sehingga setiap titik di bumi dapat melihat paling sedikit empat satelit pada setiap saat (Rohman, 2012).

Tiap satelit mengitari bumi kira-kira dalam 12 jam dengan kecepatan sekitar 11.000 Km per jam. Satelit GPS memiliki panel-panel tenaga matahari yang berguna untuk membangkitkan energi listrik yang diperlukannya dalam satelit, namun ketika satelit tidak mendapatkan sinar matahari, terdapat baterai yang dapat memenuhi kebutuhan satelit akan listrik (Rohman, 2012).

2.2.1.1 Kelebihan GPS

GPS melebihi beberapa kelebihan, diantaranya:

1. Mengetahui koordinat suatu titik.
2. Menhitung jarak dan arah dari lokasi suatu tempat.
3. Penunjuk arah (navigasi)

4. Sistem informasi Geografis
5. Mencari tempat-tempat penting dan dapat menampilkan peta *point of interest*: SPBU, Bandara, Stasiun Kereta, Tempat Ibadah, Hotel dll.

2.2.1.2 Kekurangan GPS

Dari beberapa kelebihan di atas, tentunya GPS juga memiliki beberapa kekurangan, yaitu:

1. Penggunaan GPS untuk mengetahui posisi yang mengandalkan setidaknya tiga satelit ini tidak selamanya akurat.
2. Terkadang, dibutuhkan satu satelit untuk memperbaiki sinyal yang diterima. Ketidakakuratan posisi yang ditunjukkan
3. GPS ini dipengaruhi oleh posisi satelit yang berubah dan adanya proses sinyal yang ditunda. Kecepatan sinyal GPS ini juga seringkali berubah karena dipengaruhi oleh kondisi atmosfer yang ada. Selain itu, sinyal GPS juga mudah berinterferensi dengan gelombang elektromagnetik lainnya.
4. GPS tidak dapat memetakan posisi seseorang yang berada di dalam gedung dengan tepat.

2.2.2 Wireless Fidelity (Wi-Fi)

Wireless Fidelity (Wi-Fi) atau WLAN adalah suatu jaringan area lokal yang terhubung tanpa kabel dimana media transmisinya menggunakan frekuensi radio (RF) dan *infrared* (IR), sebagai sebuah koneksi jaringan untuk pengguna dalam area sekitarnya. Area jangkauannya *Wi-Fi* dapat berjarak 100m sampai 1km bahkan lebih. Perangkat yang biasa digunakan untuk jaringan *Wi-Fi* adalah PC, *laptop*, PDA, telepon seluler, dan perangkat lain sebagainya yang dapat menggunakan Wi-Fi. Teknologi WLAN ini memiliki kegunaan yang sangat banyak. Contohnya, pengguna *laptop* dapat menggunakan jaringan *Wi-Fi* untuk melakukan *browsing* atau mengakses *e-mail* (Hartono, 2011).

2.2.2.1 Karakteristik Wifi

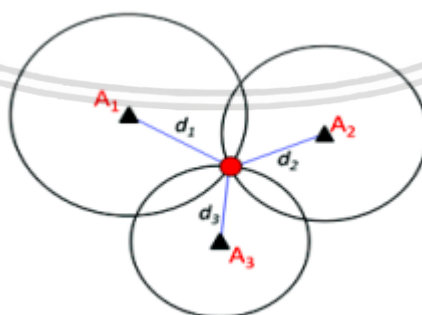
Jaringan *wifi* mempunyai karakteristik yang berbeda dengan jaringan kabel pada umumnya. Pada jaringan wifi banyak faktor yang dapat mengganggu kinerja dan keandalannya. Beberapa faktor secara teknis diantaranya adalah Panjang Gelombang (*wavelength*), frekuensi radio yang dipilih dalam transmisi data dalam jaringan, dan Amplitudo (kekuatan sinyal). Karena media transmisi Wi-Fi menggunakan sinyal radio (RF) maka banyak faktor alam yang dapat mempengaruhi seperti *free path loss*, *absorption*, pemantulan sinyal, pemecahan sinyal, pembelokan sinyal dan *line of sight* atau biasa disingkat LOS (Hartono, 2011).

2.2.3 Trilateration

Trilateration atau dalam Bahasa Indonesia trilaterasi adalah sebuah metode untuk menentukan posisi sebuah objek berdasarkan pengukuran jarak secara simultan dari tiga buah titik (dimisalkan titik A, B, dan C) yang terletak di lokasi yang diketahui. Trilaterasi merupakan metode perhitungan jarak antara dua atau lebih titik yang bersinggungan. Dengan kata lain, pada trilaterasi terdapat sebuah titik yang tidak dikenal, untuk mengetahui nilai titik yang tidak diketahui tersebut maka dilakukan pengukuran panjang dari setiap sisi pada segitiga antara titik yang tidak diketahui dengan dua atau lebih titik yang telah diketahui.

Pada sistem lokasi *indoor 2D*, *Trilateration* digunakan untuk mencari objek yang terletak di permukaan pada bidang XY. Lokasi tidak AP harus diketahui untuk pencarian jarak antara AP dan objek sehingga *2D Trilateration* dapat bekerja dengan benar. Titik di tengah akan mewakili lokasi objek yang sedang dicari. Node referensi juga dikenal sebagai AP, diberi label AP1, AP2, dan AP3, jarak antar *node* referensi dan *node* ditandai dengan label r_1 , r_2 , dan r_3 . Perpotongan antar ketika *node* adalah lokasi tidak diketahui.

Setiap *tag* dan *reader* akan terdiri dari *transceiver* dan antenna. *Tag* mentransmisikan sinyal dan kekuatan sinyal, atau dikenal sebagai RSSI (*Received Signal Strength Indicator*). Kekuatan sinyal dapat dikonversi menjadi jarak, yang kemudian dapat memberikan tiga jarak antara masing-masing AP dengan objek yang diperlukan untuk *Trilateration*. Kelemahan untuk menggunakan kekuatan sinyal adalah jarak dihitung tidak akan selalu memiliki jarak yang tepat antara AP dan objek. Alasannya adalah karena kekuatan sinyal yang datang dari objek atau AP yang lain bias mengganggu frekuensi sinyal, selain itu suhu kamar, kelembaban, konstruksi dalam bangunan dan logam juga merupakan factor yang dapat mengganggu kekuatan sinyal. Setelah jarak dihitung, *trilateration* akan digunakan untuk mencari posisi obyek yang ditandai. Cara *2D Trilateration* bekerja ditampilkan pada Gambar 2.1 (Robert J. et al., 2011).



Gambar 2.1 Trilateration 2D

Sumber: Robert J. et al. (2014)

2.2.4 Teknik Positioning berdasarkan kekuatan sinyal wifi

Dalam penelitian yang dilakukan Bose dan Heng dilakukan pengelompokan metode penentuan posisi berbasis WiFi ke dalam *Cell Identity* (Cell-ID), *Time of Arrival* (TOA), *Time Difference of Arrival* (TDOA), *Angle of Arrival* (AOA), dan kategori kekuatan sinyal (Bose et al., 2007).

Cell Identity (Cell-ID) adalah solusi dasar *wireless positioning system*. Metode *cell identity* (Cell-ID) tidak memerlukan operasi kompleks seperti waktu sinkronisasi dan beberapa AP. Metode *Time of Arrival* (TOA) mengukur jarak menggunakan waktu tempuh sinyal radio dari pemancar ke penerima.

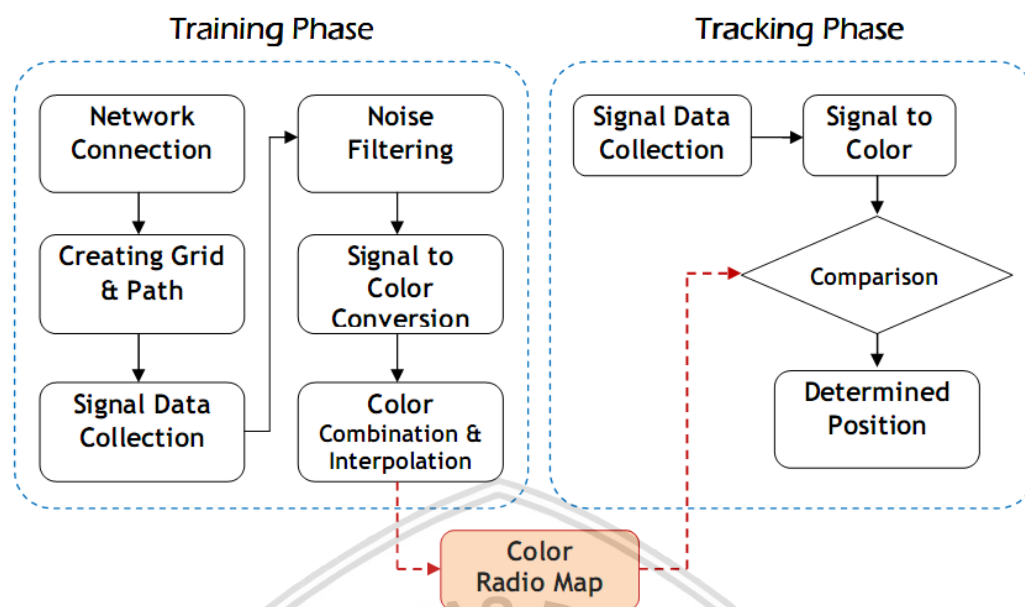
Penerapan metode ini membutuhkan waktu sinkronisasi dari pemancar ke penerima, sehingga sulit dicapai dalam waktu yang cepat. Untuk mengatasi masalah tersebut terdapat metode *Time Difference of Arrival* (TDOA) yang dikembangkan, TDOA memanfaatkan perbedaan waktu antara penerima yang satu dengan penerima yang lainnya. Artinya, TOA membutuhkan waktu sinkronisasi disisi pemancar dan penerima, sedangkan TDOA hanya membutuhkan sinkronisasi antara penerima. Metode yang terakhir adalah *Angle of Arrival* (AOA), metode ini menentukan posisi penerima dengan cara mengukur sudut penerima dari pemancar. Sebuah AP harus menggunakan antena cerdas dan mampu mounting di bawah kondisi statis (Bose et al., 2007).

2.2.5 Metode Color Radiomap Interpolation

Metode *Color Radiomap Interpolation* adalah sebuah metode yang digunakan untuk mempermudah melakukan pemetaan kekuatan sinyal, konsep dasar metode ini adalah mengkonversi nilai kekuatan sinyal kedalam warna RGB dan memprediksi nilai yang belum diketahui melalui dua titik yang telah diketahui disekitar titik yang belum diketahui dengan metode interpolasi (Arai, 2013).

2.2.5.1 Arsitektur sistem

Pada penelitian *positioning* sistem berbasis Wifi, sistem dikonfigurasi melalui dua tahap: pelatihan dan pelacakan. Tujuan dari fase pelatihan adalah membangun data informasi sinyal yang nantinya digunakan untuk membangun peta *Offline Color Radiomap*, karena pada metode ini menggunakan warna sebagai wakil dari nilai dari signal radio. Data sinyal yang diterima disusun dalam memori computer sesuai dengan ID dari AP. Sinyal data AP yang dipilih tersebut kemudian digunakan dalam bentuk langkah pemrosesan sinyal, yang melibatkan penyaringan dan *interpolation*, untuk mengurangi *noise*. Data sinyal yang telah dimodifikasi kemudian disimpan dalam *file temporary* dan diubah menjadi perwakilan warna dalam *sel* peta atau *grid* peta. Pada tahap pelacakan, lokasi target dihitung dengan membandingkan pola sinyal yang diterima target dengan pola sinyal yang telah disimpan dalam *file temporary* yang didapat dari tahap pelatihan. Selanjutnya koordinat posisi target ditampilkan dalam peta *grid* yang telah dibuat dalam tahap pelatihan. Mekanisme pengolahan data dijelaskan secara sederhana pada Gambar 2.2 (Arai, 2013).



Gambar 2.2 Data flow mekanisme pengolahan data.

Sumber: Arai & Tolle (2013)

Penjelasan pengolahan data setiap tahap, secara rinci adalah:

1. Tahap pelatihan

Langkah pertama adalah menggambar peta dua dimensi dari bangunan. Berdasarkan peta, kita membuat grid dalam bentuk persegi panjang dan pengumpulan data. Data sinyal yang telah dikumpulkan dapat disimpan untuk setiap *Access Point* (AP) pada setiap kotak. Gambar menunjukkan prosedur tahap pelatihan dan tahap pelacakan.

Informasi kekuatan sinyal dikumpulkan dalam operasi yang telah diuraikan di atas kemudian kekuatan sinyal perlu diproses untuk mengurangi *noise* serta untuk tujuan membangun model distribusi kekuatan sinyal. Data kekuatan sinyal yang dikumpulkan dari AP diproses dengan menggunakan algoritma *Kalman Filtering* adaptif untuk menghilangkan *noise* dan efek gangguan. Dalam membangun model distribusi sinyal diterapkan metode interpolasi. Kemudian distribusi sinyal pada setiap titik dalam rute dikumpulkan pada *database radiomap* dari komputer server (Arai, 2013).

2. Tahap pelacakan

Pada tahap pelacakan, data sinyal diproses secara *real-time* dengan algoritma penyaringan dari RSSI kemudian dibandingkan dengan data sinyal yang disimpan dalam *radiomap*. Berdasarkan perbandingan ini, lokasi yang optimal ditentukan dan ditandai cara yang representatif dan visualisasi warna (Arai, 2013).

2.2.5.2 Grid segment

Proses segmen *grid* dibuat untuk segmentasi daerah di dalam gedung. Ukuran segmen diasumsikan dalam jarak toleran di mana orang dapat melihat orang lain dengan jelas dalam jarak yang dipilih. Asumsi ini digunakan karena desain dalam

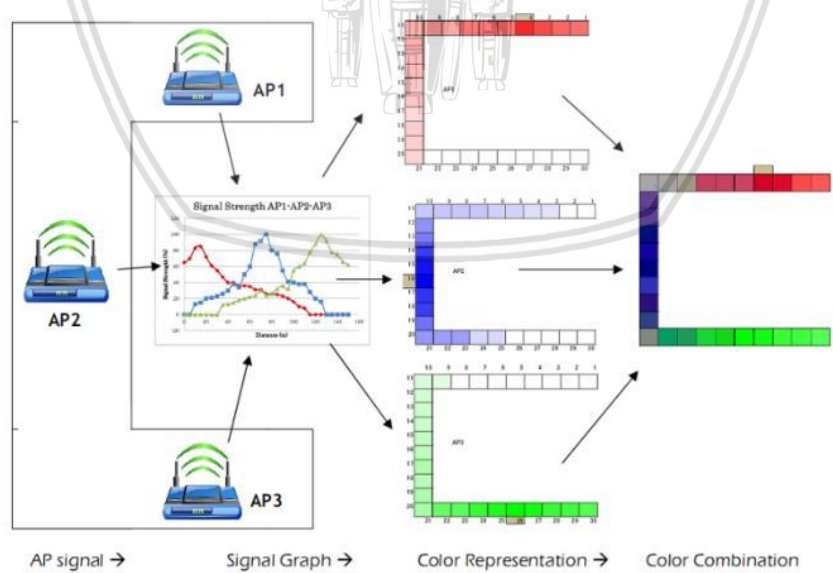


ruangan lokasi sistem penentuan lokasi dalam ruangan untuk digunakan pada penanggulangan bencana untuk memantau dan menemukan orang-orang atau pengguna dalam sisi sebuah gedung dalam situasi bencana atau situasi normal. Jika kita dapat menentukan perkiraan lokasi pengguna dalam posisi *grid*, cukup menggunakan informasi tersebut untuk mencari atau memantau pengguna. Jarak awal untuk ukuran *grid* dapat diubah sesuai dengan kebutuhan, kita dapat memperpanjang peta radio ke segmen yang lebih rinci dari *grid* untuk mencapai akurasi yang lebih baik. Hal ini dapat dilakukan dengan menggunakan metode interpolasi warna. Setelah menciptakan tata letak segmen *grid* untuk bangunan, maka sinyal informasi dikumpulkan pada setiap *grid* untuk masing-masing AP (Arai, 2013).

2.2.5.3 Representasi warna untuk warna radiomap

Sinyal yang diterima dari setiap jalur akses diubah menjadi perwakilan warna. Pada sistem penelitian terdahulu peneliti menggunakan informasi 3 sinyal dari 3 AP berbeda berdasarkan pendekatan triangulasi. Setiap titik akses memiliki warna dasar yang berbeda satu sama lain. Tiga masukan AP (AP1, AP2, dan AP3) menggunakan warna dasar RGB. Peta warna didasarkan pada informasi kekuatan sinyal yang direkam dari proses pengumpulan data sinyal. Gradiasi warna didasarkan pada nilai RGB tersebut (Arai, 2013).

Pertama, sinyal dikumpulkan dari masing-masing AP kemudian disaring sebelum diubah menjadi kumpulan peta gradiasi warna. Kemudian membuat peta radio warna baru sebagai fungsi dari jumlah dari 3 peta radio warna yang berbeda. Jumlah 3 warna berdasarkan nilai RGB-nya ringkasan dengan operasi OR (Arai, 2013).

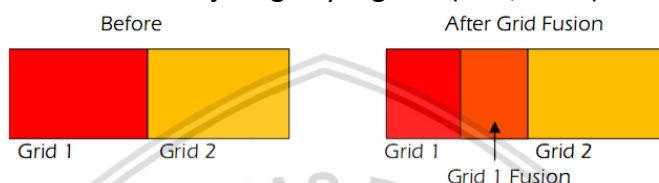


Gambar 2.3 Pembuatan peta Color Radiomap menggunakan kombinasi 3 sinyal Akses Point

Sumber: Arai & Tolle (2013)

2.2.5.4 Interpolasi dengan warna grid fusion

Ketika peta *offline* awal dengan ukuran *grid* awal tidak cukup akurat, maka kita dapat meningkatkan peta *radiomap* dengan interpolasi. Efektivitas dari metode ini adalah interpolasi menggunakan informasi warna setiap *grid* menggunakan teknik jaringan fusi. Pada awalnya sel atau jaringan berukuran panjang 5 meter dengan lebar 2,5 meter (lebar koridor). Ukuran jaringan awal ditentukan dalam nomor yang rendah pada pelatihan *offline*. Warna *radiomap* awal akan memiliki kesalahan awal juga sekitar 5 meter. Kemudian, untuk meningkatkan akurasi sistem, peta awal ditingkatkan untuk menciptakan peta baru dengan warna rinci. Ukuran *grid* awal difusi menjadi setengah ukuran *grid* lama dan nilai baru dihitung dengan interpolasi anatar dua jaringan yang ada (Arai, 2013).



Gambar 2.4 Ilustrasi Grid Fusi pada interpolasi warna

Sumber: Arai & Tolle (2013)

Fusi dari grid adalah setengah ukuran dan warna didasarkan pada kombinasi RGB dua warna yang berdekatan dengan rumus (Arai, 2013):

$$Grid_1Fusion(r, g, b) = Grid_1(r_1, g_1, b_1) + Grid_2(r_2, g_2, b_2) \quad (2,1)$$

$$r = \frac{r_1+r_2}{2}, g = \frac{g_1+g_2}{2}, \text{ and } b = \frac{b_1+b_2}{2} \quad (2,2)$$

Keterangan : r = Nilai Red
g = Nilai Green
b = Nilai Blue

Dimisalkan jika *Grid* awal 1 dan *Grid* 2 memiliki warna RGB adalah *Grid* 1 (255,0,0), dan *Grid* 2 (255, 102, 0), maka *Grid* 1 Fusion menjadi (255, 51, 0). Interpolasi untuk setiap jaringan yang berdekatan menghasilkan *grid* baru dengan lebih detail peta warna seperti yang ditunjukkan pada Gambar 2.4. Tahap kedua *grid* fusi bertujuan untuk menciptakan warna lebih detail pada *radiomap* dengan menggunakan data kekuatan sinyal yang ada dari tahap *offline* (Arai, 2013).

2.2.5.5 Penentuan lokasi dan visualisasi

Proses penentuan lokasi seseorang di dalam ruangan didasarkan pada perbandingan antara data *real-time* secara *online* dan warna yang telah ditetapkan pada tahap *offline radiomap* untuk menemukan kesalahan minimal antara dua nilai tersebut pada nilai warna RGB. Diasumsikan bahwa data kekuatan sinyal *online* dari 3 AP dikonversikan menjadi nilai HSL-nya kemudian digabungkan



menjadi satu warna dengan menggunakan proses yang sama seperti membuat peta *offline*. Kemudian fungsi setiap *grid* diperiksa untuk semua jaringan posisi *Grid*_x (r_x , g_x , b_x) untuk menemukan kesalahan minimum menggunakan rumus *mean square error* (MSE) (Arai, 2013).

$$MSE = \frac{[(r_x - r_s)^2 + (g_x - g_s)^2 + (b_x - b_s)^2]}{3} \quad (2,3)$$

Keterangan : MSE = Nilai *Mean Squared Error*

- r_x = Nilai *Red* pada *grid* data latih
- r_s = Nilai *Red* pada *grid* data uji
- g_x = Nilai *Green* pada *grid* data latih
- g_s = Nilai *Green* pada *grid* data uji
- b_x = Nilai *Blue* pada *grid* data latih
- b_s = Nilai *Blue* pada *grid* data uji

Indoor positioning system terintegrasi dengan sistem secara *real time online monitoring*. Dengan menggunakan teknik kombinasi warna untuk estimasi posisi, pemantauan dan visualisasi dari posisi estimasi menjadi lebih mudah. Teknik warna *masking* sederhana menggunakan masukan informasi warna sebagai representasi dari kombinasi 3 kekuatan sinyal pengukuran perangkat pengguna yang dipantau saat ini, kita dapat menghitung langsung menunjukkan posisi pengguna tanpa menggunakan rumus *mean square error*. Metode untuk menentukan posisi pengguna dalam kotak menggunakan kombinasi warna *layering* berdasarkan algoritma ini (Arai, 2013).

```
// initial variable
input UserSS; // rgb representative of input
user detected signal strength
var NewLayer; // create new layer image var maxGridNumber;
// maximum grid number for x=1 to maxGridNumber {

tempX = (binary(UserSS(x) XOR Grid(x)) if (decimal(tempX)
== 0)

Newlayer(x) = Grid(x) else

        Newlayer(x) = whiteColor + tempX;}
return NewLayer;
```

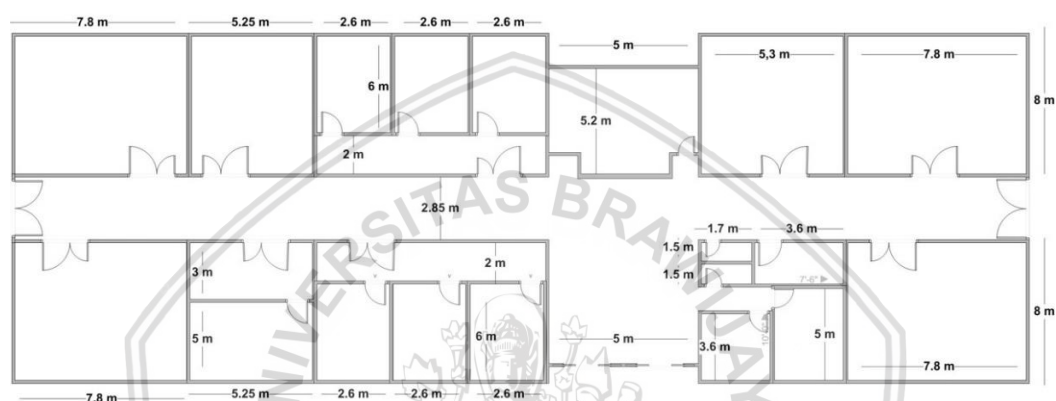
Dengan menggunakan metode *masking* ini, informasi tentang estimasi posisi pengguna otomatis ditampilkan pada layer gambar baru hanya dengan *masking input* data manual untuk warna *radiomap*. Algoritma ini bekerja untuk pelacakan *online* estimasi posisi terdekat pengguna. Pendekatan ini efektif karena tidak ada

perhitungan matematis untuk menemukan MSE tetapi hanya pengolahan grafis yang secara langsung menunjukkan *output* dari posisi pengguna ke dalam sistem pemantau.

