

**RANCANG BANGUN PROTOTIPE SISTEM NAVIGASI DALAM
RUANGAN BERBASIS *WI-FI FINGERPRINT POSITIONING* DENGAN
STUDI KASUS PADA LANTAI 3 GEDUNG FAKULTAS TEKNOLOGI
INDUSTRI UNISSULA**

LAPORAN TUGAS AKHIR

Laporan ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana
Strata 1 pada Program Studi Teknik Informatika Fakultas Teknologi Industri
Universitas Islam Sultan Agung Semarang



**DISUSUN OLEH :
LANANG BAGUS PRAKARSA
NIM 32601500969**

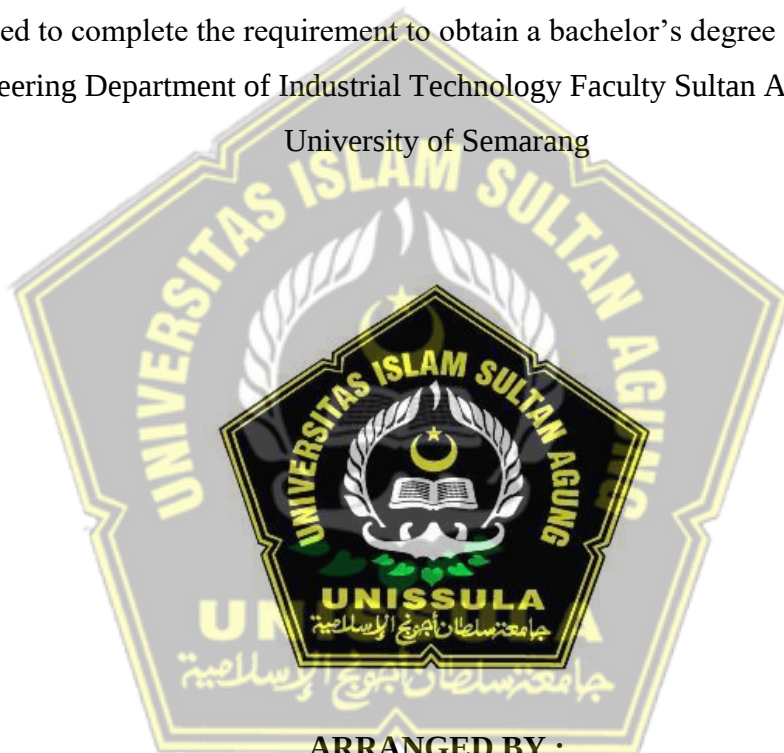
**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM SULTAN AGUNG
SEMARANG**

2022

FINAL PROJECT

DESIGN AND BUILD A PROTOTYPE OF A *WI-FI FINGERPRINT* POSITIONING SYSTEM BASED INDOOR NAVIGATION SYSTEM WITH A CASE STUDY ON THE 3rd FLOOR OF THE UNISSULA'S INDUSTRIAL TECHNOLOGY FACULTY BUILDING

Proposed to complete the requirement to obtain a bachelor's degree at Informatics
Engineering Department of Industrial Technology Faculty Sultan Agung Islamic
University of Semarang



ARRANGED BY :

LANANG BAGUS PRAKARSA

NIM 32601500969

**MAJORING OF INFORMATICS ENGINEERING
INDUSTRIAL TECHNOLOGY FACULTY
SULTAN AGUNG ISLAMIC UNIVERSITY
SEMARANG**

2022

LEMBAR PENGESAHAN PEMBIMBING

Laporan Tugas Akhir dengan judul "RANCANG BANGUN PROTOTIPE SISTEM NAVIGASI DALAM RUANGAN BERBASIS *WI-FI* FINGERPRINT POSITIONING DENGAN STUDI KASUS PADA LANTAI 3 GEDUNG FAKULTAS TEKNOLOGI INDUSTRI UNISSULA" ini disusun oleh :

Nama : Lanang Bagus Prakarsa

NIM : 32601500969

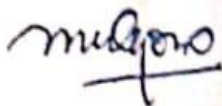
Program studi : Teknik Informatika

Telah disahkan oleh dosen pembimbing pada :

Hari :

Tanggal :

Pembimbing I



Ir. Sri Mulyono, M.Eng.
NIDN.0626066601

Pembimbing II



Badicah, S.T., M.Kom.
NIDN.0619018701

Mengetahui,

Ketua Program Studi Teknik Informatika



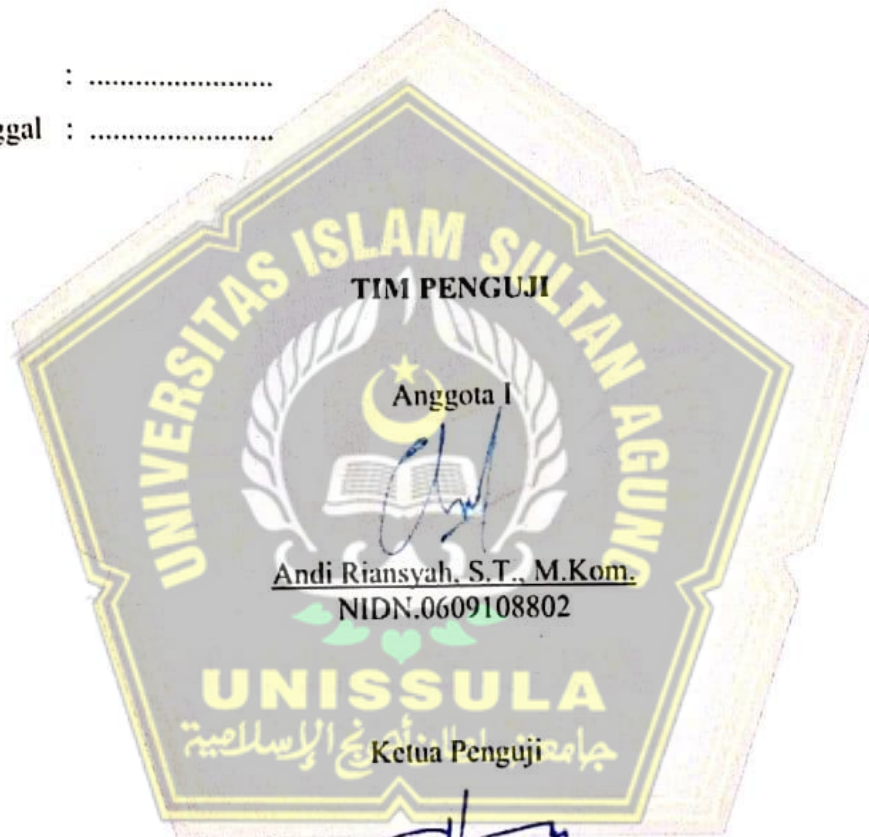
Ir. Sri Mulyono, M.Eng.
NIDN.0626066601

LEMBAR PENGESAHAN PENGUJI

Laporan Tugas Akhir dengan judul "RANCANG BANGUN PROTIPE SISTEM NAVIGASI DALAM RUANGAN BERBASIS *WI-FI FINGERPRINT POSITIONING* DENGAN STUDI KASUS PADA LANTAI 3 GEDUNG FAKULTAS TEKNOLOGI INDUSTRI UNISSULA" ini telah dipertahankan di depan dosen penguji Tugas Akhir pada :

Hari :

Tanggal :




Dedy Kurniadi, S.T., M.Kom.
NIDN.0626066601

SURAT PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertandatangan di bawah ini :

Nama : Lanang Bagus Prakarsa

NIM : 32601500969

Judul Tugas Akhir : RANCANG BANGUN PROTOTIPE SISTEM
NAVIGASI DALAM RUANGAN BERBASIS *WI-FI*
FINGERPRINT POSITIONING DENGAN STUDI
KASUS PADA LANTAI 3 GEDUNG FAKULTAS
TEKNOLOGI INDUSTRI UNISSULA

Dengan bahwa ini saya menyatakan bahwa judul dan isi Tugas Akhir yang saya buat dalam rangka menyelesaikan Pendidikan Strata Satu (S1) Teknik Informatika tersebut adalah asli dan belum pernah diangkat, ditulis ataupun dipublikasikan oleh siapapun baik keseluruhan maupun sebagian, kecuali yang secara tertulis dicatu dalam naskah ini dan disebutkan dalam daftar pustaka, dan apabila di kemudian hari ternyata terbukti bahwa judul Tugas Akhir tersebut pernah diangkat, ditulis ataupun dipublikasikan, maka saya bersedia dikenakan sanksi akademis. Demikian surat pernyataan ini saya buat dengan sadar dan penuh tanggung jawab.

UNISSULA
جامعة سلطان أبوبوع الإسلامية

Semarang, 15 Maret 2023

Yang menyatakan,



Lanang Bagus Prakarsa

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH

Saya yang bertandatangan di bawah ini :

Nama : Lanang Bagus Prakarsa

NIM : 32601500969

Program studi : Teknik Informatika

Fakultas : Teknologi Industri

Alamat asal : Dsn. Krajan, RT 13/03, Ds. Tenganan, Kec. Tenganan,
Kab. Semarang

Dengan ini menyatakan Karya Ilmiah berupa Tugas Akhir dengan judul :
**RANCANG BANGUN PROTOTYPE SISTEM NAVIGASI DALAM
RUANGAN BERBASIS WI-FI FINGERPRINT POSITIONING DENGAN
STUDI KASUS PADA LANTAI 3 GEDUNG FAKULTAS TEKNOLOGI
INDUSTRI UNISSULA**

Menyetujui menjadi hak milik Universitas Islam Sultan Agung serta memberikan Hak bebas Royalti Non-Eksklusif untuk disimpan, dialihmediakan, dikelola dan pangkalan data dan dipublikasikan di internet dan media lain untuk kepentingan akademis selama tetap mencantumkan nama penulis sebagai pemilik hak cipta. Pernyataan ini saya buat dengan sungguh-sungguh. Apabila dikemudian hari terbukti ada pelanggaran Hak Cipta/Plagiarisme dalam karya ilmiah ini, maka segala bentuk tuntutan hukum yang timbul akan saya tanggung secara pribadi tanpa melibatkan Universitas Islam Sultan Agung.

Semarang, 15 Maret 2023

Yang menyetujui

KATA PENGANTAR

Dengan mengucapkan puji syukur atas kehadiran Allah SWT berkat rahmatnya penulis dapat menyelesaikan Tugas Akhir yang berjudul **RANCANG BANGUN PROTOTYPE SISTEM NAVIGASI DALAM RUANGAN**

KATA PENGANTAR

Dengan mengucapkan puji syukur atas kehadiran Allah SWT berkat rahmatnya penulis dapat menyelesaikan Tugas Akhir yang berjudul **RANCANG BANGUN PROTOTIPE SISTEM NAVIGASI DALAM RUANGAN BERBASIS WI-FI FINGERPRINT POSITIONING DENGAN STUDI KASUS PADA LANTAI 3 GEDUNG FAKULTAS TEKNOLOGI INDUSTRI UNISSULA**. Penulisan ini tidak dapat diselesaikan tanpa adanya dukungan dari berbagai pihak baik moril maupun materiil. Karena itu, penulis ini sangat ingin menyampaikan ucapan terimakasih kepada seluruh pihak yang telah membantu penyusunan skripsi saya ini terutama :

1. Almarhumah ibunda, serta ayahanda yang telah memberikan restu, doa, semangat, keselamatan dan keberhasilan selama menempuh semua ujian.
2. Keluarga serta para sahabat yang telah memberikan semangat, motivasi, dan doa dalam mengerjakan Tugas Akhir saya.
3. Ibu Badieah, ST., M.Kom. selaku Dosen Pembimbing I yang telah meluangkan waktu dan memberi ilmu kepada penulis.
4. Bapak Ir. Sri Mulyono, M.Eng. selaku Dosen Pembimbing II yang telah meluangkan waktu dan memberi ilmu kepada penulis.
5. Para Dosen FTI Unissula yang telah memberikan banyak ilmu yang bermanfaat.

Dengan segala kerendahan hati, penulis menyadari masih terdapat kekurangan-kekurangan dari segi kualitas atau kuantitas maupun dari ilmu pengetahuan dalam menyusun laporan, sehingga penulis mengharapkan adanya saran dan kritikan bersifat membangun demi kesempurnaan laporan ini di masa mendatang.

Semarang,

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN PENGUJI	iv
SURAT PERNYATAAN KEASLIAN TUGAS AKHIR	v
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH.....	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN.....	xvi
ABSTRAK.....	xvii
BAB I.....	1
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	2
1.3. Pembatasan Masalah	2
1.4. Tujuan.....	3
1.5. Manfaat.....	3
1.6. Sistematika Penulisan.....	3
BAB II.....	6
2.1. Tinjauan Pustaka	6
2.2. Dasar Teori	7
2.2.1. Wi-fi.....	7
2.2.2. Wireless indoor positioning system.....	8
2.2.3. Fingerprint positioning	9

2.2.4.	Algoritma <i>k-NN</i> (<i>k-Nearest Neighbor</i>)	10
2.2.5.	<i>Pathfinding</i>	10
2.2.6.	Algoritma <i>pathfinding A*</i> (<i>A-star</i>).....	11
BAB III		13
3.1.	Metode Penelitian.....	13
3.1.1.	Survei lokasi penelitian.....	13
3.1.2.	Studi pustaka	15
3.1.3.	Metode pengembangan sistem.....	15
3.2.	Gambaran Sistem.....	16
3.3.	Komponen Pemposisian.....	18
3.4.	Pemposisian.....	19
3.4.1.	Mekanisme pemposisian.....	19
3.4.2.	Mekanisme 3 buah <i>transmitter</i>	21
3.4.3.	Fase <i>offline</i> pemposisian	24
3.4.4.	Fase online pemposisian	28
3.4.5.	Penentuan posisi dengan algoritma <i>k-NN</i>	29
3.5.	Navigasi.....	36
3.5.1.	Mekanisme area pencarian rute.....	36
3.5.2.	Mekanisme algoritma <i>A*</i>	39
3.5.3.	Pencarian rute dengan algoritma <i>A*</i>	44
3.6.	Perancangan <i>Database</i>	57
3.6.1.	<i>Database</i> berbasis <i>cloud</i>	57
3.6.2.	Struktur database data sampel.....	58
3.7.	Analisa Kebutuhan dan Diagram Sistem.....	60
3.7.1.	Aktor	60

3.7.2.	Use case diagram	62
3.7.3.	Data flow diagram	64
3.7.4.	Flowchart diagram	74
3.7.5.	Sequence diagram	81
3.8.	Perancangan Antarmuka Sistem.....	89
BAB IV		94
4.1.	Implementasi Sistem	94
4.1.1.	Implementasi perangkat keras.....	94
4.1.2.	Penempatan <i>transmitter</i>	96
4.1.3.	Implementasi antarmuka aplikasi sistem	98
4.2.	Pengujian Sistem dengan <i>Basis path Testing (White Box)</i>	107
4.2.1.	Pengujian pada fungsi pemposisian	107
4.2.2.	Pengujian <i>whitebox</i> pada algoritma pemposisian <i>k-NN</i>	110
4.2.3.	Pengujian <i>whitebox</i> pada algoritma <i>pathfinding A*</i>	112
4.3.	Pengujian Akurasi Pemposisian dengan Nilai Variabel “k” Berbeda ..	114
BAB V.....		119
5.1.	Kesimpulan	119
5.2.	Saran.....	119

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi algoritma <i>k-NN</i> (Suyanto, 2018)	7
Gambar 2.2 Ilustrasi <i>pathfinding</i> (Daniel, 2012)	8
Gambar 2.3 Ilustrasi <i>pathfinding</i> algoritma <i>A*</i> (Lester, 2005)	9
Gambar 3.1 Pemetaan area lantai 3 gedung FTI Unissula	10
Gambar 3.2 <i>Flowchart prototyping</i>	11
Gambar 3.3 Perangkat <i>wi-fi access point</i>	13
Gambar 3.4 Mekanisme fase <i>offline</i> dan fase <i>online</i> sistem pemindaian sidik jari	14
Gambar 3.5 Mekanisme fase <i>offline</i> dan fase <i>online</i> pada teknik <i>fingerprint positioning</i>	15
Gambar 3.6 Pemosisian dengan 2 buah <i>transmitter</i>	16
Gambar 3.7 Pemosisian dengan 3 buah <i>transmitter</i> (1)	17
Gambar 3.8 Pemosisian dengan 3 buah <i>transmitter</i> (2)	17
Gambar 3.9 Contoh pengambilan data sampel <i>wi-fi</i> di satu titik di ruang 302 ...	18
Gambar 3.10 Ilustrasi pemfilteran jaringan <i>wi-fi</i>	18
Gambar 3.11 Ilustrasi informasi daa RSS 3 <i>transmitter</i> yang telah diambil	19
Gambar 3.12 Ilustrasi pemberian label lokasi pada data sampel RSS yang telah diambil	19
Gambar 3.13 Ilustrasi pengambilan sampel <i>wi-fi</i> di seluruh lokasi pada lantai 3 gedung FTI	20
Gambar 3.14 Ilustrasi aplikasi sistem melakukan <i>scanning</i> jaringan <i>wi-fi</i> pada fase <i>online</i>	21
Gambar 3.15 Ilustrasi penyeleksian jaringan <i>wi-fi</i> fase <i>online</i>	21

Gambar 3.16 Informasi level kekuatan sinyal jaringan <i>wi-fi</i> sebagai data masukan	22
Gambar 3.17 Ilustrasi pemilihan data sampel <i>wi-fi</i> pada algoritma <i>k-NN</i>	24
Gambar 3.18 Ilustrasi perhitungan jarak <i>euclidean</i> data sampel 1 dengan data masukan	24
Gambar 3.20 <i>Graph</i> peta navigasi	27
Gambar 3.21 Atribut dinding pada sel peta navigasi	28
Gambar 3.22 Basis tata letak peta navigasi yang telah diatur atribut dindingnya	29
Gambar 3.23 Ilustrasi perhitungan skor G pada sel	30
Gambar 3.24 Ilustrasi perhitungan skot H pada sel	30
Gambar 3.25 Ilustrasi perhitungan skor F pada sel	31
Gambar 3.26 Ilustrasi fungsi <i>heuristic</i> pada algoritma A^*	31
Gambar 3.27 Ilustrasi atribut <i>parent</i> pada sel	33
Gambar 3.28 Ilustrasi pencarian rute dengan algoritma A^* (0)	33
Gambar 3.29 Ilustrasi pencarian rute dengan algoritma A^* (1)	34
Gambar 3.30 Ilustrasi pencarian rute dengan algoritma A^* (2)	34
Gambar 3.31 Ilustrasi pencarian rute dengan algoritma A^* (3)	35
Gambar 3.32 Ilustrasi pencarian rute dengan algoritma A^* (4)	35
Gambar 3.33 Ilustrasi pencarian rute dengan algoritma A^* (5)	36
Gambar 3.34 Ilustrasi pencarian rute dengan algoritma A^* (6)	36
Gambar 3.35 Ilustrasi pencarian rute dengan algoritma A^* (7)	37
Gambar 3.36 Ilustrasi pencarian rute dengan algoritma A^* (8)	37
Gambar 3.37 Ilustrasi pencarian rute dengan algoritma A^* (9)	38
Gambar 3.38 Ilustrasi pencarian rute dengan algoritma A^* (10)	38
Gambar 3.39 Ilustrasi pencarian rute dengan algoritma A^* (11)	39

Gambar 3.40 Ilustrasi pencarian rute dengan algoritma A* (12)	39
Gambar 3.41 Ilustrasi pencarian rute dengan algoritma A* (13)	40
Gambar 3.42 Ilustrasi pencarian rute dengan algoritma A* (14)	40
Gambar 3.43 Ilustrasi pencarian rute dengan algoritma A* (15)	41
Gambar 3.44 Ilustrasi pencarian rute dengan algoritma A* (16)	41
Gambar 3.45 Ilustrasi pencarian rute dengan algoritma A* (17)	42
Gambar 3.46 Ilustrasi pencarian rute dengan algoritma A* (18)	42
Gambar 3.47 Ilustrasi pencarian rute dengan algoritma A* (19)	43
Gambar 3.48 Ilustrasi arsitektur sistem dengan <i>database</i> berbasis <i>cloud</i>	44
Gambar 3.49 Alur penggunaan sistem oleh admin	46
Gambar 3.50 Alur penggunaan sistem oleh pengguna	47
Gambar 3.51 <i>Usecase diagram</i> sistem navigasi dalam ruangan	48
Gambar 3.52 <i>Data flow diagram</i> level konteks	49
Gambar 3.53 <i>Data flow diagram</i> level 0	50
Gambar 3.54 <i>Data flow diagram</i> level 1 proses 1 (autentikasi)	52
Gambar 3.55 <i>Data flow diagram</i> level 1 proses 2 (kelola data)	53
Gambar 3.56 <i>Data flow diagram</i> level 1 proses 3 (pemposisian)	55
Gambar 3.57 <i>Data flow diagram</i> level 1 proses 4 (navigasi)	57
Gambar 3.58 <i>Flowchart diagram</i> penggunaan sistem oleh admin	59
Gambar 3.59 <i>Flowchart diagram</i> penggunaan sistem oleh pengguna	60
Gambar 3.60 <i>Flowchart diagram</i> proses pemposisian	61
Gambar 3.61 <i>Flowchart diagram</i> proses navigasi	63
Gambar 3.62 <i>Sequence diagram</i> proses login	65
Gambar 3.63 <i>Sequence diagram</i> proses hapus data sampel	66
Gambar 3.64 <i>Sequence diagram</i> proses tambah data	67

Gambar 3.65 <i>Sequence diagram</i> proses pemposisian	68
Gambar 3.66 <i>Sequence diagram</i> proses navigasi	69
Gambar 3.67 Rancangan antarmuka halaman utama	71
Gambar 3.68 Rancangan antarmuka halaman login	71
Gambar 3.69 Rancangan antarmuka halaman kelola data sampel	72
Gambar 3.70 Rancangan antarmuka halaman tambah data sampel	73
Gambar 3.71 Rancangan antarmuka halaman pencarian posisi	73
Gambar 3.72 Rancangan antarmuka halaman navigasi	74
Gambar 4.1 Penempatan <i>transmitter</i>	76
Gambar 4.2 Pengaturan SSID pada aplikasi navigasi yang akan digunakan oleh sistem untuk pemposisian	77
Gambar 4.3 Tampilan halaman utama	78
Gambar 4.4 Tampilan halaman login admin	79
Gambar 4.5 Tampilan halaman pengaturan sistem oleh admin	80
Gambar 4.6 Tampilan halaman kelola data sampel	81
Gambar 4.7 Tampilan halaman tambah data sampel	82
Gambar 4.8 Tampilan halaman pengaturan SSID <i>access point</i>	83
Gambar 4.9 Tampilan halaman pengaturan nilai variabel <i>k</i>	84
Gambar 4.10 Tampilan halaman pemposisian dan navigasi	85
Gambar 4.11 Perhitungan <i>cyclomatic complexity</i> pada <i>flowchart</i> pemposisian ..	86
Gambar 4.12 Perhitungan <i>cyclomatic complexity</i> pada algoritma pemposisian <i>k-NN</i>	88
Gambar 4.13 Perhitungan <i>cyclomatic complexity</i> pada algoritma <i>pathfinding A*</i>	90

DAFTAR TABEL



DAFTAR LAMPIRAN



ABSTRAK

Area publik merupakan suatu area dimana lokasi tersebut menjadi pusat masyarakat umum untuk beraktifitas. Sebuah area publik biasanya mempunyai area yang luas dengan berbagai sub area yang terdapat di dalamnya. Contoh dari area publik adalah mall, bandara, stasiun, museum, alun-alun, dan lain-lain. Pengunjung baru suatu are publik biasanya kebingungan saat ingin menelusuri tempat-tempat di dalamnya. Akan sangat memudahkan jika suatu sistem navigasi dapat diterapkan pada fasilitas umum seperti ini. Area publik dapat berbentuk *indoor* maupun *outdoor*. Sistem navigasi berbasis satelit seperti GPS mempunyai keterbatasan jika diterapkan pada area *indoor* disebabkan oleh suatu kondisi yang disebut *line of sight* yaitu suatu kondisi dimana tidak memungkinkan terjadinya transmisi sinyal yang tidak terhalan antara perangkat dan satelit. *Wi-fi fingerprint positioning* merupakan salah satu metode alternatif dalam sistem penentuan posisi dengan memanfaatkan *wi-fi access point* sebagai *transmitter*. Jangkauan sinyal *wi-fi* yang cukup luas dan kebutuhan perangkat yang sangat minimal diharapkan dapat menjadi solusi optimal untuk membangun suatu sistem navigasi dalam ruangan.

Area publik merupakan suatu tempat

Kata Kunci : *wi-fi fingerprint positioning*, sistem navigasi dalam ruangan

ABSTRACT

The public area is an area where the location is the center of the general public for activities. A public area usually has a large area with various sub areas contained in it. Examples of public areas are malls, airports, stations, museums, squares, and others. New visitors to a public navigation sistem can be applied to

public facilities like this. Public areas can be either indoor or outdoor. Satellite-based navigation systems such as GPS have limitations when applied to indoor areas due to a condition called line of sight, which is a condition where unobstructed signal transmission between the device and satellite is not possible. Wi-fi fingerprint positioning is an alternative method in positioning systems by utilizing a wi-fi access point as a transmitter. Wi-fi signal coverage is quite wide and minimal device requirements are expected to be the optimal solution for building an indoor navigation system.

Keywords: wi-fi fingerprint positioning, indoor navigation system



BAB I

PENDAHULUAN

1.1. Latar Belakang

Tidak bisa dipungkiri bahwa saat ini kita sudah sangat jauh memasuki era digital. Segala aspek kehidupan manusia kini tak lepas dari sebuah peralihan yang disebut dengan digitalisasi. Kecanggihan teknologi digital benar-benar dapat membantu manusia mengatasi permasalahan yang dihadapi setiap waktu. Salah satu contoh manfaat teknologi digital dalam kasus permasalahan sehari-hari adalah penggunaan teknologi GPS sebagai penunjuk jalan saat kita sedang berpergian. Hari ini ketika kita ingin pergi keluar rumah hanya sekedar untuk *hang-out* saja, kita tidak perlu khawatir salah mengambil jalan ataupun tersesat karena aplikasi navigasi berbasis GPS telah terinstal di *smartphone* yang kita miliki. Dengan banyaknya aplikasi berbasis sistem navigasi yang dirancang untuk *gadget* maupun *smartphone*, akan sangat mempermudah berbagai pekerjaan manusia sehari-hari.

Saat ini sistem navigasi berbasis GPS telah meng-cover hampir seluruh tempat di dunia dengan tingkat *perincian* lokasi yang sudah sangat detail dan akurasi yang tinggi. Pasinggi dkk., pada tahun 2018 dalam *literature review*-nya mengungkapkan bahwa sistem navigasi berbasis satelit semacam GPS mampu memberikan tingkat akurasi yang tinggi untuk penggunaan di luar ruangan (*outdoor*), namun akan menemukan tantangan yang signifikan untuk penggunaan di dalam ruangan (*indoor*). Seperti yang kita tahu bahwa mall, rumah sakit, bandara, dan lain-lain merupakan area publik yang berbentuk *indoor area*. GPS sendiri tidak dirancang untuk penggunaan di dalam ruangan karena kondisi tersebut tidak memungkinkan terjadinya transmisi yang tidak terhalang antara perangkat dan satelit. Kondisi ini biasa disebut dengan *line of sight*.

Area publik merupakan suatu tempat yang menjadi pusat masyarakat umum dalam melakukan berbagai aktifitas. Area publik biasanya memiliki jangkauan area yang sangat luas dan memiliki sektor-sektor beragam di dalamnya baik *indoor*

maupun *outdoor*. Biasanya pengunjung area publik yang baru pertama kali datang ke tempat tersebut akan kebingungan mencari lokasi yang ingin dituju. Padahal sebenarnya di setiap area publik telah disediakan denah khusus untuk pengunjung, namun tidak jarang banyak pengunjung malas untuk sekedar memperhatikannya. Jika sistem navigasi dalam ruangan yang berbasis digital dapat diterapkan pada area-area publik seperti ini, tentu akan banyak membantu orang-orang yang beraktifitas di dalamnya.

1.2. Perumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dirumuskan beberapa permasalahan sebagai berikut :

1. Pengunjung baru suatu area publik kesulitan saat menelusuri tempat-tempat di dalamnya ketika hanya mengandalkan sebuah denah.
2. Sistem navigasi berbasis satelit (*satnav*) memiliki keterbatasan saat diimplementasikan pada area *indoor* yaitu tidak memungkinkan terjadinya transmisi sinyal yang tidak terhalang antara perangkat dan satelit (*line of sight*).

1.3. Pembatasan Masalah

Pada penelitian tugas akhir ini diberlakukan beberapa pembatasan masalah yaitu :

1. Jangkauan area navigasi pada prototipe sistem yang dibangun dibatasi hanya pada lantai 3 gedung Fakultas Teknologi Industri Unissula untuk faktor fleksibilitas dalam pengembangannya yang hanya berisi ruang kuliah saja dan sifatnya bebas untuk mahasiswa melakukan aktifitas.

1.4. Tujuan

Tujuan yang ingin dicapai dalam penelitian tugas akhir ini adalah untuk merancang serta membangun sebuah prototipe sistem navigasi yang dapat beroperasi di dalam ruangan menggunakan metode *wi-fi fingerprint positioning* untuk melakukan penentuan posisi, dan algoritma A* untuk melakukan pencarian rute navigasi, yang diharapkan mampu mempermudah pengunjung baru suatu *public area* dalam upaya menelusuri lokasi-lokasi yang berada dalam lingkup *public area*.