2.2.6 Peta Gedung FILKOM Universitas Brawijaya

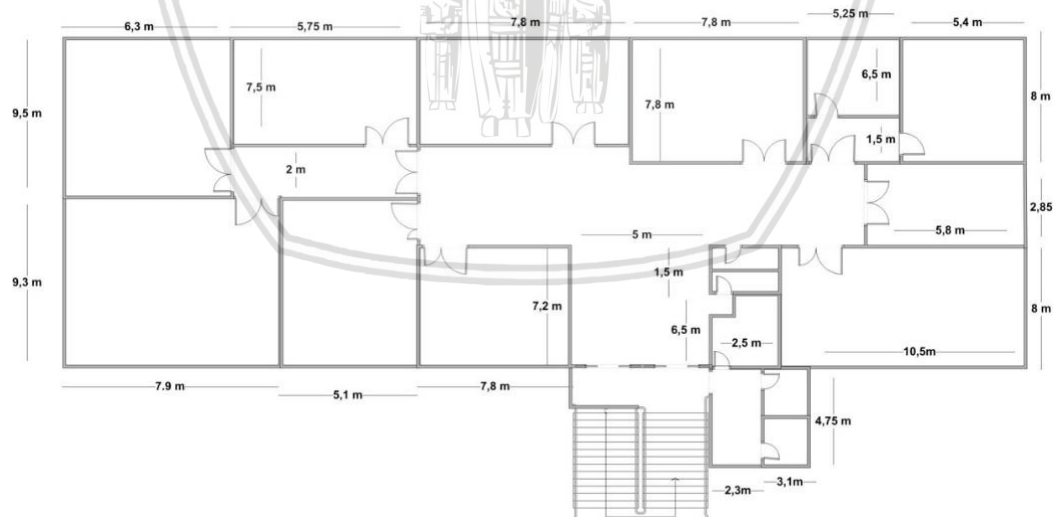
2.2.6.1 Gedung A FILKOM

Gedung A FILKOM Universitas Brawijaya merupakan gedung perkuliahan yang memiliki luas $\pm 735 \text{ m}^2$. Gedung ini merupakan gedung lama yang digunakan ruang dosen yang berada di lantai 1, dan di lantai 2 sebagai gedung perkuliahan yang digunakan oleh mahasiswa.



Gambar 2.5 Denah Gedung A lantai 1 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)



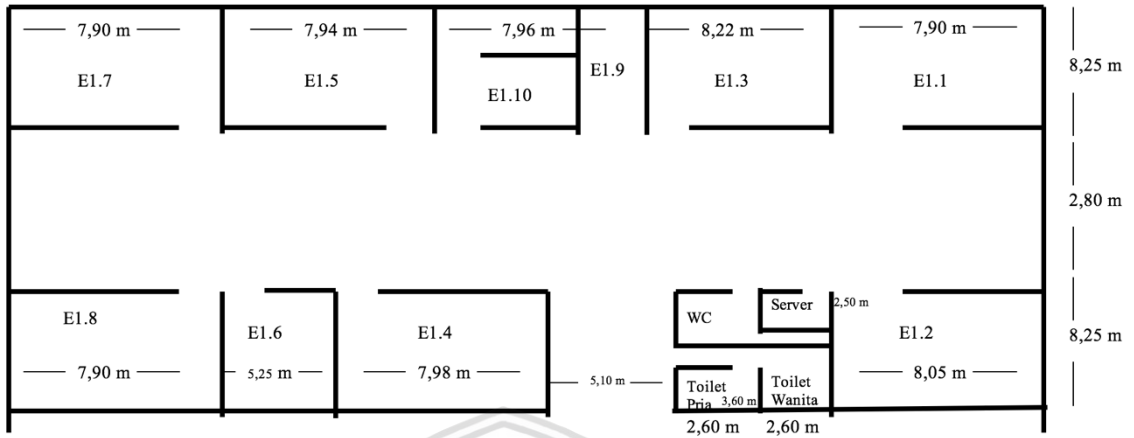
Gambar 2.6 Denah Gedung A lantai 2 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)

2.2.6.2 Gedung E FILKOM

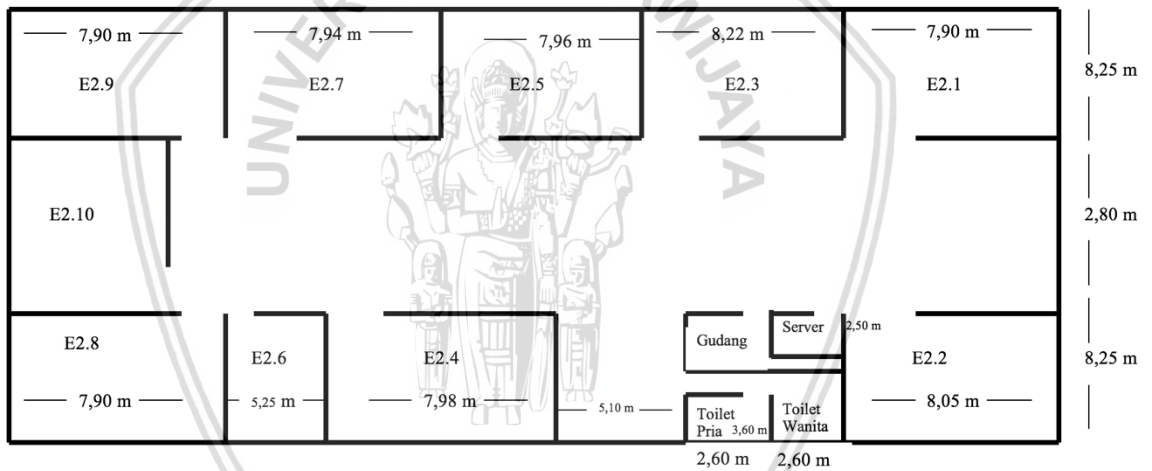
Gedung E juga merupakan pusat perkuliahan mahasiswa FILKOM. Gedung ini memiliki luas yang sama dengan gedung A FILKOM. Gedung ini berisikan ruang

kelas mahasiswa S1 dan S2, ruang pengajaran, dan Laboratorium riset mahasiswa S2.



Gambar 2.7 Denah Gedung E lantai 1 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)

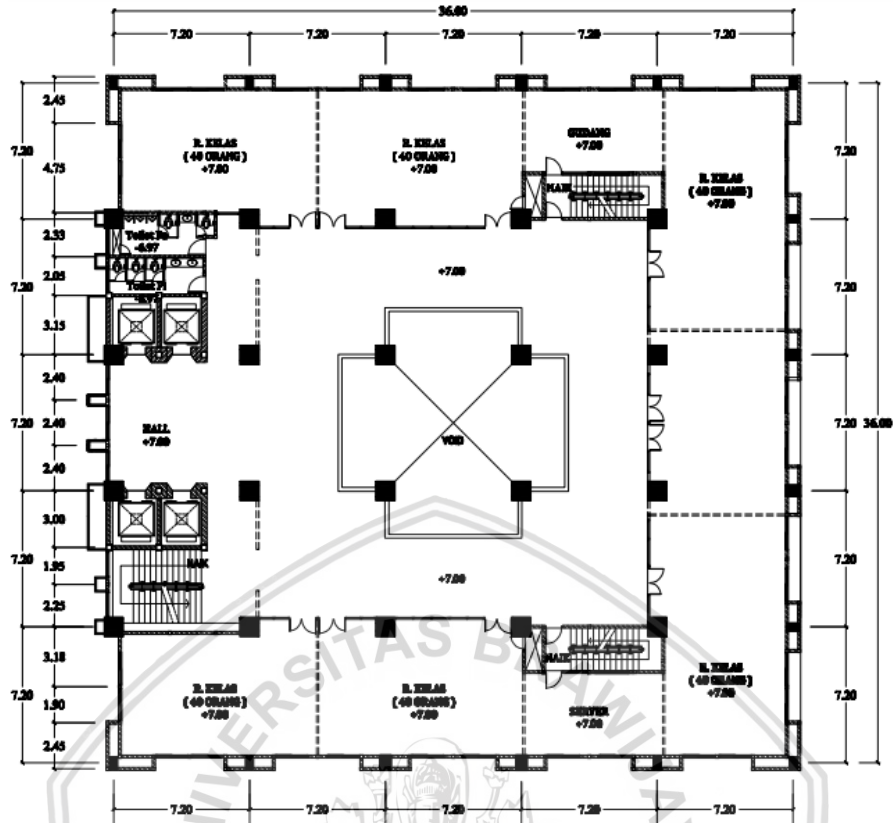


Gambar 2.8 Denah Gedung E lantai 2 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)

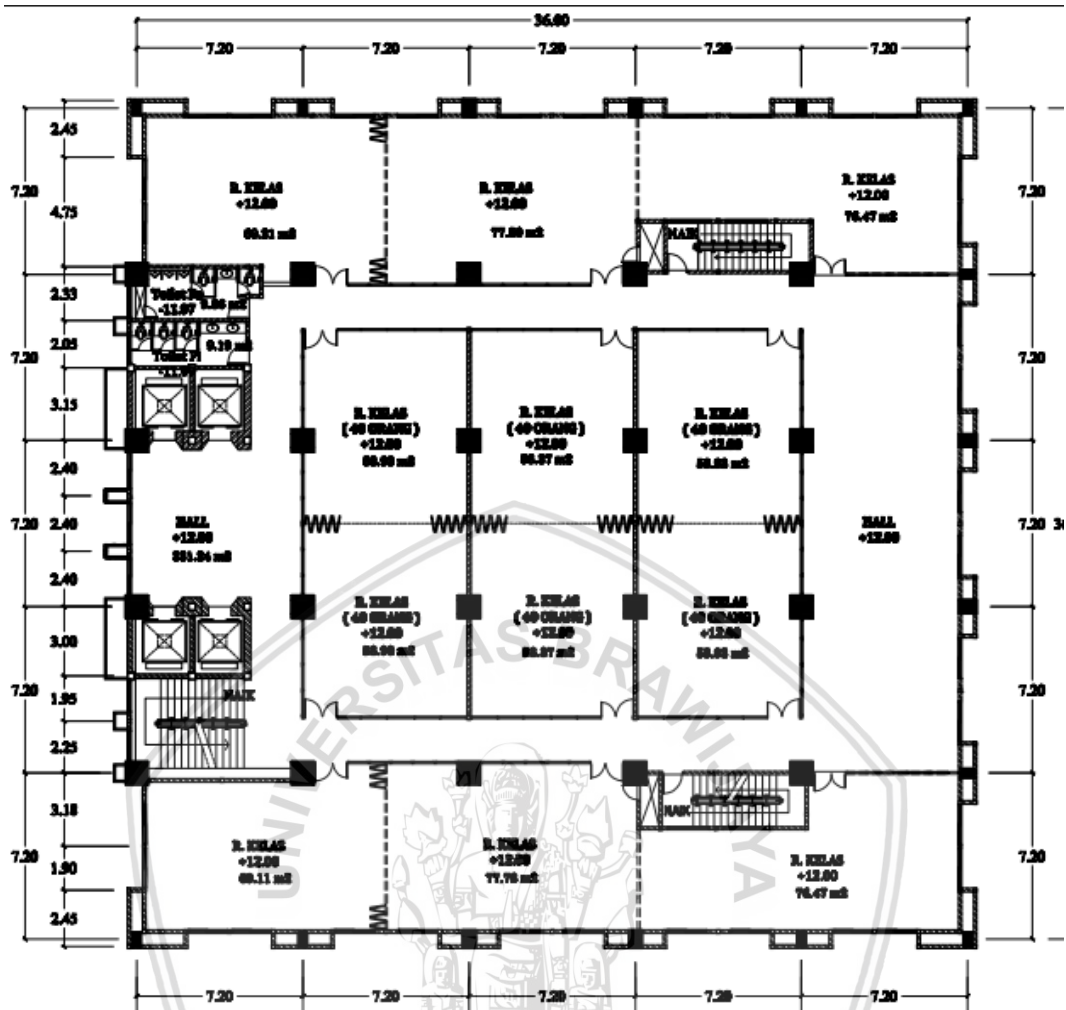
2.2.6.3 Gedung F FILKOM

Gedung F merupakan gedung baru di FILKOM, Gedung yg digunakan sebagai Gedung utama FILKOM, dimana tedapat dekanat, akademik, perkuliahan dan lain – lain.



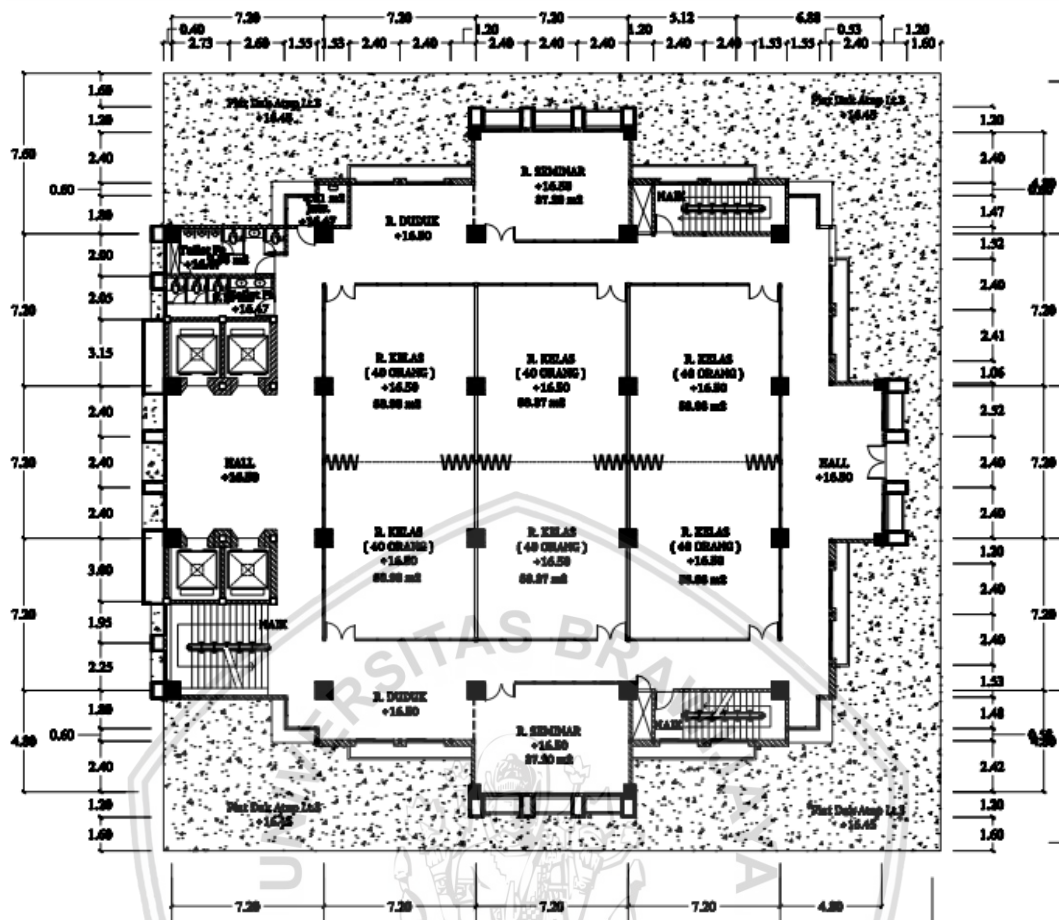
Gambar 2.9 Denah Gedung F lantai 2 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)



Gambar 2.10 Gedung F lantai 3 FILKOM Universitas Brawijaya

Sumber: Perengkapan FILKOM (2017)



Gambar 2.11 Gedung F lantai 4 FILKOM Universitas Brawijaya

Sumber: Perlengkapan FILKOM (2017)

BAB 3 METODOLOGI

Penelitian ini merupakan penelitian yang bertipe implementatif, dimana peneliti hanya akan mengimplementasikan sistem Finding Dosen untuk mencari lokasi dosen di dalam ruangan berupa aplikasi *mobile*. Pada bab ini peneliti akan mencoba menguraikan mengenai langkah-langkah yang akan dilakukan untuk memecahkan masalah yang dibahas. Tahap pertama adalah studi literatur, penulis mempelajari teori-teori yang berhubungan tentang penelitian dan metode *Color Radiomap Interpolation*. Tahap selanjutnya adalah melakukan analisa kebutuhan, kebutuhan diperoleh dari data yang sudah didapatkan. Tahap selanjutnya adalah pengumpulan data, dimana peneliti mengumpulkan data yang diperoleh dari FILKOM Universitas Brawijaya. Dari data yang diperoleh, data kemudian diolah agar dapat dijadikan dasar perancangan aplikasi yang akan dibangun. Kemudian perancangan dibuat sesuai dengan kebutuhan pengguna dan data yang ada. Setelah dihasilkan rancangan sistem yang sesuai dengan pengguna, kemudian dilakukan implementasi sistem. Setelah dihasilkan sistem Finding Dosen, pengujian dilakukan untuk membuktikan kebutuhan-kebutuhan yang sudah dibuat berjalan dengan benar. Tahap terakhir adalah penulisan kesimpulan dan saran.

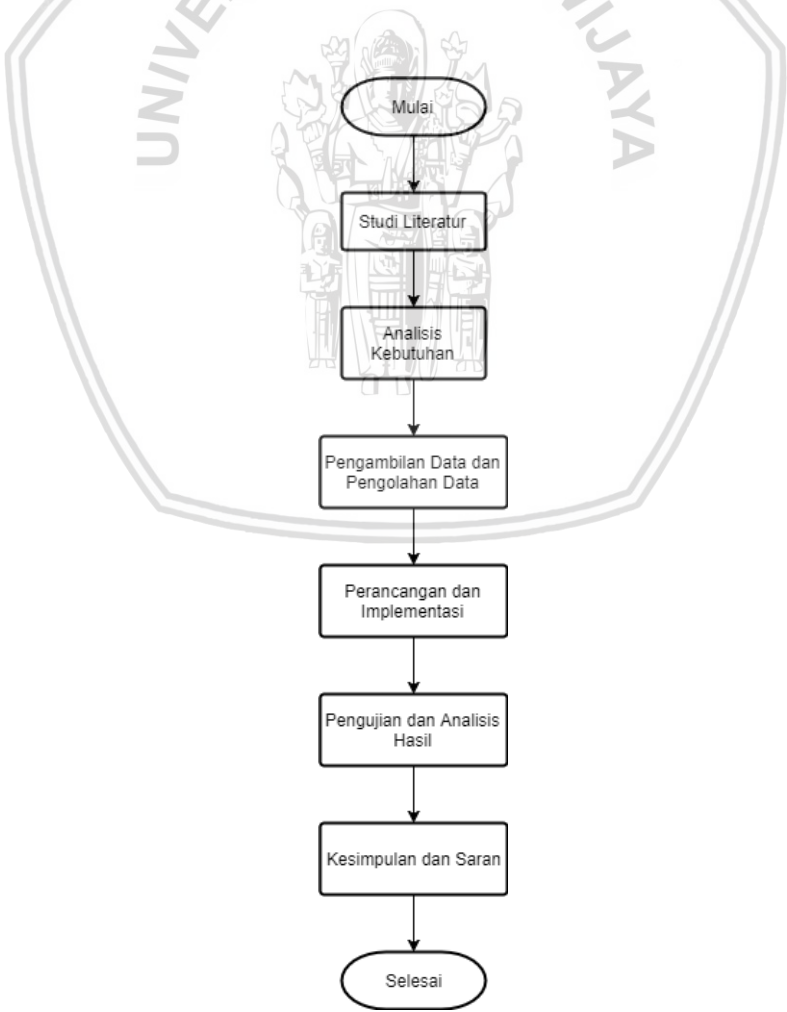


Diagram 3.1 Metodologi

3.1 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk lebih memantapkan pengetahuan dalam merealisasikan tujuan dan pemecahan masalah dalam penelitian ini. Studi ini dilakukan untuk mempelajari teori-teori yang berhubungan dengan Color Radiomap Interpolation dan Pembuatan aplikasi Finding Dosen. Pustaka yang digunakan yaitu berupa buku-buku teks yang berupa tulisan ilmiah, *handbook*, *e-book*, buku referensi Mata Kuliah dan juga jurnal-jurnal internasional baik yang berupa *hardcopy* maupun *softcopy* yang dapat menunjang penelitian yang dilakukan.

3.2 Analisis Kebutuhan

Pada tahap ini dimulai menentukan kebutuhan yang diperlukan selama proses penelitian. Analisis kebutuhan yang digunakan menggunakan metode *Object Oriented Analysis*. Tahap analisis berorientasi objek menitik beratkan pada kegiatan menemukan dan mendeskripsikan objek atau konsep dalam Penentuan Posisi Lokasi di Dalam Ruang menggunakan metode *Color Radiomap Interpolation*. Berikut adalah kebutuhan dalam penerapan rancang bangun aplikasi *Finding Dosen*.

Kebutuhan Fungsional dari aplikasi *Finding Dosen* sebagai berikut:

- a. Mahasiswa dan dosen dapat login ke aplikasi
- b. Mahasiswa dan dosen dapat melakukan pendaftaran
- c. Mahasiswa dan dosen dapat melihat profil diri sendiri
- d. Mahasiswa dan dosen dapat mengubah profil diri
- e. Mahasiswa dan dosen dapat keluar dari sesi aplikasi
- f. Mahasiswa dapat melihat daftar dosen
- g. Mahasiswa dapat mencari dosen menggunakan nama dosen
- h. Mahasiswa dapat melihat detail profil dosen
- i. Mahasiswa dapat melihat posisi dosen di dalam ruangan
- j. Dosen dapat mengubah status

3.3 Pengambilan Data dan Pengolahan Data

Pengumpulan data diperlukan untuk menjelaskan kebutuhan data apa saja yang diperlukan dalam penelitian ini. Pada penelitian ini data yang dibutuhkan adalah data mengenai beberapa AP yang ada di lingkungan FILKOM Universitas Brawijaya. Data yang dikumpulkan berupa RSSI perangkat mobile terhadap AP yang tersedia.

Data yang telah didapatkan harus diolah agar dapat dijadikan perancangan aplikasi *finding dosen*. Data kekuatan sinyal tersebut diolah untuk mengurangi *noise* agar hasilnya lebih akurat, kemudian data tersebut diolah menggunakan

metode *Color Radiomap Interpolation* untuk menggambarkan peta *offline* FILKOM Universitas Brawijaya yang nantinya peta tersebut digunakan sebagai acuan untuk mencari posisi dosen.

3.4 Perancangan dan Implementasi

Tahap perancangan aplikasi ini menggunakan metode *Object Oriented Design*, dimana dibuatkan diagram-diagram perancangan aplikasi sesuai kebutuhan yang telah dispesifikasikan. Terdapat beberapa Objek yang harus dibuat, lokasi gedung, ruangan, dan warna ruangan, serta objek Dosen yang akan dipantau dari aplikasi ini.

Pada tahap implementasi, rancangan aplikasi yang telah dibuat diimplementasikan ke dalam kode program yang dapat dimengerti oleh komputer. Implementasi aplikasi Finding Dosen ini dispesifikasikan menggunakan *platform* Android dan menggunakan Bahasa pemrograman Java. Implementasi dari data yang telah diolah yaitu peta *offline* diproses dalam aplikasi sehingga dapat ditampilkan dan digunakan sebagai penentu lokasi seorang Dosen di dalam ruangan.

3.5 Pengujian dan Analisis Hasil

Pengujian dilakukan untuk mencari kesalahan atau *error* yang ada pada saat implementasi dan memastikan aplikasi telah memenuhi kebutuhan yang telah dispesifikasikan. Teknik pengujian sistem ini menggunakan metode *white box* dan *black box* untuk menguji validasi dari aplikasi ini. Parameter lain yang dapat dijadikan pengujian adalah tingkat akurasi lokasi seorang dosen terhadap posisi sesungguhnya.

Analisis hasil diperlukan untuk mengetahui hasil yang didapatkan dari pengujian aplikasi Finding Dosen. Analisis dilakukan dengan cara menghitung tingkat akurasi posisi dosen di dalam ruangan dengan posisi aslinya. Analisis yang didapatkan diharapkan dapat menentukan penarikan simpulan pada tahap selanjutnya.

3.6 Kesimpulan dan Saran

Tahap yang terakhir adalah pembuatan kesimpulan dari semua proses penelitian yang telah dilakukan sehingga pada tahap ini dicantumkan jawaban atas rumusan masalah yang ada. Selain itu terdapat saran yang berhubungan dengan hasil yang telah dicapai pada penelitian ini sehingga dapat berguna untuk pengembangan lebih lanjut.

BAB 4 ANALISIS KEBUTUHAN

4.1 Gambaran Umum Sistem

Aplikasi *Finding Dosen* dibangun menggunakan *framework* yang telah disediakan oleh *Android*. Dalam pengembangan aplikasi pada platform *Android*, Bahasa pemrograman yang digunakan adalah *Java*. Kegunaan Bahasa pemrograman *Java* pada *framework Android* sendiri merupakan Bahasa pembuat *business logic* pada aplikasi. Sedangkan untuk pembuatan layout menggunakan Bahasa *markup XML*. DBMS merupakan salah satu komponen penting dalam aplikasi ini sebagai media penyimpanan data yang dibutuhkan dalam aplikasi ini sebagai media penyimpanan data yang berhubungan dengan data user yang menggunakan aplikasi, peta offline color radiomap, dan data pendukung lainnya. Dalam aplikasi ini, DBMS yang digunakan adalah *MySQL*. Untuk menjalankan *MySQL*, dibutuhkan server yang menyediakan DBMS *MySQL*. Untuk berkomunikasi antara aplikasi *Finding Dosen* dengan server, digunakan *REST API* sebagai *web service*, aplikasi akan melakukan permintaan *HTTP* ke server yang dibuat.



Gambar 4.1 Gambaran umum sistem

4.2 Deskripsi Umum Perangkat Lunak

Aplikasi *Finding Dosen* merupakan sebuah perangkat lunak berbasis *Android* yang berfungsi sebagai aplikasi pencarian posisi seorang Dosen Filkom Universitas Brawijaya yang berada di dalam ruangan atau gedung. Aplikasi ini dibangun untuk dua pengguna, yaitu Mahasiswa dan Dosen. Mahasiswa dapat mencari posisi dosen dalam aplikasi ini dengan tepat, mahasiswa juga dapat melihat profil dosen, dan Dosen dapat merubah statusnya.

4.3 Identifikasi Aktor

Aktor merupakan seorang ataupun sistem eksternal yang bisa berinteraksi dengan sistem. Adapun aktor pada sistem ini ditunjukkan pada tabel 4.1 dibawah ini.



Tabel 4.1 Deskripsi Aktor

No	Aktor	Deskripsi
1	Mahasiswa	Merupakan orang yang dapat melakukan <i>register</i> , <i>login</i> , <i>logout</i> , pencarian dosen, melihat profil dosen, melihat posisi dosen, melihat profil diri mengubah profil diri.
2	Dosen	Merupakan orang yang dapat melakukan <i>register</i> , <i>login</i> , <i>logout</i> , mengubah status untuk dapat dicari atau tidak, melihat profil diri, mengubah profil diri.

4.4 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus disediakan di dalam aplikasi, bagaimana aplikasi bereaksi pada masukan tertentu dan bagaimana perilaku aplikasi pada situasi tertentu. Kebutuhan fungsional yang ada pada Aplikasi Finding Dosen akan dijelaskan pada tabel 4.2.

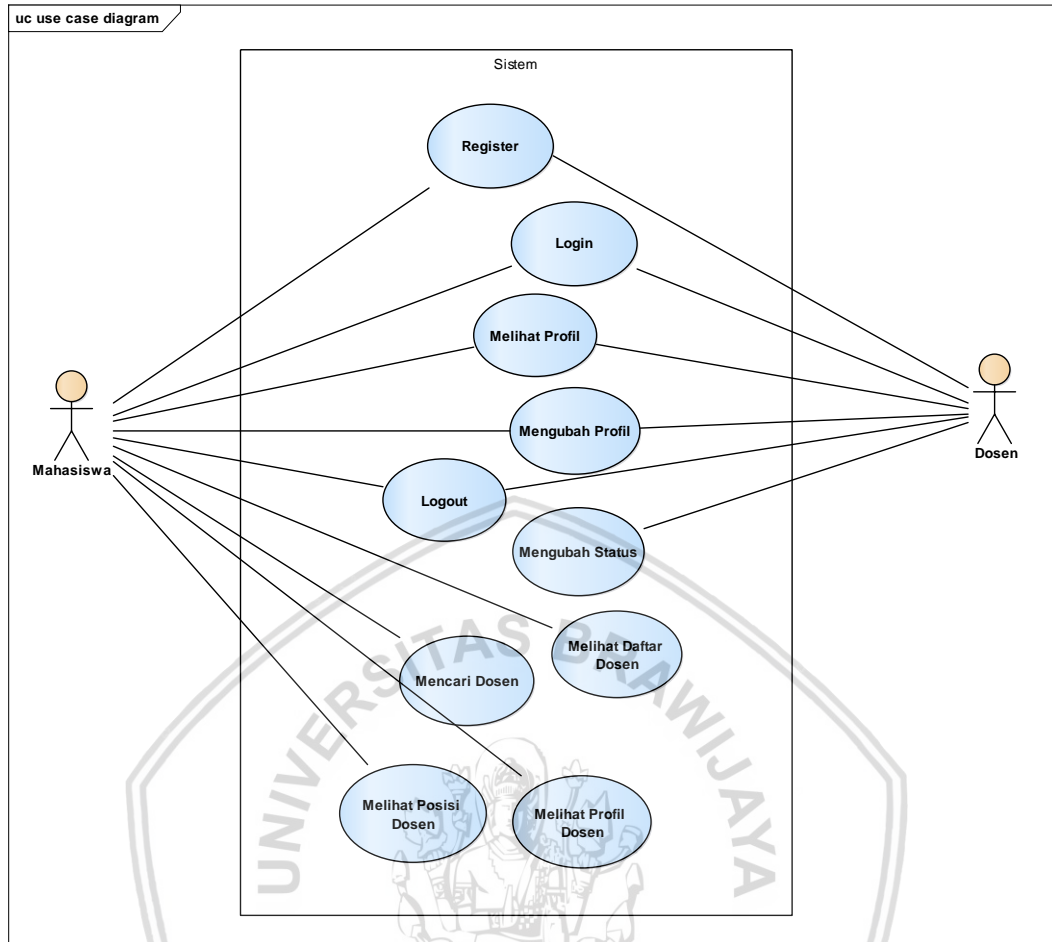
Tabel 4.2 Kebutuhan Fungsional

No	Aktor	Kode	Nama Fungsi	Deskripsi
1	Mahasiswa dan Dosen	SRS-FIND-01	Login	Aplikasi menyediakan fitur untuk melakukan login dengan cara memasukkan username dan password
2		SRS-FIND-02	Register	Aplikasi menyediakan fitur untuk melakukan register dengan memasukkan data yang diperlukan untuk pendaftaran
3		SRS-FIND-03	Melihat profil	Aplikasi menyediakan fitur untuk melihat profil diri sendiri
4		SRS-FIND-04	Mengubah profil	Aplikasi menyediakan fitur

				untuk mengubah profil
5		SRS-FIND-052	Logout	Aplikasi menyediakan fitur untuk keluar dari sesi aplikasi.
6	Mahasiswa	SRS-FIND-06	Melihat daftar dosen	Aplikasi menyediakan fitur untuk melihat daftar dosen
7		SRS-FIND-07	Mencari dosen	Aplikasi menyediakan fitur untuk mencari dosen berdasarkan nama
8		SRS-FIND-08	Melihat profil dosen	Aplikasi menyediakan fitur untuk melihat profil dosen
9		SRS-FIND-09	Melihat posisi dosen	Aplikasi menyediakan fitur untuk melihat posisi dosen yang ditampilkan dalam peta
10	Dosen	SRS-FIND-10	Mengubah status	Aplikasi menyediakan fitur untuk mengubah status pada dosen

4.5 Use Case Diagram

Use case diagram merupakan tahap *konseptualitas dalam konteks Unified Modeling Language (UML)* dimana mendeskripsikan bagaimana perangkat lunak (aplikasi) akan digunakan oleh user sehingga membentuk sistem secara teratur yang dilakukan oleh aktor. Penggambaran aplikasi *Finding Dosen* secara umum menggunakan use case diagram berikut:



Gambar 4.2 Use Case Diagram

4.6 Use Case Scenario

4.6.1 Use Case Scenario Login: SRS-FIND-01

Tabel 4.3 Use Case Scenario Login

Item	Deskripsi
Nama Use Case	Login
Deskripsi	Use case ini digunakan oleh aktor untuk dapat masuk ke dalam aplikasi dan dapat menggunakan fitur yang telah disediakan.
Aktor	Mahasiswa dan Dosen
Pra-Kondisi	Aktor memasuki halaman login
Tindakan	<ol style="list-style-type: none"> 1. Aktor memasukkan username 2. Aktor memasukkan password 3. Aktor menekan tombol login
Alternatif	<ol style="list-style-type: none"> 1. Jika salah satu kolom tidak diisi, maka akan menampilkan pesan "Semua kolom harus diisi!"



	2. Jika username belum terdaftar atau password salah, maka akan muncul pesan "Login gagal!"
Post-Kondisi	Aktor berhasil masuk ke dalam aplikasi

4.6.2 Use Case Scenario Register: SRS-FIND-02

Tabel 4.4 Use Case Scenario Register

Item	Deskripsi
Nama Use Case	Register
Deskripsi	Use case ini digunakan oleh aktor untuk mendaftarkan diri untuk dapat menggunakan aplikasi
Aktor	Mahasiswa dan Dosen
Pra-Kondisi	Aktor memasuki halaman register
Tindakan	<ol style="list-style-type: none"> 1. Aktor memasukkan Nama 2. Aktor memasukkan Nomor Identitas 3. Aktor memasukkan Email 4. Aktor memasukkan Password 5. Aktor memasukkan Nomor Telpon 6. Aktor menekan tombol Register
Alternatif	<ol style="list-style-type: none"> 1. Jika salah satu kolom tidak diisi, maka akan menampilkan pesan "Semua kolom harus diisi!" 2. Jika email sudah terdaftar, maka akan menampilkan pesan "Pendaftaran gagal"
Post-Kondisi	Aktor berhasil mendaftarkan dirinya, lalu masuk ke halaman login

4.6.3 Use Case Scenario Melihat Profil: SRS-FIND-03

Tabel 4.5 Use Case Scenario Melihat Profil

Item	Deskripsi
Nama Use Case	Melihat Profil
Deskripsi	Use case ini digunakan oleh aktor untuk melihat profil dirinya sendiri
Aktor	Mahasiswa dan Dosen
Pra-Kondisi	Aktor memasuki halaman utama

Tindakan	1. Aktor memilih menu profil
Alternatif	
Post-Kondisi	Aktor berhasil melihat profil

4.6.4 Use Case Scenario Mengubah Profil: SRS-FIND-04

Tabel 4.6 Use Case Scenario Mengubah Profil

Item	Deskripsi
Nama Use Case	Merubah Profil
Deskripsi	Use case ini digunakan oleh aktor untuk merubah profil
Aktor	Mahasiswa dan Dosen
Pra-Kondisi	Aktor memasuki halaman Profil
Tindakan	1. Aktor menekan tombol rubah profil 2. Aktor memasukkan data profil yang baru 3. Aktor menekan tombol simpan
Alternatif	1. Jika ada salah satu kolom yang kosong maka akan menampilkan pesan, "Semua kolom harus diisi!"
Post-Kondisi	Aktor berhasil merubah profil

4.6.5 Use Case Scenario Logout: SRS-FIND-05

Tabel 4.7 Use Case Scenario Logout

Item	Deskripsi
Nama Use Case	Logout
Deskripsi	Use case ini digunakan oleh aktor untuk keluar dari aplikasi
Aktor	Mahasiswa dan Dosen
Pra-Kondisi	Aktor memasuki halaman utama
Tindakan	1. Aktor memilih menu Logout
Alternatif	-
Post-Kondisi	Aktor berhasil keluar dari aplikasi

4.6.6 Use Case Scenario Melihat Daftar Dosen: SRS-FIND-06

Tabel 4.8 Use Case Scenario Melihat Daftar Dosen

Item	Deskripsi
Nama Use Case	Melihat Dosen yang Tersedia
Deskripsi	Use case ini digunakan oleh mahasiswa untuk melihat dosen yang tersedia untuk dicari posisinya
Aktor	Mahasiswa
Pra-Kondisi	Aktor telah login
Tindakan	1. Aktor memilih menu Daftar Dosen
Alternatif	1. Jika tidak ada dosen, aplikasi akan menampilkan list kosong
Post-Kondisi	Aplikasi menampilkan daftar dosen

4.6.7 Use Case Scenario Mencari Dosen: SRS-FIND-07

Tabel 4.9 Use Case Scenario Mencari Dosen

Item	Deskripsi
Nama Use Case	Mencari Dosen
Deskripsi	Use case ini digunakan oleh mahasiswa untuk mencari dosen berdasarkan nama
Aktor	Mahasiswa
Pra-Kondisi	Aktor memasuki halaman Daftar Dosen
Tindakan	1. Aktor mengetuk pada kolom cari 2. Aktor memasukkan nama dosen 3. Aktor menekan tombol cari
Alternatif	1. Jika tidak ada dosen yang dengan nama yang dimasukkan oleh mahasiswa, maka aplikasi menampilkan list kosong
Post-Kondisi	Mahasiswa berhasil mencari dosen berdasarkan nama

4.6.8 Use Case Scenario Melihat Profil Dosen: SRS-FIND-08

Tabel 4.10 Use Case Scenario Melihat Profil Dosen

Item	Deskripsi
------	-----------

Nama Use Case	Melihat Profil Dosen
Deskripsi	Use case ini digunakan oleh mahasiswa untuk melihat profil dosen
Aktor	Mahasiswa
Pra-Kondisi	Aktor memasuki halaman Daftar Dosen
Tindakan	1. Aktor memliih dosen yang tersedia
Alternatif	-
Post-Kondisi	Mahasiswa berhasil melihat profil dosen

4.6.9 Use Case Scenario Melihat Posisi Dosen: SRS-FIND-09

Tabel 4.11 Use Case Scenario Melihat Posisi Dosen

Item	Deskripsi
Nama Use Case	Melihat Posisi Dosen
Deskripsi	Use case ini digunakan oleh mahasiswa untuk melihat posisi dosen yang ditampilkan dalam peta gedung
Aktor	Mahasiswa
Pra-Kondisi	Aktor memasuki halaman Profil Dosen
Tindakan	1. Aktor menekan tombol lihat pada peta
Alternatif	-
Post-Kondisi	Mahasiswa berhasil melihat posisi dosen yang ditampilkan dalam peta gedung

4.6.10 Use Case Scenario Mengubah Status: SRS-FIND-10

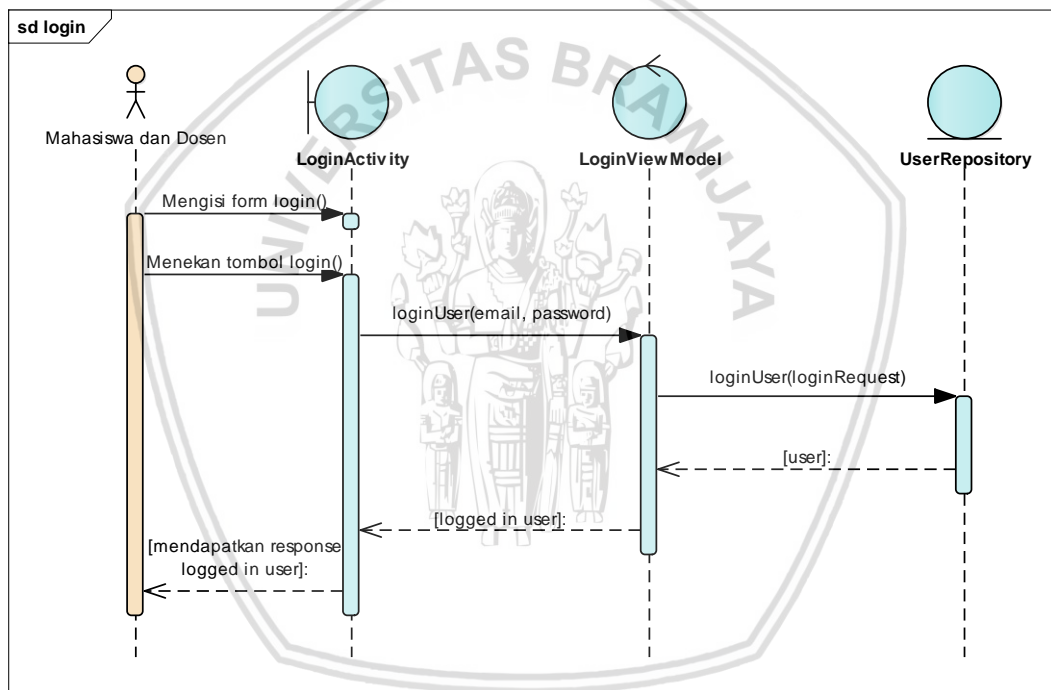
Tabel 4.12 Use Case Scenario Mengubah Status

Item	Deskripsi
Nama Use Case	Merubah Status
Deskripsi	Use case ini digunakan oleh aktor untuk merubah status agar dapat dicari atau tidak

Aktor	Dosen
Pra-Kondisi	Aktor memasuki halaman utama
Tindakan	<ol style="list-style-type: none"> 1. Aktor menekan tombol ubah status 2. Aktor memilih status 3. Aktor menambahkan keterangan status 4. Aktor menekan tombol simpan
Alternatif	1. Jika tidak lokasi tidak tersedia di dalam database, maka akan muncul pesan "Akses point tidak ditemukan!"
Post-Kondisi	Aktor berhasil merubah status

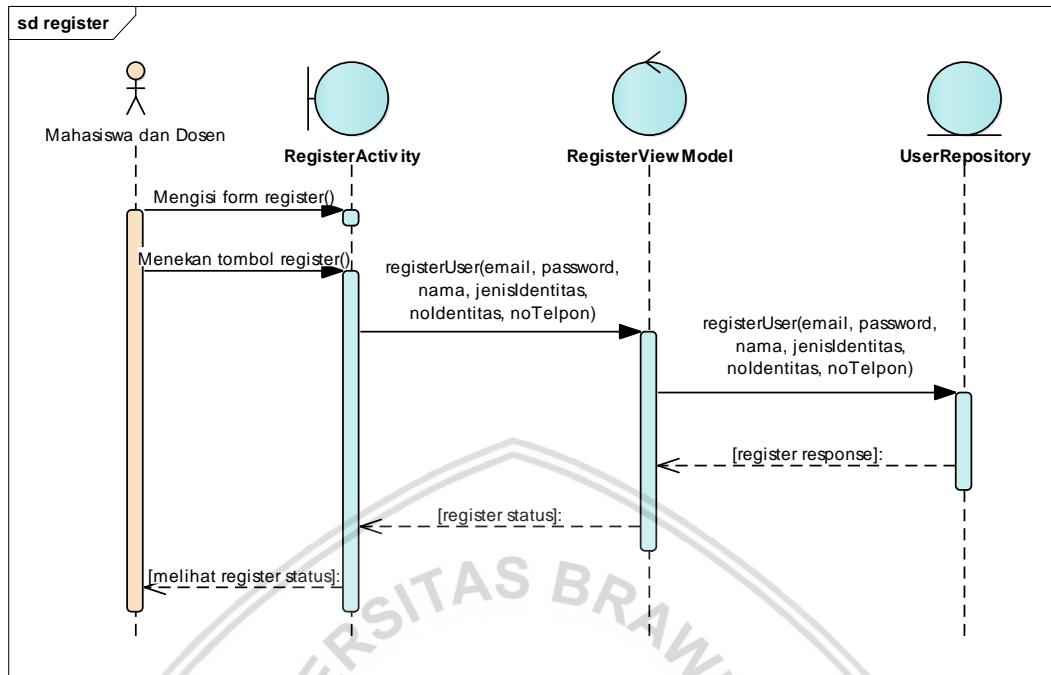
4.7 Sequence Diagram

4.7.1 Sequence Diagram Login (SRS-FIND-01)



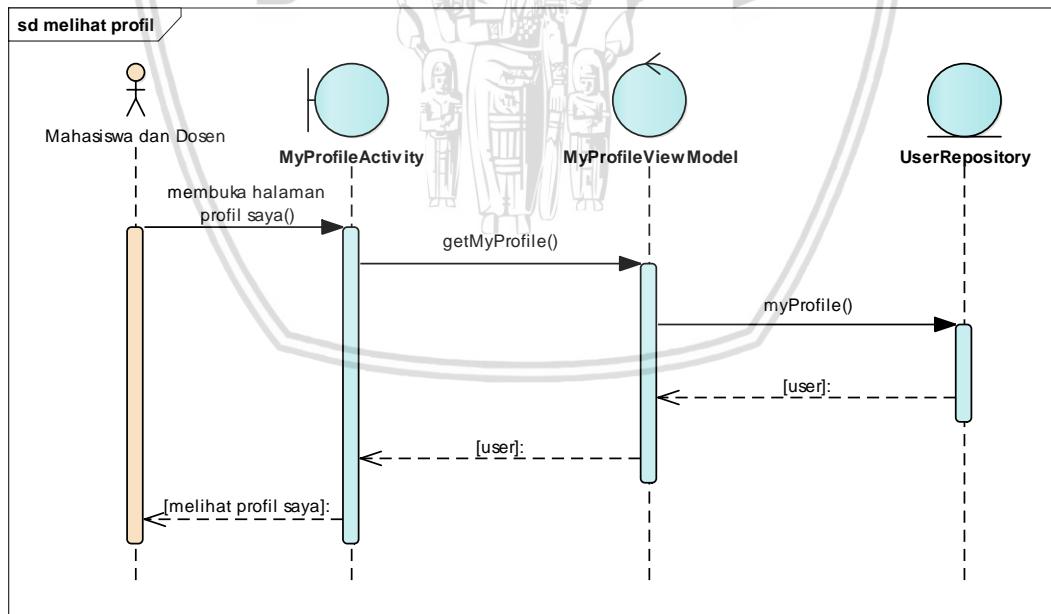
Gambar 4.3 Sequence Diagram Login

4.7.2 Sequence Diagram Register (SRS-FIND-02)



Gambar 4.4 Sequence Diagram Register

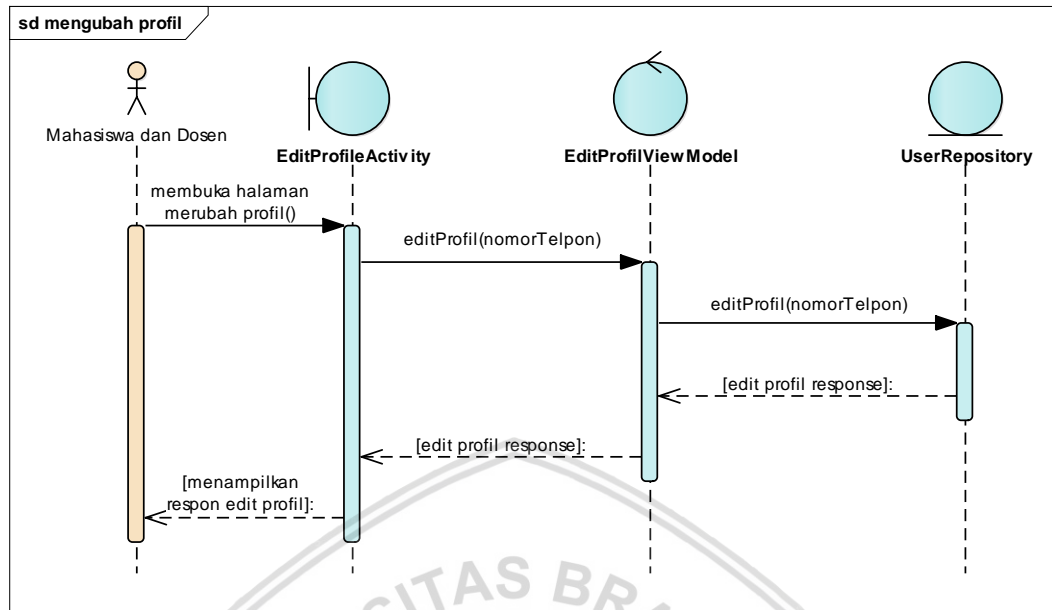
4.7.3 Sequence Diagram Melihat Profil (SRS-FIND-03)