1.5. Manfaat

Pada penelitian tugas akhir ini diharapkan akan menghasilkan beberapa manfaat yaitu :

1. Prototipe sistem navigasi dalam ruangan yang berhasil dibangun dan berfungsi dengan baik dapat dijadikan referensi untuk diterapkan dalam kasus nyata pada area publik.
2. Dapat mengetahui kelebihan dan kekurangan dari rancangan dan metode yang digunakan dalam membangun prototipe.

1.6. Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini dibagi menjadi beberapa bab, yaitu :

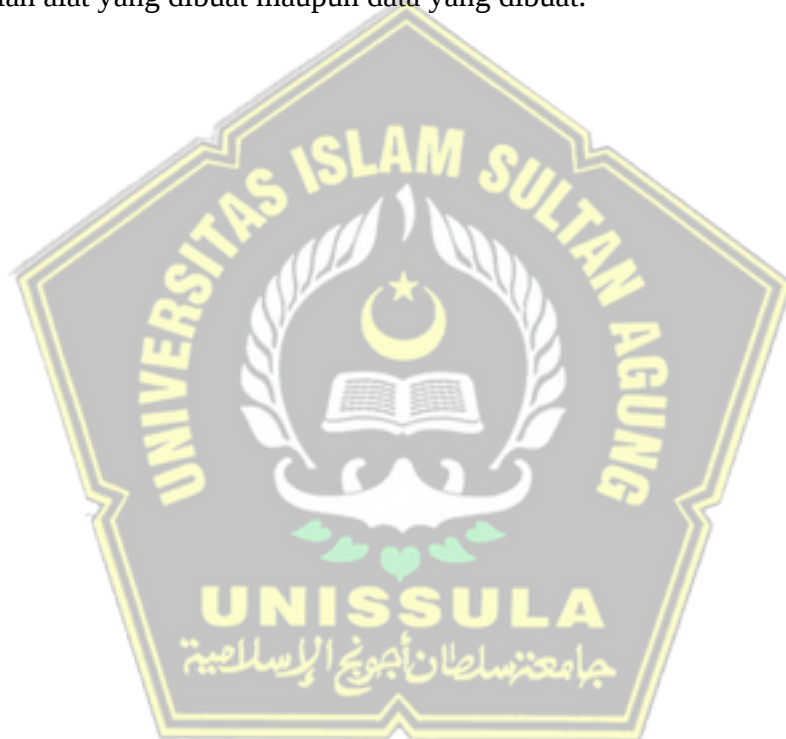
BAB 1: PENDAHULUAN, pada tahap ini menampilkan latar belakang, perumusan masalah, pembatasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB 2: TINJAUAN PUSTAKA DAN DASAR TEORI, pada tahap ini menampilkan tentang konsep dan prinsip dasar guna memecahkan permasalahan pada tugas akhir dengan bersumber dari berbagai referensi yang ada. Menguraikan

hal-hal yang ada pada tugas akhir secara relevan sesuai penelitian yang dilakukan didukung dengan adanya index atau notasi keterangan sumber referensi didapat.

BAB 3: METODE PENELITIAN, pada bab ini di sampaikan metode yang digunakan untuk melakukan perancangan sistem serta pendekatan guna mendapatkan solusi dari permasalahan yang ada. Dapat berupa perhitungan, simulasi dalam komputer dan desain alat.

BAB 4: HASIL DAN ANALISIS PENELITIAN, memuat hasil penelitian dan pengujian alat yang dibuat maupun data yang dibuat.





BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Menurut Deng dkk. (2013) seperti dikutip oleh Xia dkk. (2017) menjelaskan bahwa potensi pasar teknologi *indoor positioning* telah meningkat pesat dengan luasnya distribusi *wi-fi*. Presisi yang tinggi, konsumsi daya rendah, dan biaya rendah menjadikannya area yang sangat menarik dalam penelitian teknologi *indoor positioning*. Menurut standar IEEE 802.11, kartu jaringan nirkabel *wi-fi* dan *access point* (AP) memiliki fungsi mengukur intensitas sinyal frekuensi radio. Oleh karena itu, user dapat memanfaatkan perangkat seluler seperti *smartphone*, laptop, PC tablet, dan lainnya untuk menjalankan *indoor positioning* dengan *wi-fi* dan algoritma tertentu.

Sampai saat ini, sebagian besar metode penentuan posisi dalam ruangan didasarkan pada teknologi *fingerprint matching* dikombinasikan dengan metode lain. Dalam artikel jurnal “*Indoor Fingerprint positioning Based on wi-fi: An Overview*” Xia dkk. (2017) menjelaskan jika dibandingkan dengan sistem pemosisian lain, teknologi *wi-fi Fingerprint positioning* memiliki keuntungan dari biaya rendah dan presisi tinggi. Karena penyebaran yang luas dan penggunaan *wi-fi* di seluruh dunia, teknologi pemosisian sidik jari (*fingerprint positioning*) dapat diterapkan pada setiap skenario dalam ruangan di mana jaringan *wi-fi* berada dikerahkan tanpa perangkat keras tambahan, yang membuat biaya teknologi rendah. Teknologi ini menggunakan kekuatan sinyal *wi-fi* untuk memodelkan dan mengukur, tanpa harus mengidentifikasi lokasi yang tepat dari AP. Dalam lingkungan *indoor* yang kompleks, dalam kondisi biaya rendah, atribut ruang-waktu seperti *angle and time of arrival* dapat menghasilkan kesalahan besar; Namun, intensitas sinyal relatif stabil, sehingga membuat akurasi posisi metode ini lebih besar dari yang lain.

Pada tahun 2013, Bai dkk. (2013) mengusulkan metode yang dikombinasikan dengan algoritma *k-NN* (*k-Nearest Neighbor*) dan teknologi *fingerprint positioning*; metode *triangular positioning* didasarkan pada model *classic path loss* dimana mereka menemukan bahwa akurasi metode *positioning* mereka meningkat 0,5 meter dibandingkan dengan sistem *Ekahau real-time positioning*.

Musthafa dkk. (2016) mengimplementasikan teknologi *Indoor Localization* dengan metode *Clustering Filtered k-Nearest Neighbors* dalam membangun sistem navigasi *indoor* yang beroperasi pada *smartphone* android untuk studi kasus pada gedung bertingkat. Pada hasil pengujiannya menunjukkan bahwa sistem ini memiliki akurasi pendeteksian posisi yang tergolong tinggi sebesar 88,953%.

Permana dkk. (2013) menggunakan metode *Iterative Deepening A** (IDA*) untuk melakukan pencarian rute dalam membangun aplikasi *Indoor Positioning System*. Hasil pengujiannya menunjukkan bahwa algoritma pencarian ini dapat menghitung jarak terpendek dari semua titik lokasi yang ada ke semua titik yang disediakan.

2.2. Dasar Teori

2.2.1. Wi-fi

Wi-fi (*wireless fidelity*) adalah standar jaringan area lokal nirkabel, yang merupakan bentuk peralatan penghubung jaringan area lokal yang digunakan di ruang dengan skala kecil melalui sinyal frekuensi radio. wi-fi tidak hanya dapat mengidentifikasi interkoneksi perangkat, tetapi juga dapat terhubung ke jaringan yang lebih luas melalui wi-fi. wi-fi adalah standar IEEE 802.11b, menggunakan gelombang elektromagnetik teknologi frekuensi radio (RF) sebagai pembawa data. Karena frekuensi tinggi (2,4 GHz dan 5 GHz) dan radius transmisi sinyal wi-fi yang pendek (sekitar 100 m), sulit untuk menjangkau seluruh area yang luas dengan wi-fi dan hanya dapat digunakan pada *hotspot* lokal seperti apartemen, kantor, pasar, dll. (Xia dkk., 2017)

2.2.2. *Wireless indoor positioning system*

WIPS merupakan sebuah sistem yang khusus dirancang untuk menentukan posisi di dalam ruangan. WIPS mulai dikembangkan karena teknologi penentuan lokasi berbasis satelit (*satnav*) seperti GPS (*Global Positioning System*) tidak dirancang untuk penggunaan di dalam ruangan karena kondisi tersebut tidak memungkinkan terjadinya transmisi yang tidak terhalang (*Line of sight*) antar perangkat dan satelit. Tidak seperti *satnav* yang menggunakan satelit, ciri dari WIPS yaitu menggunakan teknologi *wireless* lainnya seperti WLAN, *infrared*, *bluetooth*, dan RFID.

Sebuah WIPS dapat menyediakan beberapa jenis informasi sesuai dengan kebutuhan user. Informasi tersebut dapat berupa lokasi absolut, relatif, atau *proximity*. Informasi absolut dinyatakan dalam bentuk koordinat, misalnya koordinat GPS. Informasi relatif diperoleh berdasarkan referensi sebuah objek lain, misalnya objek A berada di sebelah kiri objek B. Informasi *proximity* dinyatakan bentuk simbolik, misalnya dapur, kantor dan kelas. Sebuah sistem juga dapat mengombinasikan jenis informasi untuk memperkaya informasi, misalnya sebuah lokasi absolut ditambah dengan informasi *proximity* sebagai tambahan informasi.

WIPS dapat dikategorikan berdasarkan beberapa kriteria. Berdasarkan penggunaan infrastruktur yang telah ada, WIPS dapat dikelompokkan menjadi *infrastructure-based* dan *non-infrastructure-based*. penggunaan infrastruktur yang telah ada dapat menurunkan biaya, namun di sisi lain pembangunan infrastruktur khusus dapat memberikan kebebasan desain yang bisa berpengaruh pada kualitas sistem.

Kriteria lain yang dapat digunakan adalah arsitektur atau topologi. Sebuah WIPS minimal memiliki dua buah komponen: komponen *transmitter* dan komponen penghitung. Berdasarkan arsitektur, WIPS dapat dikelompokkan menjadi *self-positioning* dan *remote-positioning*. Pada arsitektur *self-positioning*, perangkat mobile akan menerima sinyal dari beberapa *transmitter*, berdasarkan sinyal yang

diterima tersebut, perangkat *mobile* akan melakukan kalkulasi, sedangkan pada arsitektur *remote-positioning*, perangkat *mobile* berperan sebagai *transmitter* dan beberapa perangkat pengukur menerima sinyal tersebut. Hasil pengukuran dikumpulkan pada sebuah *master station* untuk dilakukan perkiraan posisi. (Pasinggi dkk., 2018)

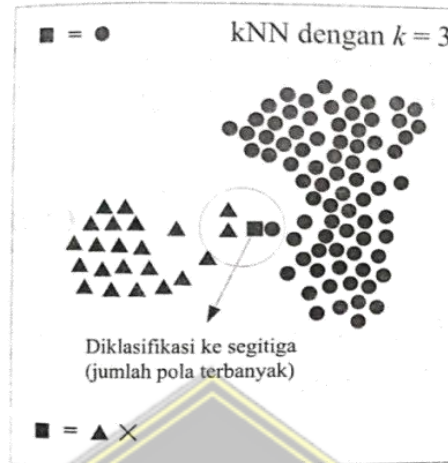
2.2.3. *Fingerprint positioning*

Ada banyak teknik yang dapat diimplementasikan jika menyoal tentang sistem penentuan posisi. Namun jika menyangkut soal sistem penentuan posisi di dalam ruangan maka teknik *fingerprint positioning* merupakan salah satu yang sangat populer dikarenakan faktor kemudahan pengembangannya serta kebutuhan perangkat yang sangat minimal. *Fingerprint positioning* merupakan salah satu teknik dalam penentuan lokasi. Mirip seperti teknologi *fingerprint* (pencocokan sidik jari), teknik ini mencocokkan fitur sinyal pada sebuah lokasi dengan data hasil pengukuran agar dapat mengetahui dimana posisi perangkat saat ini berada.

Terdapat dua fase dalam penggunaan teknik ini yaitu fase *online* dan fase *offline*. Pada fase *offline* dilakukan survei pada lokasi. Data yang diambil berupa koordinat/label disertai data hasil pengukuran kekuatan sinyal dari beberapa *transmitter* atau biasa disebut RSS (*Received Signal Strength*). Dan pada fase *online*, yaitu pada saat penentuan lokasi, RSS yang diterima dicocokkan dengan kumpulan data yang diperoleh pada fase *offline*.

Setidaknya ada lima algoritma yang dapat digunakan untuk melakukan perhitungan penentuan lokasi pada teknik ini: metode probabilistik, *k-nearest neighbor* (kNN), *neural networks*, *support vector machine* (SVM), dan *smallest M-vertex polygon* (SMP). (Pasinggi dkk., 2018)

2.2.4. Algoritma k -NN (k -Nearest Neighbor)



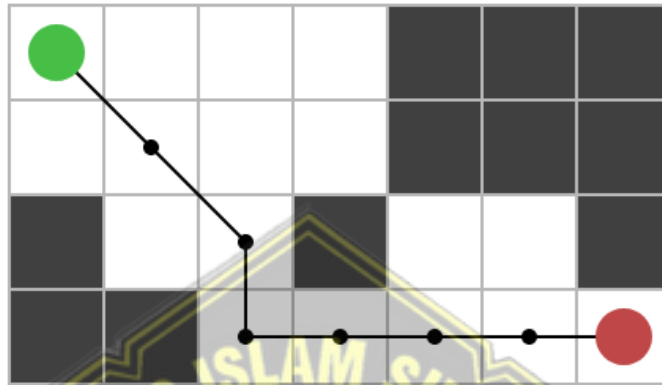
Gambar 2.1 Ilustrasi algoritma k -NN (Suyanto, 2018)

Algoritma k -NN merupakan cabang dari sebuah teknik *machine learning* yaitu *Instance-Based Learning* (IBL). IBL dikenal sebagai *lazy learner* atau *pembelajaran malas* karena IBL tidak melakukan proses belajar (dari data latih) melainkan secara langsung melakukan klasifikasi berdasarkan jumlah data tetangga yang memiliki jarak terdekat dengan pola masukan. Metode kNN bekerja dengan cara mencari sejumlah k pola (diantara semua pola latih yang ada di semua kelas) yang terdekat dengan pola masukan, kemudian menentukan kelas keputusan berdasarkan jumlah pola terbanyak diantara k pola tersebut (*voting*) (Suyanto, 2018).

2.2.5. Pathfinding

Pathfinding umumnya bertujuan untuk mencari rute terpendek di antara dua titik/posisi. Contoh penerapannya yaitu seperti *transit planning*, *telephone traffic routing*, *maze navigation*, dan *robot path planning*. Dengan naiknya kebutuhan industri game, *pathfinding* menjadi begitu populer. Game seperti *role-playing* dan *real-time strategy* biasanya mempunyai karakter yang diutus dalam suatu misi dari lokasi saat itu menuju lokasi yang telah ditentukan. Masalah paling umum *pathfinding* dalam video game yaitu bagaimana caranya untuk menghindari hambatan atau dinding secara cerdas serta mencari jalur yang paling efisien

melewati berbagai medan. Solusi awal untuk masalah *pathfinding* dalam *computer games*, seperti *depth first search*, *iterative deepening*, *breadth first search*, *dijkstra algorithm*, *best first search*, *A* algorithm*, dan *iterative deepening A**. (Xiao Cui, 2011)

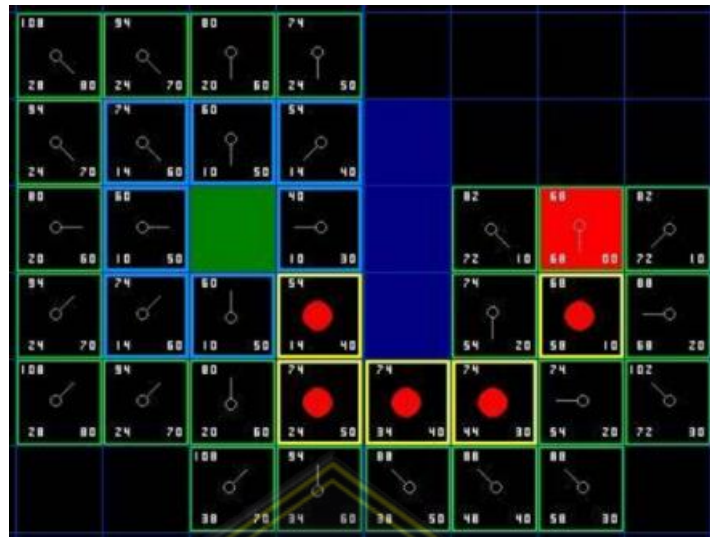


Gambar 2.2 Ilustrasi *pathfinding* (Daniel, 2012)

2.2.6. Algoritma *pathfinding* A* (A-star)

A* adalah salah satu teknik terbaik dan populer yang digunakan dalam *pathfinding* dan *graph travelsals*. Secara informal, algoritma pencarian A*, tidak seperti teknik traversal lainnya, algoritma ini memiliki "otak". Artinya adalah benar-benar algoritma pintar yang memisahkannya dari algoritma konvensional lainnya. (Cahyadi dkk., 2017) Dengan fungsi *heuristic*nya, kinerja algoritma A* dalam pengecekan node terbukti lebih efektif jika dibanding algoritma pencarian lainnya seperti *Dijkstra* dan *Floyd Warshall*.

Ketika A* melintasi *graph*, ia mengikuti jalur dengan biaya terendah yang diketahui, menjaga antrian prioritas yang diurutkan dari segmen jalur alternatif di sepanjang jalan. Jika, pada titik mana pun, segmen jalur yang dilalui memiliki biaya lebih tinggi daripada segmen jalur yang ditemui lainnya, segmen jalur tersebut akan meninggalkan segmen jalur berbiaya lebih tinggi dan sebaliknya melintasi segmen jalur berbiaya lebih rendah. Proses ini berlanjut hingga tujuan tercapai.



Gambar 2.2 Ilustrasi pathfinding algoritma A* (Lester, 2005)



BAB III

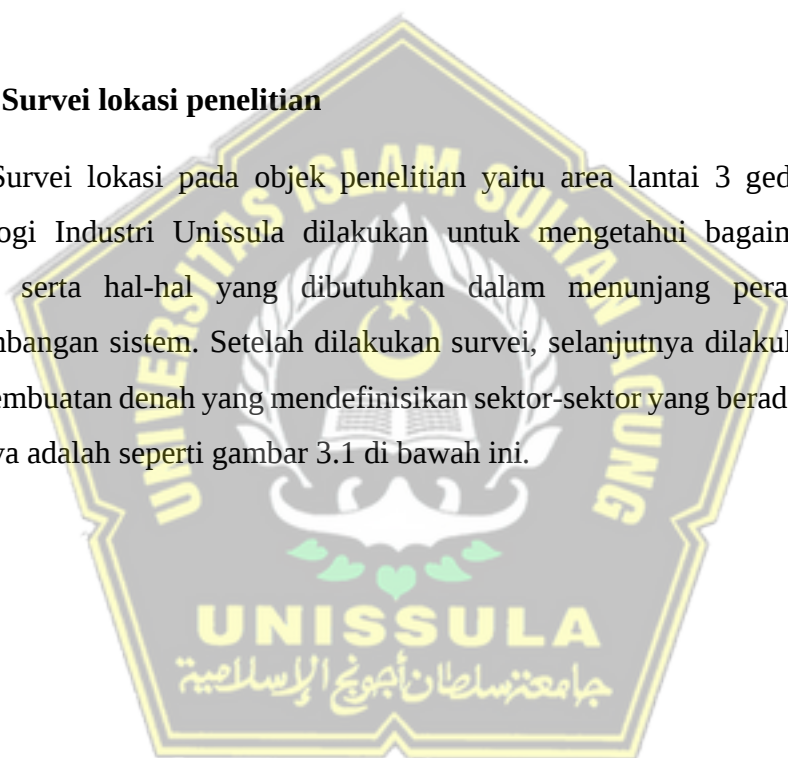
METODE PENELITIAN

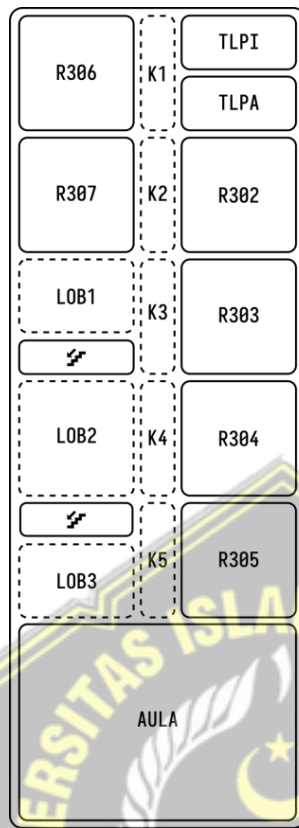
3.1. Metode Penelitian

Metode penelitian yang dipakai sebagai pendekatan untuk mendapatkan solusi dari penelitian tugas akhir ini yaitu sebagai berikut ini.

3.1.1. Survei lokasi penelitian

Survei lokasi pada objek penelitian yaitu area lantai 3 gedung Fakultas Teknologi Industri Unissula dilakukan untuk mengetahui bagaimana kondisi, medan, serta hal-hal yang dibutuhkan dalam menunjang perancangan dan pengembangan sistem. Setelah dilakukan survei, selanjutnya dilakukan pemetaan serta pembuatan denah yang mendefinisikan sektor-sektor yang berada di dalamnya. Hasilnya adalah seperti gambar 3.1 di bawah ini.





Keterangan :

- R302 = Ruang 302
- R303 = Ruang 303
- R304 = Ruang 304
- R305 = Ruang 305
- R306 = Ruang 306
- R307 = Ruang 307
- LOB1 = Lobi 1
- LOB2 = Lobi 2
- LOB3 = Lobi 3
- AULA = Ruang Aula
- TLPI = Toilet Putri
- TLPA = Toilet Putra
- K1 = Koridor 1
- K2 = Koridor 2
- K3 = Koridor 3
- K4 = Koridor 4
- K5 = Koridor 5

Gambar 3.1 Pemetaan area lantai 3 gedung FTI Unissula

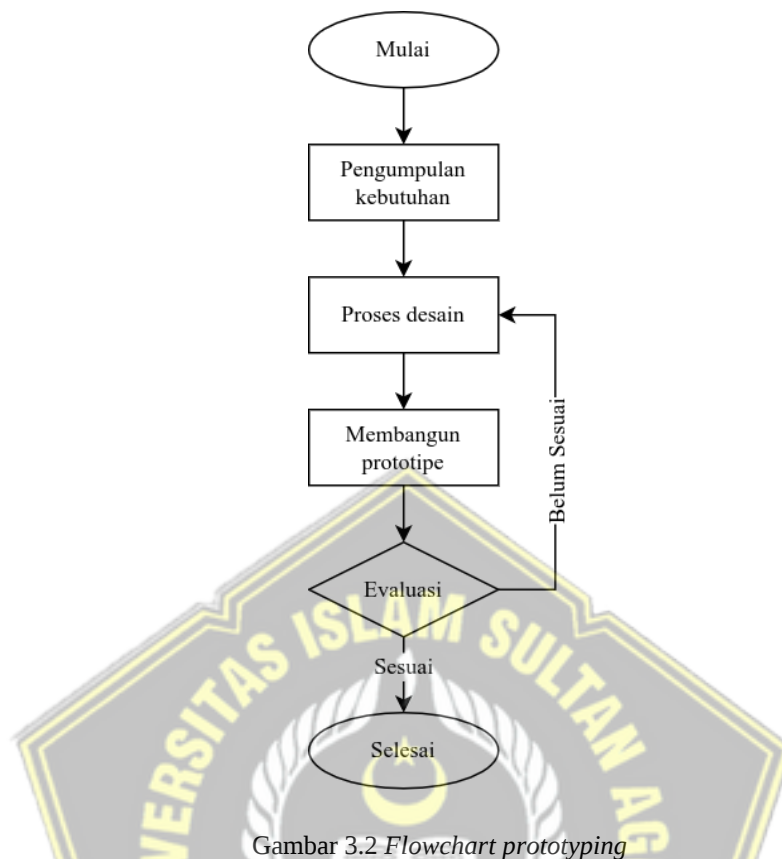
3.1.2. Studi pustaka

Studi pustaka ini merupakan kegiatan mengumpulkan, memahami, mempelajari serta menerapkan berbagai materi dan metode yang dibutuhkan dalam sebuah penelitian. Sumber yang digunakan diantaranya dapat diperoleh dari buku, jurnal, *website*, dan dari berbagai sumber yang berhubungan dengan masalah yang sedang diteliti.

3.1.3. Metode pengembangan sistem

Prototyping dimulai dengan pengumpulan kebutuhan, melibatkan pengembang dan pengguna sistem untuk menentukan tujuan, fungsi, dan kebutuhan operasional sistem. Langkah-langkah dalam *prototyping* adalah sebagai berikut :

1. Pengumpulan kebutuhan
2. Proses desain yang cepat
3. Membangun prototipe
4. Evaluasi dan perbaikan



Gambar 3.2 *Flowchart prototyping*

3.2. Gambaran Sistem

Sistem yang dirancang serta dibangun peneliti merupakan sebuah prototipe sistem penentuan posisi dan navigasi yang diharapkan mampu beroperasi di dalam suatu ruangan (*indoor*) dengan baik. Sistem yang dibangun berlokasi pada lantai 3 gedung Fakultas Teknologi Industri Unissula sebagai studi kasus objek penelitian. Sistem memiliki dua fungsi utama yaitu pemposisian dan navigasi. Sebuah teknologi pemposisian pada dasarnya membutuhkan dua komponen utama yaitu *transmitter* dan komponen penghitung. Pada sistem yang dibangun peneliti memanfaatkan *wi-fi access point* sebagai *transmitter*, serta *smartphone* android sebagai komponen penghitung sekaligus sebagai perangkat dimana user berinteraksi dengan sistem.

Pemposisian merupakan tahap awal sebelum sistem dapat menjalankan fungsi navigasi. Pada tahap pemposisian, sistem melakukan proses perhitungan untuk mengetahui dimana posisi *user* saat ini berada di area lantai 3. Setelah posisi

diketahui, selanjutnya user dapat memilih lokasi yang ingin dituju. Lokasi yang telah di-*input*kan oleh user, kemudian diproses oleh sistem untuk dicari rute efektif dari posisi awal menuju ke lokasi tujuan. Rute yang dihasilkan selanjutnya akan disajikan kepada *user*.



3.3. Komponen Pemposisian

Terdapat dua komponen utama dalam sebuah sistem pemposisian yaitu komponen pemancar sinyal (*transmitter*) dan komponen penghitung.

1) Komponen pemancar sinyal (*transmitter*)

Pada kasus pemposisian dalam ruangan (*indoor*), komponen *transmitter* dapat memanfaatkan teknologi telekomunikasi *wireless* (nirkabel) seperti *infrared*, *bluetooth*, atau *wi-fi*. Namun untuk dapat menjangkau skala area yang luas, teknologi *wi-fi* merupakan pilihan yang lebih efektif. Perangkat *transmitter* dalam jaringan *wi-fi* biasa disebut dengan *access point*.



Gambar 3.3 Perangkat *wi-fi access point*

2) Komponen penghitung

Sedangkan untuk komponen penghitung, peneliti memanfaatkan *smartphone* dengan android sebagai sistem operasinya. *Smartphone* android pada umumnya mampu menjalin sebuah interkoneksi melalui jaringan *wi-fi*, sehingga telah memenuhi syarat untuk dijadikan sebagai komponen penghitung. Sistem operasi android mempunyai kelas yang bernama “*Wi-fiManager*”. Kelas ini menyediakan

API (*Application Programming Interface*) untuk mengelola semua aspek konektivitas dengan *wi-fi*.

3.4. Pemposisian

Sistem yang dirancang pada penelitian ini meingimplementasikan teknik *fingerprint positioning* untuk dapat melakukan pemposisian. Berikut ini dijelaskan proses-proses yang terdapat pada teknik *fingerprint positioning*.