Gambar 4.5 Sequence Diagram Melihat Profil

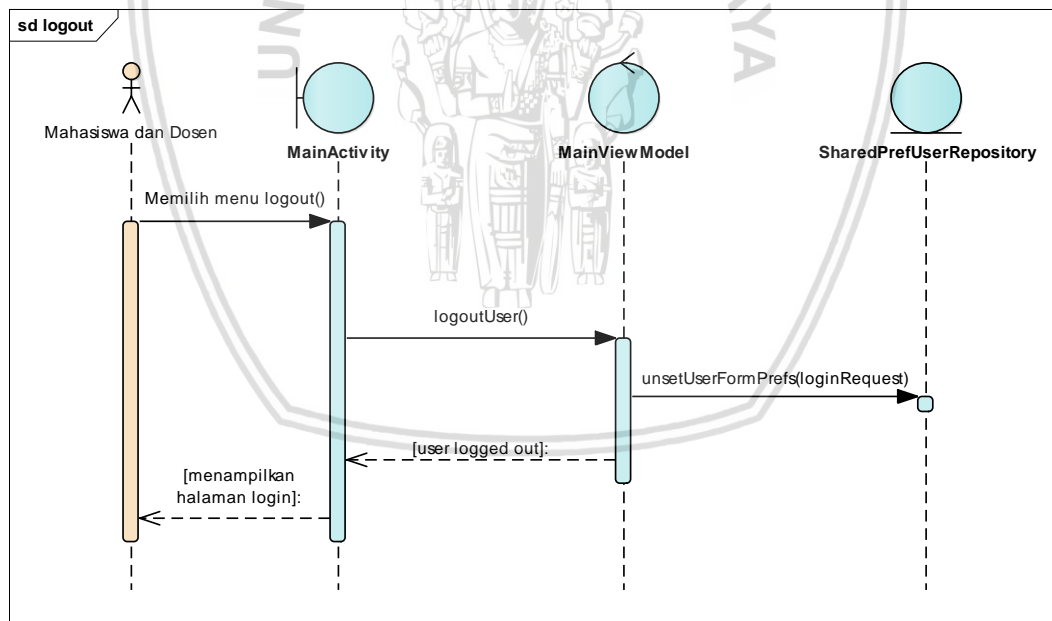


4.7.4 Sequence Diagram Mengubah Profil (SRS-FIND-04)



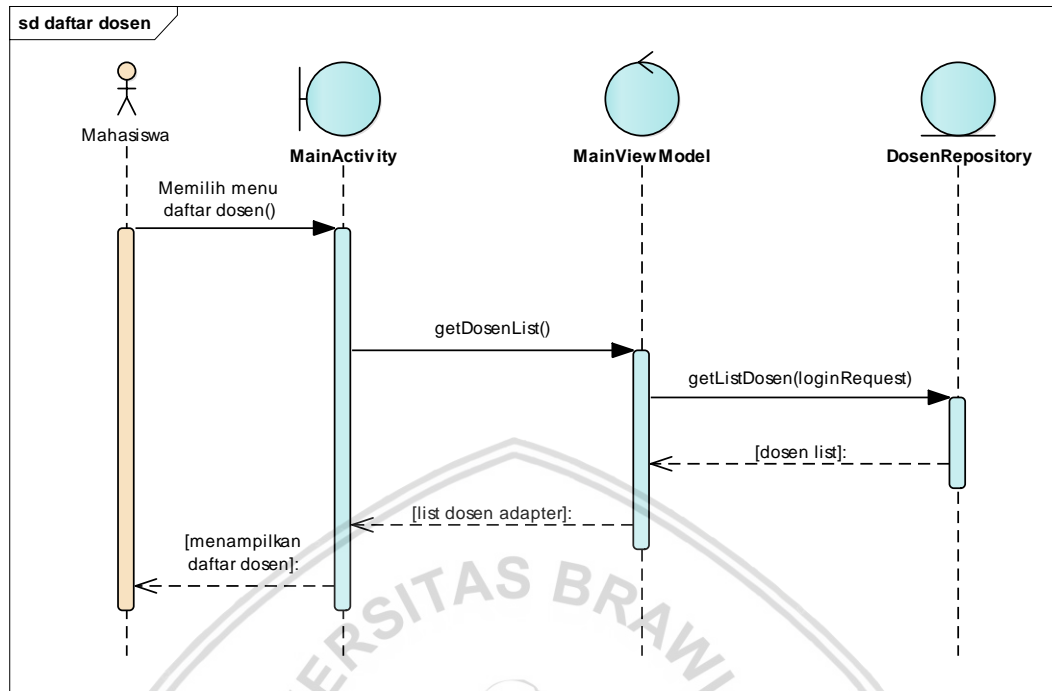
Gambar 4.6 Sequence Diagram Mengubah Profil

4.7.5 Sequence Diagram Logout (SRS-FIND-05)



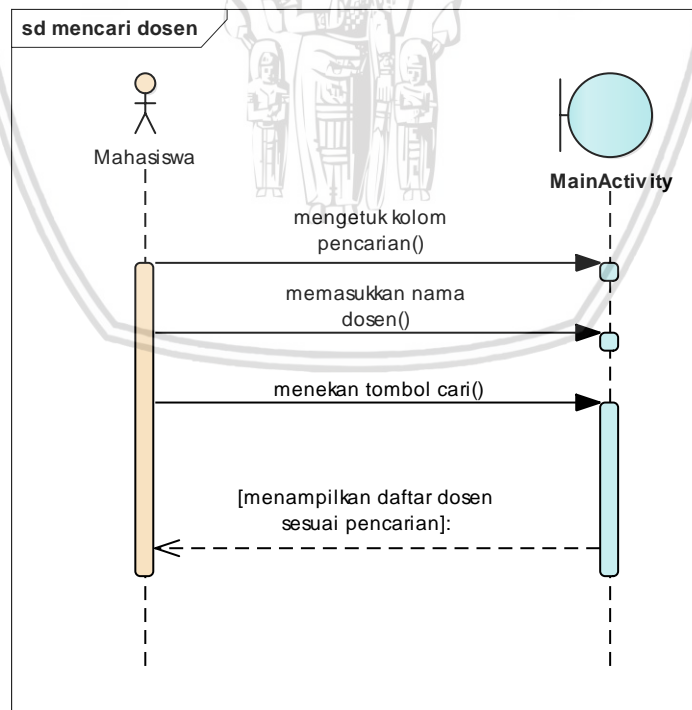
Gambar 4.7 Sequence Diagram Logout

4.7.6 Sequence Diagram Melihat Daftar Dosen (SRS-FIND-06)



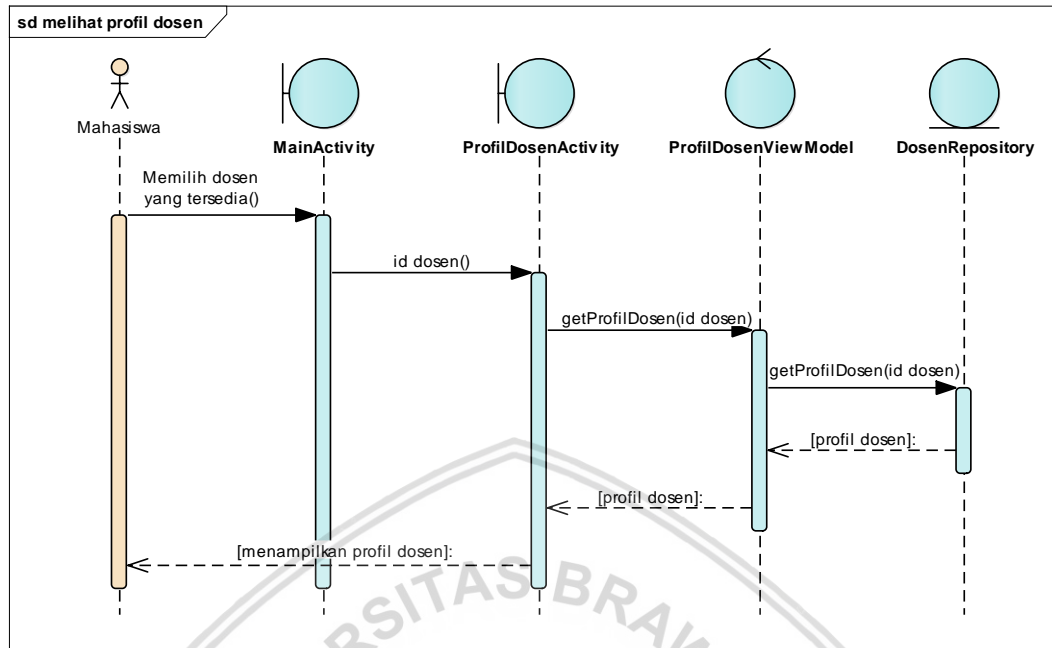
Gambar 4.8 Sequence Diagram Melihat Daftar Dosen

4.7.7 Sequence Diagram Mencari Dosen (SRS-FIND-07)



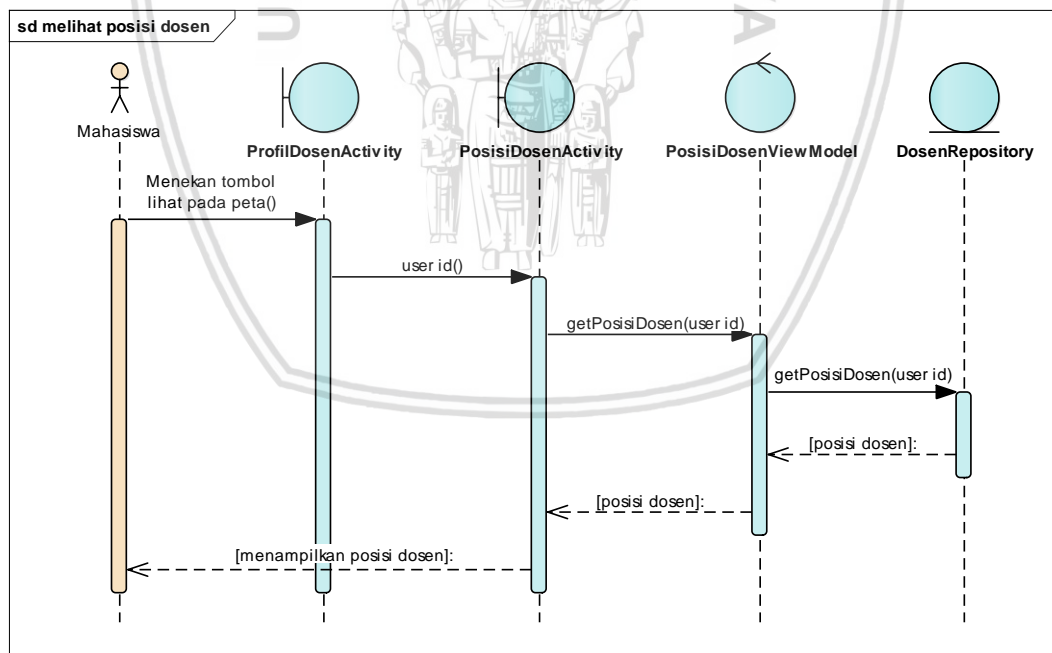
Gambar 4.9 Sequence Diagram Mencari Dosen

4.7.8 Sequence Diagram Melihat Profil Dosen (SRS-FIND-08)



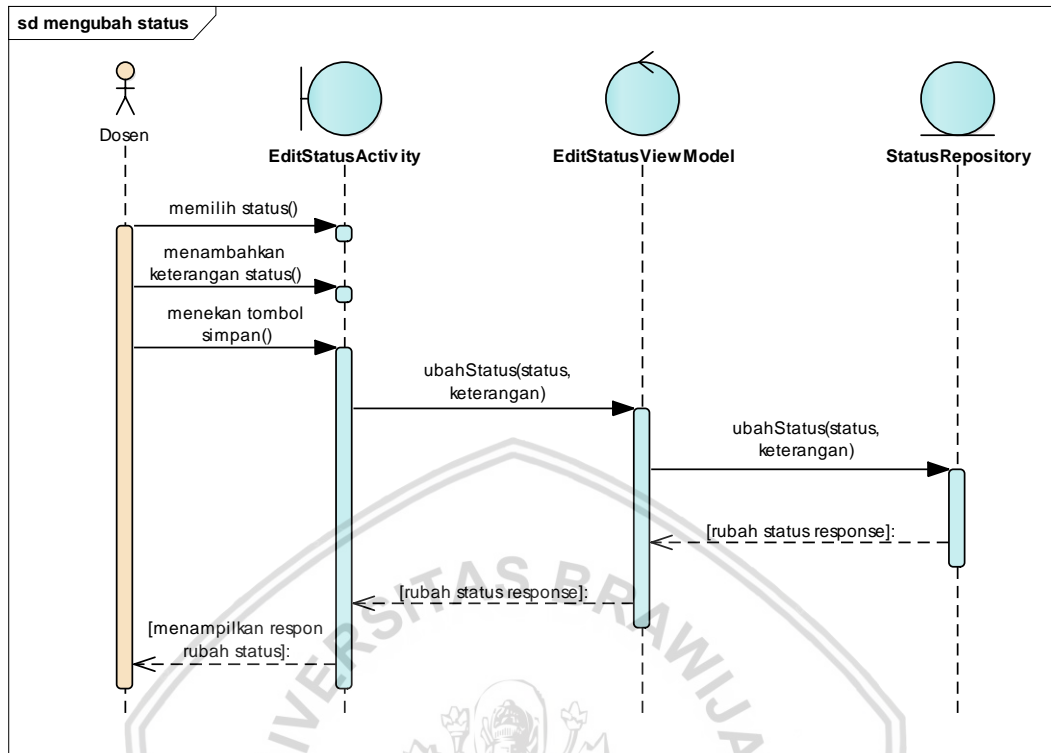
Gambar 4.10 Sequence Diagram Melihat Profil Dosen

4.7.9 Sequence Diagram Melihat Posisi Dosen (SRS-FIND-09)



Gambar 4.11 Sequence Diagram Melihat Posisi Dosen

4.7.10 Sequence Diagram Mengubah Status (SRS-FIND-10)



Gambar 4.12 Sequence Diagram Mengubah Status

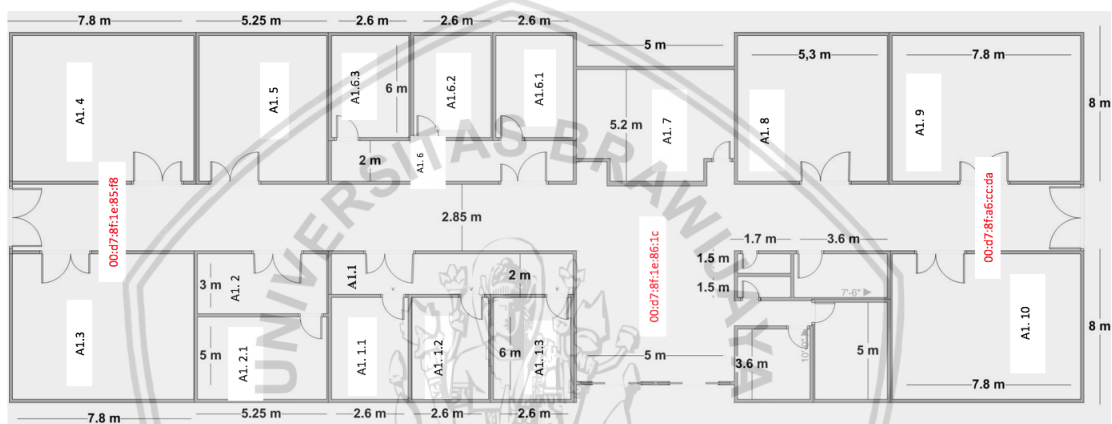
BAB 5 PENGAMBILAN DAN PENGOLAHAN DATA

5.1 Pengambilan Data

Proses pengambilan data dilakukan dengan mencari lokasi akses point (AP) dari masing-masing gedung yang ada di FILKOM Universitas Brawijaya. Kemudian, membuat peta *grid* dari setiap gedung, setelah itu mengambil data RSSI dari setiap *grid* yang dibuat. Data yang diambil pada penelitian ini adalah data dengan denah pada tahun 2017.

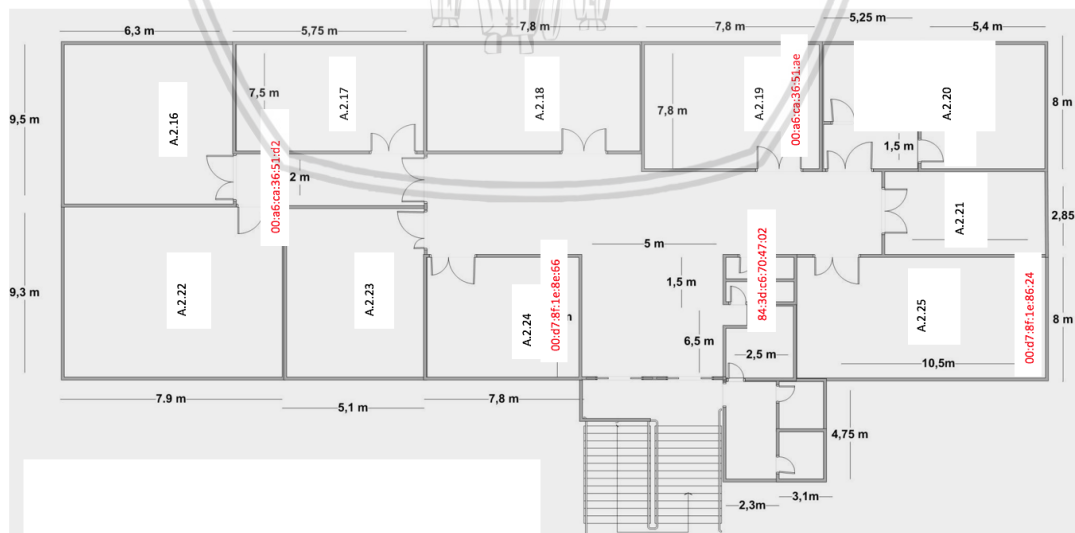
5.1.1 Peletakkan Access Point

Berikut adalah titik peletakkan access point yang ada pada gedung-gedung di FILKOM Universitas Brawijaya.