3.4.1. Mekanisme pemposisian

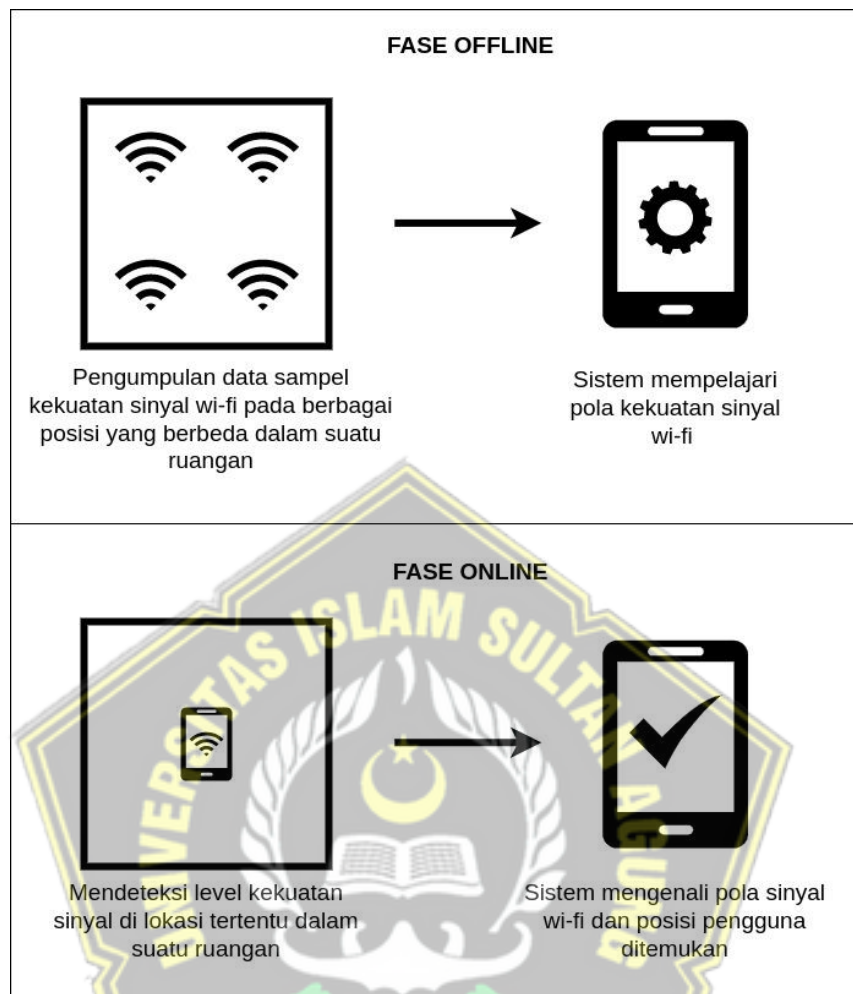
Alasan mengapa teknik ini disebut dengan teknik *fingerprint positioning* adalah karena teknik pemposisian tersebut mirip dengan cara kerja sistem pemindaian sidik jari atau biasa disebut dengan *fingerprint*. Sistem pemindaian sidik jari biasanya digunakan untuk melakukan identifikasi otomatis menggunakan pola sidik jari. Terdapat dua fase yaitu fase *offline* dan *online*. Awalnya pada fase *offline*, perlu dilakukan pengumpulan data sidik jari seseorang selama beberapa kali dengan berbagai posisi yang berbeda agar sistem dapat mempelajari pola sidik jari tersebut. Baru kemudian pada fase *online*, sistem akan dapat mengenali sidik jari seseorang dalam sekali pemindaian.





Gambar 3.4 Mekanisme fase *offline* dan fase *online* sistem pemindaian sidik jari

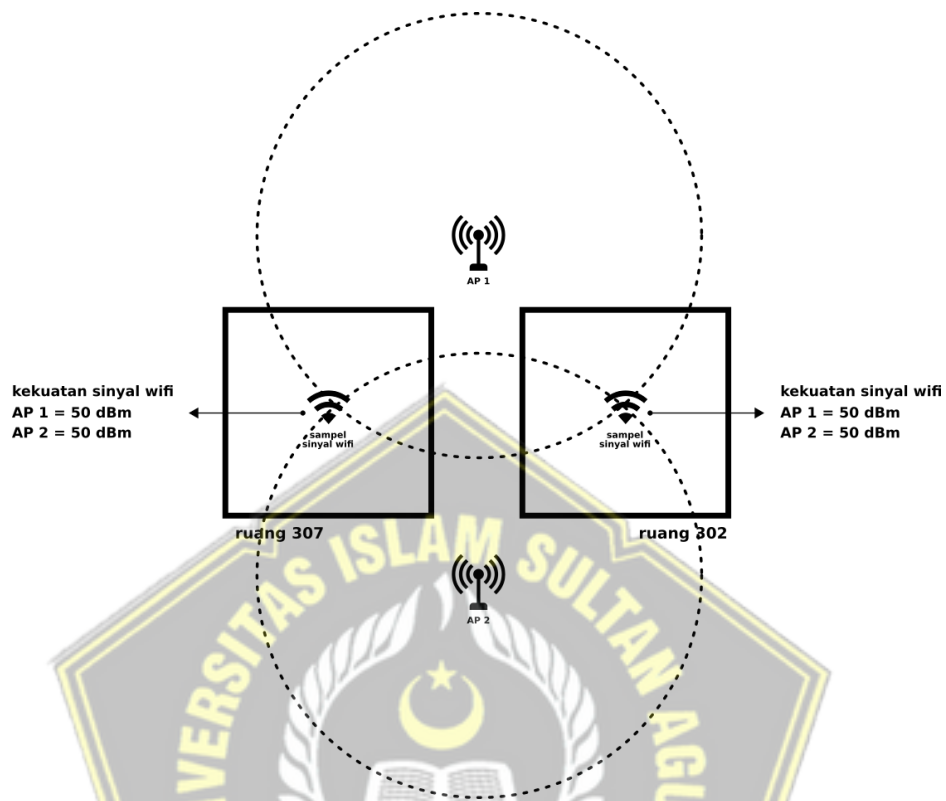
Cara kerja teknik *fingerprint positioning* serupa dengan sistem pemindaian sidik jari, hanya saja objek yang digunakan untuk mengidentifikasi sebuah lokasi adalah level kekuatan sinyal *wi-fi* yang terdeteksi di sekitar area lokasi tersebut.



Gambar 3.5 Mekanisme fase *offline* dan fase *online* pada teknik *fingerprint positioning*

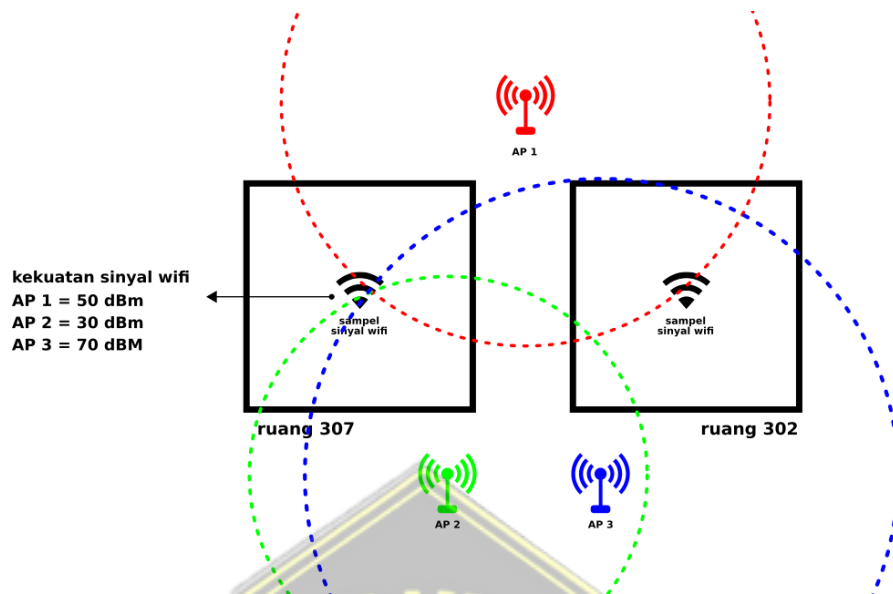
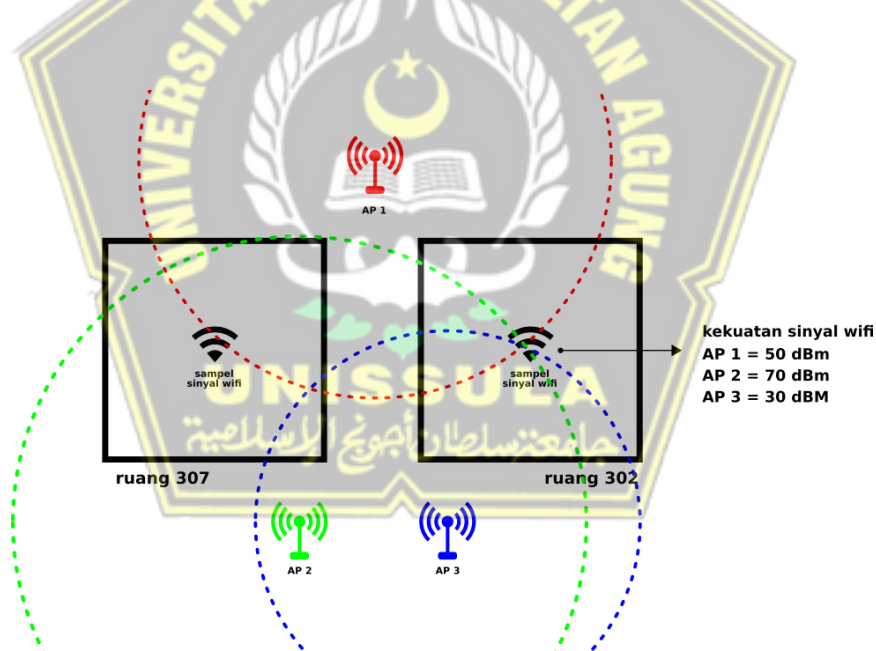
3.4.2. Mekanisme 3 buah *transmitter*

Transmitter atau *wi-fi access point* berfungsi sebagai alat pemancar sinyal *wi-fi*, sedangkan sinyal *wi-fi* merupakan parameter yang digunakan untuk menentukan posisi. Pada penelitian ini, sistem pemosisian akan dirancang menggunakan 3 buah *transmitter*. Sistem wajib memiliki minimal 3 buah *transmitter* agar setiap sampel sinyal *wi-fi* yang diambil memiliki data yang unik. Jika hanya 2 atau 1 *transmitter* yang digunakan, maka yang terjadi adalah banyak sampel yang memiliki data sinyal yang sama namun lokasi/ruangannya berbeda, dan akan mengakibatkan sistem tidak dapat melakukan pemosisian.



Gambar 3.6 Pemposisian dengan 2 buah *transmitter*

Pada gambar di atas dapat diketahui bahwa sampel di ruang 307 dan ruang 302 memiliki data yang sama, sehingga sistem tidak dapat melakukan pemposisian karena syarat utamanya adalah data sampel yang dimiliki haruslah unik. Hal tersebut mirip dengan konsep pemindaian sidik jari yang setiap orang mempunyai sidik jari yang berbeda, sehingga dapat digunakan sebagai objek untuk diidentifikasi.

Gambar 3.7 Pemosisian dengan 3 buah *transmitter* (1)Gambar 3.8 Pemosisian dengan 3 buah *transmitter* (2)

Dapat diketahui pada kedua gambar di atas, bahwa sistem pemosisian dengan menggunakan 3 buah *transmitter* menghasilkan sampel *wi-fi* yang unik. Tidak seperti sebelumnya ketika menggunakan hanya 2 buah *transmitter* dimana

masih terdapat data sampel yang sama. Jika ingin mendapatkan hasil pemosisian yang lebih baik, dapat digunakan lebih dari 3 buah *transmitter*.

3.4.3. Fase *offline* pemosisian

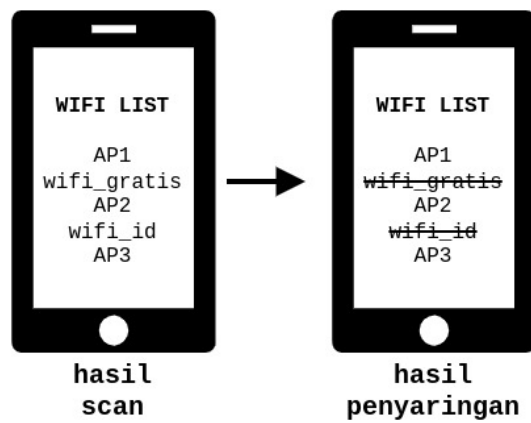
Pada fase *offline* ini intinya adalah melakukan pengumpulan data sampel sinyal *wi-fi* pada setiap titik lokasi yang berada di lantai 3. Data sampel yang diambil berupa level kekuatan sinyal *wi-fi* yang diterima oleh *smartphone* android atau biasa disebut dengan RSS (*Received Signal Strength*). Seluruh data sampel yang diambil kemudian akan disimpan ke dalam *database*. Berikut ini dijelaskan simulasi proses-proses yang dilakukan pada fase *offline*.

1. Pilih satu titik lokasi pada sebuah ruangan, lalu lakukan *scanning* jaringan *wi-fi* dengan *smartphone android* menggunakan aplikasi yang telah dibangun.



Gambar 3.9 Contoh pengambilan data sampel *wi-fi* di satu titik di ruang 302

2. Aplikasi akan melakukan pemfilteran jaringan *wi-fi*, karena sangat memungkinkan terdapat jaringan *wi-fi* lain yang terdeteksi oleh *smartphone* dan bukan bagian dari kebutuhan sistem.



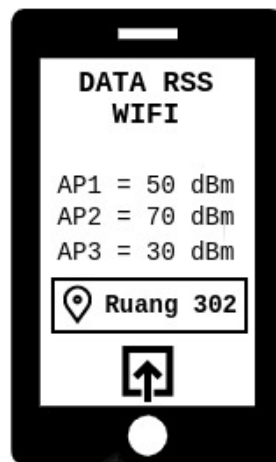
Gambar 3.10 Ilustrasi pemfilteran jaringan wi-fi

3. Jaringan wi-fi yang telah terfilter, kemudian diambil informasi level kekuatan sinyalnya.



Gambar 3.11 Ilustrasi informasi daa RSS 3 transmitter yang telah diambil

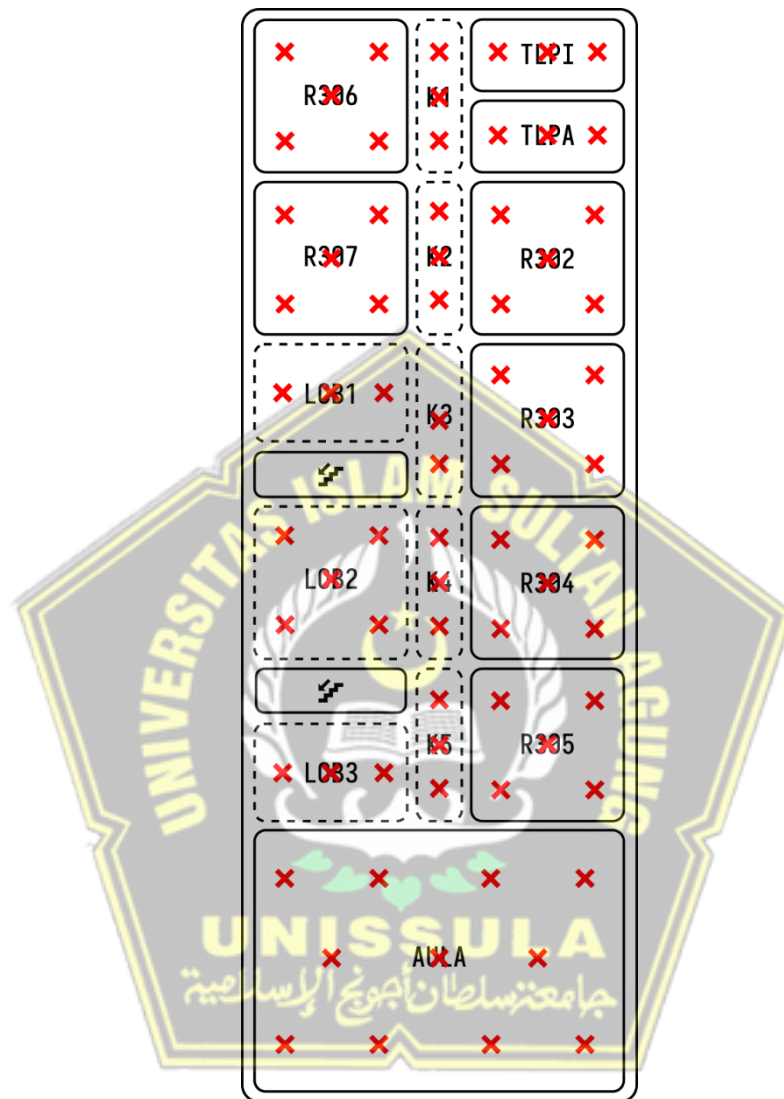
4. Kemudian data sampel wi-fi tersebut diberi label dimana lokasi/ruangan diambil dan disimpan ke dalam *database*.



Gambar 3.12 Ilustrasi pemberian label lokasi pada data sampel RSS yang telah diambil



5. Setelah selesai, kemudian lanjut menambahkan data sampel sinyal *wi-fi* di titik lain hingga semua lokasi terliputi.



Gambar 3.13 Ilustrasi pengambilan sampel *wi-fi* di seluruh lokasi pada lantai 3 gedung FTI

3.4.4. Fase online pemposisian

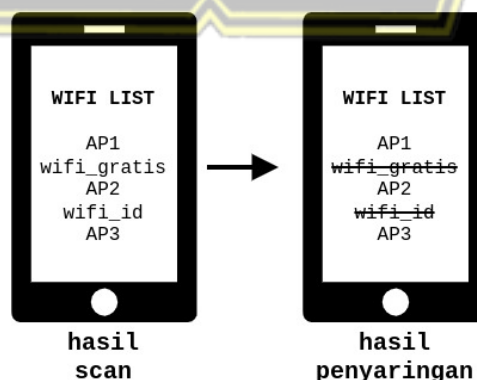
Setelah seluruh data sampel *wi-fi* di seluruh titik lokasi lantai 3 telah terkumpul, maka sistem sudah siap untuk memasuki fase *online* pemposisian. Pada fase ini posisi dari user dapat diketahui. Sistem akan melakukan perhitungan tingkat kemiripan antara sampel *wi-fi* dari user dengan seluruh data sampel yang tersimpan pada *database*. Berikut ini dijelaskan proses-proses yang terjadi pada fase *online*.

1. User berada di suatu lokasi/ruang di area lantai 3 gedung FTI Unissula. Untuk mengetahui posisi saat ini, pertama aplikasi sistem akan melakukan *scanning* jaringan *wi-fi*.



Gambar 3.14 Ilustrasi aplikasi sistem melakukan scanning jaringan *wi-fi* pada fase online

2. Selanjutnya aplikasi akan menyeleksi jaringan *wi-fi* yang dibutuhkan berdasarkan data SSID dari data jaringan *wi-fi* yang terdeteksi.



Gambar 3.15 Ilustrasi penyeleksian jaringan *wi-fi* fase online

3. Selanjutnya data level kekuatan sinyal dari jaringan *wi-fi* yang terseleksi akan digunakan sebagai data masukan.



Gambar 3.16 Informasi level kekuatan sinyal jaringan *wi-fi* sebagai data masukan

4. Data masukan yang didapatkan selanjutnya akan dikalkulasi dan ditentukan posisinya menggunakan algoritma *k-NN*. Proses bagaimana algoritma *k-NN* menentukan posisi dijelaskan pada sub poin 3.4.4.

3.4.5. Penentuan posisi dengan algoritma *k-NN*

Pada poin ini akan dijelaskan bagaimana proses penentuan posisi fase *online* dengan mengimplementasikan algoritma *k-NN*. Sebelumnya pada fase *offline* telah terkumpul data sampel *wi-fi* dari setiap titik lokasi lantai 3 dan telah tersimpan di *database*. Contoh data sampel pada database adalah seperti tabel berikut ini.

Tabel 3.1 Contoh database data sampel *wi-fi*

Database data sampel <i>wi-fi</i>				
Lokasi	Sampel <i>wi-fi</i>	RSS		
		AP 1	AP 2	AP 3
Ruang 302	Sampel 1	- 20 dBm	- 30 dBm	- 40 dBm
	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm

	Sampel 3	- 19 dBm	- 30 dBm	- 46 dBm
	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm
	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm
Ruang 303	Sampel 1	- 30 dBm	- 22 dBm	- 30 dBm
	Sampel 2	- 32 dBm	- 22 dBm	- 30 dBm
Dst.	Dst.	Dst.	Dst.	Dst.

Kemudian pada fase *online*, telah didapatkan juga data masukan yang diambil dari titik lokasi pengguna. Contoh data masukan adalah seperti tabel berikut ini.

Tabel 3.2 Contoh data masukan

Data masukan		
AP1	AP2	AP3
- 50 dBm	- 30 dBm	- 70 dBm

Selanjutnya untuk mengetahui posisi dari data masukan tersebut adalah dengan cara mengetahui tingkat kemiripan antara data masukan dengan setiap data sampel yang tersimpan di *database* data sampel. Tingkat kemiripan data diukur berdasarkan jarak kedekatan antar data. Sehingga semakin kecil jarak kedekatan antar data, maka tingkat kemiripannya semakin tinggi. Jarak kedekatan antar data dihitung menggunakan persamaan *euclidean*.

$$E(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

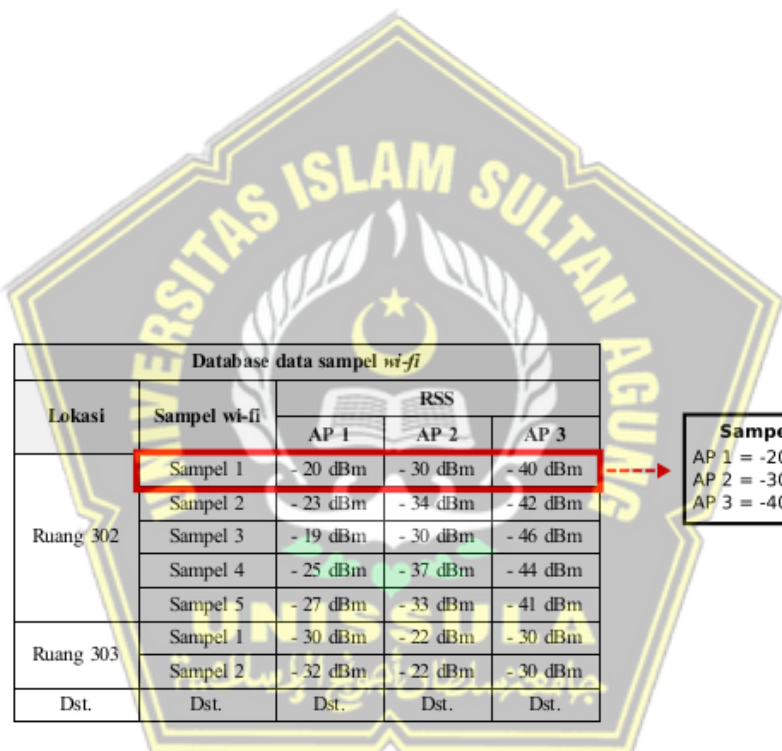
Keterangan :

- x sebagai data masukan
- y sebagai data sampel

- $(x_1, x_2, \dots, x_n)$ dan $(y_1, y_2, \dots, y_n)$ sebagai nilai level sinyal *wi-fi access point* dari (AP 1, AP2, ..., AP n)

Berikut ini merupakan contoh langkah-langkah bagaimana algoritma *k-NN* menentukan posisi dari sebuah data masukan.

1. Mulai dengan memilih sebuah data sampel dari database data sampel *wi-fi* yang telah dikumpulkan pada fase offline sebelumnya.

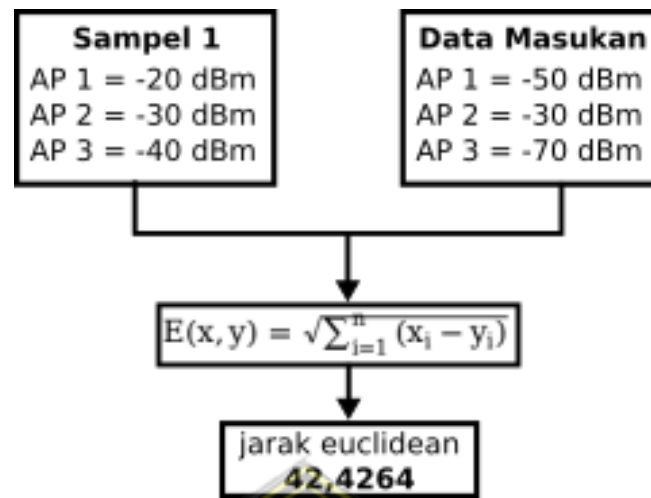


Database data sampel <i>wi-fi</i>				
Lokasi	Sampel <i>wi-fi</i>	RSS		
		AP 1	AP 2	AP 3
Ruang 302	Sampel 1	- 20 dBm	- 30 dBm	- 40 dBm
	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm
	Sampel 3	- 19 dBm	- 30 dBm	- 46 dBm
	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm
	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm
Ruang 303	Sampel 1	- 30 dBm	- 22 dBm	- 30 dBm
	Sampel 2	- 32 dBm	- 22 dBm	- 30 dBm
Dst.	Dst.	Dst.	Dst.	Dst.

Sampel 1
 AP 1 = -20 dBm
 AP 2 = -30 dBm
 AP 3 = -40 dBm

Gambar 3.17 Ilustrasi pemilihan data sampel *wi-fi* pada algoritma *k-NN*

2. Lakukan perhitungan antara data sampel dan data masukan menggunakan persamaan euclidean.



Gambar 3.18 Ilustrasi perhitungan jarak euclidean data sampel 1 dengan data masukan

Pada perhitungan di atas dihasilkan nilai jarak euclidean antara data sampel 1 dan data masukan yaitu 42, 4264.



3. Ulangi perhitungan untuk seluruh data sampel yang ada pada database.

Tabel 3.3 Database data sampel *wi-fi* dengan jarak euclidean antara data sampel dan data masukan

Database data sampel <i>wi-fi</i>					
Lokasi	Sampel <i>wi-fi</i>	RSS			Jarak Euclidean
		AP 1	AP 2	AP 3	
Ruang 302	Sampel 1	- 20 dBm	- 30 dBm	- 40 dBm	42.426407
	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm	39.10243
	Sampel 3	- 19 dBm	- 30 dBm	- 46 dBm	39.20459
	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm	36.742348
	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm	37.134888
Ruang 303	Sampel 1	- 30 dBm	- 22 dBm	- 30 dBm	41.53312
	Sampel 2	- 32 dBm	- 22 dBm	- 30 dBm	44.586994
dst.	dst.	dst.	dst.	dst.	dst.

4. Sortir database offline secara menurun berdasarkan jarak euclidean.

Tabel 3.4 Database data sampel *wi-fi* diurutkan secara menurun berdasarkan nilai jarak euclidean

Database data sampel <i>wi-fi</i>					
Lokasi	Sampel <i>wi-fi</i>	RSS			Jarak Euclidean
		AP 1	AP 2	AP 3	
Ruang 302	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm	36.742348
Ruang 302	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm	37.134888
Ruang 302	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm	39.10243
Ruang 302	Sampel 3	- 19 dBm	- 30 dBm	- 44 dBm	39.20459
Ruang 303	Sampel 1	- 30 dBm	- 20 dBm	- 35 dBm	41.53312
Ruang 302	Sampel 1	- 20 dBm	- 30 dBm	- 40 dBm	42.426407
Ruang 303	Sampel 2	- 32 dBm	- 22 dBm	- 30 dBm	44.586994
dst.	dst.	dst.	dst.	dst.	dst.

5. Seleksi sampel yang memiliki jarak euclidean paling dekat dengan data masukan berdasarkan variabel “k”. Nilai dari variabel “k” dapat disesuaikan sesuai dengan kebutuhan sistem.



Tabel 3.5 Penyeleksian sejumlah “k” (k=5) data sampel yang memiliki jarak euclidean terkecil

Database data sampel wi-fi (“k”=5)					
Lokasi	Sampel wi-fi	RSS			Jarak Euclidean
		AP 1	AP 2	AP 3	
Ruang 302	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm	36.742348
Ruang 302	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm	37.134888
Ruang 302	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm	39.10243
Ruang 302	Sampel 3	- 19 dBm	- 30 dBm	- 44 dBm	39.20459
Ruang 303	Sampel 1	- 30 dBm	- 20 dBm	- 35 dBm	41.53312
Ruang 302	Sampel 1	- 20 dBm	- 30 dBm	- 40 dBm	42.426407
Ruang 303	Sampel 2	- 32 dBm	- 22 dBm	- 30 dBm	44.586994
dst.	dst.	dst.	dst.	dst.	dst.

6. Lakukan *voting* berdasarkan lokasi yang paling sering muncul pada data yang telah terseleksi sebelumnya.

Tabel 3.6 Proses *voting* pada sejumlah (k=5) data sampel

Database data sampel wi-fi					
Lokasi	Sampel wi-fi	RSS			Jarak Euclidean
		AP 1	AP 2	AP 3	
Ruang 302	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm	36.742348
Ruang 302	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm	37.134888
Ruang 302	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm	39.10243
Ruang 302	Sampel 3	- 19 dBm	- 30 dBm	- 44 dBm	39.20459
Ruang 303	Sampel 1	- 30 dBm	- 20 dBm	- 35 dBm	41.53312

7. Klasifikasikan lokasi data masukan berdasarkan hasil dari *voting*.

Database data sampel <i>wi-fi</i>					
Lokasi	Sampel <i>wi-fi</i>	RSS			Jarak Euclidean
		AP 1	AP 2	AP 3	
Ruang 302	Sampel 4	- 25 dBm	- 37 dBm	- 44 dBm	36.742348
Ruang 302	Sampel 5	- 27 dBm	- 33 dBm	- 41 dBm	37.134888
Ruang 302	Sampel 2	- 23 dBm	- 34 dBm	- 42 dBm	39.10243
Ruang 302	Sampel 3	- 19 dBm	- 30 dBm	- 44 dBm	39.20459
Ruang 303	Sampel 1	- 30 dBm	- 20 dBm	- 35 dBm	41.53312

→ Ruang 302 = 4 ✓

→ Ruang 303 = 1 ✗

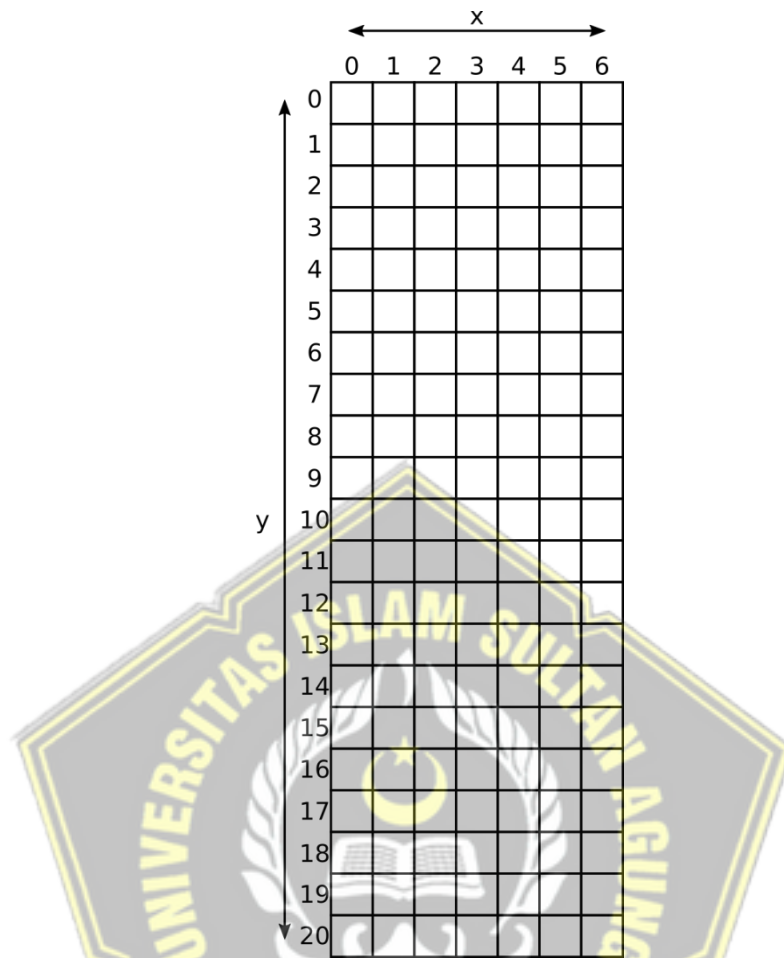
Gambar 3.19 Proses akhir algoritma *k-NN* melakukan *voting*

3.5.Navigasi

Setelah posisi pengguna telah diketahui, maka fungsi navigasi pada sistem siap untuk dijalankan. Pengguna akan memilih lokasi/ruangan yang ingin dituju. Lalu aplikasi akan mencari rute efektif dari posisi awal pengguna yang mengarah ke lokasi/ruangan tujuan. Fungsi untuk pencarian rute pada sistem ini mengimplementasikan algoritma pencarian *A-star* (A^*). Kemudian rute yang dihasilkan akan ditampilkan pada tampilan peta aplikasi.

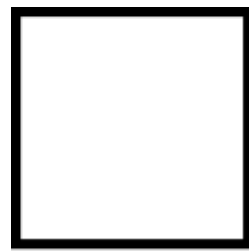
3.5.1. Mekanisme area pencarian rute

Peta navigasi yang dirancang akan dibangun di atas *graph* berbentuk kumpulan sel kotak seperti pada tampilan di bawah ini.

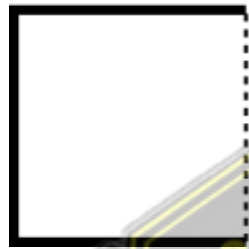


Gambar 3.20 *Graph* peta navigasi

Setiap sel mempunyai atribut dinding di ke-empat sisinya. Atribut dinding tersebut dapat diatur menjadi “ada” atau “tidak ada”. Jika terdapat dinding di salah satu sisi sel, maka sel tidak dapat dilewati melalui sisi tersebut. Namun jika tidak ada dinding, maka sel dapat dilewati melalui sisi tersebut.



dindingAtas = ADA
 dindingKiri = ADA
 dindingBawah = ADA
 dindingKanan = ADA

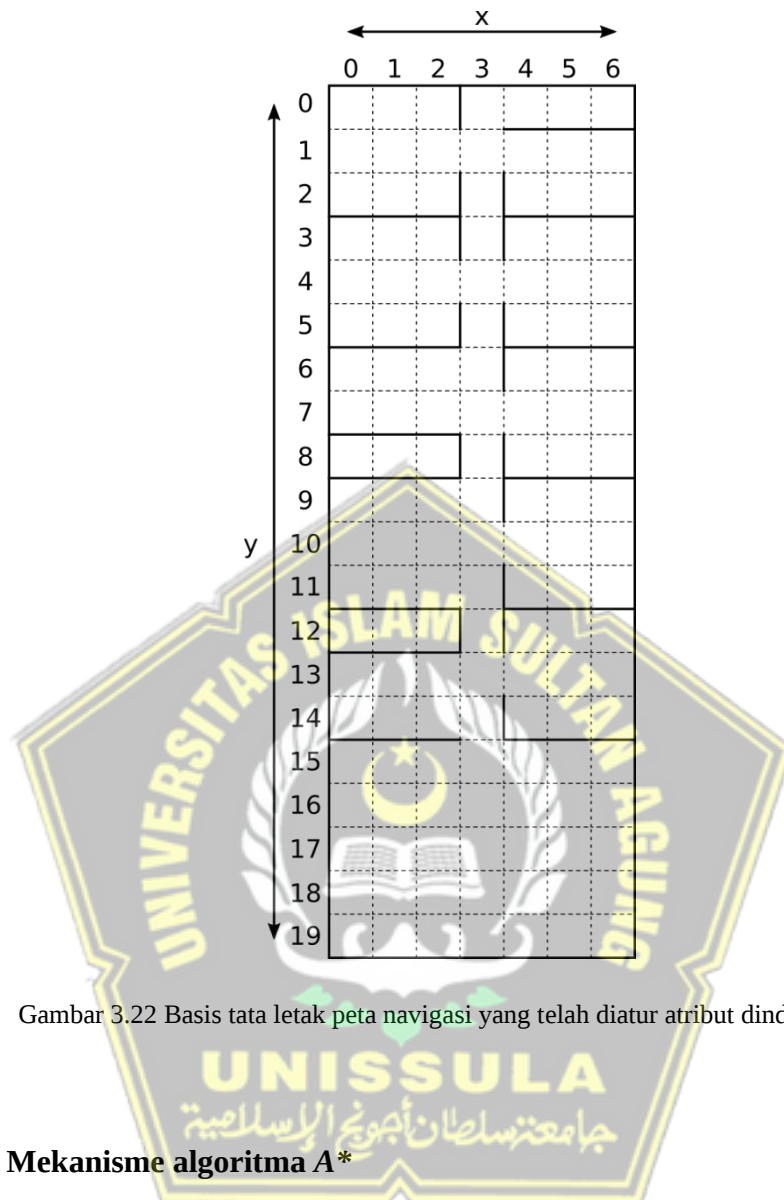


dindingAtas = ADA
 dindingKiri = ADA
 dindingBawah = ADA
 dindingKanan = TIDAK ADA

Gambar 3.21 Atribut dinding pada sel peta navigasi

Selanjutnya, ukuran dan tata letak sel direkayasa sedemikian rupa hingga tampilannya menyerupai peta yang menggambarkan lantai 3 gedung FTI Unissula.





Gambar 3.22 Basis tata letak peta navigasi yang telah diatur atribut dindingnya

3.5.2. Mekanisme algoritma A*

Sistem mengimplementasikan algoritma A* untuk melakukan pencarian rute dari posisi pengguna menuju lokasi tujuan. Terdapat tiga poin utama dalam metode pencarian A*.

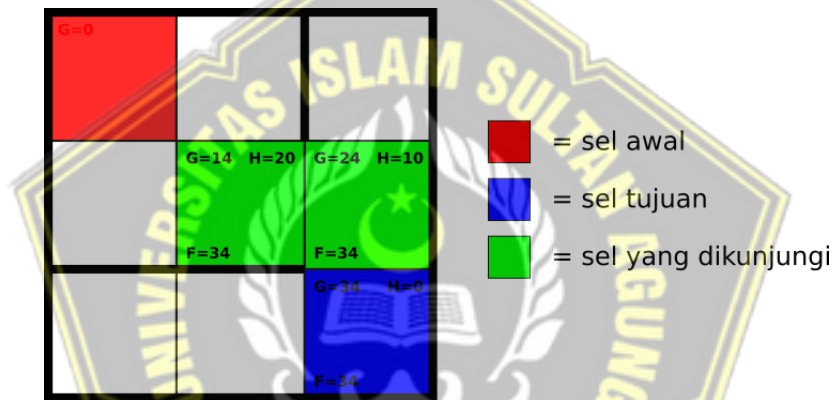
1. Perhitungan skor rute

Pada aturan A*, setiap sel mempunyai atribut F, G, dan H. Dalam upaya mencari rute terpendek, A* akan memilih sel dengan skor F paling kecil sebagai langkah berikutnya, dimana persamaan untuk menghitung skor dari F adalah :

$$F = G + H \quad (3)$$

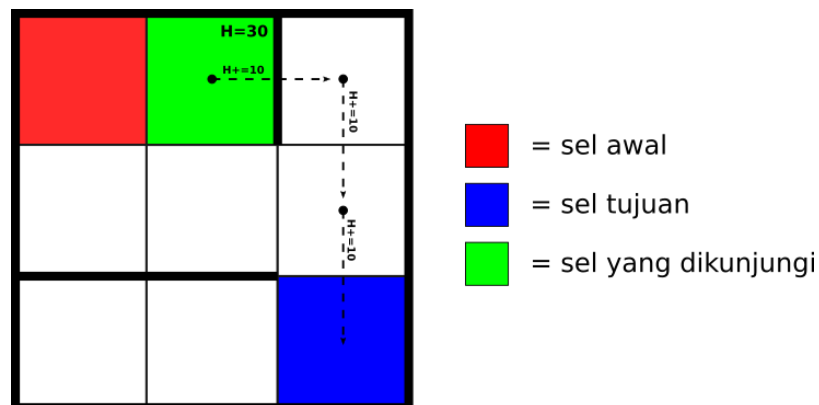
Keterangan :

- G adalah biaya pergerakan yang dikenakan dari titik sel awal menuju sel saat ini. Setiap pergerakan dengan arah vertikal dan horisontal maka skor dari G akan ditambahkan 10 ($G += 10$). Sedangkan untuk pergerakan dengan arah diagonal maka skor G ditambahkan 14 ($G += 14$).

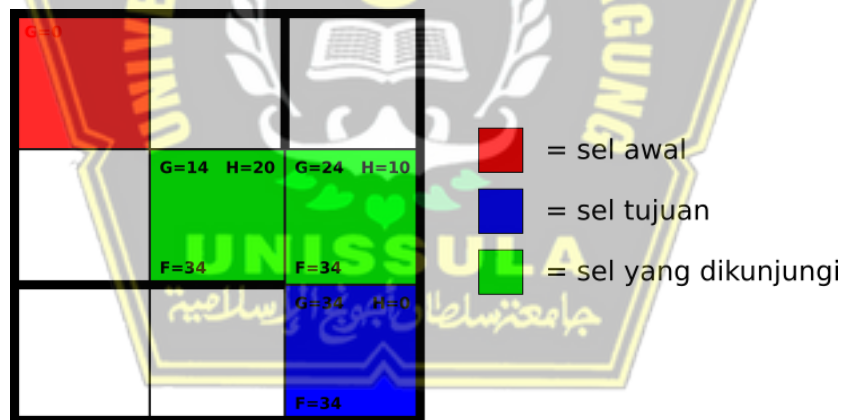


Gambar 3.23 Ilustrasi perhitungan skor G pada sel

- H adalah estimasi/perkiraan biaya pergerakan antara sel saat ini dengan sel tujuan. Skor dari H dihitung menggunakan rumus manhattan distance. Karena hanya sebuah estimasi, perhitungan skor atribut H tidak terlalu mementingkan ketepatan rute dan hanya melakukan perhitungan secara cepat dan sederhana. Perhitungan H tidak mengabaikan ada atau tidaknya dinding, serta hanya menggunakan pergerakan secara vertikal dan horisontal.

Gambar 3.24 Ilustrasi perhitungan skot H pada sel

- F adalah total skor dari sebuah sel.

Gambar 3.25 Ilustrasi perhitungan skor F pada sel

Satu hal yang spesial dari A^* adalah terdapatnya atribut H yang merupakan inisial dari fungsi *heuristic*. Inti dari *heuristic* adalah estimasi/perkiraan. Saat kita sedang berpergian dan tahu dimana lokasi yang ingin kita tuju, maka kita akan senantiasa membuat perkiraan jalan mana yang akan dilalui.



Gambar 3.26 Ilustrasi fungsi heuristic pada algoritma A*

Dengan adanya fungsi tersebut, proses pencarian menjadi lebih efektif karena komputer tidak perlu mengecek semua jalan yang tersedia untuk mencari rute yang tepat menuju lokasi tujuan.

Pada perancangan sistem ini, nilai atribut H pada sel dihitung menggunakan rumus *manhattan distance*, dengan persamaan sebagai berikut :

$$H = |x_1 - x_2| + |y_1 - y_2| \quad (4)$$

Keterangan :

- H adalah skor *heuristic*
- x_1 adalah koordinat x dari sel saat ini
- x_2 adalah koordinat x dari sel tujuan
- y_1 adalah koordinat y dari sel saat ini
- y_2 adalah koordinat y dari sel tujuan

2. Penyeleksian sel

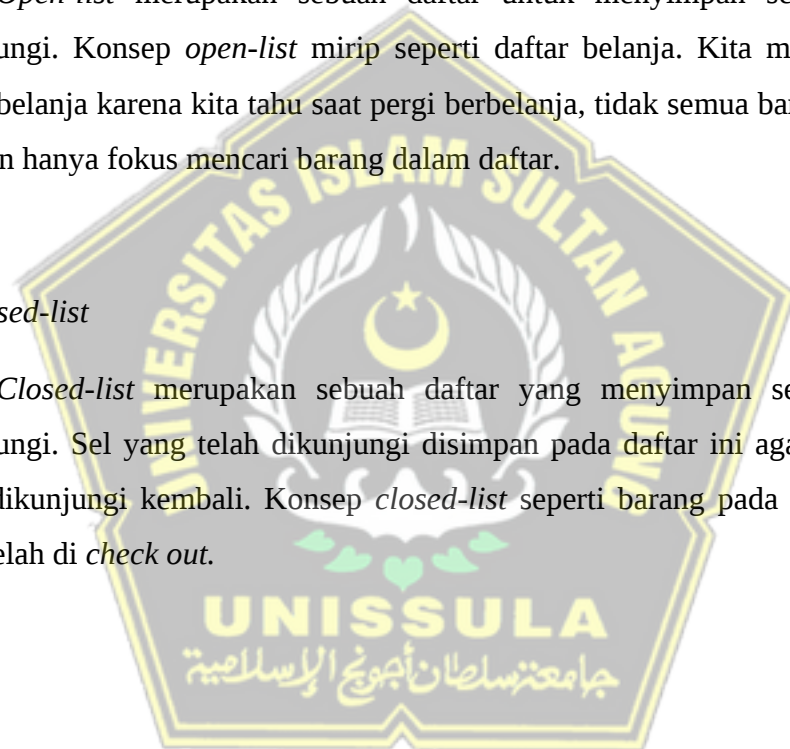
Pada proses penelusuran dalam mencari rute terbaik, sel akan dicek dan dilakukan seleksi, A^* menggunakan dua buah *list* yang sangat krusial dalam proses penyeleksian sel ini, yaitu *open-list* dan *closed-list*.

- *Open-list*

Open-list merupakan sebuah daftar untuk menyimpan sel yang ingin dikunjungi. Konsep *open-list* mirip seperti daftar belanja. Kita menulis sebuah daftar belanja karena kita tahu saat pergi berbelanja, tidak semua barang akan kita beli dan hanya fokus mencari barang dalam daftar.

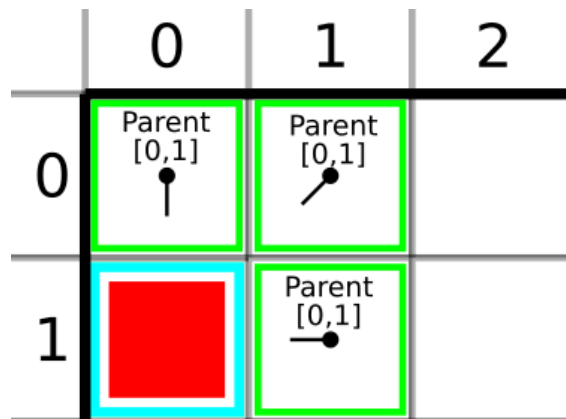
- *Closed-list*

Closed-list merupakan sebuah daftar yang menyimpan sel yang telah dikunjungi. Sel yang telah dikunjungi disimpan pada daftar ini agar sel tersebut tidak dikunjungi kembali. Konsep *closed-list* seperti barang pada daftar belanja yang telah di *check out*.



3. Penelusuran rute

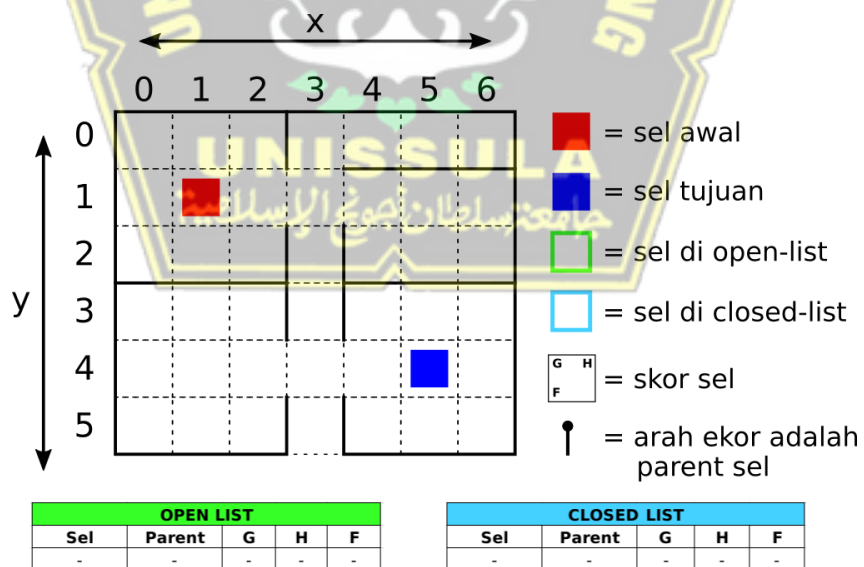
Selain atribut penskoran rute (G, H, dan F), sel memiliki atribut lain yaitu *parent*. *Parent* adalah darimana sebuah sel berasal. Atribut *parent* nantinya akan berguna untuk menampilkan hasil dari rute terpendek yang berhasil ditemukan.



Gambar 3.27 Ilustrasi atribut parent pada sel

3.5.3. Pencarian rute dengan algoritma A*

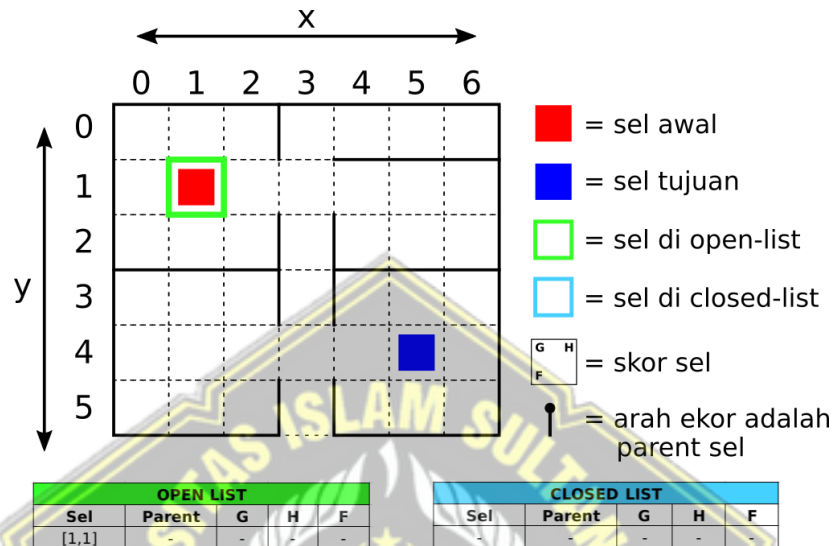
Pada penjelasan di bawah ini akan disimulasikan bagaimana sistem mencari rute dari sel awal menuju sel tujuan dengan menggunakan metode algoritma A*. Berikut ini dimisalkan sel awal berada pada koordinat sel [1,1] dan sel tujuan berada pada koordinat sel [5,4].



Gambar 3.28 Ilustrasi pencarian rute dengan algoritma A* (0)

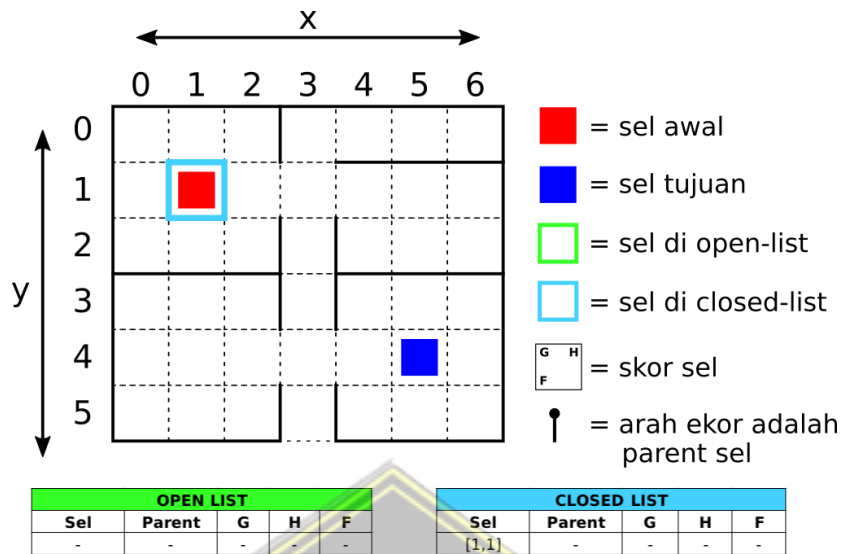
Memulai pencarian rute dengan langkah-langkah berikut ini :

1. Algoritma A* akan selalu melakukan pencarian selama *open-list* tidak kosong. Langkah pertama, masukkan sel awal yaitu sel [1,1] ke dalam *open-list*.



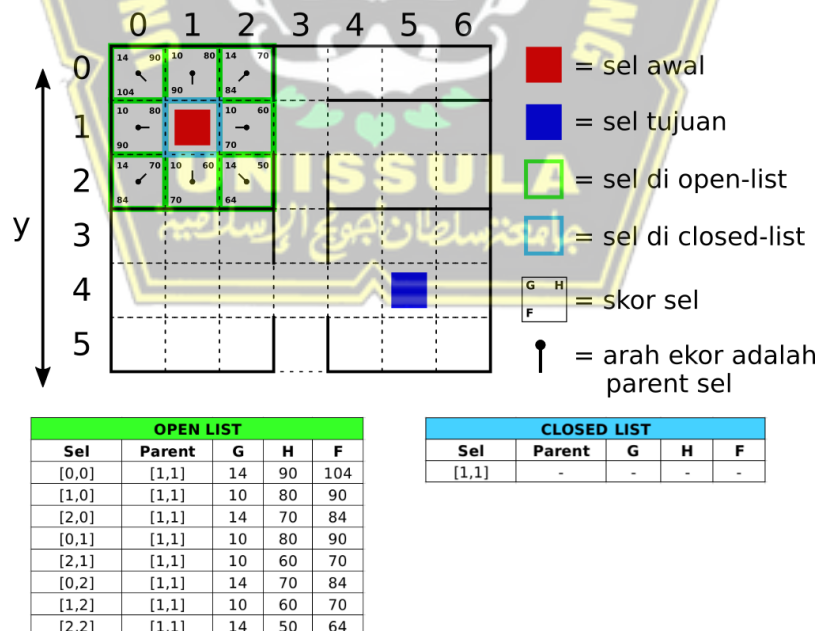
Gambar 3.29 Ilustrasi pencarian rute dengan algoritma A* (1)

2. A* akan memilih sel dengan skor F paling kecil yang tersedia di open-list untuk dikunjungi berikutnya. Sel [1,1] mempunyai skor F paling kecil serta satu-satunya sel yang tersedia di *open-list* saat ini. Keluarkan sel [1,1] dari *open-list* dan pindahkan ke *closed-list*.



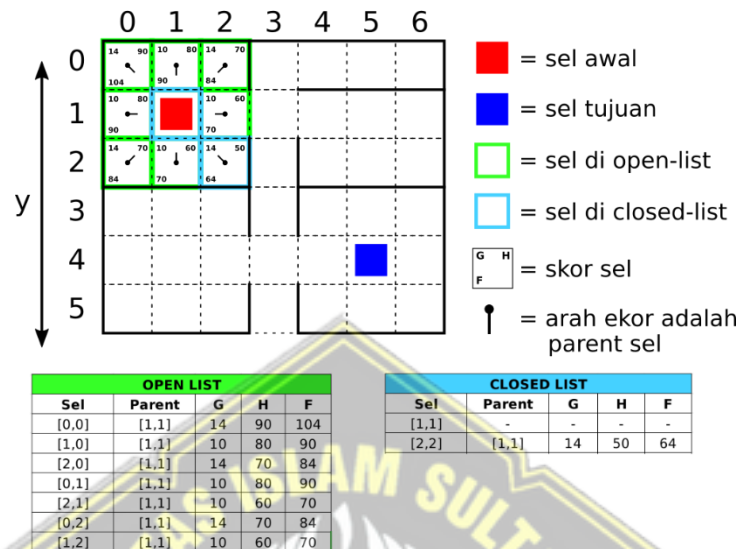
Gambar 3.30 Ilustrasi pencarian rute dengan algoritma A* (2)

3. Selanjutnya adalah mengecek ketersediaan sel tetangga mana yang dapat dikunjungi berikutnya. Masukkan semua sel tetangga ke dalam *open-list*, kecuali bagian sel yang terhalang oleh dinding. Hitung skor masing-masing sel, dan set *parent*-nya dengan sel awal atau sel [1,1].



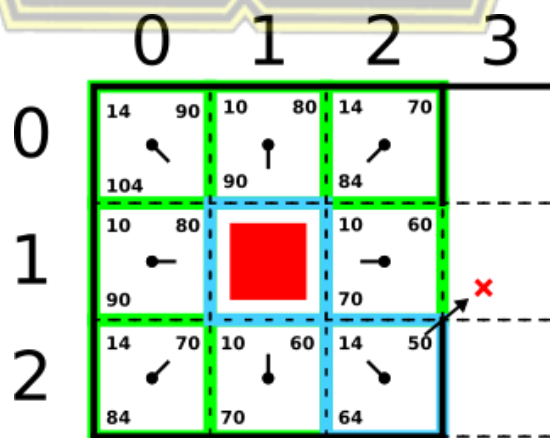
Gambar 3.31 Ilustrasi pencarian rute dengan algoritma A* (3)

4. Pada *open-list*, sel [2,2] mempunyai skor F paling kecil, maka selanjutnya sel [2,2] dipindahkan ke *closed-list*.



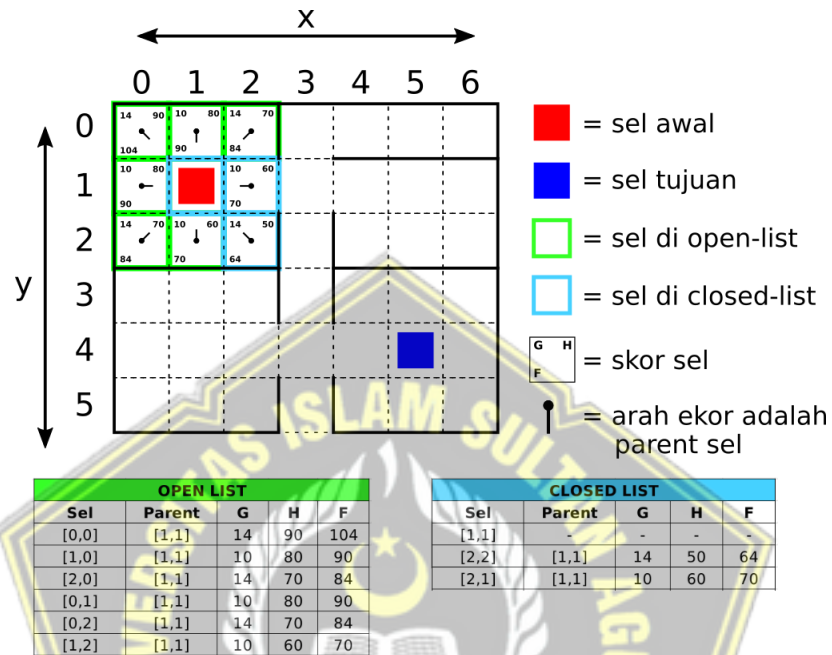
Gambar 3.32 Ilustrasi pencarian rute dengan algoritma A* (4)

5. Cek ketersediaan tetangga dari sel [2,3]. Kita abaikan sel [2,1], [1,2] karena telah berada pada *open-list*, dan sel [1,1] yang telah berada pada *closed-list*. Kita abaikan juga sel yang terhalang oleh dinding. Sel [3,1] yang berada di sisi diagonal masih terhalang oleh dinding maka kita abaikan juga. Maka kita selesai dengan sel [2,3].