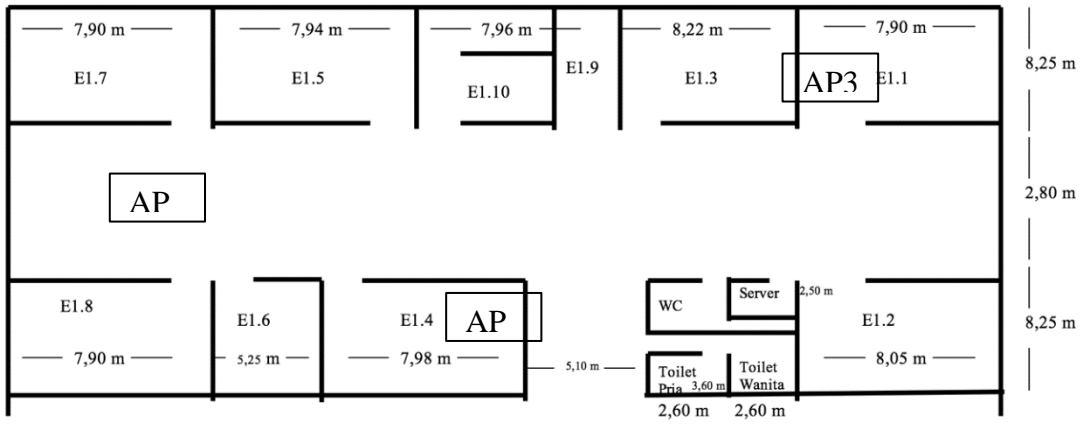
Gambar 5.1 Peta AP Gedung A Lantai 1 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



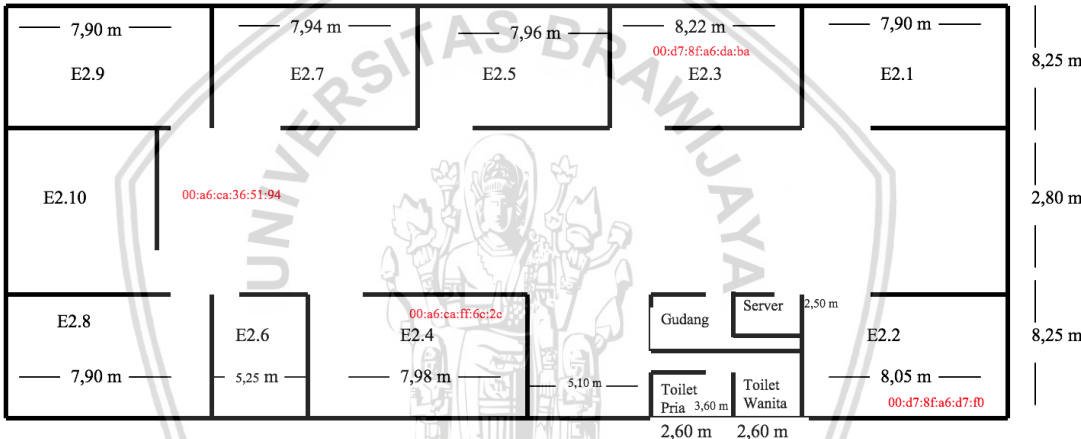
Gambar 5.2 Peta AP Gedung A Lantai 2 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



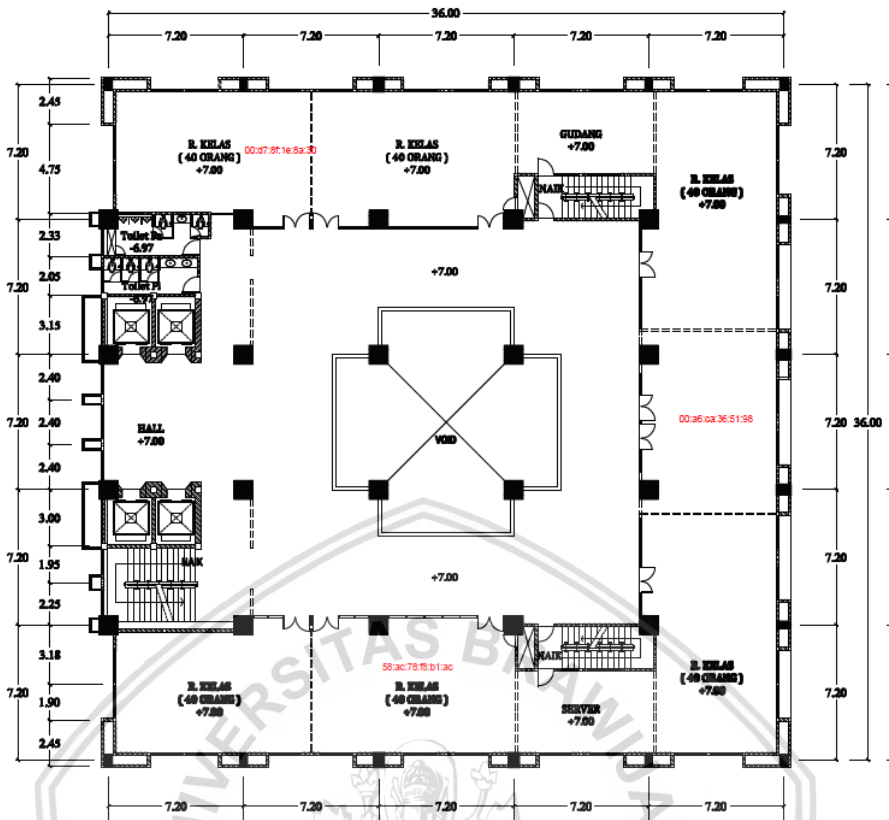
Gambar 5.3 Peta AP Gedung E Lantai 1 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



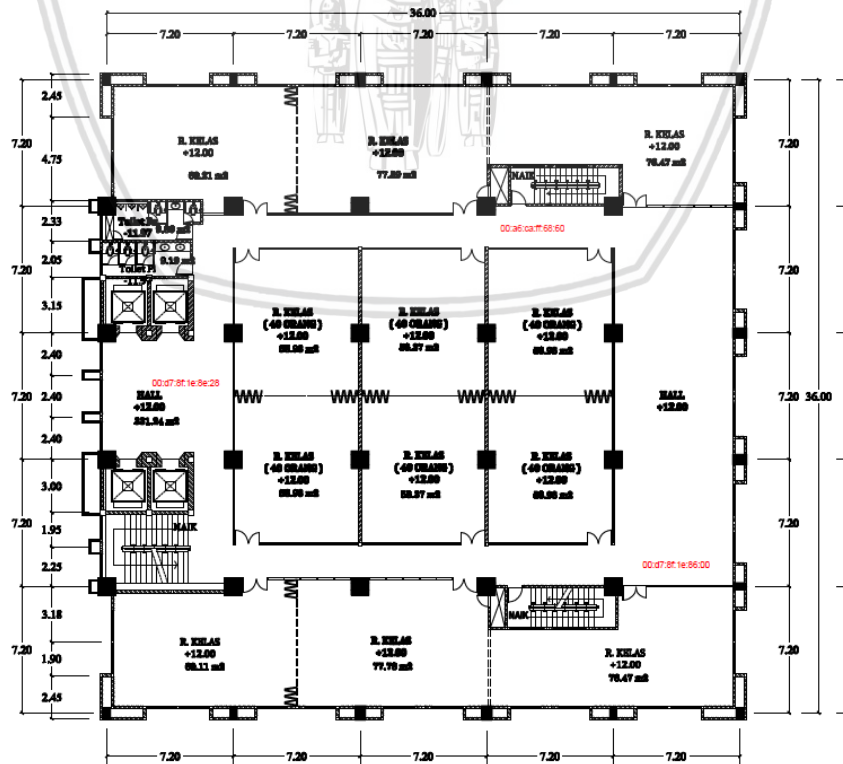
Gambar 5.4 Peta AP Gedung E Lantai 2 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



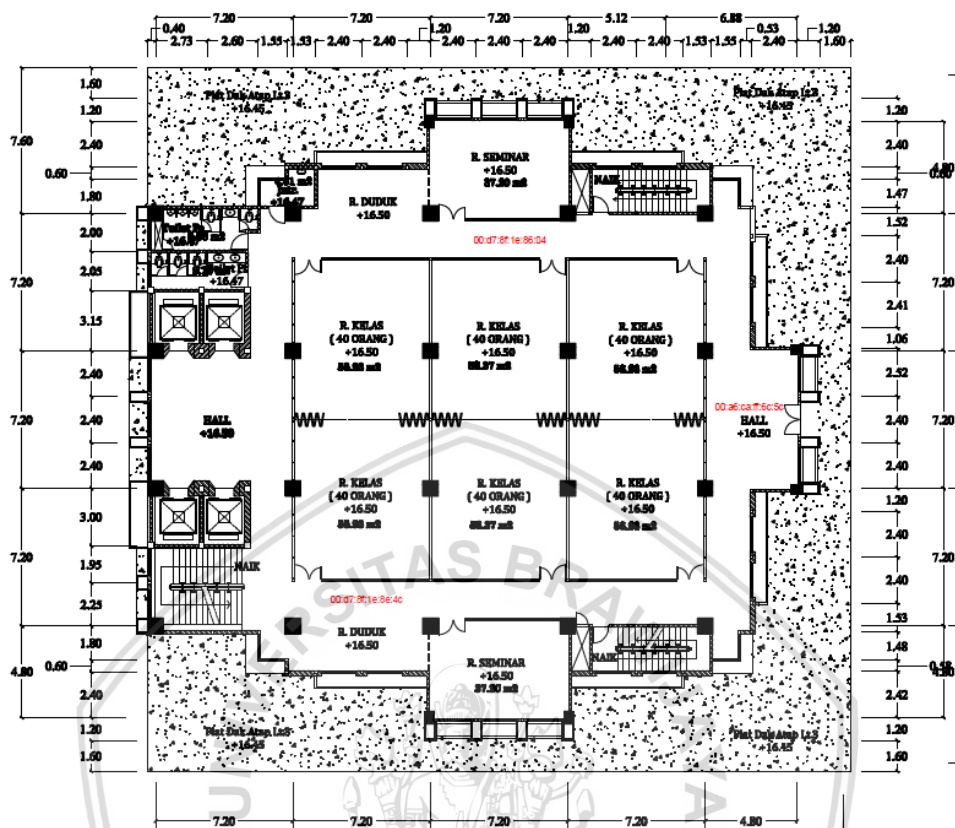
Gambar 5.5 Peta AP Gedung F Lantai 2 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



Gambar 5.6 Peta AP Gedung F Lantai 3 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)



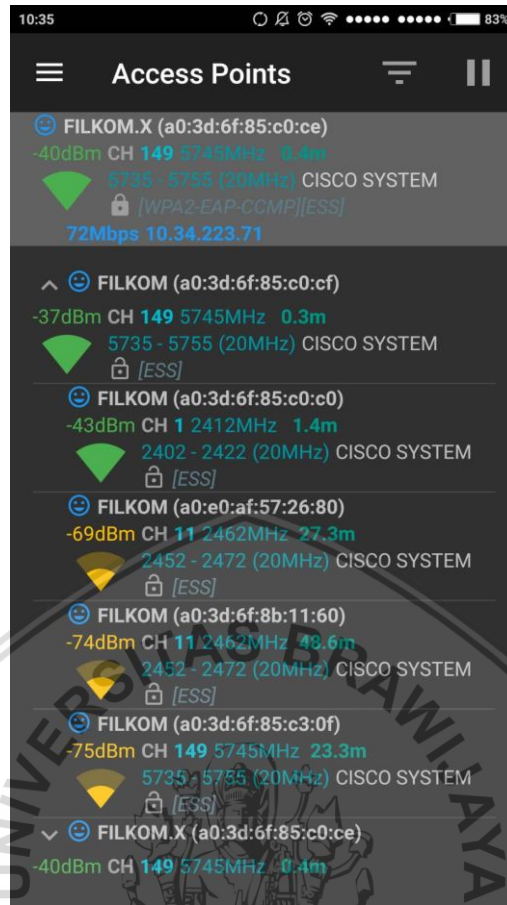
Gambar 5.7 Peta AP Gedung F Lantai 4 FILKOM Universitas Brawijaya

Sumber: PSIK FILKOM (2017)

5.1.2 Proses Pengambilan Data

Pengambilan data dilakukan dengan cara *scanning* kekuatan sinyal wifi dengan aplikasi *Wifi Analyzer* yang ditampilkan pada Gambar 5.5. Data diambil dari ruangan yang ada di gedung-gedung FILKOM Universitas Brawijaya pada setiap grid yang telah ditentukan, setiap grid yang dibuat tidak memiliki ukuran yang sama karena pada setiap ruangan dalam gedung tidak memiliki ukuran yang panjang dan lebar yang sama. Pembuatan grid tidak dibuat melewati tembok, karena data kekuatan sinyal wifi bersifat tidak linier antara data di dalam sisi tembok dan di luar sisi tembok. Dari aplikasi *Wifi Analyzer*, didapatkan data wifi yang ada di setiap grid, kemudian data diexport dalam bentuk file teks yang nantinya dapat diolah.





Gambar 5.5 Wifi Analyzer

5.2 Pengolahan Data

Pengolahan data dimulai dari proses rekapitulasi data RSSI yang dikelompokkan berdasarkan ruangan dan AP, data yang diambil kemudian diolah adalah data yang berasal hanya dari 3 AP reference pada setiap lantai pada gedung, AP reference merupakan 3 AP hasil reduksi seluruh AP pada setiap lantai, sehingga mewakili AP yang lain.

Langkah awal pengolahan data adalah menormalisasi nilai RSSI yang berupa data dengan nilai negatif (-dbm), tujuannya untuk menjadikan nilai RSSI dalam bentuk nilai positif, persamaan yang digunakan untuk melakukan normalisasi adalah:

$$N = \text{JarakRSSI} + (x + \text{MinRSSI}) \tag{5.1}$$

Keterangan: N = Normalisasi
 JarakRSSI = Jarak maksimal yang didapat dari AP reference
 x = Nilai dari RSSI
 MinRSSI = 0

Pada penelitian ini, nilai dari JarakRSSI tergantung dari scan yang didapat dari masing-masing lantai tiap gedung, yaitu pencarian jarak antara RSSI tertinggi dan terendah, dan nilai MinRSSI yang di-scan adalah 0. Tahap selanjutnya adalah

pengolahan data yang telah dinormalisasi kedalam level warna RGB. Cara mengkonversikan RSSI ke dalam level warna dengan persamaan berikut:

$$L = \left(\frac{255}{\text{JarakRSSI}} \right) * N \quad (5.2)$$

Keterangan: L = Level Warna
 JarakRSSI = Jarak maksimal yang didapat dari AP reference
 N = Normalisasi

Setelah data dimasukkan dalam level maka akan dihasilkan nilai dalam bentuk angka desimal yang merupakan nilai dari komponen RGB peta warna. Selanjutnya nilai RSSI akan dikonversikan secara manual kedalam warna dan menjadi peta warna berdasarkan komposisi warna yang telah ditemukan dari proses pengolahan nilai RSSI.

5.2.1 Data Sampel

Data sampel berisikan contoh pengolahan data kekuatan *wifi* dari pengumpulan, perhitungan, pembuatan peta, dan perhitungan tingkat akurasi hingga proses interpolasi. Pada Tabel 5.1 akan ditampilkan proses pengambilan data pada Ruang A1.2 pada Gedung A Lantai 1.

Tabel 5.1 Data kekuatan sinyal wifi

BSSID	Scan A1.2 A										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
0e:c3:9b:ef:14:75	-88	-88	-88	-88	-87	-87	-87	-87	0	0	-70
a0:3d:6f:85:c0:c0	-55	-55	-55	-54	-54	-56	-56	-56	-56	-55	-55,2
a0:3d:6f:85:c0:c1	-55	-55	-54	-54	-55	-54	-54	-55	-55	-57	-54,8
a0:3d:6f:85:c0:ce	-65	-65	-65	-65	-65	-64	-64	-63	-65	-66	-64,7
a0:3d:6f:85:c0:cf	-65	-65	-65	-65	-65	-64	-64	-63	-64	-65	-64,5
a0:3d:6f:85:c3:00	-74	-77	-77	-77	-77	0	0	0	0	-76	-45,8
a0:3d:6f:85:c3:01	-75	-75	-75	-75	-74	-77	-77	-75	-70	-74	-74,7
a0:3d:6f:85:c3:0e	-87	-87	-86	-86	-86	-86	-87	-86	-86	-86	-86,3
a0:3d:6f:85:c3:0f	-87	-87	-87	-86	-86	-86	-87	0	0	-85	-69,1
a0:3d:6f:8b:11:61	-80	-80	-80	-80	-80	-80	-80	-80	-80	-79	-79,9
a0:3d:6f:8f:96:00	-80	0	0	0	0	-83	-83	-83	-83	0	-41,2
a0:3d:6f:8f:96:01	-79	-79	0	0	0	0	0	-81	-81	-81	-40,1
a0:e0:af:57:26:80	-82	-82	-82	-82	-80	-80	-80	-80	0	-77	-72,5
a0:e0:af:57:26:81	-83	-83	-83	-83	-83	-83	-82	-82	-82	0	-74,4
c2:9f:db:2b:b5:68	-88	-88	-88	-88	-89	-89	-89	-89	0	0	-70,8

Dala Tabel 5.1 di atas merupakan data hasil scanning pada ruangan A1.2 Grid A, pada tabel di atas dilihat bahwa proses *scanning* dilakukan sebanyak 10 kali, kolom data yang kosong menandakan bahwa AP tidak terdeteksi pada saat

scanning, sehingga nilai kolom kosong namun dalam perhitungan rata-rata tetap dijumlahkan dengan nilai "0".

Tabel 5.2 Rata-rata dan pembulatan

Access Point \ Ruang Scan	A1.2 A	A1.2 A
AP1	-54,8	-55
AP2	-74,7	-75
AP3	-74,4	-74

Keterangan:

AP1	a0:3d:6f:85:c0:c1	FILKOM.X 2,4 GHz
AP2	a0:3d:6f:85:c3:01	FILKOM.X 2,4 GHz
AP3	a0:e0:af:57:26:81	FILKOM.X 2,4 GHz

Tabel 5.2 adalah hasil rata-rata dari scan yang dilakukan dan dilakukan pembulatan, access point yang digunakan sebagai *reference* sebanyak tiga buah yaitu *access point* yang memiliki SSID FILKOM.X dengan frekuensi 2,4 GHz. *Access point* yang dipilih menggunakan frekuensi 2,4 GHz dikarenakan kompatibilitas beberapa perangkat saat ini yang belum memiliki *wifi adapter* yang menangkap *access point* dengan frekuensi 5 GHz.

Tabel 5.3 Normalisasi

Access Point \ Ruang Scan	A1.2 A	A1.2 A
AP1	-55	25
AP2	-75	5
AP3	-74	6

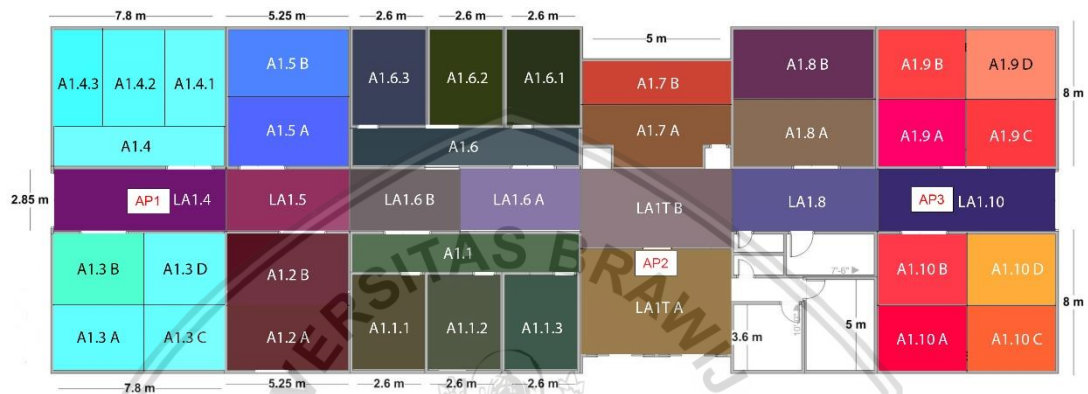
Pada Tabel 5.3, terlihat proses normalisasi nilai RSSI yang awalnya adalah nilai negative dirubah menjadi nilai positif, normalisasi dilakukan dengan menggunakan persamaan 5.1.

Tabel 5.4 Pelevelan RGB

Access Point \ Ruang Scan	A1.2 A	A1.2 A
AP1 / R	79,6875	80
AP2 / G	15,9375	16
AP3 / B	19,125	19

Pada Tabel 5.4, terlihat proses pelevelan dari nilai hasil normalisasi menjadi warna RGB, proses pelevelan menggunakan persamaan 5.2 dengan nilai JarakRSSI dalam kasus ini adalah 80 dan nilai MinRSSI adalah 0, hasil pelevelan kemudian dibulatkan. Hasil pembulatan dari pelevelan tersebut digunakan sebagai komponen nilai RGB yang digunakan untuk memberikan warna pada grid sampling yang telah ditentukan.

Dari data yang telah diolah di atas, maka komponen warna RGB dapat divisualisasikan dalam setiap grid peta, dalam sampel data ini ditampilkan peta warna gedung A lantai 1 FILKOM Universitas Brawijaya pada gambar 5.6:



Gambar 5.6 Color Radiomap Gedung A Lantai 1

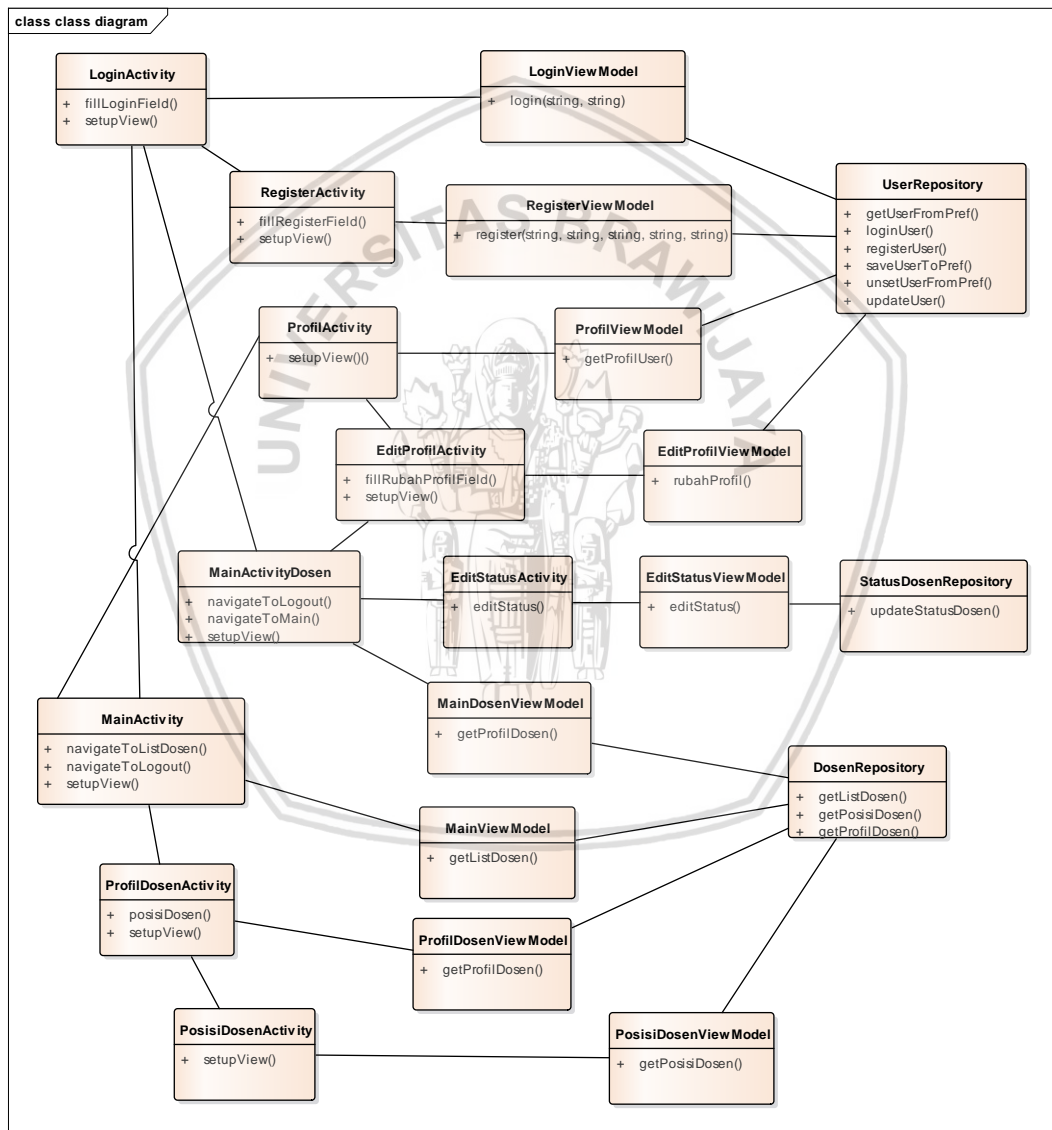


BAB 6 PERANCANGAN DAN IMPLEMENTASI

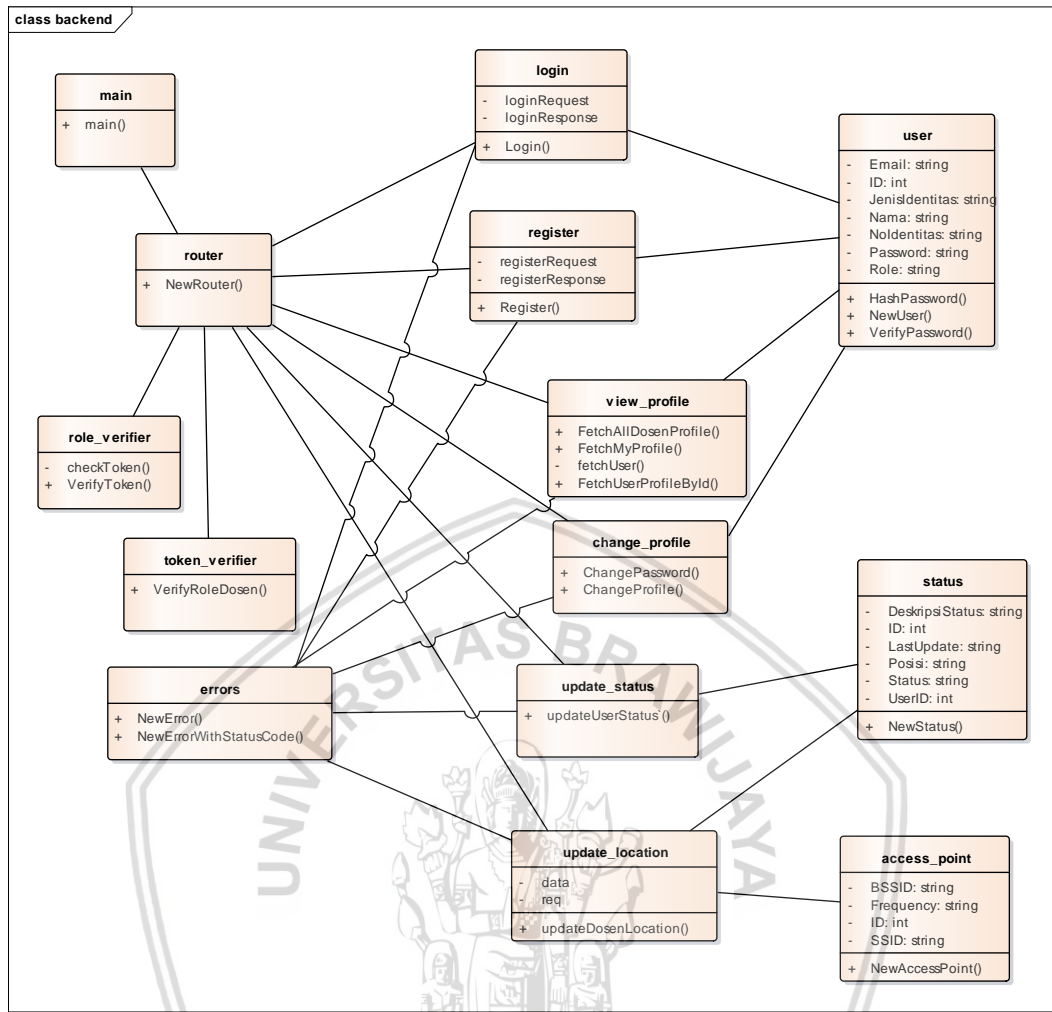
6.1 Perancangan

6.1.1 Class Diagram

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem sehingga dapat memberikan gambaran dari sistem dan relasi-relasi yang ada di dalamnya. *Class* diagram pada perancangan aplikasi Finding Dosen ditujukan oleh Diagram 6.1 berikut.