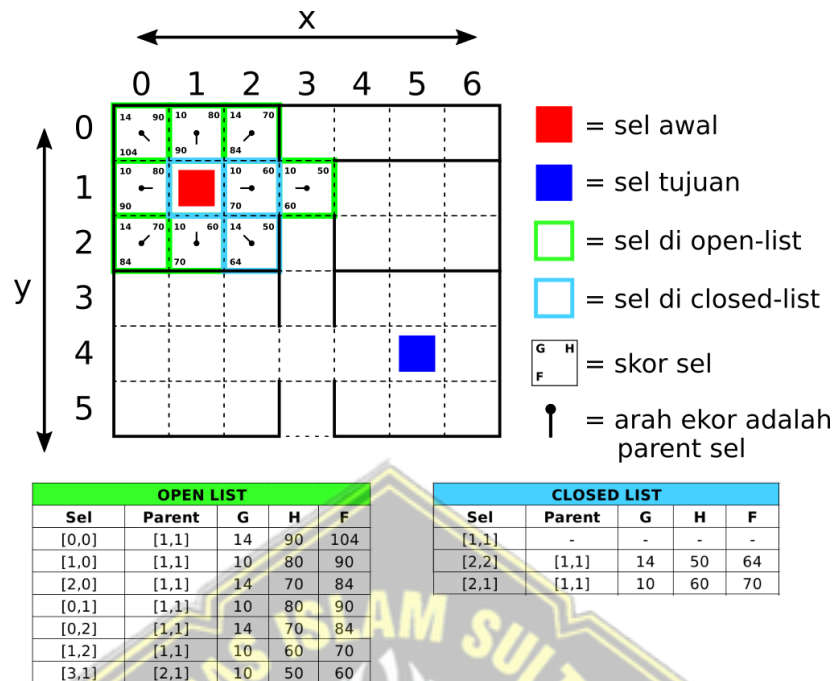
Gambar 3.33 Ilustrasi pencarian rute dengan algoritma A* (5)

6. Selanjutnya pada *open-list*, sel [2,1] memiliki skor F paling kecil maka kita pindahkan ke *closed-list*.



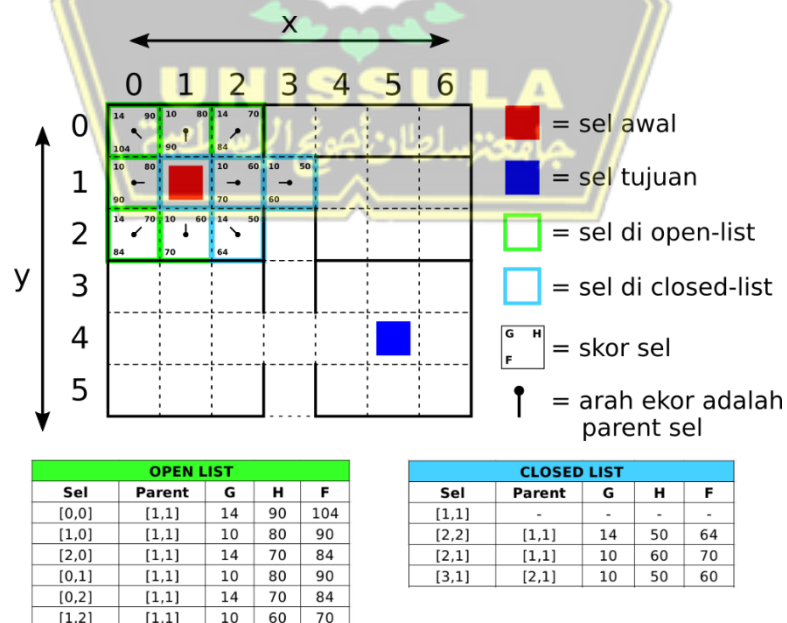
Gambar 3.34 Ilustrasi pencarian rute dengan algoritma A* (6)

7. Cek tetangga dari sel[2,1], dan tersedia sel[3,1]. Hitung skor dan masukkan ke dalam *open-list*.



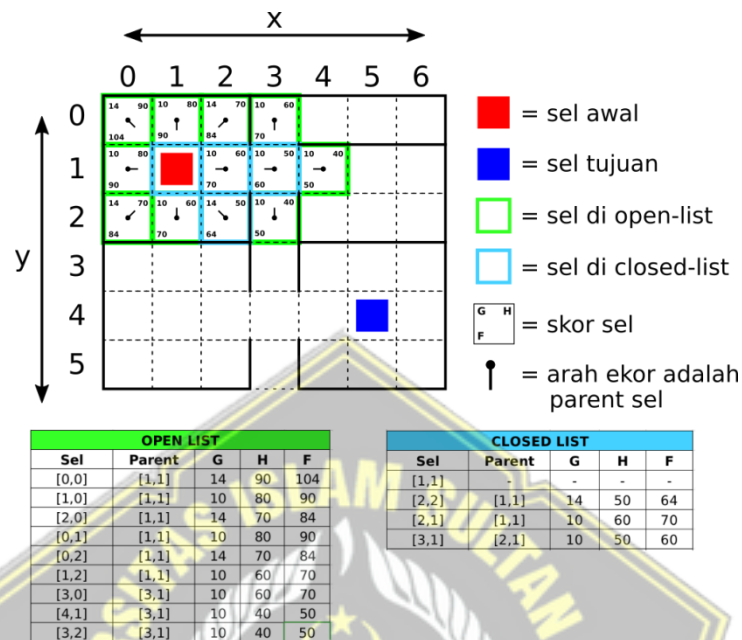
Gambar 3.35 Ilustrasi pencarian rute dengan algoritma A* (7)

8. Selanjutnya memilih sel dengan skor F terkecil. Sel[3,1] memiliki skor F terkecil, maka kita pindahkan ke *closed-list*



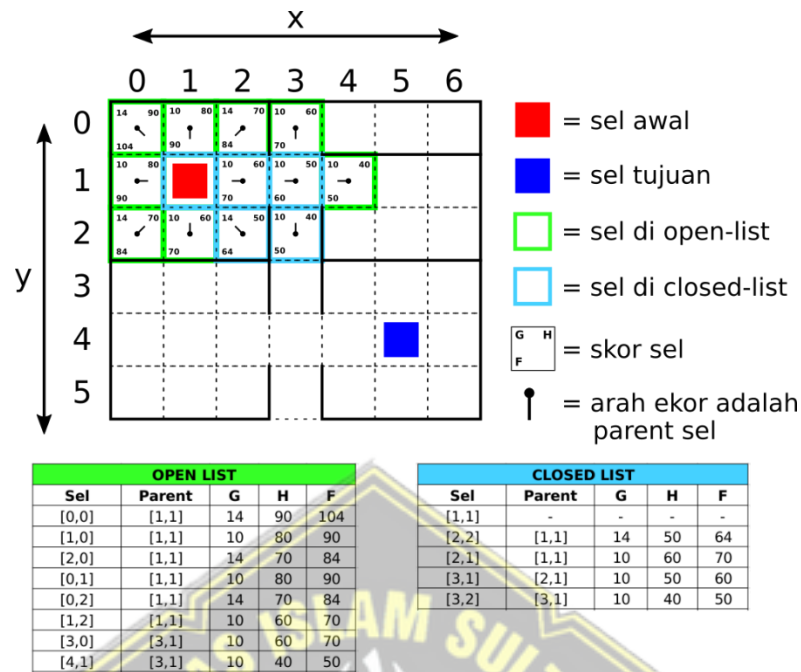
Gambar 3.36 Ilustrasi pencarian rute dengan algoritma A* (8)

9. Selanjutnya kita cek tetangga sel[3,1] yang tersedia. Hitung skor sel, atur parent, dan masukkan ke dalam *open-list*.



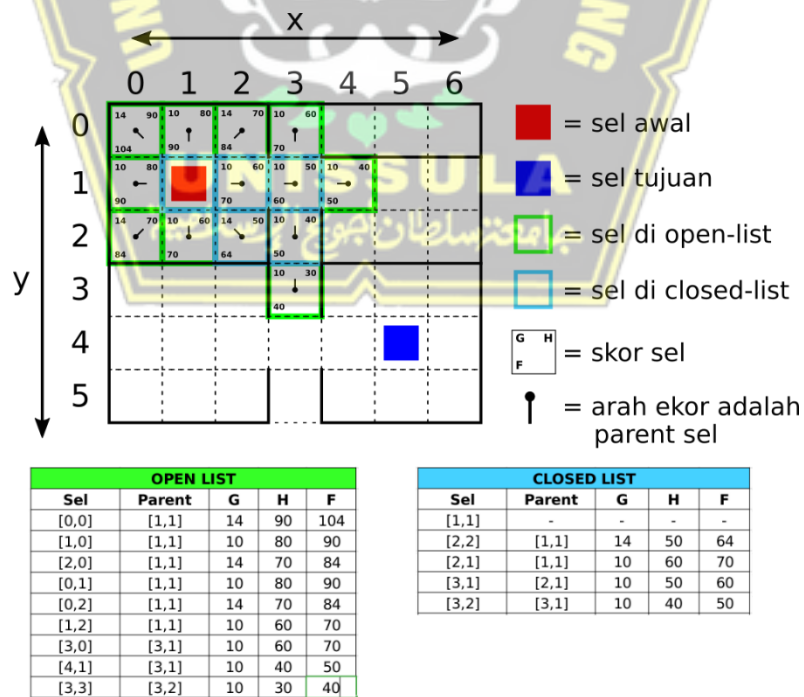
Gambar 3.37 Ilustrasi pencarian rute dengan algoritma A* (9)

10. Selanjutnya pada *open-list*, pilih sel dengan skor F terkecil. Sel[3,2] memiliki skor F terkecil, maka kita pindahkan ke *closed-list*.



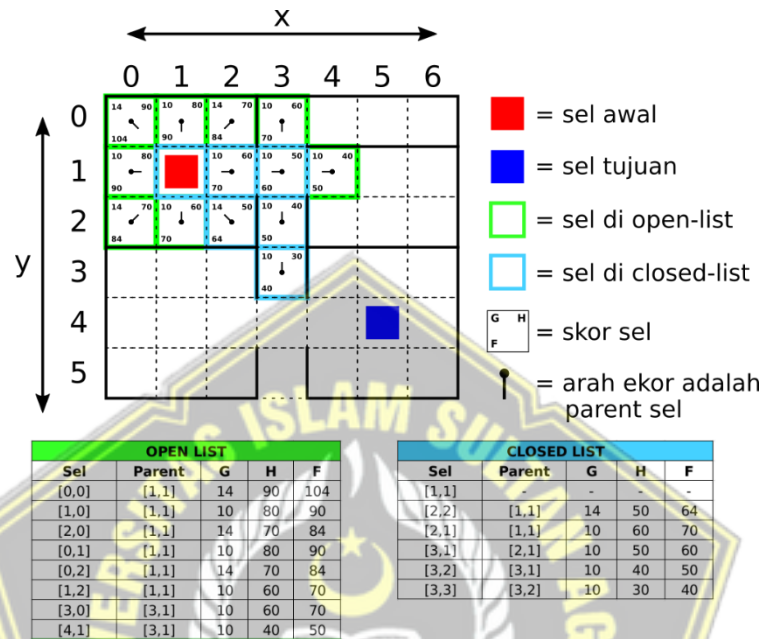
Gambar 3.38 Ilustrasi pencarian rute dengan algoritma A* (10)

11. Cek tetangga sel[3,2]. Sel[3,3] tersedia, maka hitung skor, atur parent, dan masukkan ke *open-list*.



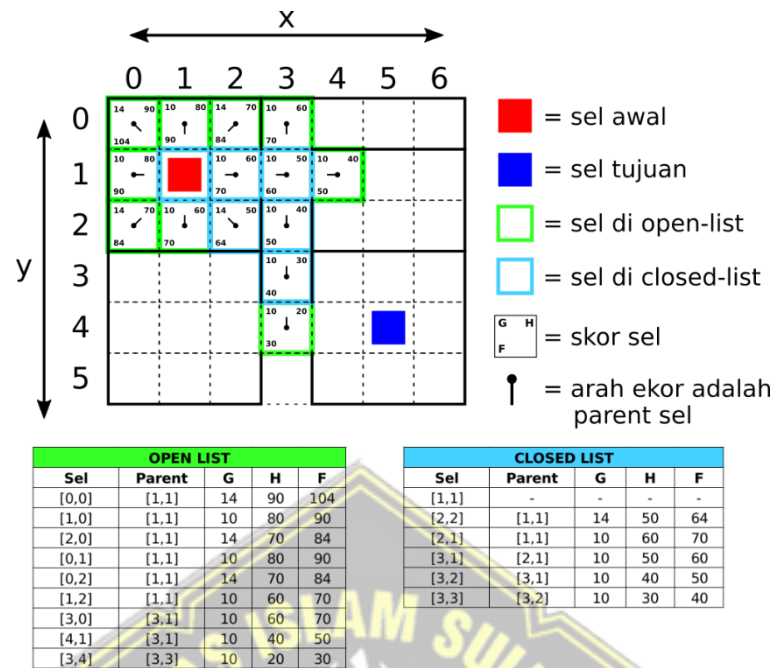
Gambar 3.39 Ilustrasi pencarian rute dengan algoritma A* (11)

12. Selanjutnya cek sel dengan skor terkecil pada open-list. Sel[3,3] memiliki skor terkecil, maka kita pindahkan ke *closed-list*.



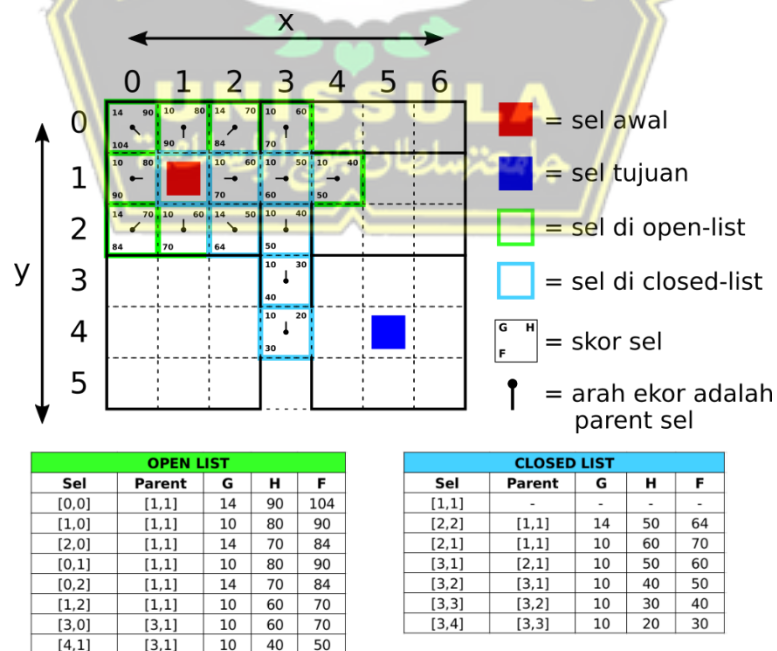
Gambar 3.40 Ilustrasi pencarian rute dengan algoritma A* (12)

13. Cek tetangga sel[3,3]. Tersedia sel[3,4], hitung skor, atur parent, dan masukkan ke *open-list*.



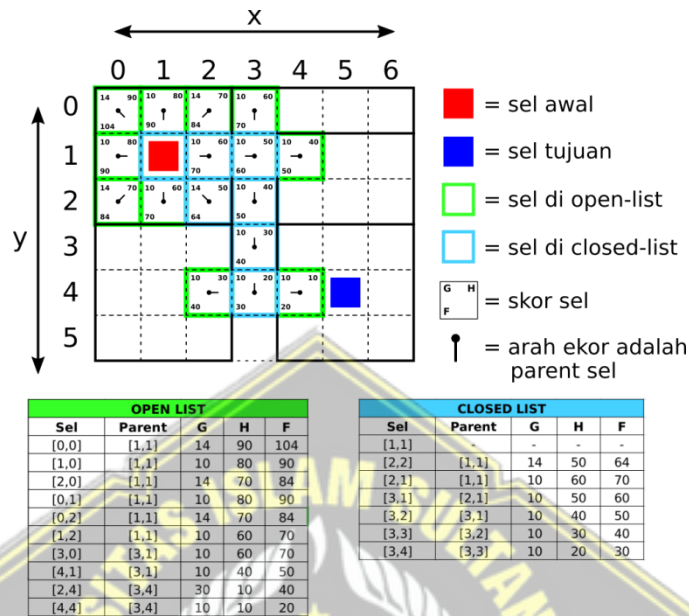
Gambar 3.41 Ilustrasi pencarian rute dengan algoritma A* (13)

14. Selanjutnya cek sel dengan skor F terkecil pada *open-list*. Sel[3,4] memiliki skor F terkecil, maka kita pindahkan ke dalam *closed-list*.



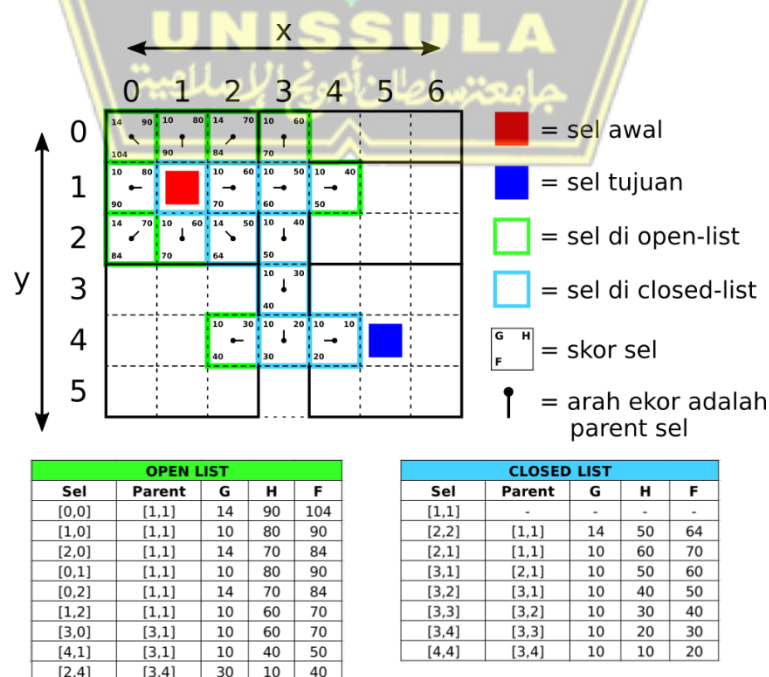
Gambar 3.42 Ilustrasi pencarian rute dengan algoritma A* (14)

15. Cek tetangga dari sel[3,4]. Tersedia sel[2,4] dan sel[4,4], maka hitung skor sel, atur parent, dan masukkan ke dalam *open-list*.



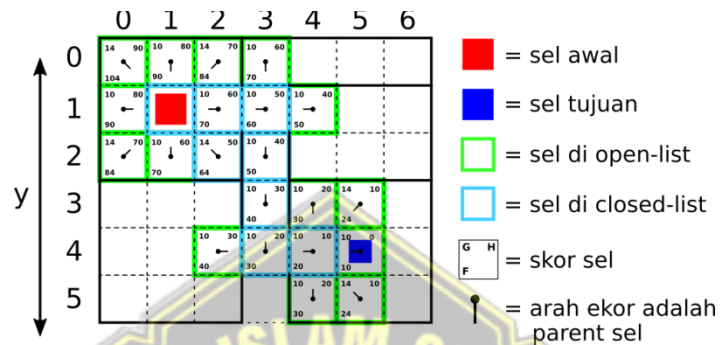
Gambar 3.43 Ilustrasi pencarian rute dengan algoritma A* (15)

16. Selanjutnya pilih sel dengan skor F terkecil pada *open-list*. Sel[4,4] memiliki skor F paling kecil, maka kita pindahkan ke dalam *closed-list*.



Gambar 3.44 Ilustrasi pencarian rute dengan algoritma A* (16)

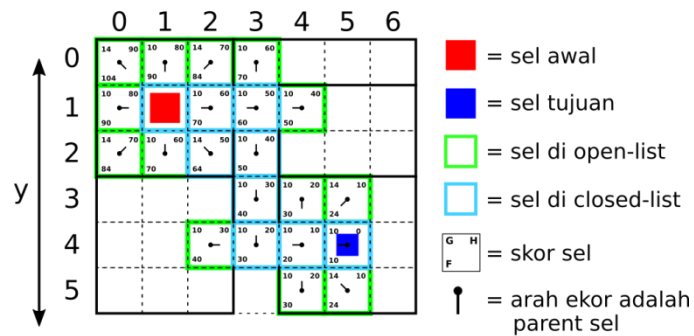
17. Cek tetangga dari sel[4,4]. Tersedia 5 buah sel tetangga, maka kita hitung skor, atur parent, dan masukkan ke dalam *open-list*.



OPEN LIST					CLOSED LIST				
Sel	Parent	G	H	F	Sel	Parent	G	H	F
[0,0]	[1,1]	14	90	104	[1,1]	-	-	-	-
[1,0]	[1,1]	10	80	90	[2,2]	[1,1]	14	50	64
[2,0]	[1,1]	14	70	84	[2,1]	[1,1]	10	60	70
[0,1]	[1,1]	10	60	70	[3,1]	[2,1]	10	50	60
[0,2]	[1,1]	14	70	84	[3,2]	[3,1]	10	40	50
[1,2]	[1,1]	10	60	70	[3,3]	[3,2]	10	30	40
[3,0]	[3,1]	10	60	70	[3,4]	[3,3]	10	20	30
[4,1]	[3,1]	10	40	50	[4,4]	[3,4]	10	10	20
[2,4]	[3,4]	30	10	40					
[4,3]	[4,4]	10	20	30					
[5,3]	[4,4]	14	10	24					
[5,4]	[4,4]	10	0	10					
[4,5]	[4,4]	10	20	30					
[5,5]	[4,4]	14	10	24					

Gambar 3.45 Ilustrasi pencarian rute dengan algoritma A* (17)

18. Selanjutnya cek sel dengan skor F terkecil pada open-list. Sel[5,4] memiliki skor F terkecil, maka kita pindahkan ke dalam *closed-list*.

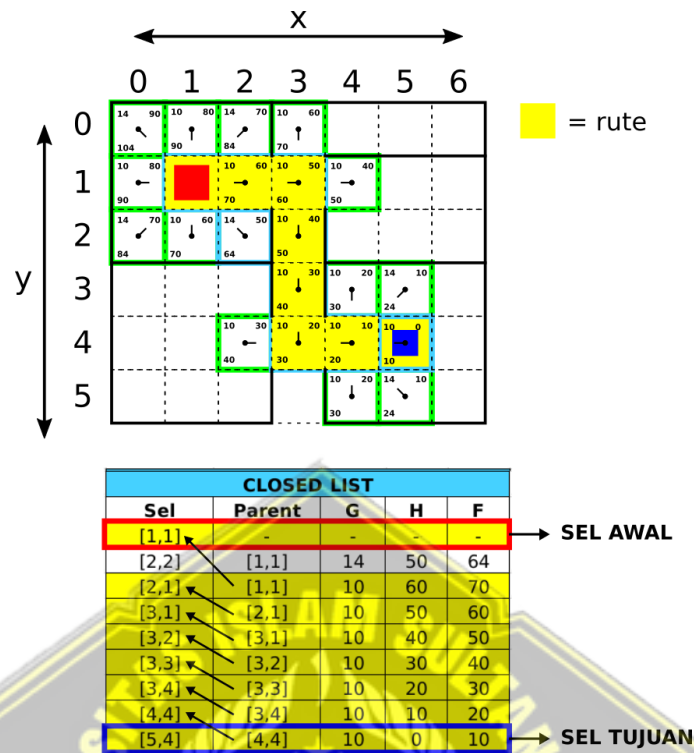


OPEN LIST				
Sel	Parent	G	H	F
[0,0]	[1,1]	14	90	104
[1,0]	[1,1]	10	80	90
[2,0]	[1,1]	14	70	84
[0,1]	[1,1]	10	80	90
[0,2]	[1,1]	14	70	84
[1,2]	[1,1]	10	60	70
[3,0]	[3,1]	10	60	70
[4,1]	[3,1]	10	40	50
[2,4]	[3,4]	30	10	40
[4,3]	[4,4]	10	20	30
[5,3]	[4,4]	14	10	24
[4,5]	[4,4]	10	20	30
[5,5]	[4,4]	14	10	24

CLOSED LIST				
Sel	Parent	G	H	F
[1,1]	-	-	-	-
[2,2]	[1,1]	14	50	64
[2,1]	[1,1]	10	60	70
[3,1]	[2,1]	10	50	60
[3,2]	[3,1]	10	40	50
[3,3]	[3,2]	10	30	40
[3,4]	[3,3]	10	20	30
[4,4]	[3,4]	10	10	20
[5,4]	[4,4]	10	0	10

Gambar 3.46 Ilustrasi pencarian rute dengan algoritma A* (18)

19. Sel[5,4] yang merupakan sel tujuan telah masuk ke dalam *closed-list*, maka pencarian dihentikan. Selanjutnya menelusuri rute yang telah dilalui untuk sampai ke sel tujuan dengan menggunakan atribut parent. Dimulai dari sel tujuan hingga sampai kembali ke sel awal. Hasil dari penelusuran tersebut merupakan rute yang berhasil ditemukan.

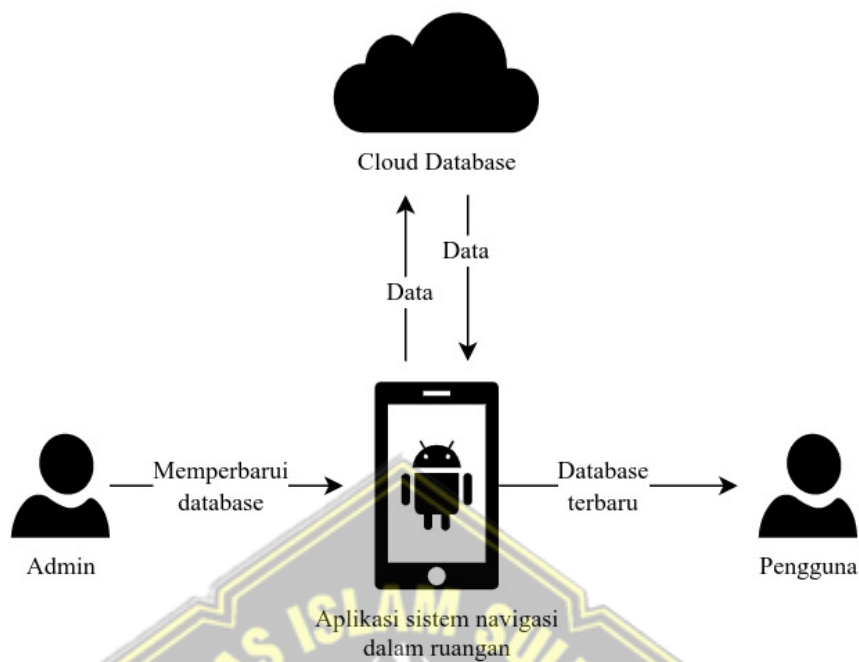


Gambar 3.47 Ilustrasi pencarian rute dengan algoritma A* (19)

3.6. Perancangan Database

3.6.1. Database berbasis cloud

Pada fungsi pemosisian, database yang berisi data sampel wi-fi merupakan bagian utama dalam proses penentuan posisi. Jika data sampel pada database cacat atau tidak lengkap, maka proses pemosisian dipastikan tidak dapat menghasilkan hasil yang akurat. Jadi ketika sistem digunakan oleh pengguna, database pada aplikasi wajib dalam kondisi memiliki data yang optimal terhadap kebutuhan sistem.



Gambar 3.48 Ilustrasi arsitektur sistem dengan *database* berbasis *cloud*

Dalam proses pengembangan sistem, database akan selalu mengalami perubahan data dalam rangka mewujudkan kinerja sistem yang baik. Layanan penyimpanan berbasis *cloud* akan sangat memudahkan pada proses ini karena membuat sistem lebih fleksibel dalam perubahan dan penyajian data. Dengan menggunakan database berbasis *cloud*, penggunapun tidak perlu mencopot dan memasang kembali aplikasi versi terbaru hanya untuk mendapatkan database sampel *wi-fi* yang telah diperbarui. Aplikasi akan secara otomatis memperbarui database selama perangkat terhubung ke internet.

3.6.2. Struktur database data sampel

Data yang disimpan pada database adalah kumpulan data sampel *wi-fi* yang ditambahkan oleh admin. Berikut ini merupakan tabel struktur database data sampel *wi-fi*.

Tabel 3.7 Struktur database data sampel *wi-fi*

Field	Tipe	Keterangan
id	<i>string</i>	<i>Primary key</i>
lokasi	<i>string</i>	Lokasi dimana diambilnya data sampel <i>wi-fi</i>
ap1	<i>number</i>	Level kekuatan sinyal <i>access point</i> 1
ap2	<i>number</i>	Level kekuatan sinyal <i>access point</i> 2
ap3	<i>number</i>	Level kekuatan sinyal <i>access point</i> 3
waktu_input	<i>date</i>	Waktu kapan ditambahnya data sampel

Pada tabel 3.7 struktur data sampel *wi-fi* di atas dapat diketahui memiliki 6 buah *field*. *Field id* memiliki fungsi sebagai *primary key* untuk setiap data sampel yang setiap nilainya bersifat unik.