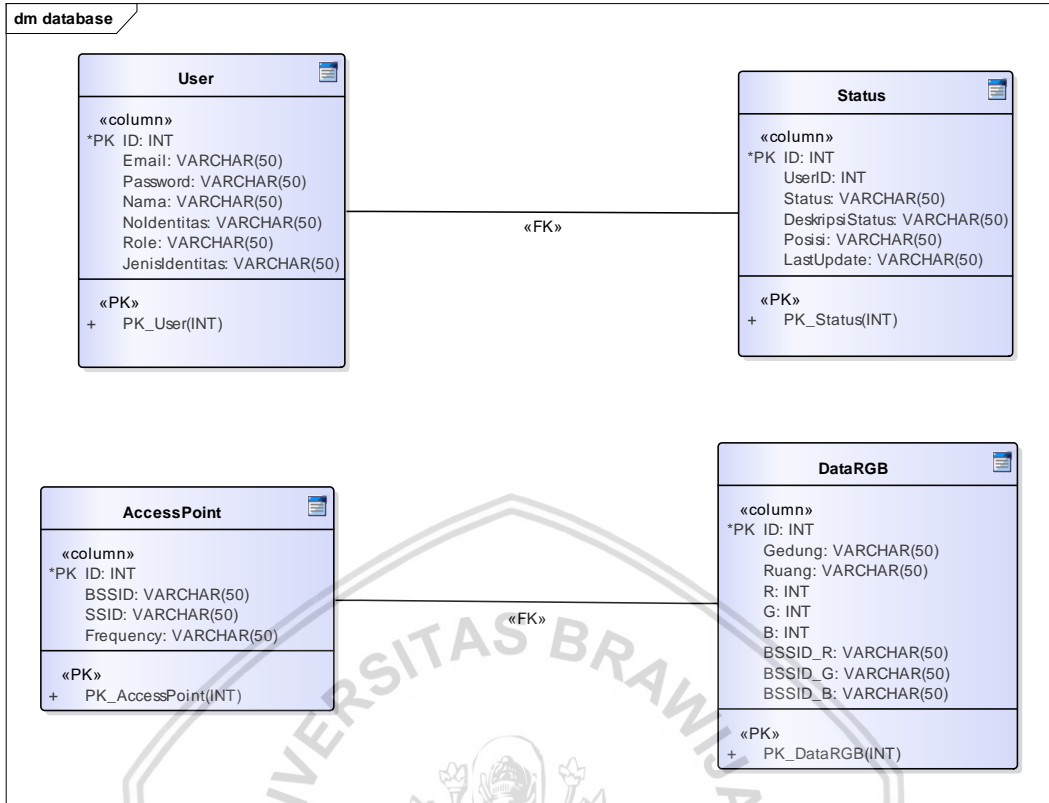
Gambar 6.1 Class Diagram Aplikasi Finding Dosen



Gambar 6.2 Class Diagram Backend Finding Dosen

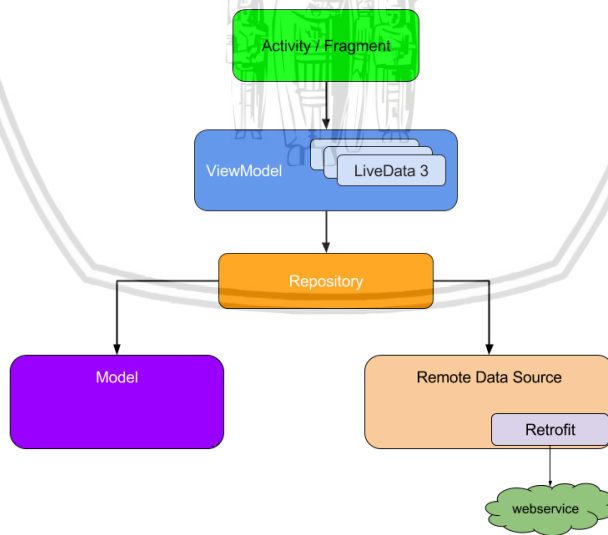
6.1.2 Physical Data Model

Aplikasi Finding Dosen memanfaatkan penyimpanan lokal yang digunakan untuk menyimpan data pengguna yang telah masuk ke dalam aplikasi dan untuk menyimpan data peta gedung. Gambar *physical data model* ditunjukkan oleh Diagram 6.2 berikut.



Gambar 6.3 Physical Data Model Finding Dosen

6.1.3 Perancangan Arsitektur



Gambar 6.4 Arsitektur Komponen Aplikasi Android

Sumber: Android Developer (2017)

Pada penelitian ini menggunakan pola MVVM (*Model-View-ViewModel*). Pola MVVM mendukung data dua arah yang mengikat antara *View* and *View-*

Model. Ini memungkinkan perbanyak perubahan secara otomatis, di dalam keadaan *View-Model* ke *View*. Umumnya, *View-Model* menggunakan pola pengamat untuk menginformasikan perubahan dalam Model-View pada Model. View adalah komponen yang berinteraksi langsung dengan pengguna seperti XML, *Activity*, *fragment*. *View-Model* bertanggung jawab untuk mengekspos metode, perintah, dan sifat lainnya yang membantu mempertahankan keadaan tampilan, memanipulasi model sebagai hasil tindakan pada tampilan, dan aktivitas dalam tampilan itu sendiri. *Model* mewakili seperangkat kelas yang menggambarkan logika bisnis dan data. Ini juga mendefinisikan aturan bisnis untuk data yang berarti bagaimana data bisa diubah dan dimanipulasi. *Model* pada arsitektur ini terbagi menjadi dua bagian yaitu model yang ada pada lokal aplikasi android dan model untuk *remote* yaitu model yang terhubung dengan *web service*.

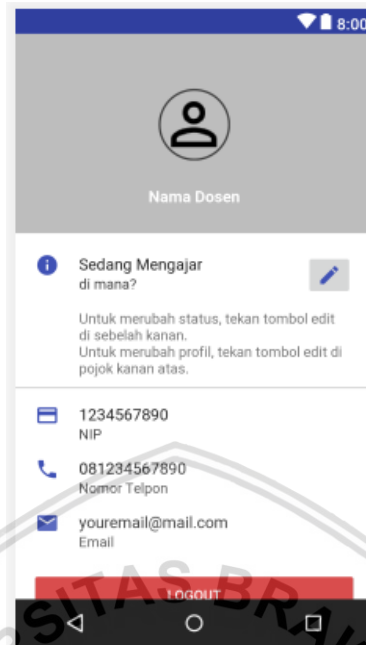
6.1.4 Perancangan Antar Muka

6.1.4.1 Rancangan Antar Muka Halaman Login



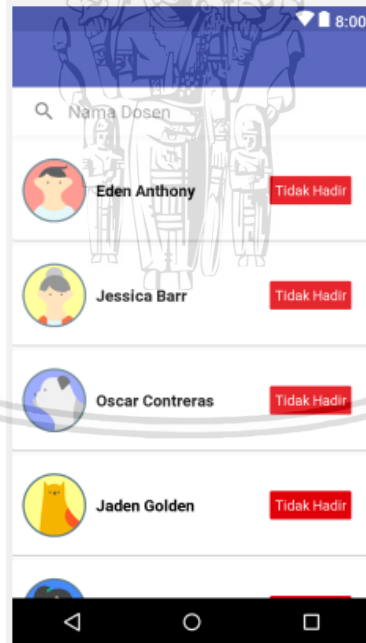
Gambar 6.5 Rancangan Antar Muka Halaman Login

6.1.4.2 Rancangan Antar Muka Halaman Utama Dosen



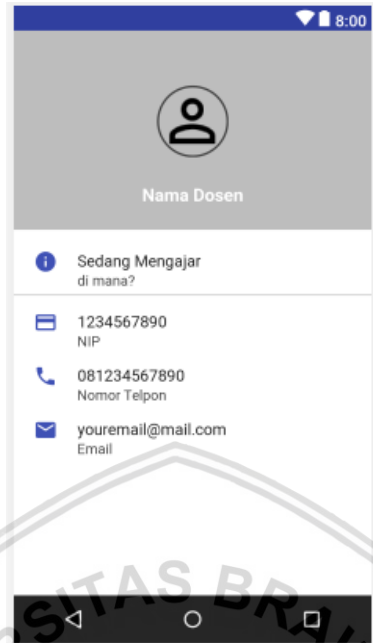
Gambar 6.6 Rancangan Antar Muka Halaman Utama Dosen

6.1.4.3 Rancangan Antar Muka Daftar Dosen



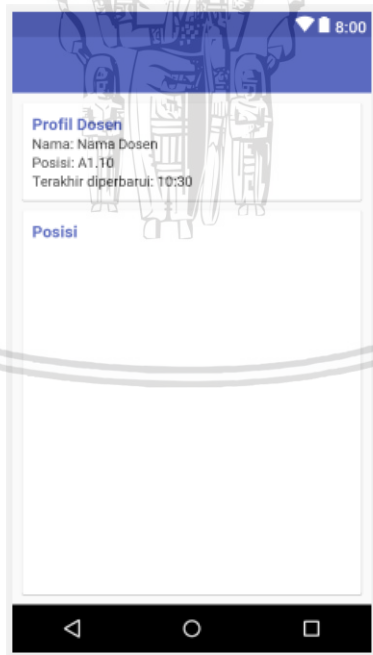
Gambar 6.7 Rancangan Antar Muka Daftar Dosen

6.1.4.4 Rancangan Antar Muka Profil Dosen



Gambar 6.8 Rancangan Antar Muka Profil Dosen

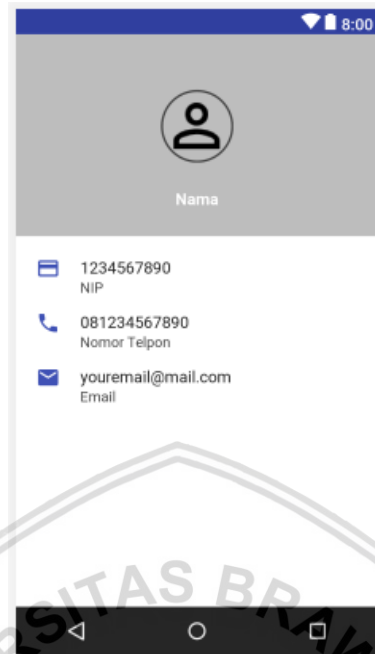
6.1.4.5 Rancangan Antar Muka Pencarian Posisi Dosen



Gambar 6.9 Rancangan Antar Muka Pencarian Posisi Dosen

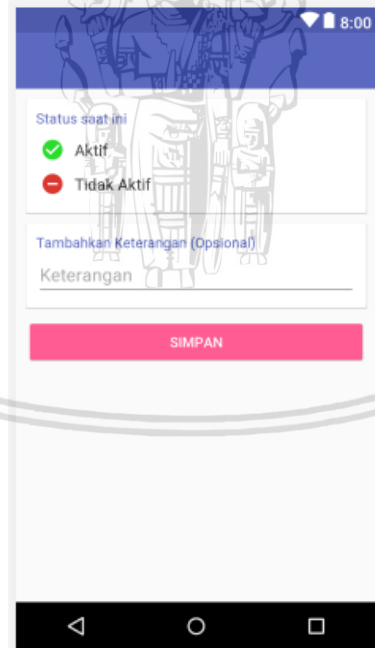


6.1.4.6 Rancangan Antar Muka Halaman Profil



Gambar 6.10 Rancangan Antar Muka Halaman Profil

6.1.4.7 Rancangan Antar Muka Mengubah Status



Gambar 6.11 Rancangan Antar Muka Mengubah Status



6.2 Implementasi

6.2.1 Implementasi Kode Program Back End

Pada implementasi kode program *back end*, diambil 3 fungsi utama dari aplikasi *Finding Dosen* yaitu, melihat posisi dosen, melihat daftar dosen, dan merubah status dosen.

```

View_profile.go

func FetchDosenByID(w http.ResponseWriter, r *http.Request) {
    userID := pat.Param(r, "id")

    dosen, err := fetchDosen(userID)

    if err != nil {
        errors.NewError("Can't fetch profile",
            http.StatusInternalServerError).WriteTo(w)
        return
    }

    json.NewEncoder(w).Encode(dosen)
}

func fetchDosen(id string) (DosenResponse, error) {
    var dosen DosenResponse

    query := `
        SELECT users.id as UserID, users.nama, users.email,
        users.jenis_identitas as JenisIdentitas,
        users.no_identitas as NoIdentitas, users.no_telpon as
        NoTelpon,
        status.status, status.desc_status as DeskripsiStatus,
        status.ket_status as KetStatus
        FROM users JOIN status
        WHERE users.id = ? AND users.id = status.user_id;
    `

    if err := models.Dbm.SelectOne(&dosen, query, id); err != nil
    {
        return DosenResponse{}, err
    }

    return dosen, nil
}

```

Gambar 6.12 Kode Program Melihat Posisi Dosen Back End

Kode program di atas digunakan untuk melihat posisi dosen berdasarkan *id user*. Data dosen diambil dari *database* kemudian dibeikan respon berupa objek Dosen dalam format *json*.

```

View_profile.go
func FetchAllDosenProfile(w http.ResponseWriter, r *http.Request) {
    var users []DosenResponseAll

    role := "dosen"

    query := `SELECT users.id as UserID, users.nama,
status.status
FROM users JOIN status
WHERE role=? AND users.id = status.user_id;`

    if _, err := models.Dbm.Select(&users, query, role); err !=
nil {
        errors.NewError("can't fetch profile",
http.StatusInternalServerError).WriteTo(w)
        return
    }

    json.NewEncoder(w).Encode(map[string][]DosenResponseAll{
        "data": users,
    })
}

```

Gambar 6.13 Kode Program Melihat Daftar Dosen Back End

Kode program di atas digunakan untuk melihat seluruh daftar dosen yang ada pada *database* yang kemudian dikembalikan respon dalam bentuk *array json*.

```

update_location.go
func UpdateDosenLocation(w http.ResponseWriter, r *http.Request) {

    lastUpdate := time.Now()
    formattedLastUpdate := lastUpdate.Format("2006-01-02 15:04")

    userID := r.Context().Value("user_id").(int)
    var status models.Status
    queryUser := "SELECT * FROM status WHERE user_id = ?"
    if err := models.Dbm.SelectOne(&status, queryUser, userID);
err != nil {

        errors.NewErrorWithStatusCode(http.StatusInternalServerError)
        .WriteTo(w)
        return
    }
    parsedDate, err := time.Parse("2006-01-02 15:04",
formattedLastUpdate)
    if err != nil {
        return
    }

    var req req

```

```
decoder := json.NewDecoder(r.Body)
if err := decoder.Decode(&req); err != nil {

errors.NewErrorWithStatusCode(http.StatusBadRequest).WriteTo(
w)
    return
}

apRequests := req.Data

var apDb []models.AccessPoint

query := "SELECT * FROM access_point"

if _, err := models.Dbm.Select(&apDb, query); err != nil {
    errors.NewError("can't fetch access points",
http.StatusInternalServerError).WriteTo(w)
    return
}

// filter AP yang sama dengan AP yang ada di DB
listAp := make(map[string]int)

for _, apReq := range apRequests {
    for _, apData := range apDb {
        if apReq.BSSID == apData.BSSID {
            if value, ok := listAp[apReq.BSSID]; ok {
                value = value + apReq.Level
                listAp[apReq.BSSID] = value
            } else {
                listAp[apReq.BSSID] = apReq.Level
            }
        }
    }
}

if len(listAp) == 0 {
    // TODO UPDATE POSISI DOSEN TO ""
    status.Posisi = "Tidak ada"
    status.LastUpdate = parsedDate
    if _, err := models.Dbm.Update(&status); err != nil {

errors.NewErrorWithStatusCode(http.StatusInternalServerError)
.WriteTo(w)
        return
    }

    errors.NewError("Tidak ada Access Point yang sesuai!",
http.StatusBadRequest).WriteTo(w)
    return
}
```

```
// Deklarasi variabel untuk perhitungan data RGB
jarakRSSI := 80 // batas jarak dari 0 - maksimum nilai level
dbm yang diterima
minRSSI := 0

// nilai normalisasi
var normalisasi1 int
var normalisasi2 int
var normalisasi3 int

// level RGB
var level1 int
var level2 int
var level3 int

// lokasi gedung
var lokasiGedung string
var posisi string

// Hitung rata2 tiap AP
average := make(map[string]int)

fmt.Println(len(listAp))

for ap, dataLevel := range listAp {
    // Gedung A Lantai 1
    // A1 AP1 a0:3d:6f:85:c0:c1
    if ap == "a0:3d:6f:85:c0:c1" {
        average["A1AP1"] = dataLevel / 10
    }

    // A1 AP2 a0:3d:6f:85:c3:01
    if ap == "a0:3d:6f:85:c3:01" {
        average["A1AP2"] = dataLevel / 10
    }

    // A1 AP3 a0:e0:af:57:26:81
    if ap == "a0:e0:af:57:26:81" {
        average["A1AP3"] = dataLevel / 10
    }

    // Gedung A Lantai 2
    // A2 AP1 a0:3d:6f:8b:11:61
    if ap == "a0:3d:6f:8b:11:61" {
        average["A2AP1"] = dataLevel / 10
    }

    // A2 AP2 a0:3d:6f:8f:96:01
    if ap == "a0:3d:6f:8f:96:01" {
        average["A2AP2"] = dataLevel / 10
    }

    // A2 AP3 a0:3d:6f:8b:0f:21
```

```
    if ap == "a0:3d:6f:8b:0f:21" {
        average["A2AP3"] = dataLevel / 10
    }

    // Gedung E Lantai 1
    // E1 AP1 a0:3d:6f:8b:0b:c1
    if ap == "a0:3d:6f:8b:0b:c1" {
        average["E1AP1"] = dataLevel / 10
    }

    // E1 AP2 a0:3d:6f:89:22:c1
    if ap == "a0:3d:6f:89:22:c1" {
        average["E1AP2"] = dataLevel / 10
    }

    // E1 AP3 a0:3d:6f:89:22:e1
    if ap == "a0:3d:6f:89:22:e1" {
        average["E1AP3"] = dataLevel / 10
    }

    // Gedung E Lantai 2
    // E2 AP1 a0:3d:6f:8b:0d:81
    if ap == "a0:3d:6f:8b:0d:81" {
        average["E2AP1"] = dataLevel / 10
    }

    // E2 AP2 a0:e0:af:9a:12:01
    if ap == "a0:e0:af:9a:12:01" {
        average["E2AP2"] = dataLevel / 10
    }

    // E2 AP3 a0:e0:af:99:e4:e1
    if ap == "a0:e0:af:99:e4:e1" {
        average["E2AP3"] = dataLevel / 10
    }
}

fmt.Printf("average A1AP1: %d\n", average["A1AP1"])
fmt.Printf("average A1AP2: %d\n", average["A1AP2"])
fmt.Printf("average A1AP3: %d\n", average["A1AP3"])

// Menentukan posisi gedung
gedung := make(map[string]int)

gedung["A1"] = average["A1AP1"] + average["A1AP2"] +
average["A1AP3"]
gedung["A2"] = average["A2AP1"] + average["A2AP2"] +
average["A2AP3"]
gedung["E1"] = average["E1AP1"] + average["E1AP2"] +
average["E1AP3"]
gedung["E2"] = average["E2AP1"] + average["E2AP2"] +
average["E2AP3"]
```

```

// AP Gedung A Lantai 1
if gedung["A1"] < gedung["A2"] {
    normalisasi1 = jarakRSSI + (minRSSI + average["A1AP1"])
    normalisasi2 = jarakRSSI + (minRSSI + average["A1AP2"])
    normalisasi3 = jarakRSSI + (minRSSI + average["A1AP3"])
    lokasiGedung = "A1"
} else if gedung["A2"] < gedung["A1"] {
    normalisasi1 = jarakRSSI + (minRSSI + average["A2AP1"])
    normalisasi2 = jarakRSSI + (minRSSI + average["A2AP2"])
    normalisasi3 = jarakRSSI + (minRSSI + average["A2AP3"])
    lokasiGedung = "A2"
}

// Pelevelan persamaan (255/jarakRSSI)*normalisasi, nilai 255
didapatkan dari max nilai RGB
level1 = (255 / jarakRSSI) * normalisasi1 // R
level2 = (255 / jarakRSSI) * normalisasi2 // G
level3 = (255 / jarakRSSI) * normalisasi3 // B

// bandingkan data level RGB yang telah dihitung dengan data
RGB Training di DB
var dataRgb []models.DataRgb

queryRgb := "SELECT * FROM data_rgb WHERE gedung=?"

if _, err := models.Dbm.Select(&dataRgb, queryRgb,
lokasiGedung); err != nil {
    errors.NewError("can't fetch rgb data",
http.StatusInternalServerError).WriteTo(w)
    return
}

fmt.Printf("R: %d\n", level1)
fmt.Printf("G: %d\n", level2)
fmt.Printf("B: %d\n", level3)

// hitung mean squared error
var mseData []float64

// difference = rgb training - rgb testing
for _, rgb := range dataRgb {
    // Mean Squared Error
    difLevel1 := rgb.LevelR - level1
    difLevel2 := rgb.LevelG - level2
    difLevel3 := rgb.LevelB - level3
    mse := math.Sqrt(math.Pow(float64(difLevel1), 2) +
math.Pow(float64(difLevel2), 2) + math.Pow(float64(difLevel3), 2))
    mseData = append(mseData, mse)
}

if len(mseData) == 0 {
    errors.NewError("Data MSE kosong",
http.StatusInternalServerError).WriteTo(w)
}

```

```

        return
    }

    // hitung nilai minimum
    var minimumValue = mseData[0]
    var idxMinimum = 0

    for i, value := range mseData {
        if value < minimumValue {
            minimumValue = value
            idxMinimum = i
        }
    }

    // posisi dosen dilihat dari data RGB
    posisi = dataRgb[idxMinimum].Ruang
    //TODO UPDATE POSISI DOSEN
    status.Posisi = posisi
    status.LastUpdate = parsedDate
    if _, err := models.Dbm.Update(&status); err != nil {
        errors.NewErrorWithStatusCode(http.StatusInternalServerError)
        .WriteTo(w)
        return
    }
    var response updateResponse
    response.Posisi = posisi
    response.Mse = minimumValue

    json.NewEncoder(w).Encode(response)
}

```

Gambar 6.14 Kode Program Mengubah Status Back End

Kode program di atas digunakan untuk mengubah status dosen dan menentukan posisi dosen. Data *Access Point* yang didapatkan melalui *scanning* pada *smartphone* yang digunakan dosen dibandingkan dengan data yang ada pada database, kemudian dilakukan perhitungan dengan mengkonversi nilai sinyal wifi menjadi data RGB. Data RGB yang didapatkan dibandingkan dengan data RGB pada grid yang ada, kemudian dicari nilai terkecil dari *Mean Squared Error* untuk mendapatkan posisi dosen yang sebenarnya.