Field lokasi pada data sampel merupakan label lokasi dimana diambilnya data sampel. *Field lokasi* ini memiliki tipe data *String*. Data dari *field lokasi* akan diinput oleh admin menggunakan *form* dengan pilihan lokasi di lantai 3 Fakultas Teknologi Industri yang telah disediakan oleh aplikasi sistem.

Field ap1, ap2, dan ap3 berisi data kekuatan sinyal dari *access point wi-fi* yang tersedia pada sistem. Tipe dari *field* tersebut adalah *number*. Data dari *field* ini diambil melalui perangkat android dari hasil pemindaian jaringan *wi-fi* yang memiliki informasi level kekuatan sinyal.

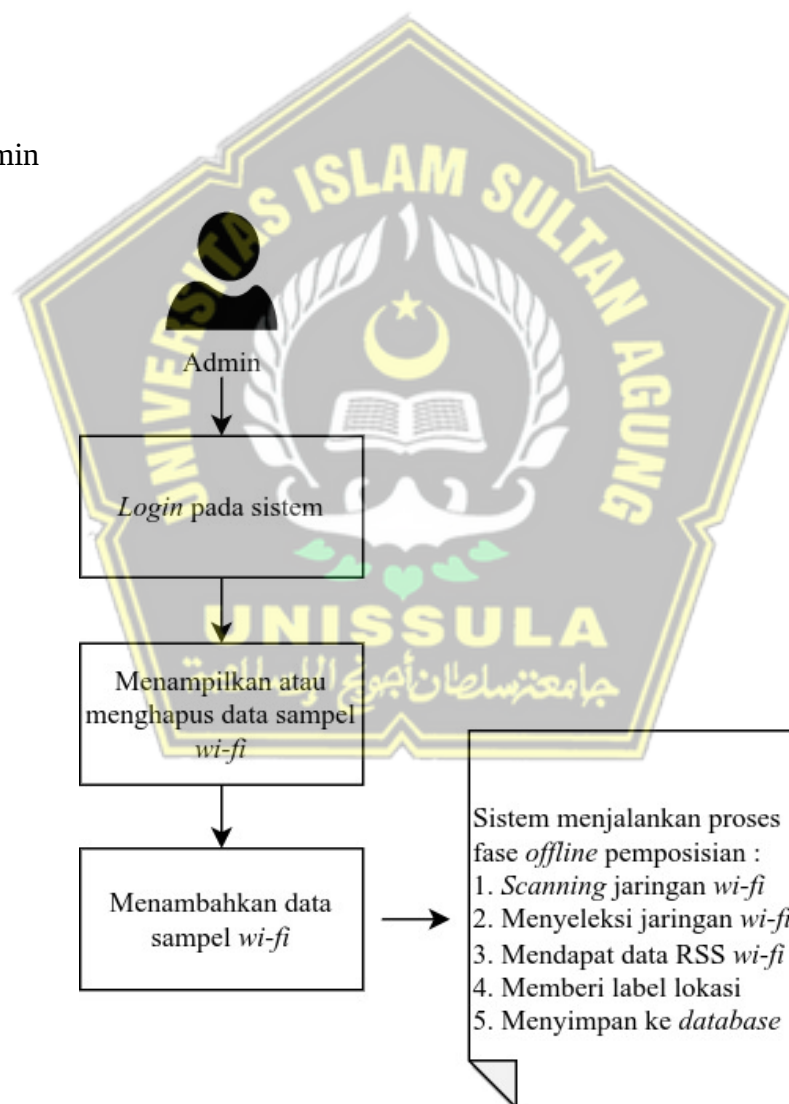
Field waktu_input berisi data kapan waktu diambilnya data sampel *wi-fi*. *Field* tersebut memiliki tipe *number*. Data dari *field* ini akan secara otomatis dihasilkan oleh aplikasi melalui sitem waktu pada android.

3.7. Analisa Kebutuhan dan Diagram Sistem

3.7.1. Aktor

Sistem yang akan dibangun memiliki 2 aktor yaitu admin dan pengguna. Kedua aktor tersebut dapat melakukan beberapa aktivitas pada sistem, yaitu sebagai berikut.

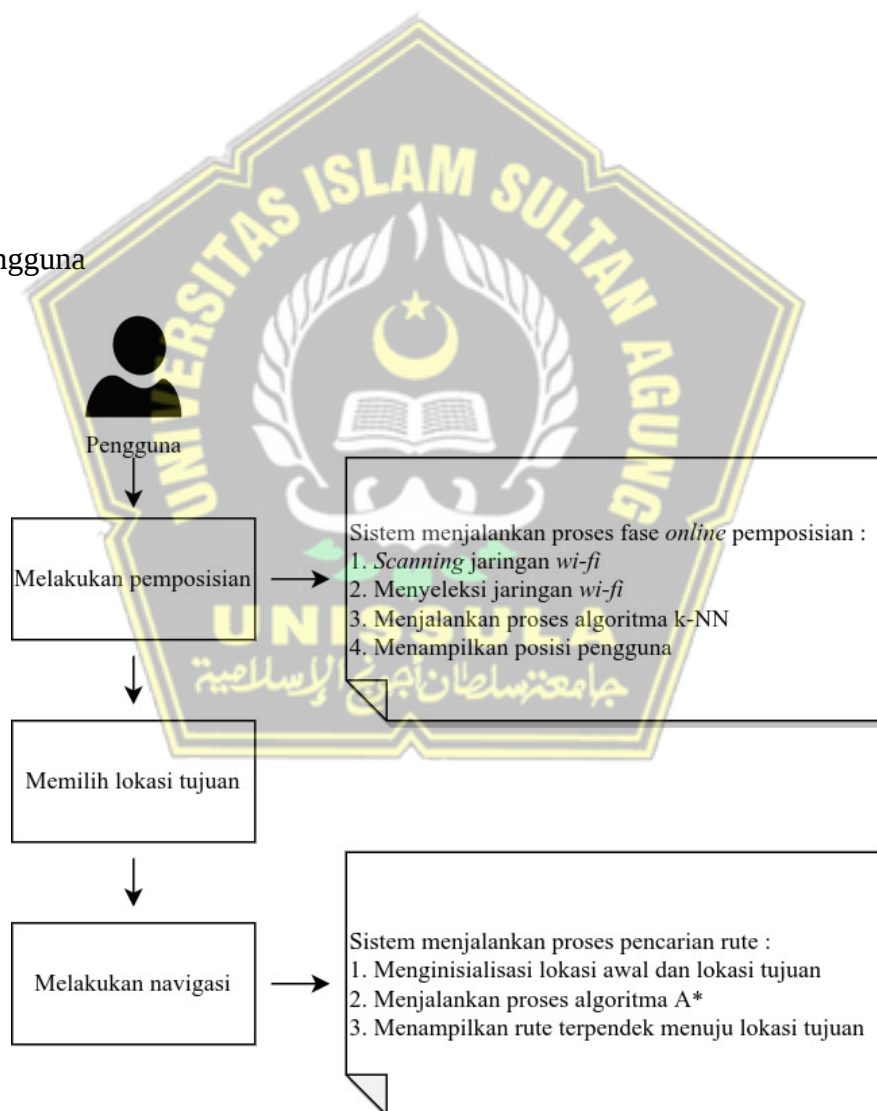
1. Admin



Gambar 3.49 Alur penggunaan sistem oleh admin

Admin dapat melakukan pembaruan pada *cloud database*, yang terintegrasi dengan seluruh aplikasi yang dipasang pada *smartphone android* baik oleh admin maupun pengguna. Pada sistem ini admin memiliki 2 pengoperasian yaitu menambah data sampel *wi-fi* dan menghapus data sampel *wi-fi* pada *cloud database*.

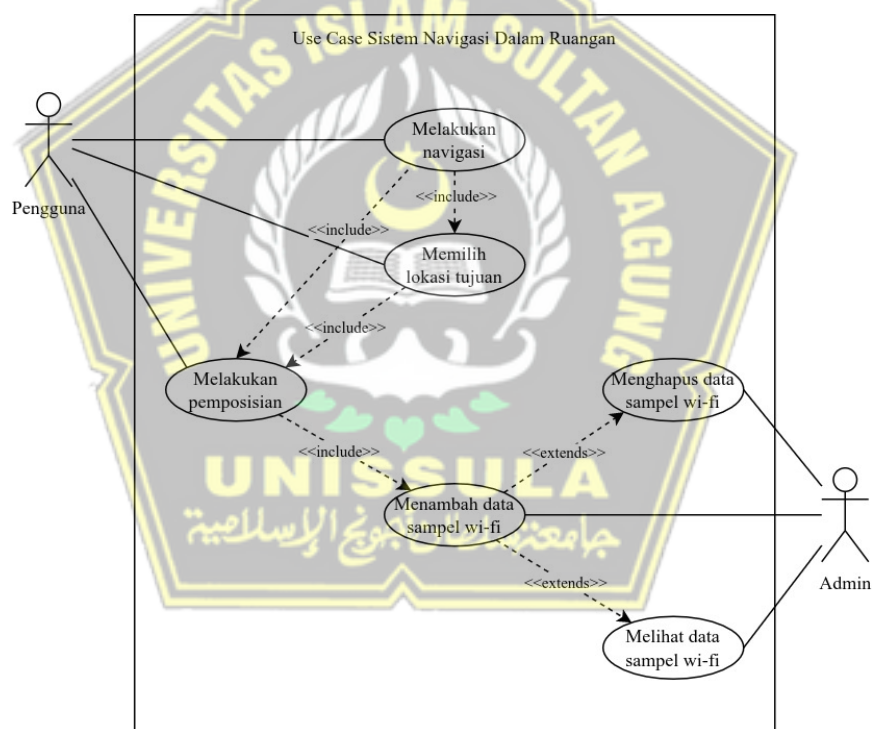
2. Pengguna



Gambar 3.50 Alur penggunaan sistem oleh pengguna

Pengguna dapat mengetahui posisi dimana saat ini berada pada lantai 3 gedung Fakultas Teknologi Industri Unissula dengan mengoperasikan fungsi pemposisian pada sistem. Kemudian fungsi navigasi dapat dioperasikan selanjutnya setelah posisi pengguna saat ini telah diketahui. Pengguna dapat mengetahui rute terpendek dari posisi saat ini menuju ruangan atau lokasi telah dipilih untuk dikunjungi.

3.7.2. Use case diagram



Gambar 3.51 Usecase diagram sistem navigasi dalam ruangan

Gambar 3.49 merupakan use case diagram dari prototipe sistem navigasi dalam ruangan. Terdapat dua aktor dalam diagram tersebut yaitu pengguna dan admin. Pengguna memiliki 3 use case yaitu melakukan pemposisian, memilih

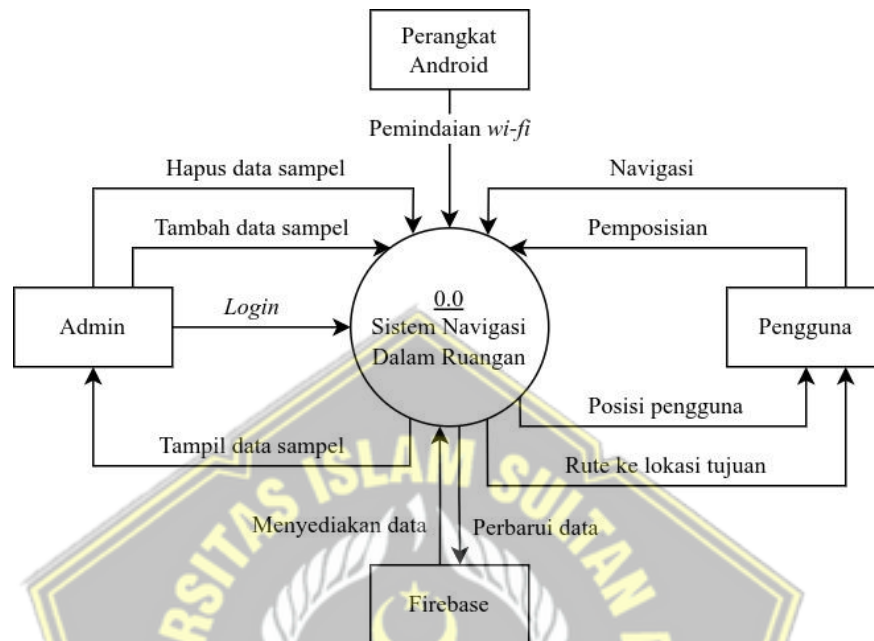
lokasi tujuan, dan melakukan navigasi. Sedangkan admin memiliki 3 kasus penggunaan yaitu melihat, menambah, dan menghapus data sampel *wi-fi*.

Use case pengguna pada bagian melakukan pemposisian merupakan *use case* utama untuk dapat memilih lokasi tujuan dan melakukan navigasi. Pengguna dapat melakukan pemposisian selama tersedia data sampel yang ditambahkan oleh admin. Admin dapat melihat serta menghapus data sampel yang telah ditambahkan.



3.7.3. Data flow diagram

1. Data flow diagram level konteks

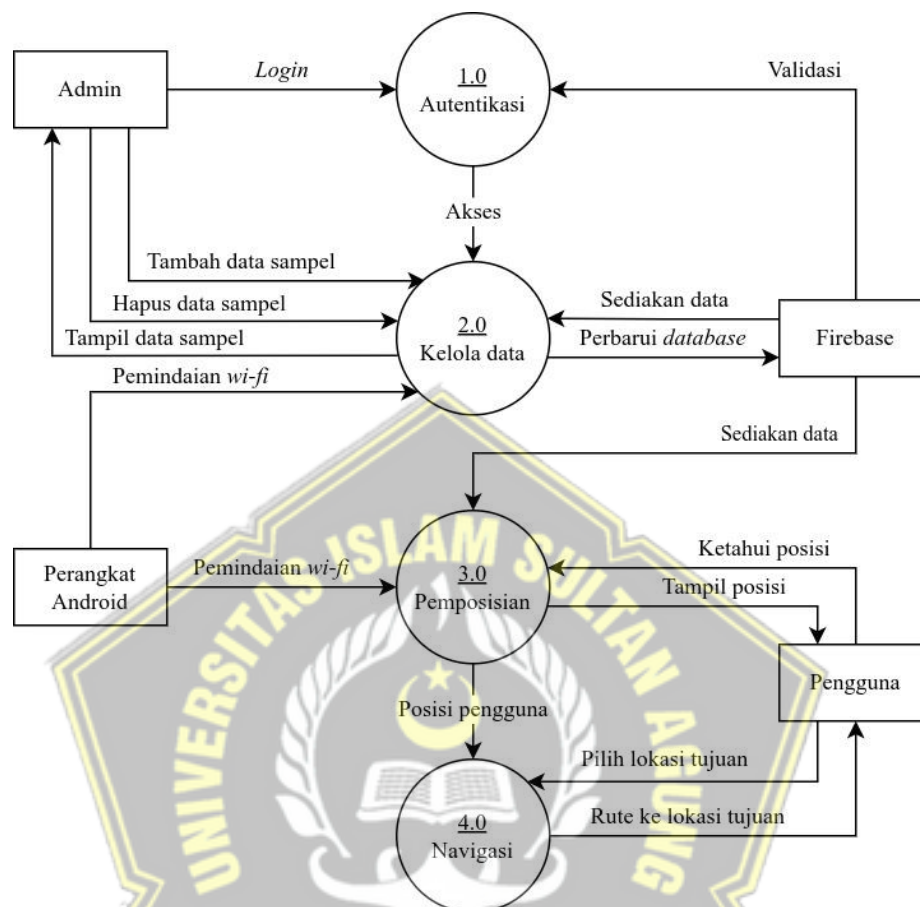


Gambar 3.52 Data flow diagram level konteks

Gambar 3.50 merupakan *Data flow diagram* sistem navigasi dalam ruangan pada level konteks. Dapat diketahui dari diagram di atas bahwa sistem memiliki 4 terminator yaitu admin, pengguna, perangkat android, dan firebase.

Admin berurusan dengan proses-proses yang berkaitan dengan data yang dibutuhkan untuk kesiapan sistem. Pengguna berurusan dengan proses utama pada sistem yaitu pemposisian dan navigasi. Perangkat android berurusan dengan pemindaian jaringan *wi-fi* untuk mendapatkan data RSS sebagai atribut dalam data sampel dan data masukan. Firebase berurusan dengan segala proses *back-end*.

2. Data flow diagram level 0



Gambar 3.53 Data flow diagram level 0

Gambar 3.51 merupakan *Data flow diagram* sistem navigasi dalam ruangan pada level 0. Pada diagram tersebut terdapat 4 fungsi yaitu autentikasi, kelola data, pemposisian, dan navigasi.

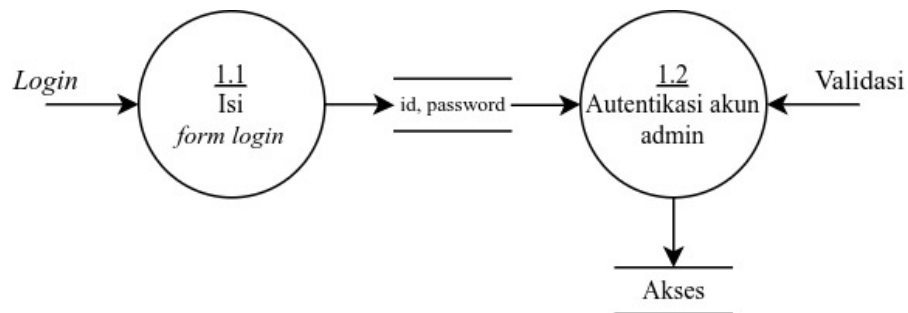
Proses 1.0 (autentikasi), berisi proses untuk keperluan login admin. Terdapat dua terminator yang terlibat dalam proses ini yaitu admin dan firebase. Admin sebagai subjek yang membutuhkan untuk dapat masuk ke dalam sistem, sedangkan firebase merupakan subjek yang akan mem*Validation* informasi login dari admin. *Output* dari proses 1.0 adalah akses untuk admin dapat melakukan pengelolaan data pada sistem.

Proses 2.0 (kelola data), berisi proses untuk keperluan pengelolaan data sampel *wi-fi* pada database sistem. Terdapat 3 terminator yang terlibat pada fungsi ini yaitu admin, perangkat android, dan firebase. Admin sebagai subjek yang melakukan pengelolaan data sampel *wi-fi*. Perangkat android akan memberikan data RSS dari hasil pemindaian jaringan *wi-fi*, yang akan digunakan oleh admin untuk melakukan tambah data sampel pada database. Firebase akan menjalankan perintah dari admin dalam memperbarui database.

Proses 3.0 (pemosisian), berisi proses untuk keperluan penentuan dimana posisi pengguna berada. Terdapat 3 terminator pada fungsi ini yaitu pengguna, perangkat android, dan firebase. Pengguna sebagai subjek yang melakukan perintah pemosisian. Perangkat android akan memberikan data RSS dari hasil pemindaian jaringan *wi-fi*, yang akan digunakan sebagai data masukan. Firebase akan menyediakan database yang berisi data sampel *wi-fi* untuk proses penentuan posisi. *Output* dari proses 3.0 adalah posisi pengguna berada.

Proses 4.0 (navigasi), berisi proses untuk keperluan penentuan rute dari posisi pengguna menuju lokasi tujuan. Terdapat 1 terminator pada fungsi ini yaitu pengguna. Pengguna sebagai subjek untuk memilih lokasi yang ingin dituju. *Output* dari proses 4.0 adalah rute dari posisi pengguna menuju lokasi yang dituju.

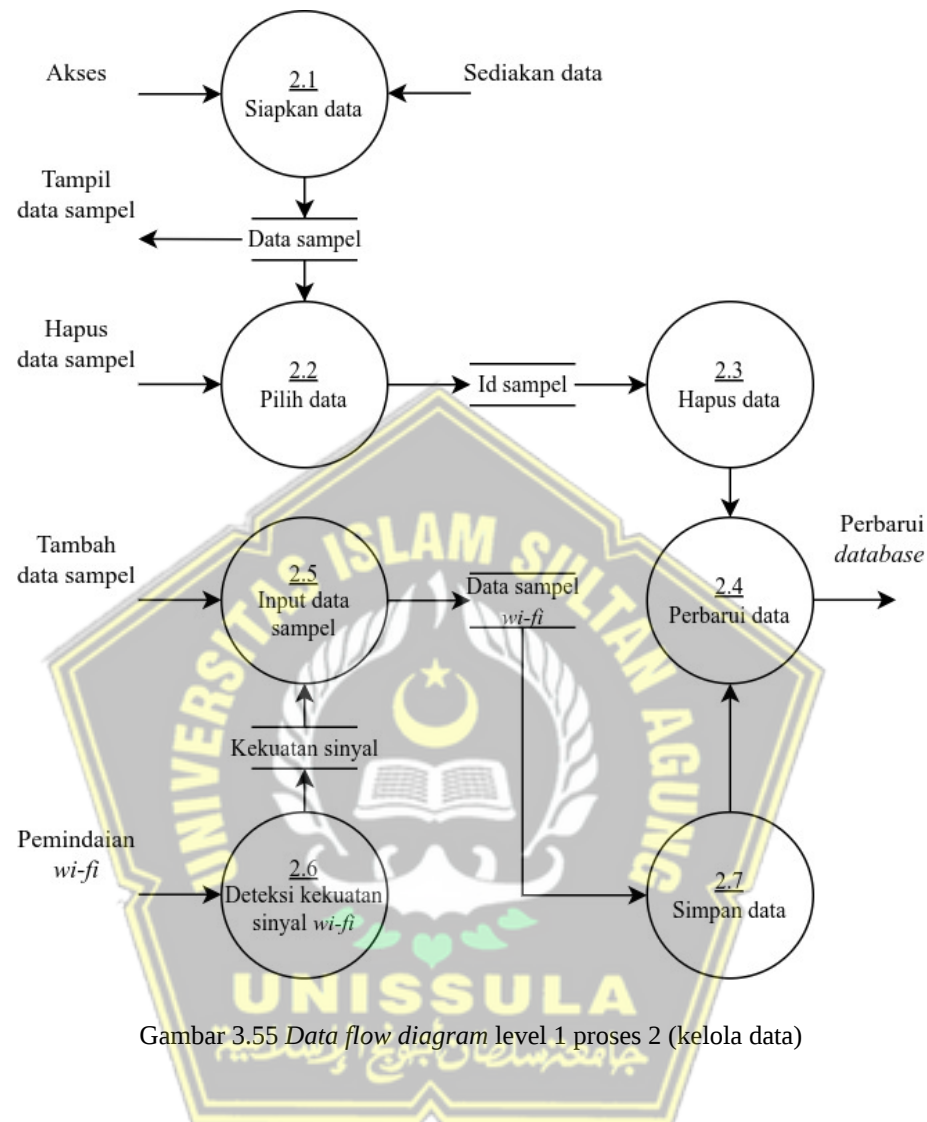
3. Data flow diagram level 1 proses 1 (autentikasi)



Gambar 3.54 Data flow diagram level 1 proses 1 (autentikasi)

Gambar 3.52 merupakan *Data flow diagram* level 1 yang menggambarkan secara lebih rinci proses 1.0 (autentikasi). Diawali dari proses 1.1 dimana admin mengisi *form login*. Proses tersebut menghasilkan data id dan password yang kemudian akan digunakan untuk proses *Validation* oleh firebase pada proses 1.2. Jika data login admin *Valid*, maka admin akan diberikan akses untuk mengelola database pada proses selanjutnya.

4. Data flow diagram level 1 proses 2 (kelola data)



Gambar 3.55 Data flow diagram level 1 proses 2 (kelola data)

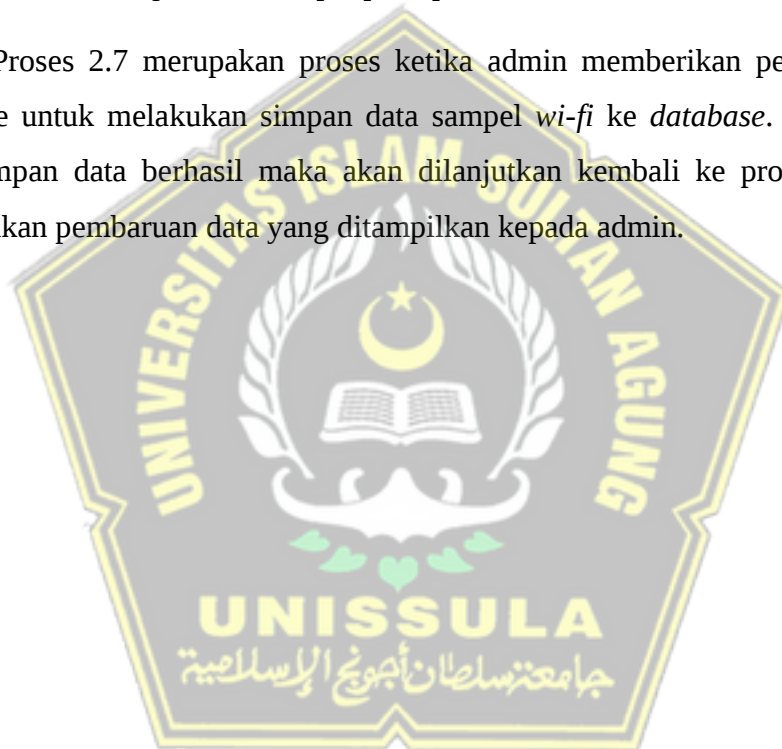
Gambar 3.53 merupakan *Data flow diagram* level 1 yang menjelaskan secara rinci proses 2.0 yaitu kelola data. Diawali dengan proses 2.1 yaitu proses penyiapan data, begitu admin diberikan akses untuk dapat melakukan pengelolaan data, sistem akan mengunduh database dari *firebase*. Proses 2.1 akan menghasilkan *output* kumpulan data sampel *wi-fi* untuk kemudian ditampilkan kepada admin.

Proses 2.2 merupakan proses pemilihan data sampel *wi-fi* yang dilakukan admin jika ingin menghapusnya dari database. *Output* dari proses ini adalah id sampel yang ingin dihapus. Id sampel ini kemudian digunakan pada proses 2.3

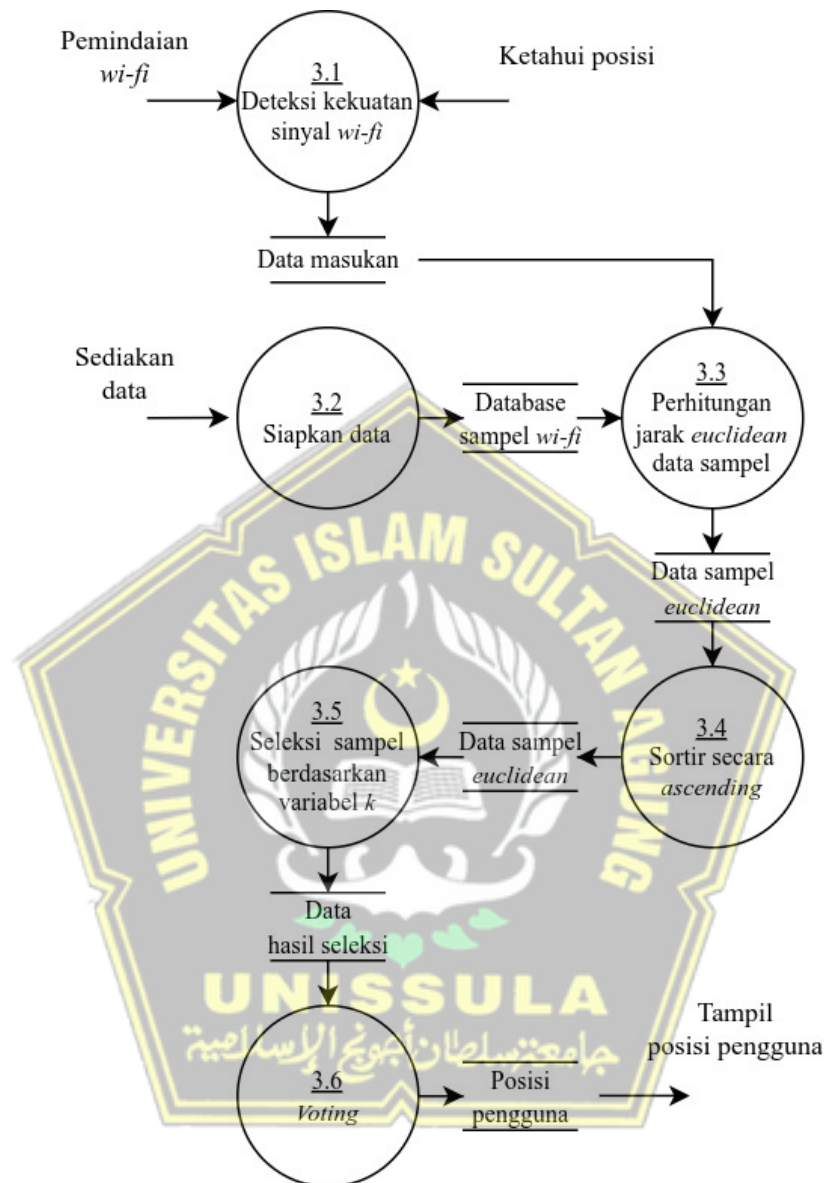
sebagai *query* untuk menjalankan perintah hapus pada *database*. Begitu data sampel *wi-fi* yang dipilih berhasil dihapus dari *database*, maka sistem melanjutkan ke proses 2.4 yaitu memperbarui data yang ditampilkan kepada admin.