6.2.2 Implementasi Kode Program Front End Android

Pada implementasi kode program *front end android*, diambil 3 fungsi utama dari aplikasi Finding Dosen yaitu, melihat profil dosen, melihat daftar dosen, dan merubah status dosen.

MainActivity.java

```

public class MainActivity extends AppCompatActivity implements
    SearchView.OnQueryTextListener, SearchView.OnCloseListener {

```



```
private static final String TAG = "MainActivity";

@BindView(R.id.mainActivity_toolbar)
Toolbar toolbar;
@BindView(R.id.mainActivity_searchView)
SearchView searchView;
@BindView(R.id.mainActivity_recyclerView)
RecyclerView recyclerView;
ActionBar actionBar;

private MainViewModel mainViewModel;

private List<DosenAdapter> dosenList;
private FastItemAdapter<DosenAdapter> fastDosenAdapter;

private List<DosenAdapter> dosenFilterList;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    ButterKnife.bind(this);

    if (toolbar != null) setSupportActionBar(toolbar);
    if (getSupportActionBar() != null) {
        actionBar = getSupportActionBar();
        actionBar.setTitle("Daftar Dosen");
    }

    mainViewModel = ViewModelProviders.of(this,
        FindingDosenApplication.getDataComponent()
            .getMainViewModelFactory()).get(MainViewModel.class);

    // get Dosen List from View Model
    mainViewModel.getDosenList().observe(this,
        this::lihatDaftarDosen);
    }

private void lihatDaftarDosen(List<DosenAdapter> listDosen) {
    dosenList = listDosen;
    fastDosenAdapter.set(dosenList);
    fastDosenAdapter.notifyDataSetChanged();
}
}
```

Gambar 6.15 Kode Program Melihat Daftar Dosen Android

Kode program di atas merupakan kelas MainActivity, dimana terdapat fungsi untuk melihat daftar dosen, fungsi tersebut dijalankan ketika *activity* melakukan *observe* ke *view model*, kemudian kembalian dari *view model* ditampilkan pada *activity*.

PosisiDosenActivity.java

```

public class PosisiDosenActivity extends AppCompatActivity {

    @BindView(R.id.posisiDosen_namaDosen)
    TextView tvNamaDosen;
    @BindView(R.id.posisiDosen_lokasiDosen)
    TextView tvLokasiDosen;
    @BindView(R.id.posisiDosen_lastUpdate)
    TextView tvTerakhirDiperbarui;
    @BindView(R.id.posisiDosen_imageViewPosisi)
    ImageView imageViewPosisi;
    @BindView(R.id.posisiDosen_toolbar)
    Toolbar toolbar;
    ActionBar actionBar;

    private PosisiDosenViewModel viewModel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_posisi_dosen);

        ButterKnife.bind(this);

        if (toolbar != null) setSupportActionBar(toolbar);
        if (getSupportActionBar() != null) {
            actionBar = getSupportActionBar();
            actionBar.setTitle("Posisi Dosen");
            actionBar.setDisplayHomeAsUpEnabled(true);
        }

        Intent intent = getIntent();
        int userId = intent.getIntExtra("userId", 0);

        viewModel = ViewModelProviders.of(this,
FindingDosenApplication.getDataComponent().getPosisiDosenViewModelFactory()).get(PosisiDosenViewModel.class);
        viewModel.getPosisiDosen(userId).observe(this,
this::lihatPosisiDosen);
    }

    private void lihatPosisiDosen(PosisiDosen posisiDosen) {
        tvNamaDosen.setText(posisiDosen.getNama());
        tvLokasiDosen.setText(posisiDosen.getPosisi());
        tvTerakhirDiperbarui.setText(posisiDosen.getLastUpdate());
imageViewPosisi.setImageDrawable(ContextCompat.getDrawable(this,
R.drawable.ic_close_grey_24dp));
    }
}

```

Gambar 6.16 Kode Program Melihat Posisi Dosen Android

Kode program di atas merupakan kelas `PosisiDosenActivity`, dimana terdapat fungsi untuk melihat profil dosen, fungsi tersebut dijalankan ketika dosen dipilih dari halaman daftar dosen, kemudian id dari dosen dikirimkan untuk mengambil informasi profil dosen dengan melakukan memanggil *method* `lihatPosisiDosen` yang ada di *view model*, kemudian *Activity* melakukan *observe* ke *view model*, kembalian dari *view model* ditampilkan pada *activity*.

>EditStatusActivity.java

```
public class EditStatusActivity extends AppCompatActivity {
    private List<ScanResult> scanResults = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_edit_status);

        ButterKnife.bind(this);

        if (toolbar != null) setSupportActionBar(toolbar);
        if (getSupportActionBar() != null) {
            actionBar = getSupportActionBar();
            actionBar.setTitle("Edit Status");
            actionBar.setDisplayHomeAsUpEnabled(true);
        }

        editStatusViewModel = ViewModelProviders.of(this,
            FindingDosenApplication.getDataComponent().getEditStatusViewModelFactory()).get(EditStatusViewModel.class);

        Dexter.withActivity(this)
            .withPermissions(
                Manifest.permission.ACCESS_COARSE_LOCATION,
                Manifest.permission.ACCESS_FINE_LOCATION)
            .withListener(new MultiplePermissionsListener() {
                @Override
                public void
                onPermissionsChecked(MultiplePermissionsReport report) {
                    if (report.areAllPermissionsGranted()) {
                        for (int i = 0; i < 10; i++) {
                            scanCount = i + 1;
                            accessWifiManager();
                            Log.d(TAG, "accessWifiManager: " +
                                scanCount);
                        }
                    }
                }
            })

        @Override
        public void
        onPermissionRationaleShouldBeShown(List<PermissionRequest>
            permissions, PermissionToken token) {
```

```
        }
    }).check();

    private void ubahStatusDosen() {
        if (!scanResults.isEmpty()) {
            List<AccessPointRequest> listApRequest = new
ArrayList<>();
            for (ScanResult result : scanResults) {
                AccessPointRequest apRequest = new
AccessPointRequest();
                apRequest.setBSSID(result.BSSID);
                apRequest.setSSID(result.SSID);
                apRequest.setLevel(result.level);
                listApRequest.add(apRequest);
            }
        }

        private void accessWifiManager() {
            final WifiManager mWifiManager = (WifiManager)
getApplicationContext().getSystemService(WIFI_SERVICE);

            if (mWifiManager.getWifiState() ==
WifiManager.WIFI_STATE_ENABLED) {

                // register WiFi scan results receiver
                IntentFilter filter = new IntentFilter();

                filter.addAction(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION);

                registerReceiver(new BroadcastReceiver() {
                    @Override
                    public void onReceive(Context context, Intent intent)
                {

                    List<ScanResult> resultList =
mWifiManager.getScanResults();
                    scanResults.addAll(resultList);
                    int N = scanResults.size();

                    Log.v(TAG, "Wi-Fi Scan Results ... Count:" + N);
                }
            }, filter);

            // start WiFi Scan
            mWifiManager.startScan();
        }
    }
}
```

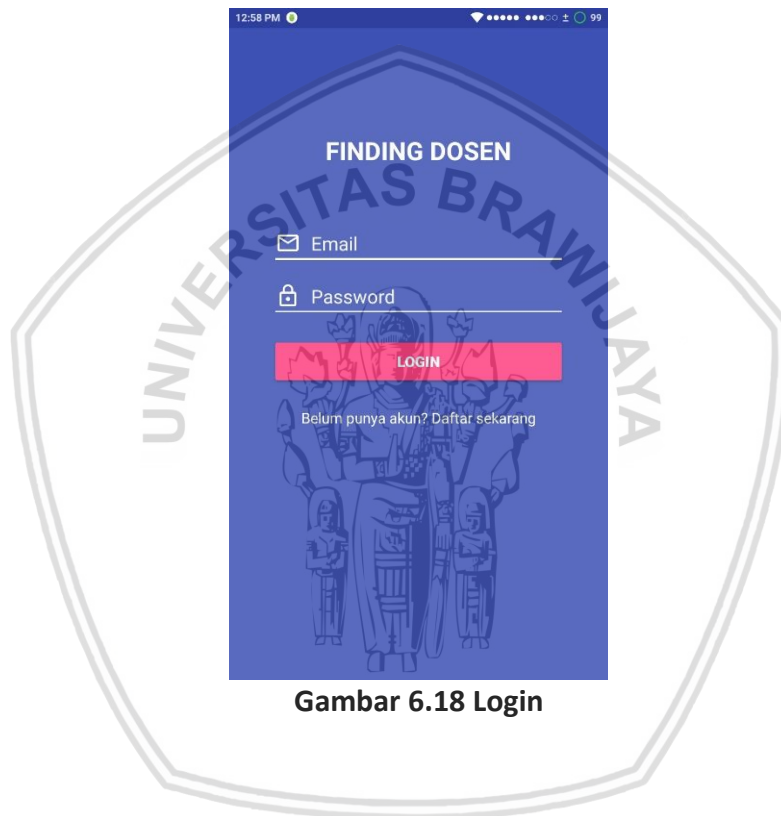
Gambar 6.17 Kode Program Merubah Status Android

Kode program di atas merupakan kelas `EditStatusActivity`, dimana terdapat fungsi untuk merubah status dosen, fungsi tersebut dijalankan ketika dosen memilih status, jika memilih status "Aktif" maka akan melakukan *scanning wifi* sebanyak 10 kali, kemudian hasil *scan* tersebut dikirim ke *server* dengan memanggil ubah status pada *view model*, kemudian *Activity* melakukan *observe* ke *view model*, kembalian dari *view model* ditampilkan pada *activity*.

6.2.3 Implementasi Antar Muka Aplikasi

6.2.3.1 Login

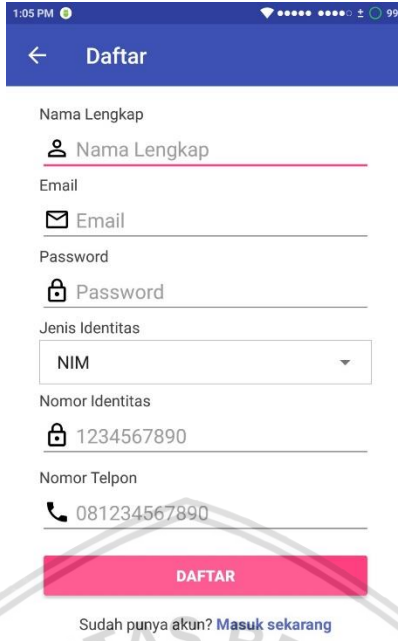
Berikut adalah tampilan Login dari Aplikasi *Finding Dosen*



Gambar 6.18 Login

6.2.3.2 Register

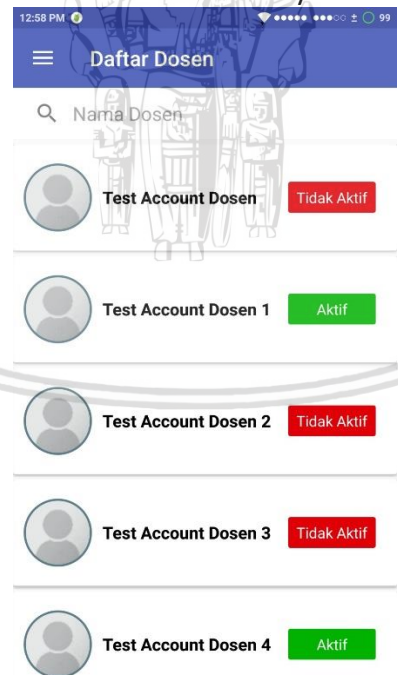
Berikut adalah tampilan Register dari Aplikasi *Finding Dosen*



Gambar 6.19 Register

6.2.3.3 Daftar Dosen

Berikut adalah tampilan utama dari mahasiswa yaitu melihat daftar dosen

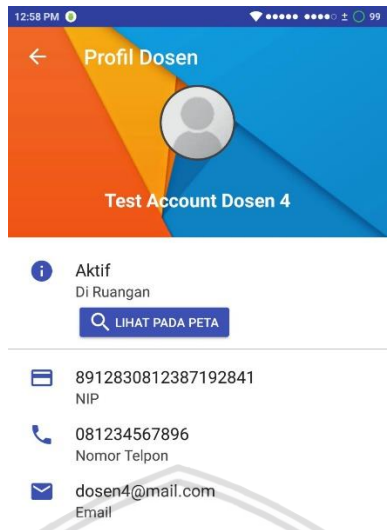


Gambar 6.20 Daftar Dosen

6.2.3.4 Profil Dosen

Berikut adalah tampilan profil dosen dari dosen yang dipilih

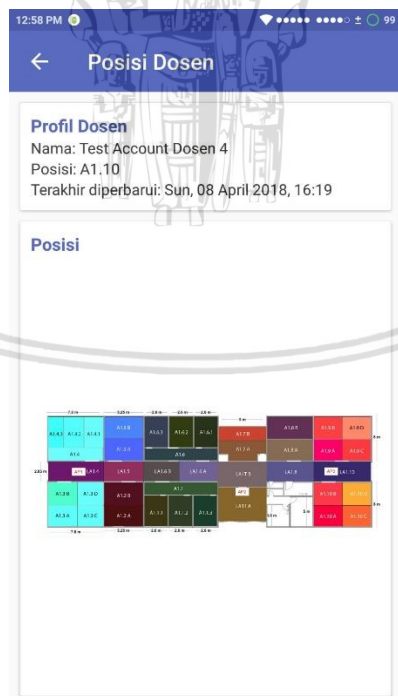




Gambar 6.21 Profil Dosen

6.2.3.5 Posisi Dosen

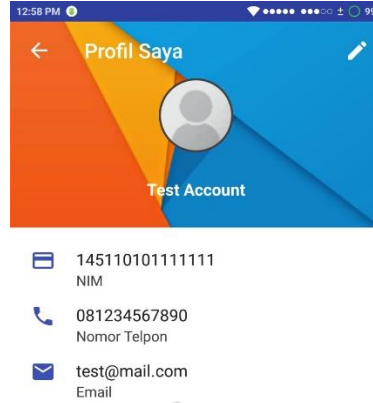
Berikut adalah tampilan posisi dosen dari dosen yang dipilih



Gambar 6.22 Posisi Dosen

6.2.3.6 Melihat Profil

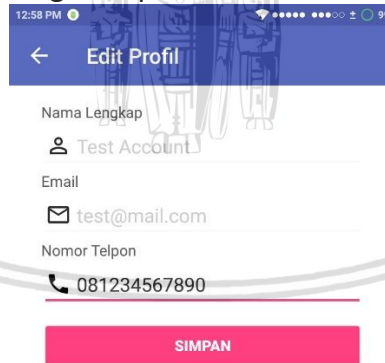
Berikut adalah tampilan melihat profil diri sendiri



Gambar 6.23 Melihat Profil

6.2.3.7 Mengubah Profil

Berikut adalah tampilan mengubah profil



Gambar 6.24 Mengubah Profil



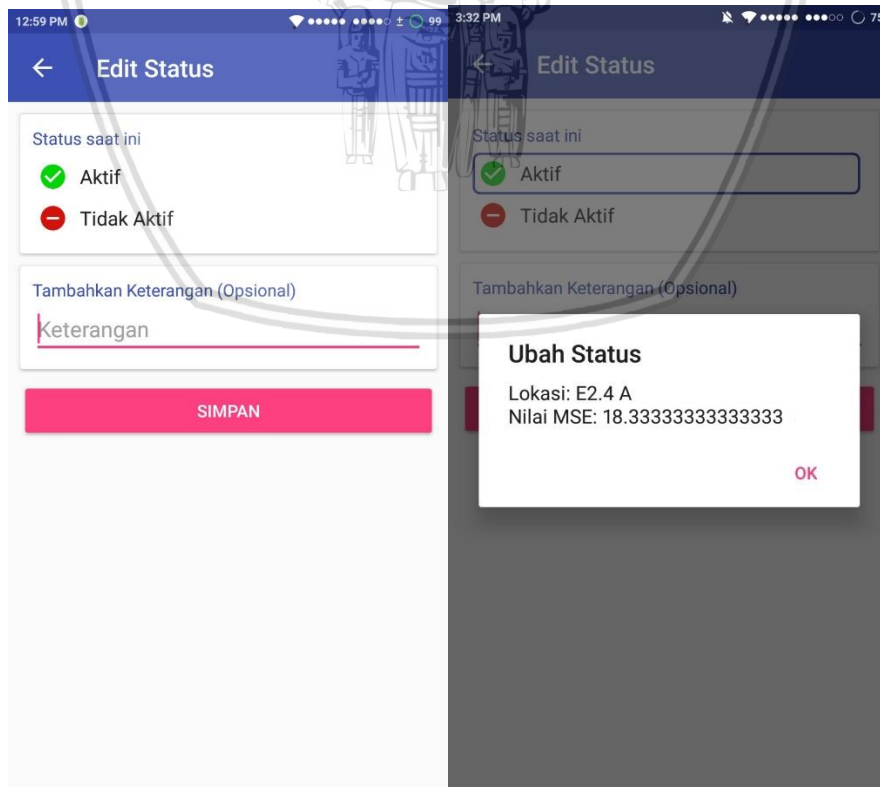
6.2.3.8 Tampilan Utama Dosen

Berikut adalah tampilan halaman utama dosen



Gambar 6.25 Halaman Utama Dosen

6.2.3.9 Mengubah Status



Gambar 6.26 Mengubah Status

BAB 7 PENGUJIAN DAN ANALISIS HASIL

7.1 Pengujian Unit

Pengujian unit dilakukan untuk menguji *method* yang ada pada Aplikasi Finding dosen menggunakan metode *white box*. Pengujian ini dilakukan dengan mengidentifikasi *Pseudocode* dan digambarkan berupa Basis Path serta identifikasi jalur independen yang mungkin dilewati pada setiap pengaksesan *method* yang terdiri dari **Lihat Daftar Dosen, Lihat Posisi Dosen dan Merubah Status Dosen.**

7.1.1 Method Lihat Daftar Dosen

1. Pseudocode

```

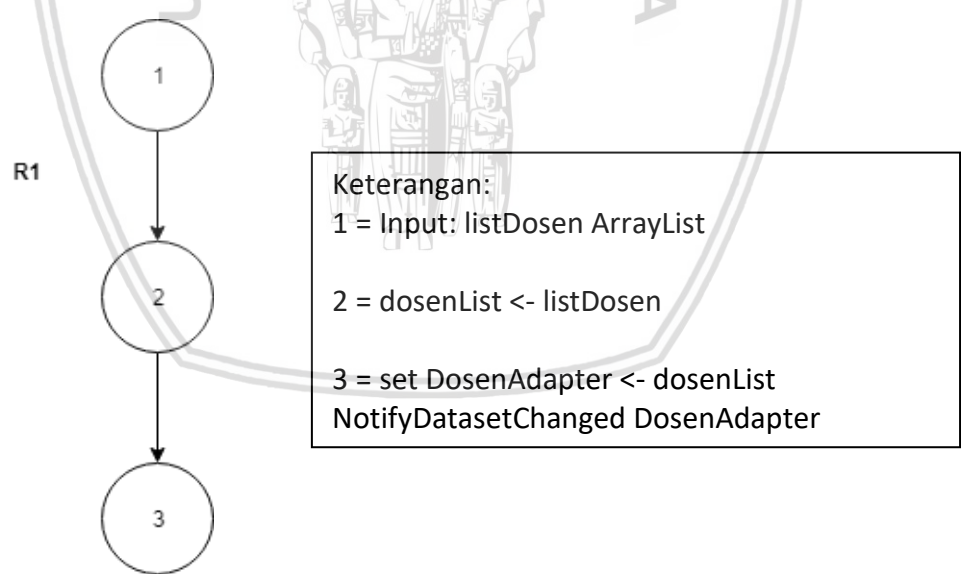
Input: listDosen ArrayList

dosenList <- listDosen

set DosenAdapter <- dosenList
NotifyDatasetChanged DosenAdapter
    
```

2. Basis Path Testing

a. Flow Graph



Gambar 7.1 Flow Graph 7.1.1.1 Method Lihat Daftar Dosen

b. Cyclomatic Complexity

Node = 3, Edge = 2, Predicate Node = 0

$V(G) = 1$ Region (R1)

$V(G) = \text{Edge} - \text{Node} + 2$

$$= 2 - 3 + 2$$

$$= 1$$

$$V(G) = \text{Predicate Node} + 1$$

$$= 0 + 1$$

$$= 1$$

c. Independent Path

$$\text{Jalur 1} = 1 - 2 - 3$$

7.1.2 Method Lihat Posisi Dosen

1. Pseudocode

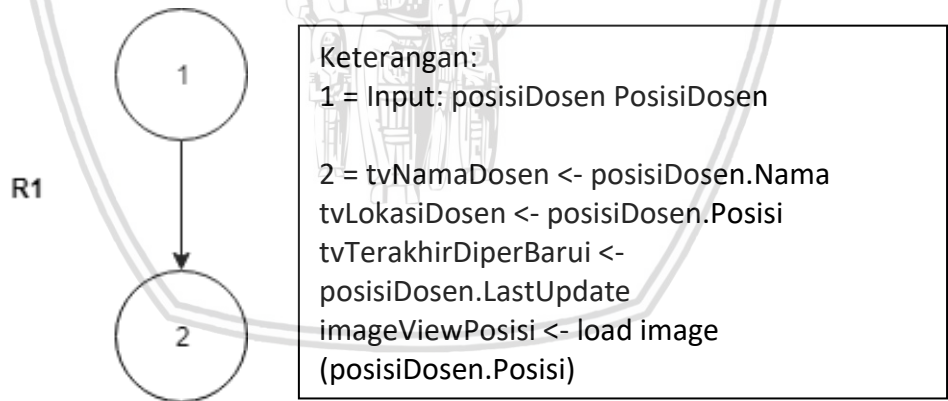
```

Input: posisiDosen PosisiDosen

tvNamaDosen <- posisiDosen>Nama
tvLokasiDosen <- posisiDosen.Posisi
tvTerakhirDiperBarui <- posisiDosen.LastUpdate
imageViewPosisi <- load image (posisiDosen.Posisi)
    
```