Proses 2.5 merupakan proses *input* untuk penambahan data sampel *wi-fi*. Sistem akan melanjutkan ke proses 2.6 yaitu menjalankan perintah untuk melakukan pemindaian jaringan *wi-fi* pada perangkat android, yang kemudian menghasilkan *output* data kekuatan sinyal *wi-fi* (RSS). Data ini akan digunakan untuk kebutuhan *input* data sampel pada proses 2.5.

Proses 2.7 merupakan proses ketika admin memberikan perintah kepada firebase untuk melakukan simpan data sampel *wi-fi* ke *database*. Begitu proses menyimpan data berhasil maka akan dilanjutkan kembali ke proses 2.4 untuk melakukan pembaruan data yang ditampilkan kepada admin.



5. *Data flow diagram level 1 proses 3 (pemposisian)*



Gambar 3.56 *Data flow diagram level 1 proses 3 (pemposisian)*

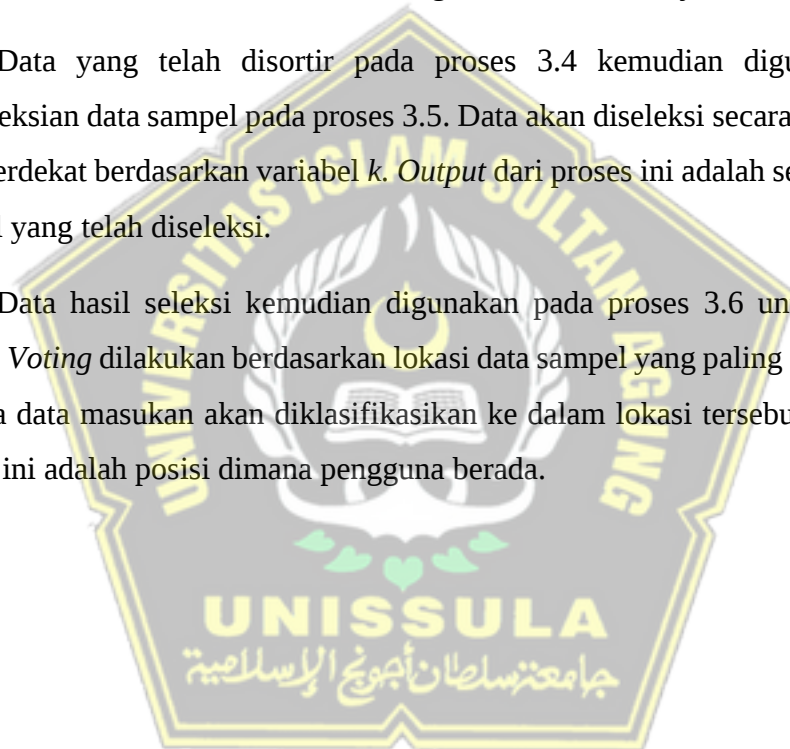
Gambar 3.54 merupakan *Data flow diagram level 1* yang menggambarkan secara rinci proses 3.0 (pemposisian). Diawali dari proses 3.1 dimana sistem memberi perintah perangkat android untuk melakukan pemindaian jaringan wi-fi. Proses tersebut menghasilkan *output* data kekuatan sinyal dari posisi pengguna saat ini yang kemudian akan digunakan sebagai data masukan.

Proses 3.2 adalah proses penyiapan data dimana sistem mengunduh *database* terbaru yang telah disediakan oleh *firebase*. *Output* dari proses ini adalah kumpulan data sampel *wi-fi*. Data ini akan digunakan pada proses 3.3 untuk dilakukan perhitungan jarak *euclidean* berdasarkan data masukan.

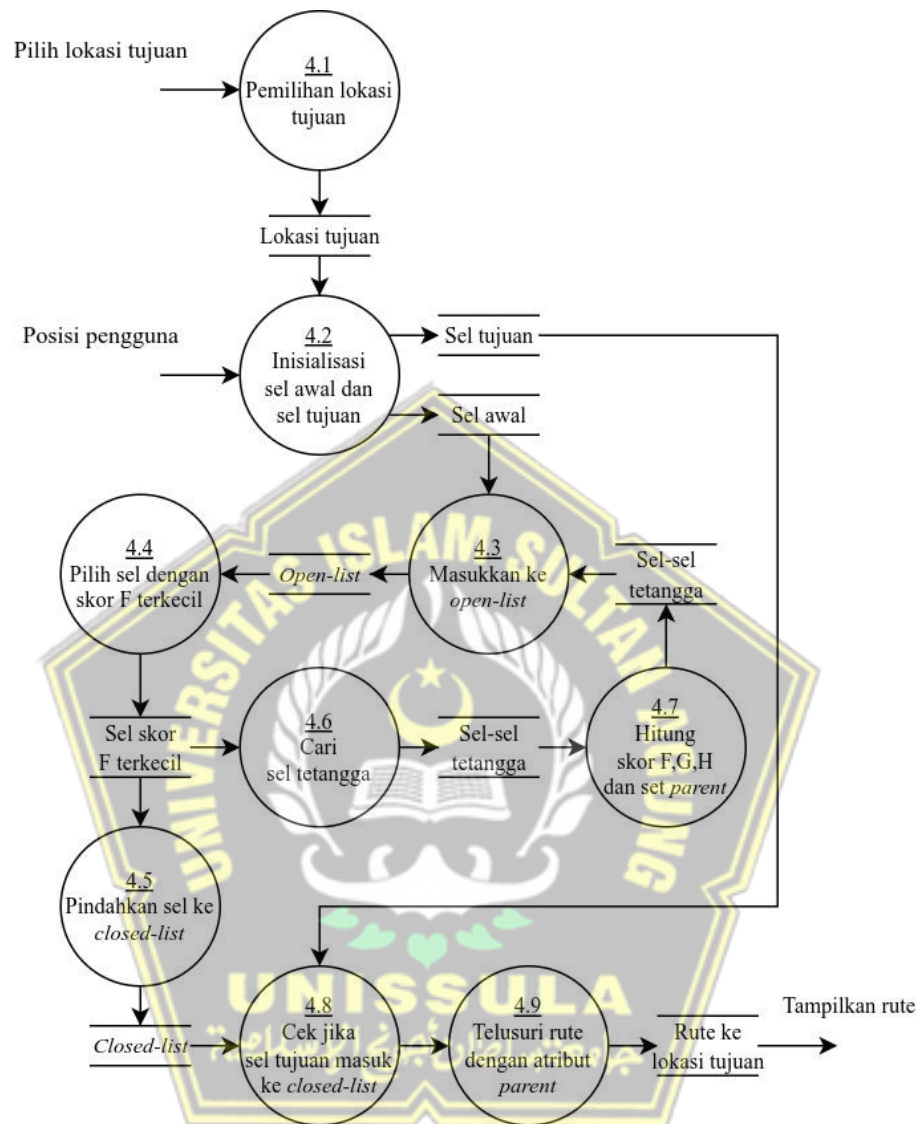
Proses 3.3 adalah proses perhitungan jarak *euclidean* pada seluruh data sampel berdasarkan data masukan. *Output* dari proses ini adalah kumpulan data sampel yang telah diberi skor jarak *euclidean*. Data ini kemudian digunakan pada proses 3.4 untuk disortir secara *ascending* berdasarkan skor jarak *euclidean*-nya.

Data yang telah disortir pada proses 3.4 kemudian digunakan untuk penyeleksian data sampel pada proses 3.5. Data akan diseleksi secara urut dari skor jarak terdekat berdasarkan variabel k . *Output* dari proses ini adalah sejumlah k data sampel yang telah diseleksi.

Data hasil seleksi kemudian digunakan pada proses 3.6 untuk dilakukan *voting*. *Voting* dilakukan berdasarkan lokasi data sampel yang paling sering muncul, dimana data masukan akan diklasifikasikan ke dalam lokasi tersebut. *Output* dari proses ini adalah posisi dimana pengguna berada.



6. Data flow diagram level 1 proses 4 (navigasi)



Gambar 3.57 Data flow diagram level 1 proses 4 (navigasi)

Gambar 3.55 merupakan *Data flow diagram level 1* yang menggambarkan secara rinci proses 4.0 (navigasi). Diawali dari proses 4.1 dimana pengguna memilih lokasi tujuan. Kemudian dilanjutkan ke proses 4.2 dimana posisi pengguna dan lokasi tujuan diinisialisasi menjadi sel awal dan sel tujuan pada *grid layout* sel peta lantai 3.

Proses 4.3 adalah proses memasukkan sel ke dalam *open-list*. Untuk langkah pertama sel awal akan dimasukkan ke dalam *open-list*. *Output* dari proses ini adalah *open-list* yang sementara hanya berisi sel awal.

Proses 4.4 adalah proses pemilihan sel dengan skor F terkecil. Pada *output* yang dihasilkan sebelumnya, *open-list* hanya berisi sel awal yang memiliki skor $F=0$. Maka pada proses saat ini yang dipilih adalah sel awal. Sel awal merupakan *output* proses saat ini.

Sel yang memiliki skor F terkecil kemudian digunakan pada proses 4.5 yaitu pemindahan sel tersebut dari *open-list* ke dalam *closed-list*. Selanjutnya pada proses 4.6, sel yang telah dipindahkan dilakukan pencarian sel tetangga yang dapat dikunjungi. *Output* dari proses 4.6 adalah kumpulan sel tetangga dari sel sebelumnya.

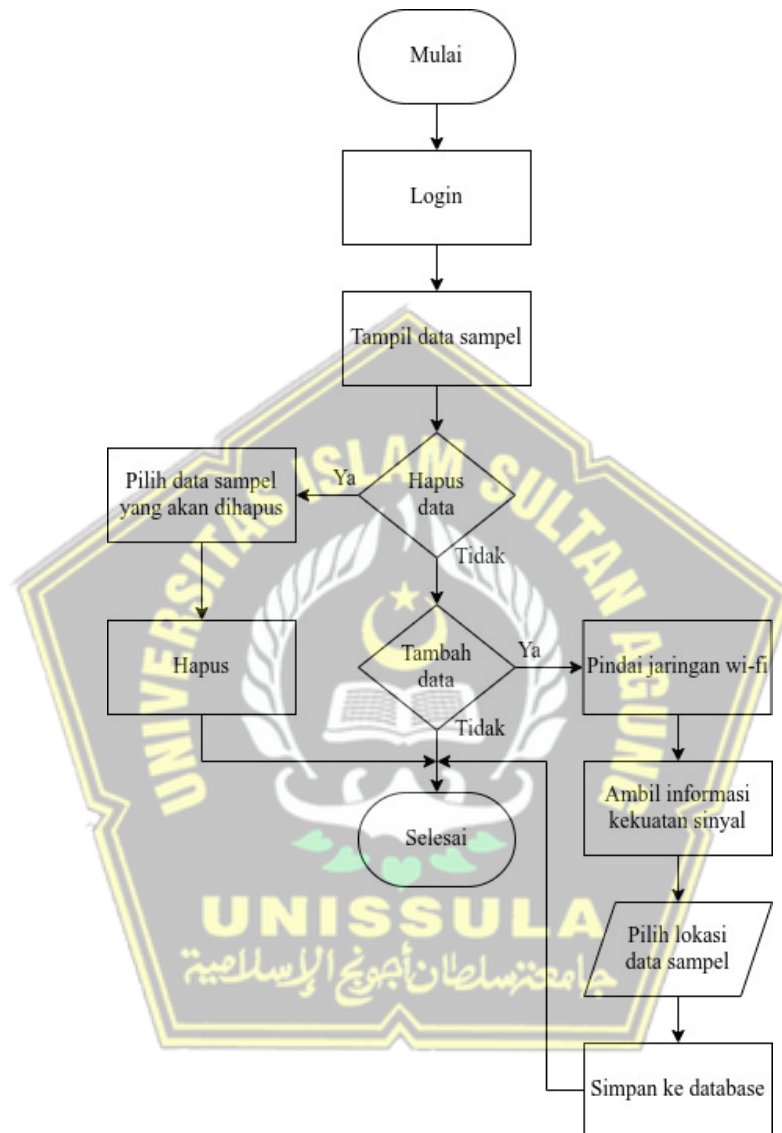
Proses 4.7 adalah proses perhitungan atribut skor F, G, H dan pemberian atribut *parent* pada kumpulan sel tetangga yang didapat pada proses sebelumnya. *Output* dari proses ini kemudian digunakan kembali pada proses 4.3 yaitu dimasukkan ke dalam *open-list*.

Proses 4.8 adalah proses dimana dilakukan pengecekan pada setiap sel yang dimasukkan pada *closed-list*. Jika yang dimasukkan bukan merupakan sel tujuan maka proses-proses sebelumnya diulang hingga sel tujuan ditemukan. Namun jika sel tujuan telah ditemukan dan masuk ke dalam *closed-list*, maka dilanjutkan ke proses selanjutnya yaitu proses 4.9.

Proses 4.9 adalah proses dimana dilakukan penelusuran rute sel yang berada pada *closed-list*. Penelusuran rute dilakukan berdasarkan atribut *parent* yang dimulai dari sel tujuan hingga kembali menuju sel awal. *Output* dari proses ini adalah rute dari posisi pengguna menuju lokasi yang ingin dituju.

3.7.4. Flowchart diagram

1. Flowchart diagram penggunaan sistem oleh admin



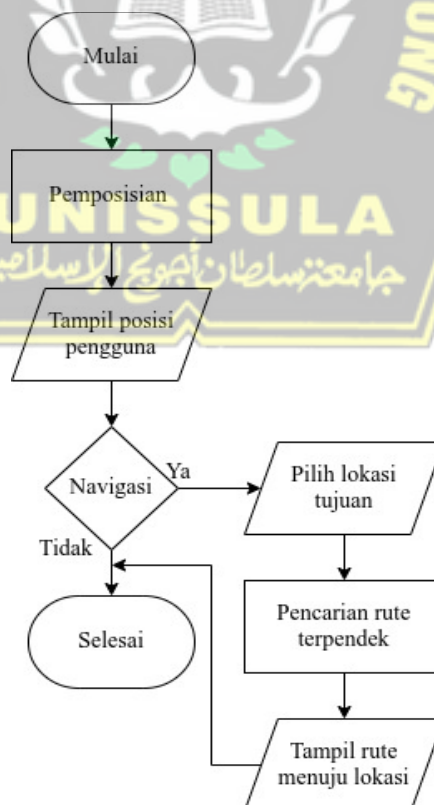
Gambar 3.58 Flowchart diagram penggunaan sistem oleh admin

Admin mengawali penggunaan sistem dengan *login* sebagai admin terlebih dahulu. Setelah *login* berhasil, maka admin akan masuk ke halaman kelola data sampel. Terdapat tombol hapus di setiap kotak informasi data sampel *wi-fi*. Jika

admin ingin menghapus data sampel, maka admin dapat memilih data sampel yang ingin dihapus dan menekan tombol hapus.

Pada halaman kelola data sampel terdapat juga tombol tambah data sampel. Admin dapat menekan tombol tambah data dan masuk ke halaman tambah data sampel. Untuk menambah data sampel *wi-fi*, admin perlu menekan tombol pindai jaringan *wi-fi* agar perangkat dapat melakukan pemindaian jaringan *wi-fi*. Setelah perangkat selesai melakukan pemindaian jaringan *wi-fi*, maka aplikasi akan secara otomatis mengambil informasi kekuatan sinyal dari jaringan *wi-fi* yang dibutuhkan untuk penambahan data sampel. Selanjutnya admin perlu memilih dimana lokasi data sampel tersebut diambil. Langkah terakhir untuk menyimpan data sampel adalah menekan tombol simpan, maka sistem akan menyimpan data sampel tersebut ke dalam *database*.

2. Flowchart diagram penggunaan sistem oleh pengguna

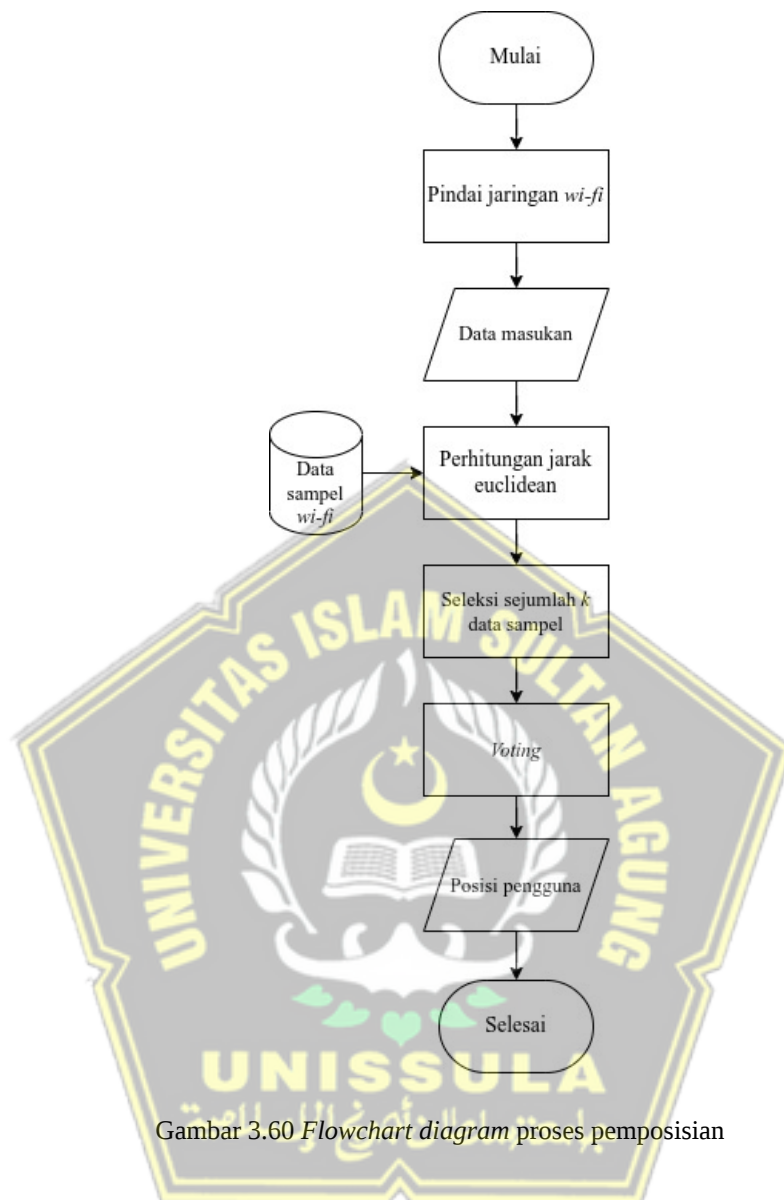


Gambar 3.59 *Flowchart diagram* penggunaan sistem oleh pengguna

Pengguna dapat langsung melakukan pencarian posisi dengan menekan tombol cari posisi pada halaman utama. Sistem akan melakukan penentuan dimana posisi pengguna berada yang kemudian hasilnya akan ditampilkan kepada pengguna. Selanjutnya jika pengguna ingin menjalankan fungsi navigasi, maka pengguna perlu untuk memilih kemana lokasi yang ingin dituju. Setelah lokasi dipilih, maka sistem akan melakukan pencarian rute terpendek dari posisi pengguna menuju lokasi tujuan. Selanjutnya rute yang dihasilkan dari proses pencarian akan ditampilkan kepada pengguna.

3. *Flowchart diagram* proses pemosisian





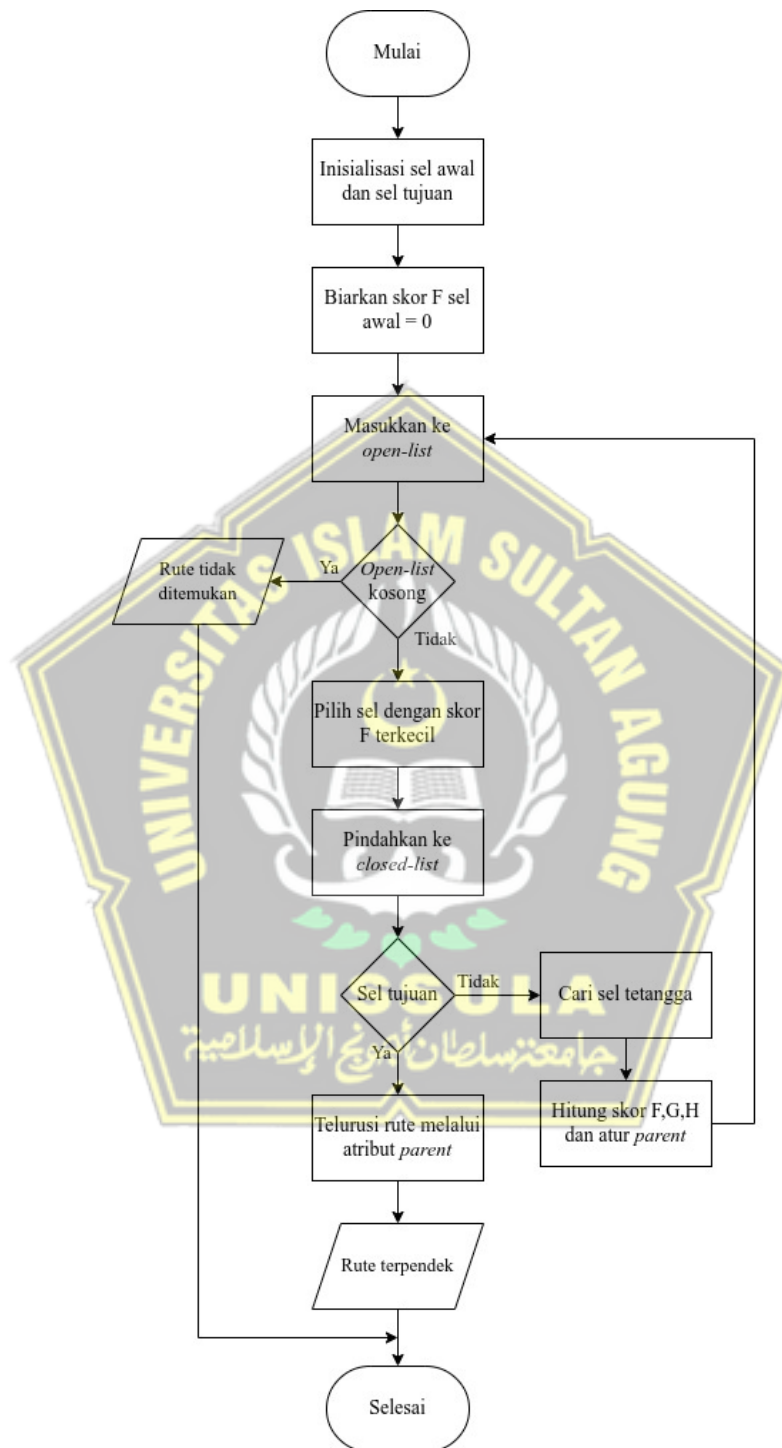
Gambar 3.60 Flowchart diagram proses pemposisian

Proses pemposisian dijelaskan lebih rinci pada *flowchart* diagram di atas. Diawali dengan dilakukannya pemindaian jaringan *wi-fi* terlebih dahulu oleh aplikasi sistem melalui perangkat android. Pemindaian yang dilakukan akan menghasilkan informasi kekuatan sinyal dimana pengguna berada dan akan digunakan sebagai data masukan. Setelah data masukan didapatkan, maka selanjutnya adalah melakukan perhitungan jarak euclidean data sampel pada *database* terhadap data masukan. Setelah semua data sampel selesai dilakukan perhitungan, maka selanjutnya akan diseleksi sejumlah data sampel yang memiliki jarak euclidean

terdekat sebanyak nilai dari variabel k . Hasil dari penyeleksian tersebut akan dilakukan *voting* berdasarkan label lokasi yang paling sering muncul. Hasil dari *voting* merupakan posisi dimana pengguna berada. Langkah terakhir adalah menampilkan posisi tersebut kepada pengguna.



4. Flowchart diagram proses navigasi

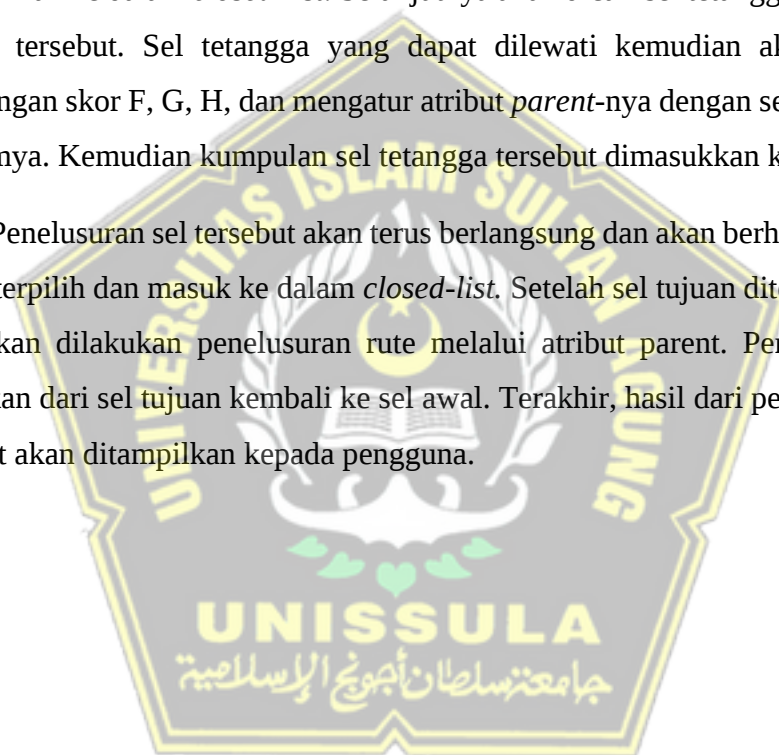


Gambar 3.61 Flowchart diagram proses navigasi

Proses pencarian rute dijelaskan lebih rinci melalui *usecase* diagram di atas. Langkah pertama adalah melakukan inisialisasi posisi pengguna dan lokasi tujuan menjadi sel awal dan sel tujuan pada peta *grid layout* sel. Selanjutnya sel awal akan diberi skor 0 pada atribut F, dan dimasukkan ke dalam *open-list*.

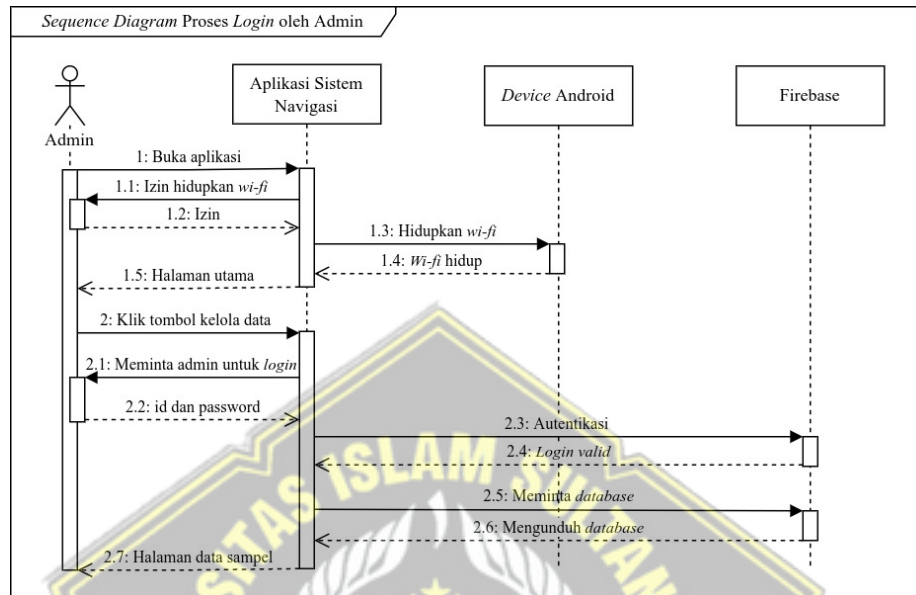
Sistem akan melakukan penelusuran rute selama *open-list* tidak kosong. Langkah awal penelusuran sel adalah memilih sel dengan skor F paling kecil yang ada pada *open-list*. Sel yang terpilih kemudian akan dikeluarkan dari *open-list* dan dipindahkan ke dalam *closed-list*. Selanjutnya akan dicari sel tetangga dari sel yang terpilih tersebut. Sel tetangga yang dapat dilewati kemudian akan dilakukan perhitungan skor F, G, H, dan mengatur atribut *parent*-nya dengan sel yang terpilih sebelumnya. Kemudian kumpulan sel tetangga tersebut dimasukkan ke *open-list*.

Penelusuran sel tersebut akan terus berlangsung dan akan berhenti hingga sel tujuan terpilih dan masuk ke dalam *closed-list*. Setelah sel tujuan ditemukan, maka akan dilakukan penelusuran rute melalui atribut *parent*. Penelusuran rute dilakukan dari sel tujuan kembali ke sel awal. Terakhir, hasil dari penelusuran rute tersebut akan ditampilkan kepada pengguna.