2. Basis Path Testing

a. Flow Graph



Gambar 7.2 Flow Graph Lihat Posisi Dosen

b. Cyclomatic Complexity

$$\text{Node} = 2, \text{Edge} = 1, \text{Predicate Node} = 0$$

$$V(G) = 1 \text{ Region (R1)}$$

$$V(G) = \text{Edge} - \text{Node} + 2$$

$$= 1 - 2 + 2$$

$$= 1$$

$$V(G) = \text{Predicate Node} + 1$$

$$= 0 + 1$$

$$= 1$$

- c. Independent Path
- Jalur 1 = 1 – 2

7.1.3 Method Ubah Status Dosen

1. Pseudocode

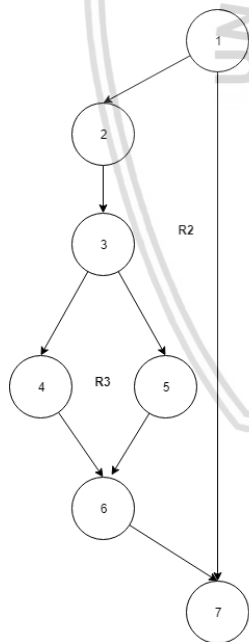
```

IF scanResults not empty
  Observe Response from ubahLokasi ViewModel
  IF Response NOT NULL
    Dialog message <- "Lokasi"
  ELSE
    Dialog message <- "Lokasi tidak ditemukan"
  END IF
END IF

```

2. Basis Path Testing

a. Flow Graph



Keterangan:

- 1 = IF scanResults not empty
- 2 = Observe Response from ubahLokasi ViewModel
- 3 = IF Response NOT NULL
- 4 = Dialog message <- "Lokasi"
- 5 = ELSE
- Dialog message <- "Lokasi tidak ditemukan"
- 6 = END IF
- 7 = END IF

Gambar 7.3 Flow Graph Mengubah Status Dosen

b. Cyclomatic Complexity

Node = 7, Edge = 8, Predicate Node = 2

V(G) = 3 Region (R1, R2, R3)



$$\begin{aligned}
 V(G) &= \text{Edge} - \text{Node} + 2 \\
 &= 8 - 7 + 2 \\
 &= 3
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= \text{Predicate Node} + 1 \\
 &= 2 + 1 \\
 &= 3
 \end{aligned}$$

c. Independent Path

$$\text{Jalur 1} = 1 - 7$$

$$\text{Jalur 2} = 1 - 2 - 3 - 4 - 6 - 7$$

$$\text{Jalur 3} = 1 - 2 - 3 - 5 - 6 - 7$$

7.2 Hasil Pengujian Unit

Berdasarkan kasus uji yang telah ditentukan, maka diperoleh hasil pengujian unit yang ditampilkan di dalam Tabel

7.2.1 Method Lihat Daftar Dosen

Tabel 7.1 Hasil Pengujian Unit Method Lihat Daftar Dosen

No.	Kasus Uji	Data Input	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	1 - 2 - 3	Posisi Dosen	Menampilkan posisi Dosen	Menampilkan posisi Dosen	Valid

7.2.2 Method Lihat Posisi Dosen

Tabel 7.2 Hasil Pengujian Unit Method Lihat Posisi Dosen

No.	Kasus Uji	Data Input	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	1 - 2	List Dosen	Menampilkan daftar Dosen	Menampilkan posisi Dosen	Valid

7.2.3 Method Ubah Status Dosen

Tabel 7.3 Hasil Pengujian Unit Method Ubah Status Dosen

No.	Kasus Uji	Data Input	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	1 - 7	Scan Result Kosong	Tidak menampilkan apa - apa	Tidak menampilkan apa - apa	Valid



2	1 – 2 – 3 – 4 – 6 – 7	Scan Result	Menampilkan dialog Lokasi Dosen	Menampilkan dialog Lokasi Dosen	Valid
3	1 – 2 – 3 – 5 – 6 – 7	Scan Result	Menampilkan dialog Lokasi Tidak Ditemukan	Menampilkan dialog Lokasi Tidak Ditemukan	Valid

7.3 Analisis Hasil Pengujian Unit

Berdasarkan hasil pengujian unit dilakukan dengan melihat Tabel 7.1, Tabel 7.2, Tabel 7.3 bahwa semua kasus uji mengeluarkan menghasilkan keluaran yang valid atau sesuai dengan hasil yang diharapkan. Dengan demikian dapat disimpulkan bahwa semua pengujian unit telah terpenuhi.

7.4 Pengujian Validasi

7.4.1 Pengujian Kebutuhan Fungsional

Tabel 7.4 Pengujian Kebutuhan Fungsional

No	Kode Kebutuhan	Nama Fungsi	Jenis Pengujian	Alat Uji
1	SRS-FIND-01	Login	Black Box (Requirement Testing)	Smartphone
2	SRS-FIND-02	Register	Black Box (Requirement Testing)	Smartphone
3	SRS-FIND-03	Melihat profil	Black Box (Requirement Testing)	Smartphone
4	SRS-FIND-04	Merubah profil	Black Box (Requirement Testing)	Smartphone
5	SRS-FIND-05	Logout	Black Box (Requirement Testing)	Smartphone
6	SRS-FIND-06	Melihat daftar dosen	Black Box (Requirement Testing)	Smartphone
7	SRS-FIND-07	Mencari dosen	Black Box (Requirement Testing)	Smartphone

8	SRS-FIND-08	Melihat profil dosen	Black Box (Requirement Testing)	Smartphone
9	SRS-FIND-09	Melihat posisi dosen	Black Box (Requirement Testing)	Smartphone
10	SRS-FIND-10	Merubah status	Black Box (Requirement Testing)	Smartphone

7.5 Hasil Pengujian Validasi

7.5.1 Hasil Pengujian Kebutuhan Fungsional

Tabel 7.5 Tabel Hasil Pengujian Kebutuhan Fungsional

No	Nama Fungsi	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Status
1	Login	Pengujian dengan memasukkan email dan password	Sistem memverifikasi login dan masuk ke halaman utama	Sistem memverifikasi login dan masuk ke halaman utama	Valid
2	Register	Pengujian dengan memasukkan data diri berupa nama, email, password, jenis identitas, nomor identitas, dan nomor telpon	Sistem mendaftarkan pengguna dan data pengguna disimpan	Sistem mendaftarkan pengguna dan data pengguna disimpan	Valid
3	Melihat profil	Pengujian dengan masuk ke halaman Profil Saya	Aplikasi menampilkan profil diri	Aplikasi menampilkan profil diri	Valid
4	Merubah profil	Pengujian dengan mengisikan	Sistem merubah profil diri dan	Sistem merubah profil diri dan	Valid

		nomor telpon yang baru	perubahan disimpan	perubahan disimpan	
5	Logout	Pengujian dengan menekan tombol login	Sistem mengeluarkan sesi pengguna dan menampilkan halaman login	Sistem mengeluarkan sesi pengguna dan menampilkan halaman login	Valid
6	Melihat daftar dosen	Pengujian dengan masuk ke halaman utama mahasiswa	Aplikasi menampilkan daftar dosen	Aplikasi menampilkan daftar dosen	Valid
7	Mencari dosen	Pengujian dengan memasukkan nama dosen yang dicari	Aplikasi menampilkan daftar dosen berdasarkan nama yang dicari	Aplikasi menampilkan daftar dosen berdasarkan nama yang dicari	Valid
8	Melihat profil dosen	Pengujian dilakukan dengan memilih dosen yang ada di dalam daftar dosen	Aplikasi menampilkan profil dosen	Aplikasi menampilkan profil dosen	Valid
9	Melihat posisi dosen	Pengujian dengan menekan tombol lihat pada peta yang ada di profil dosen	Aplikasi menampilkan posisi dosen dan menampilkan peta gedung	Aplikasi menampilkan posisi dosen dan menampilkan peta gedung	Valid
10	Merubah status	Pengujian dengan merubah status dosen saat ini	Aplikasi menyimpan perubahan status yang dilakukan oleh dosen	Aplikasi menyimpan perubahan status yang dilakukan oleh dosen	Valid

7.6 Analisis Hasil Pengujian Validasi

Analisis hasil pengujian validasi dilakukan dengan melihat kesesuaian hasil dari Aplikasi Finding Dosen dengan kebutuhan yang telah didefinisikan sebelumnya. Berdasarkan hasil pengujian pada Tabel 7.5 bahwa semua kasus uji bernilai valid. Dengan demikian dapat ditarik kesimpulan bahwa implementasi Aplikasi *Finding* Dosen telah sesuai dengan kebutuhan yang didefinisikan pada tahap Analisis Kebutuhan.

7.7 Pengujian Akurasi

Pengujian akurasi digunakan untuk mengetahui tingkat akurasi yang dimiliki oleh Aplikasi *Finding* Dosen.

Tabel 7.6 Pengujian Akurasi

No.	Lokasi Sebenarnya	Lokasi Finding Dosen	Status
1	A1.8 A	A1.8 A	Valid
2	A1.3 D	A.1.4.1	Tidak Valid

Pengujian dilakukan dengan cara membandingkan posisi sebenarnya dengan posisi yang ditampilkan di dalam aplikasi. Tabel 7.6 adalah sebuah *template* pengujian yang dilakukan untuk mencatat hasil dari pengujian yang dilakukan.

7.8 Hasil Pengujian Akurasi

Berdasarkan pengujian yang dilakukan melalui perbandingan posisi dosen yang sebenarnya dengan lokasi yang ditampilkan pada aplikasi *Finding* Dosen. Simulasi pengujian akurasi dibuktikan dengan perhitungan di bawah ini.

Tabel 7.7 Data Pengujian Akurasi Sinyal Wifi di A1.8 A

MAC	Scan A1.8 A										Rata-Rata
	1	2	3	4	5	6	7	8	9	10	
00:6b:f1:21:fc:81	-83	0	0	0	0	0	0	0	0	0	-8.3
2a:a4:3c:0d:38:f8	-91	-90	0	0	0	-90	-90	-90	-90	0	-54.1
a0:3d:6f:85:c0:c0	-57	0	0	0	-79	-79	-75	-75	-75	-75	-51.5
a0:3d:6f:85:c0:c1	-77	0	0	0	0	-76	-75	-74	-75	0	-37.7
a0:3d:6f:85:c0:ce	-72	0	0	0	0	0	0	0	0	0	-7.2
a0:3d:6f:85:c0:cf	-72	0	0	0	0	0	0	0	0	0	-7.2
a0:3d:6f:85:c3:00	-62	-62	-65	-65	-66	-67	-67	0	0	-64	-51.8
a0:3d:6f:85:c3:01	-65	0	-65	-66	-65	-66	-65	0	0	-67	-45.9
a0:3d:6f:85:c3:0e	-77	-76	-75	-75	-75	-75	-75	-75	-75	-75	-75.3
a0:3d:6f:85:c3:0f	-73	-75	-75	-76	-79	-77	-75	-75	-75	-75	-75.5

a0:3d:6f:8b:0f:20	-67	0	0	-66	-66	-66	-58	-58	-60	-60	-50.1
a0:3d:6f:8b:0f:21	-67	-68	-68	-66	-66	-66	-58	-58	-58	-58	-63.3
a0:e0:af:57:26:80	-54	-54	-52	-52	-54	-56	-56	0	0	-55	-43.3
a0:e0:af:57:26:81	-52	-53	-54	-54	-55	-51	-51	-52	-53	-54	-52.9
a0:e0:af:57:26:8e	-59	-57	-57	-58	-59	-56	-55	-57	-58	-58	-57.4
a0:e0:af:57:26:8f	-58	-57	-57	-58	-58	-59	-55	-56	-57	-57	-57.2
a0:e0:af:99:e4:e0	-78	-78	-82	-82	-82	0	-79	-79	-79	-79	-71.8
a0:e0:af:99:e4:ee	-85	-85	-84	-83	-83	-83	-83	-84	-86	-86	-84.2
a0:e0:af:99:e4:ef	-83	-85	-84	-83	-83	-83	-88	-88	-88	-88	-85.3
e8:f2:e2:2d:59:e2	-85	-85	-87	-88	-84	-83	-83	0	0	0	-59.5

Data di atas merupakan hasil *scanning* sinyal *wifi* sebanyak 10 kali di dalam ruangan A1.8 pada grid A, seluruh hasil *scanning* dilakukan rata-rata untuk melanjutkan ke tahap berikutnya.

Tabel 7.8 Pengujian Akurasi Rata-rata dan pembulatan

Access Point \ Ruang Scan	A1.8 A	A1.8 A
AP1	-37,7	-38
AP2	-45,9	-46
AP3	-52,9	-53

Keterangan:

- AP1 a0:3d:6f:85:c0:c1 FILKOM.X 2,4 GHz
- AP2 a0:3d:6f:85:c3:01 FILKOM.X 2,4 GHz
- AP3 a0:e0:af:57:26:81 FILKOM.X 2,4 GHz

Tabel 7.8 adalah hasil rata-rata dari scan yang dilakukan dan dilakukan pembulatan, access point yang digunakan sebagai *reference* sebanyak tiga buah yaitu *access point* yang memiliki SSID FILKOM.X dengan frekuensi 2,4 GHz. *Access point* yang dipilih menggunakan frekuensi 2,4 GHz dikarenakan kompatibilitas beberapa perangkat saat ini yang belum memiliki *wifi adapter* yang menangkap *access point* dengan frekuensi 5 GHz.

Tabel 7.9 Pengujian Akurasi Normalisasi

Access Point \ Ruang Scan	A1.8 A	A1.8 A
AP1	-38	42



AP2	-46	34
AP3	-53	27

Pada Tabel 7.9, terlihat proses normalisasi nilai RSSI yang awalnya adalah nilai negative dirubah menjadi nilai positif, normalisasi dilakukan dengan menggunakan persamaan 5.1.

Tabel 7.10 Pengujian Akurasi Pelevelan RGB

Access Point \ Ruang Scan	A1.8 A	A1.8 A
AP1 / R	133,875	134
AP2 / G	108,375	108
AP3 / B	86,0625	86

Pada Tabel 7.10, terlihat proses pelevelan dari nilai hasil normalisasi menjadi warna RGB, proses pelevelan menggunakan persamaan 5.2 dengan nilai JarakRSSI dalam kasus ini adalah 80 dan nilai MinRSSI adalah 0, hasil pelevelan kemudian dibulatkan. Hasil pembulatan dari pelevelan tersebut digunakan sebagai komponen nilai RGB yang digunakan untuk memberikan warna pada grid sampling yang kemudian dibandingkan dengan nilai RGB pada peta yang telah dibuat sebelumnya.

Untuk mengetahui tingkat akurasi aplikasi *Finding Dosen*, dilakukan dengan cara mencari nilai minimal dari perhitungan *Mean Squared Error* yang ada pada persamaan 2.3 antara data latih dengan data uji. Hasil perhitungan *Mean Squared Error* pada ruangan A1.8 ditampilkan pada tabel di bawah ini.

Tabel 7.11 Pengujian Akurasi Mean Squared Error

No.	RGB Data Uji	RGB Data Latih	A1.8 A	A1.8 B	Mean Squared Error Minimal	Hasil
			137	102		
		108		48		
		86		89		
1	A1.8 A	134	3	1586,33	3	A1.8 A
		108				
		86				



2	A1.8 B	105	1465,33	4,33	4,33	A1.8 B
		50				
		89				

Dari Tabel 7.11 didapatkan hasil minimal dari perhitungan *mean squared error* yang membuktikan perbandingan lokasi dosen sesungguhnya dengan lokasi yang ditampilkan di dalam aplikasi *Finding Dosen*. Maka didapatkan akumulasi hasil yang dinyatakan pada tabel berikut, hasil pengujian akurasi nantinya akan dihitung dengan persamaan akurasi sehingga didapatkan presentasi tingkat akurasi.

Tabel 7.12 Hasil Pengujian Akurasi

No.	Lokasi Sebenarnya	Lokasi Finding Dosen	Status
1	A1.8 A	A1.8 A	Valid
2	A1.9 A	A1.9 A	Valid
2	A1.10 B	A1.10 B	Valid
3	A1.3 D	A1.4.1	Tidak Valid
4	A1.5 A	A1.5 A	Valid
5	LA1T A	LA1T A	Valid
6	A2.16 C	LA2.17	Tidak Valid
7	A2.22 C	A2.22 C	Valid
8	A2.24 B	A2.24 B	Valid
9	A2.20 A	A2.20 A	Valid
10	LA2T B	LA2T B	Valid
11	E1.7 C	E1.7 C	Valid
12	E.5 D	E1.5 D	Valid
13	LE1T A	LE1T A	Valid
14	E1.3 A	E1.3 A	Valid
15	E1.2 A	E1.2 A	Valid
16	E2.9 B	E2.9 B	Valid
17	E2.5 D	E2.5 D	Valid
18	E2.1 C	E2.1 C	Valid
19	E2.2 A	E2.2 A	Valid



20	LE2.7	LE2.7	Valid
21	E2.8 D	E2.8 D	Valid
22	E2.4 A	E2.4 A	Valid
23	F2.9 A	F2.9 A	Valid
24	F2.8 B	F2.8 B	Valid
25	F2.5 A	F2.5 A	Valid
26	F2 Hall I	F2 Hall H	Tidak Valid
27	F2.1 A	F2.1 A	Valid
28	F3.11 A	F3.11 A	Valid
29	F3.3 A	F3.3 A	Valid
30	F3.12 B	F3.12 B	Valid
31	F3.8 A	F3.8 A	Valid
32	F3.7 C	F3 Hall E	Tidak Valid
34	F3.5 B	F3.5 B	Valid
35	F4 Hall J	F4 Hall J	Valid
36	F4.2 B	F4.2 B	Valid
37	F4.4 A	F4.4 A	Valid
38	F4.6 A	F4.6 A	Valid
39	F4.5 B	F4.5 B	Valid
40	F4 Hall E	F4 Hall E	Valid

7.9 Analisis Hasil Pengujian Akurasi

Analisis hasil pengujian akurasi dilakukan dengan memperhitungkan tingkat akurasi posisi dosen yang sebenarnya dengan posisi yang ditampilkan pada aplikasi. Akurasi dihitung dengan persamaan 7.1.

$$Akurasi (\%) = \left(\frac{Jumlah\ valid}{Jumlah\ sample} \right) \times 100 \quad (7.1)$$

$$\begin{aligned} Akurasi (\%) &= 36/40 \times 100 \\ &= 0,9 \times 100 \\ &= 90 \end{aligned}$$

Dari hasil pengujian akurasi yang ditampilkan pada Tabel 7.12, didapatkan jumlah valid sebesar 36 dari 40 sampel, disimpulkan bahwa akurasi yang didapatkan sebesar 90 %. Dengan hasil yang demikian menunjukkan fungsi

Aplikasi Finding Dosen dapat berjalan dengan baik. Akurasi pada aplikasi Finding Dosen tidak mencapai 100% akurat bukan dikarenakan kegagalan fungsional, namun beberapa titik yang tidak valid dikarenakan jarak nilai RGB yang sangat berdekatan sehingga perhitungan *Mean Squared Error* menghasilkan nilai yang terkecil pada ruangan lain, selain itu interferensi yang ada pada jaringan *wifi* juga mempengaruhi perhitungan nilai RGB yang tidak sesuai.



BAB 8 PENUTUP

8.1 Kesimpulan

Kesimpulan dari penelitian ini menjawab pertanyaan pada rumusan masalah yang terdiri dari dua poin dan berikut penjelasannya:

1. Perancangan dan Implementasi dari aplikasi *Finding Dosen* dilakukan dengan menganalisis kebutuhan yang menghasilkan gambaran umum sistem, identifikasi aktor, kebutuhan fungsional yang terdiri dari 10 kebutuhan. Kebutuhan fungsional tersebut dimodelkan ke dalam *use case diagram*, *use case scenario* dan *sequence diagram*.

Sebelum perancangan dibuat, dilakukan pengambilan data dan pengolahan data dari masing - masing ruangan di FILKOM Universitas Brawijaya khususnya pada Gedung A, Gedung E, dan Gedung F pada lantai 2 – 4. Data yang didapatkan diolah dan dibuat peta RGB sebagai dasar untuk penentuan lokasi dosen. Perancangan dilakukan dengan merancang *class diagram*, *physical data model*, merancang arsitektur aplikasi, dan merancang antar muka dari aplikasi *Finding Dosen*. Arsitektur yang digunakan pada aplikasi *Finding Dosen* adalah MVVM (*Model, View, View Model*).

Implementasi dilakukan dengan 2 tahap yaitu implementasi *back end* menggunakan Bahasa *Go* dan implementasi *front end* di aplikasi android menggunakan Bahasa *Java*. *Web service* juga dibuat agar aplikasi *Finding Dosen* dapat terhubung dengan *back end*, tipe *web service* yang digunakan adalah *REST API* atau *Restful Web Service* dimana komunikasi *resource* direpresentasikan dalam bentuk format *JSON*. Selain itu, implementasi antar muka juga dilakukan berdasarkan perancangan yang sudah dibuat.

2. Tingkat akurasi yang diperoleh dari pengujian aplikasi *Finding Dosen* sebesar 90%. Sehingga aplikasi *Finding Dosen* yang menggunakan metode *Color Radiomap Interpolation* memiliki tingkat akurasi yang baik.

8.2 Saran

Saran yang dapat diberikan untuk mengembangkan lebih lanjut dari aplikasi *Finding Dosen* adalah:

1. Aplikasi dapat dikembangkan lebih lanjut untuk pencarian posisi secara *real time* dan dikembangkan juga dalam platform *iOS*.
2. Mencakup seluruh ruangan dan gedung yang ada di FILKOM Universitas Brawijaya.
3. Menggunakan metode lain untuk meningkatkan akurasi ketika jarak nilai RGB antar *grid* sangat berdekatan.

DAFTAR PUSTAKA

- Ananda, M. R., Tolle, H. & Suharsono, A. 2016. *Implementasi Metode Color Radiomap Interpolation Pada Aplikasi Mobile Estimasi Posisi Pengguna (Studi Kasus Gedung A dan E PTIIK UB)*. Doro Jurnal. Volume 7 - Number 10.
- Arai, K. & Tolle, H. 2013. *Color Radiomap Interpolation for Efficient Fingerprint WiFi-based Indoor Location Estimation*. International Journal of Advanced Research in Artificial Intelligence (IJARAI), Vol. 2, No. 3.
- Bose, A. & Foh, C.H. 2007. *A practical path loss model for indoor WiFi positioning enhancement*. 2007 6th International Conference on Information, Communications & Signal Processing, 1-5.
- Developer, Android. 2017. *Android and Architecture*. [online] Tersedia di: <<https://android-developers.googleblog.com/2017/05/android-and-architecture.html>> [Diakses 9 September 2017]
- Hartono, R. & Purnomo, A. 2011. *Wireless Network 802.11*. D3 TI FMIPA UNS.
- Rashid, T., Chowdhury, S. S. & Nawal, M. F. 2016. *Composite Indoor Localization and Positioning System Based on Wi-Fi Repeater Technology*. Region 10 Conference (TENCON), 2016 IEEE. Singapura.
- Robert J., Arthur M., Kevin T. & Bobby, Z. 2011. *Indoor Positioning System*. EE 4820.
- Rohman, A. 2012. *Sejarah GPS dan Fungsinya*. [online] Tersedia di: <<http://ilmuti.org/2012/06/sejarah-gps-dan-fungsinya/>> [Diakses 22 Mei 2017]
- Susantok M., Kurniawan A. & Azwar H. 2013. *Thesis projects: Wifi Positioning System (WPS) Menggunakan Algoritma Neural Network Backpropagation di Area Kampus Politeknik Caltex Riau*, Jurnal Teknik Elektro dan Komputer, Vol.I, No. 2.
- Widyaman, C. & Kusumawardani, S. S. 2012. *Indoor Localization berbasis RSS Fingerprinting Menggunakan IEEE 802.11n*. ICITEE 2012, ISSN: 2008-6578. Yogyakarta.