3.7.5. Sequence diagram

1. Sequence diagram proses login



Gambar 3.62 Sequence diagram proses login

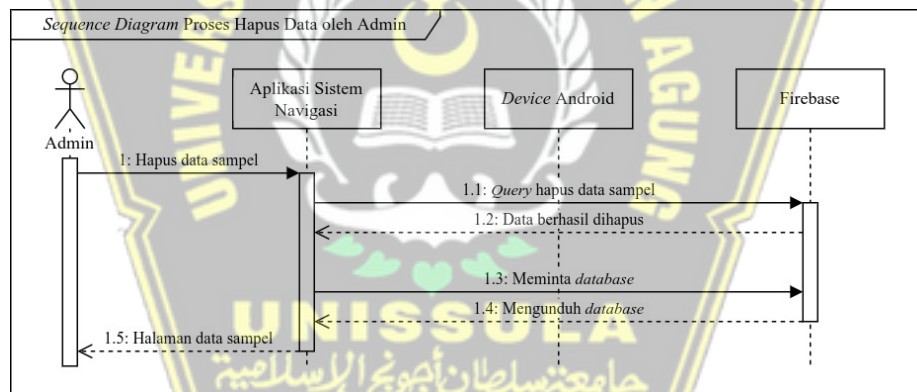
Sequence diagram pada gambar 3.60 menjelaskan secara rinci bagaimana urutan proses login yang dilalui oleh admin untuk mendapat akses pengelolaan pada aplikasi untuk kesiapan sistem. Diawali dengan proses 1, admin membuka aplikasi sistem navigasi dalam ruangan. Begitu aplikasi terbuka lalu dilanjutkan ke proses 1.1, pada proses ini aplikasi akan meminta persetujuan admin untuk menghidupkan layanan *wi-fi* pada android. Proses 1.2, merupakan respon dari admin ke aplikasi bahwa aplikasi mendapatkan izin untuk menghidupkan layanan *wi-fi* pada android.

Proses 1.3, merupakan proses dimana aplikasi memberikan perintah pada sistem android untuk menghidupkan layanan *wi-fi*. Dilanjutkan ke proses 1.4, yang merupakan respon sistem ke aplikasi bahwa layanan *wi-fi* telah dihidupkan. Setelah layanan *wi-fi* hidup, maka aplikasi melanjutkan ke proses 1.5, yaitu menampilkan halaman utama pada tampilan antarmuka aplikasi.

Proses 2, merupakan aksi dari admin untuk masuk ke halaman kelola data yang akan direspon oleh aplikasi di proses 2.1. Selanjutnya admin akan memasukkan id dan password login pada proses 2.2. Informasi login admin yang diterima aplikasi akan diteruskan ke *firebase* untuk dilakukan autentikasi. Proses 2.4, merupakan respon dari *firebase* ke aplikasi bahwa informasi login dari admin *Valid*.

Proses 2.5, adalah proses aplikasi meminta *firebase* untuk menyiapkan data yang dibutuhkan. Selanjutnya pada proses 1.4, aplikasi akan mengunduh *database* begitu data siap dan akan ditampilkan pada admin melalui halaman data sampel pada proses 1.5.

2. Sequence diagram proses hapus data sampel



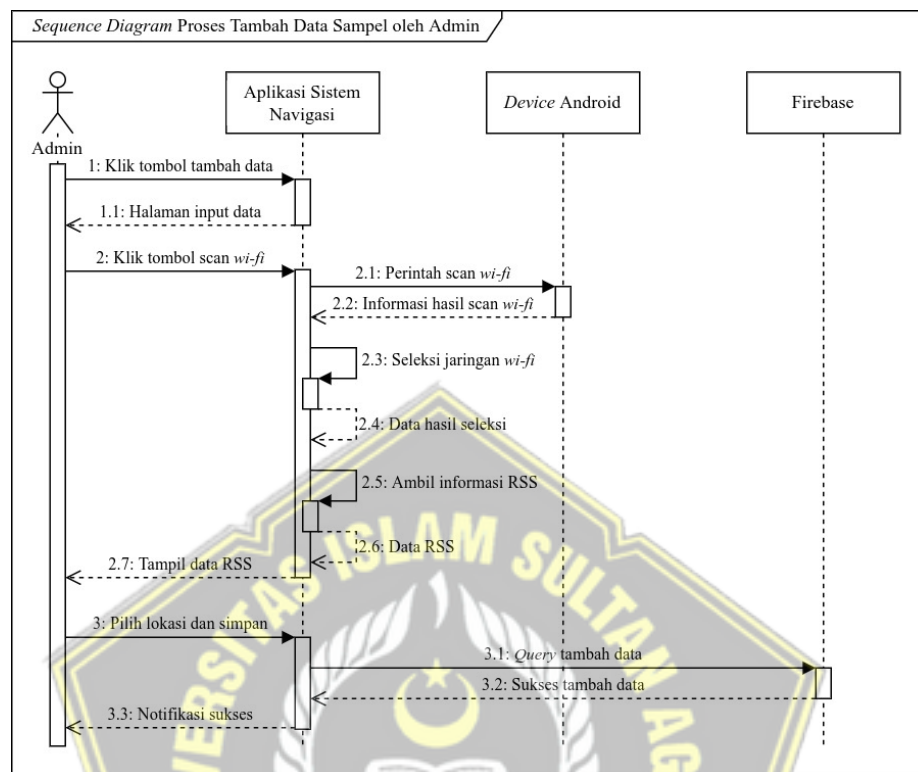
Gambar 3.63 Sequence diagram proses hapus data sampel

Sequence diagram pada gambar 3.61 menjelaskan secara rinci urutan proses yang dilalui admin ketika melakukan hapus data sampel. Diawali dari proses 1, admin menekan tombol hapus pada data sampel yang telah dipilih untuk dihapus. Kemudian aplikasi akan mengirim *query* hapus data sampel ke *firebase*, berdasarkan id dari data sampel yang telah dipilih admin. Begitu aplikasi mendapatkan respon bahwa *query* hapus data berhasil, selanjutnya aplikasi akan

melakukan pembaruan data untuk ditampilkan pada admin melalui proses 1.3, 1.4, dan 1.5.



3. Sequence diagram proses tambah data



Gambar 3.64 Sequence diagram proses tambah data

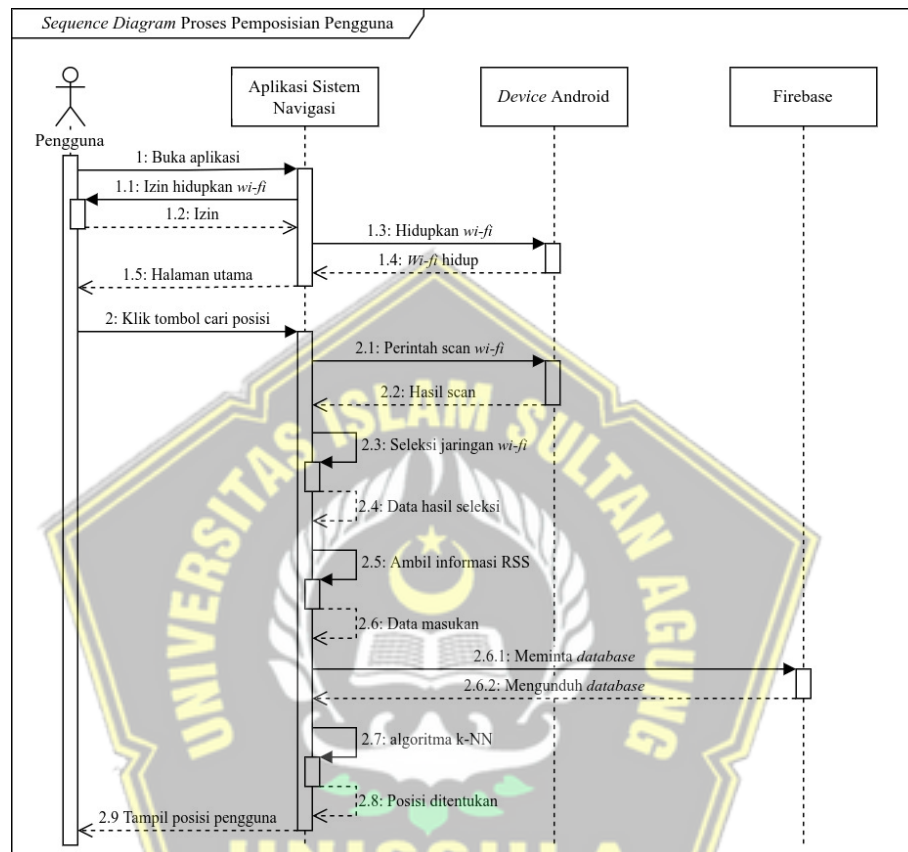
Sequence diagram pada gambar 3.61 menjelaskan secara rinci urutan proses yang dilalui admin ketika melakukan tambah data sampel. Diawali dari proses 1, admin menekan tombol tambah data dan akan muncul halaman *input data*. Selanjutnya pada proses 2, admin menekan tombol *scan wi-fi*, untuk melakukan pemindaian jaringan *wi-fi*.

Pada proses 2.1, aplikasi akan melakukan pemindaian jaringan *wi-fi* dengan mengirim perintah *scan wi-fi* pada sistem android dan informasi hasil *scan* akan diterima aplikasi pada proses 2.2. Proses 2.3 hingga 2.6 merupakan pengolahan data hasil *scan* oleh aplikasi yang kemudian data RSS yang didapatkan akan ditampilkan kepada admin pada proses 2.7.

Setelah data RSS didapatkan, selanjutnya pada proses 3 admin memilih dimana lokasi data RSS tersebut diambil. Data tersebut kemudian disimpan ke

dalam *database* melalui *query* yang dikirim aplikasi ke *firebase*. Admin akan mendapatkan notifikasi jika data sampel berhasil ditambahkan.

4. Sequence diagram proses pemposisian



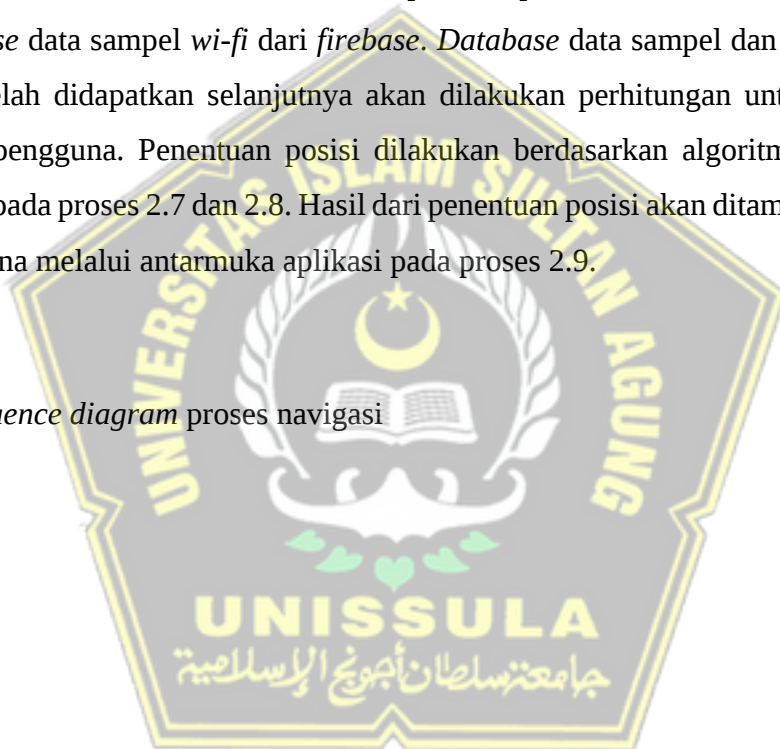
Gambar 3.65 Sequence diagram proses pemposisian

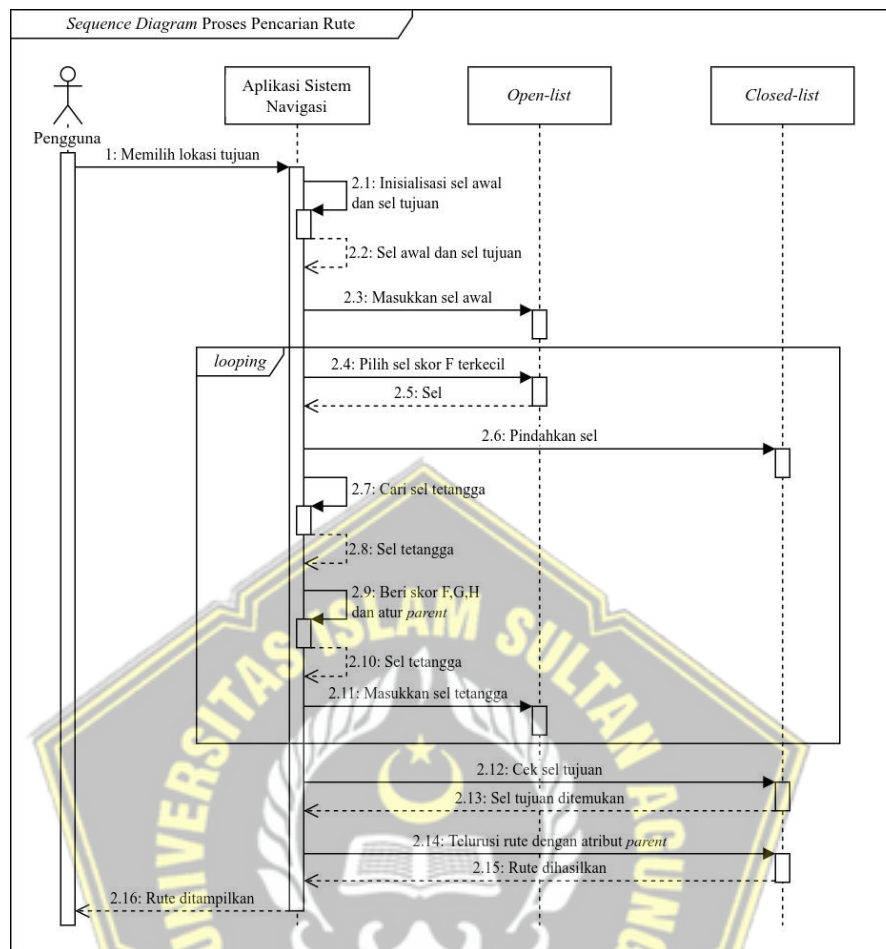
Sequence diagram pada gambar 3.62 menggambarkan secara rinci proses penentuan dimana posisi pengguna berada. Diawali dengan proses 1, dimana pengguna membuka aplikasi sistem navigasi. Selanjutnya pada proses 1.1 dan 1.2, aplikasi akan meminta persetujuan dari pengguna untuk menghidupkan layanan *wi-fi* pada perangkat android. Setelah pengguna mengizinkan, selanjutnya pada proses 1.3 dan 1.4 aplikasi akan mengirim perintah ke sistem android untuk menghidupkan layanan *wi-fi*. Setelah serangkaian proses tadi, aplikasi akan menampilkan halaman utama kepada pengguna.

Proses 2 merupakan aksi oleh pengguna yaitu menekan tombol cari posisi untuk mengetahui posisi pengguna saat ini berada. Setelah pada proses 2.1 dan 2.2, aplikasi akan mengirim perintah kepada sistem android untuk melakukan pemindaian jaringan *wi-fi* dan aplikasi akan menerima respon berupa informasi jaringan *wi-fi* yang terdeteksi. Proses 2.3 hingga 2.6 merupakan serangkaian proses yang dilakukan aplikasi untuk mengambil informasi data RSS sebagai data masukan.

Proses 2.6.1 dan 2.6.2 adalah proses aplikasi meminta dan mengunduh *database* data sampel *wi-fi* dari *firebase*. *Database* data sampel dan data masukan yang telah didapatkan selanjutnya akan dilakukan perhitungan untuk penentuan posisi pengguna. Penentuan posisi dilakukan berdasarkan algoritma *k-NN* yang terjadi pada proses 2.7 dan 2.8. Hasil dari penentuan posisi akan ditampilkan kepada pengguna melalui antarmuka aplikasi pada proses 2.9.

5. *Sequence diagram* proses navigasi





Gambar 3.66 *Sequence diagram* proses navigasi

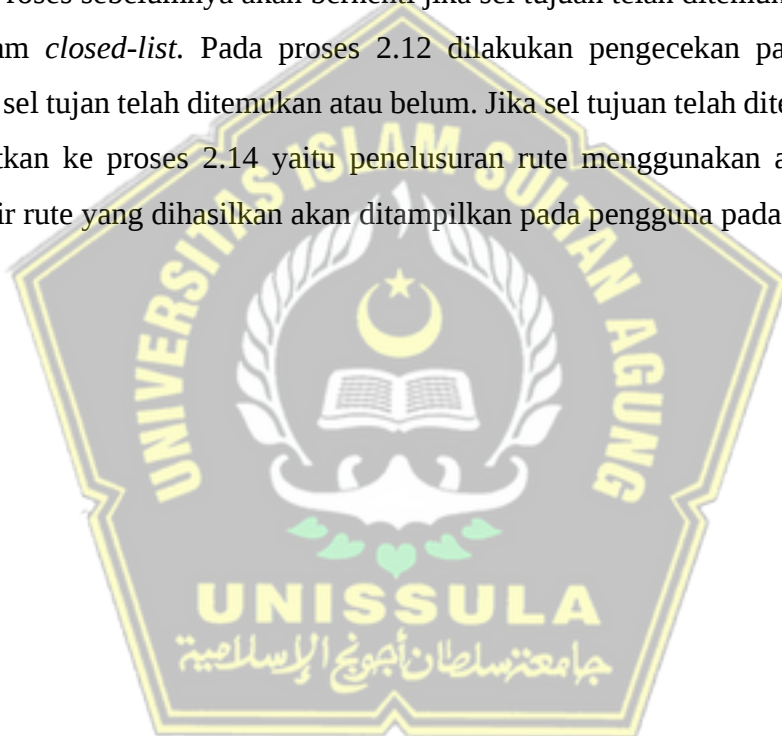
Sequence diagram pada gambar 3.63 menggambarkan secara rinci urutan proses pencarian rute dari posisi pengguna menuju lokasi tujuan. Diawali dengan proses 1 dimana pengguna memilih lokasi yang ingin dituju. Selanjutnya pada proses 2.1 dan 2.2 akan dilakukan proses inisialisasi sel awal dan sel tujuan pada peta di aplikasi. Kemudian sel awal yang telah diinisialisasi akan dimasukkan ke *open-list* dengan membiarkan skor atribut $F = 0$.

Proses selanjutnya adalah penelusuran rute yang dijalankan berdasarkan algoritma A^* . Proses penelusuran rute akan berlangsung selama *open-list* tidak kosong. Dimulai dengan memilih sel dengan skor F paling kecil yang ada di *open-*

list yang terjadi pada proses 2.4 dan 2.5. Pada proses 2.6, sel yang terpilih kemudian dikeluarkan dari *open-list* dan dipindahkan ke dalam *closed-list*.

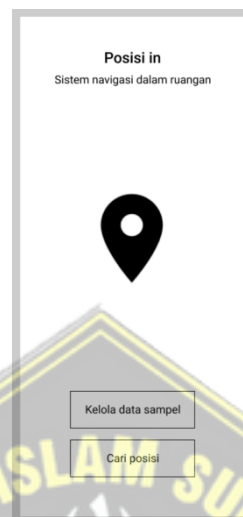
Langkah berikutnya pada proses 2.7 yaitu mencari kumpulan sel tetangga dari sel yang terpilih sebelumnya. Kumpulan sel tetangga tersebut kemudian dilakukan perhitungan skor F, G, H serta diatur atribut *parent*-nya dengan sel yang terpilih sebelumnya. Selanjutnya kumpulan sel tetangga tersebut dimasukkan ke dalam *open-list*.

Proses sebelumnya akan berhenti jika sel tujuan telah ditemukan dan masuk ke dalam *closed-list*. Pada proses 2.12 dilakukan pengecekan pada *closed-list* apakah sel tujuan telah ditemukan atau belum. Jika sel tujuan telah ditemukan, maka dilanjutkan ke proses 2.14 yaitu penelusuran rute menggunakan atribut *parent*. Terakhir rute yang dihasilkan akan ditampilkan pada pengguna pada proses 2.16.



3.8. Perancangan Antarmuka Sistem

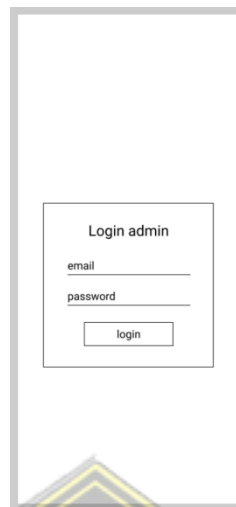
1. Halaman utama



Gambar 3.67 Rancangan antarmuka halaman utama

Pada halaman utama, menampilkan nama dan ikon dari aplikasi. Terdapat dua buah tombol pada bagian bawah yaitu tombol kelola data sampel, dan tombol cari posisi. Tombol kelola data sampel akan mengarahkan ke halaman kelola data sampel, sedangkan tombol cari posisi akan mengarahkan ke halaman pencarian posisi.

2. Halaman *login*



Login admin

email _____

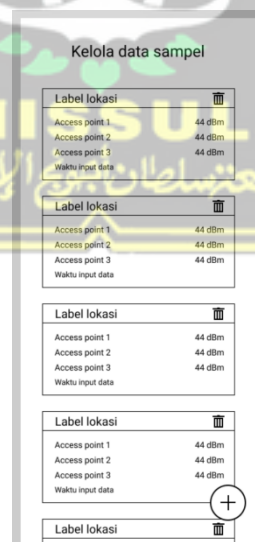
password _____

login

Gambar 3.68 Rancangan antarmuka halaman *login*

Pada halaman *login*, terdapat tampilan *form* untuk *login* admin. Terdapat 2 buah *form email* dan *password* yang harus diisi oleh admin untuk dapat masuk ke halaman kelola data. Terdapat pula 1 buah tombol *login* aksi *login* admin ke aplikasi.

3. Halaman kelola data sampel



Kelola data sampel

Label lokasi	
Access point 1	44 dBm
Access point 2	44 dBm
Access point 3	44 dBm
Waktu input data	

Label lokasi	
Access point 1	44 dBm
Access point 2	44 dBm
Access point 3	44 dBm
Waktu input data	

Label lokasi	
Access point 1	44 dBm
Access point 2	44 dBm
Access point 3	44 dBm
Waktu input data	

Label lokasi	
Access point 1	44 dBm
Access point 2	44 dBm
Access point 3	44 dBm
Waktu input data	

+

Gambar 3.69 Rancangan antarmuka halaman kelola data sampel

Pada halaman kelola data sampel, menampilkan kumpulan informasi dengan tampilan *card-view* untuk setiap data sampel. Informasi yang tertera pada setiap *card-view* adalah label lokasi data sampel, kekuatan sinyal per-*access point*, dan waktu *input* data sampel, serta tombol hapus sampel. Terdapat pula tombol tambah data sampel pada bagian kanan bawah yang akan menampilkan halaman tambah data sampel.



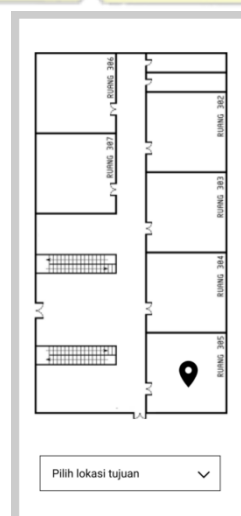
4. Halaman tambah data sampel



Gambar 3.70 Rancangan antarmuka halaman tambah data sampel

Pada halaman tambah data sampel, terdapat 3 buah *form* data kekuatan sinyal (data RSS) dari *access point* 1, 2, dan 3. *Form* ini akan secara otomatis terisi jika pengguna menekan tombol *scan wi-fi*. Terdapat pula *form* pilih lokasi dimana data sampel diambil. Tombol simpan data sampel akan menyimpan data sampel yang telah diinput ke dalam *database*.

5. Halaman pencarian posisi



Gambar 3.71 Rancangan antarmuka halaman pencarian posisi

Pada halaman pencarian posisi menampilkan peta lantai 3 Fakultas Teknologi Industri serta ikon kecil lokasi yang menunjukkan posisi dimana pengguna berada. Terdapat pula *form* pilih lokasi tujuan, untuk pengguna memilih lokasi yang ingin dilakukan pencarian rute ke lokasi tersebut.

6. Halaman navigasi



Gambar 3.72 Rancangan antarmuka halaman navigasi

Halaman pencarian rute merupakan halaman yang sama dengan halaman pencarian posisi, namun dengan tambahan tampilan rute dari posisi pengguna menuju lokasi yang ingin dituju oleh pengguna.

BAB IV

HASIL DAN ANALISIS PENELITIAN

4.1. Implementasi Sistem

4.1.1. Implementasi perangkat keras

Sebuah sistem pemosisian membutuhkan dua komponen utama, yakni komponen *transmitter* dan komponen penghitung. Pada penelitian ini, komponen *transmitter* menggunakan *wi-fi access point*, sedangkan komponen penghitung menggunakan *smartphone* android.

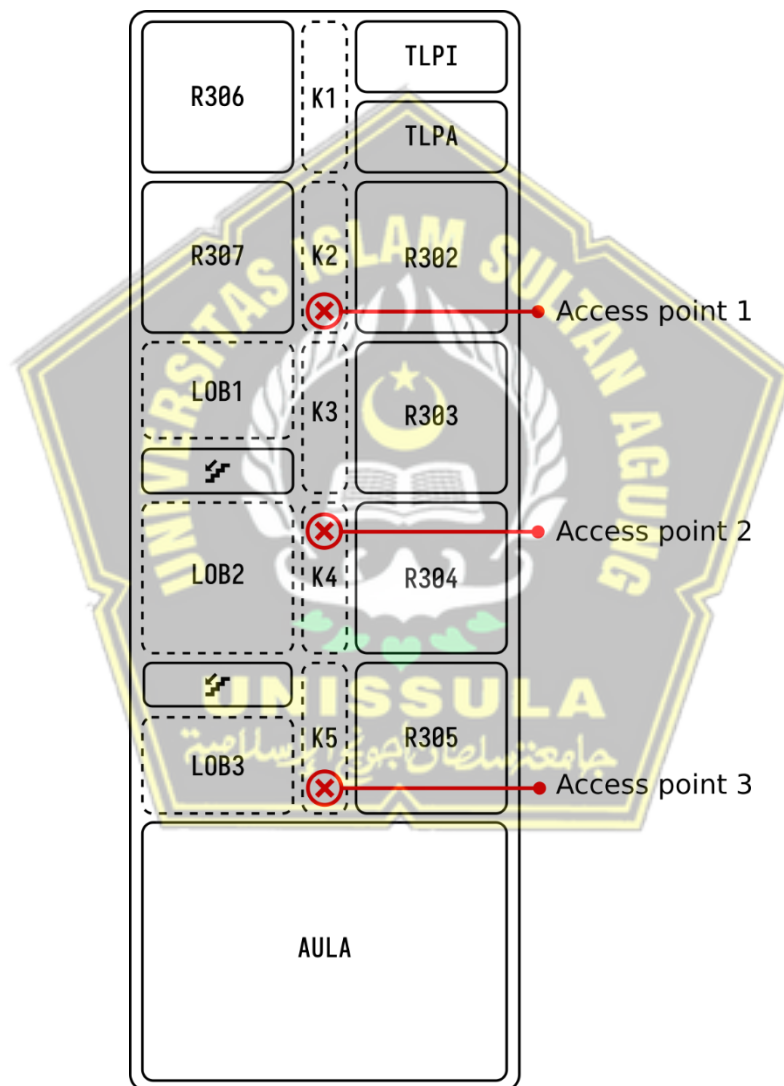
Tabel 4.1 Tabel implementasi perangkat keras sistem

No	Komponen	Perangkat keras	Gambar
1	Transmitter	3 buah ZTE ZXV10 W300S	
2	Penghitung	Samsung Galaxy A20	



4.1.2. Penempatan *transmitter*

Penempatan *transmitter* perlu direncanakan dengan baik agar sinyal *wi-fi* dari ketiga *access point* dapat menjangkau seluruh area pada lantai 3 gedung Fakultas Teknologi Industri. Setelah dilakukan beberapa percobaan variasi penempatan *transmitter*, didapatkan penempatan yang paling optimal yaitu sebagai berikut.



Gambar 4.1 Penempatan *transmitter*

Pada implementasi sistem yang menggunakan 3 *transmitter* tersebut, maka pada aplikasi navigasi perlu diatur nama SSID dari *wi-fi* tiap *transmitter* yang akan digunakan. Contoh pada gambar 4.2 berikut ini merupakan tampilan dari daftar *wi-fi* yang tersedia di sekitar lantai 3 hasil dari pemindaian jaringan *wi-fi* dengan menggunakan aplikasi WiFi Analyzer di android. Jaringan *wi-fi* yang digunakan untuk fungsi pemosisian pada sistem navigasi adalah jaringan dengan nama SSID : “My_Speedy@757C”, “LT3-1”, dan “My_Speedy@B600”.



Gambar 4.2 Pengaturan SSID pada aplikasi navigasi yang akan digunakan oleh sistem untuk pemosisian

4.1.3. Implementasi antarmuka aplikasi sistem

1. Tampilan halaman utama



Gambar 4.3 Tampilan halaman utama

Pada halaman utama antarmuka aplikasi sistem terdapat dua buah tombol utama yaitu tombol untuk deteksi posisi dan tombol pengaturan sistem. Tombol deteksi posisi akan mengarahkan pengguna ke halaman pemosisian. Tombol pengaturan sistem akan menampilkan halaman login untuk admin dapat masuk ke dalam halaman pengaturan sistem.



2. Tampilan halaman login admin



Gambar 4.4 Tampilan halaman login admin

Pada halaman login terdapat dua buah *form* yaitu *form* email dan password yang harus dilengkapi oleh admin agar dapat masuk ke dalam pengaturan sistem.

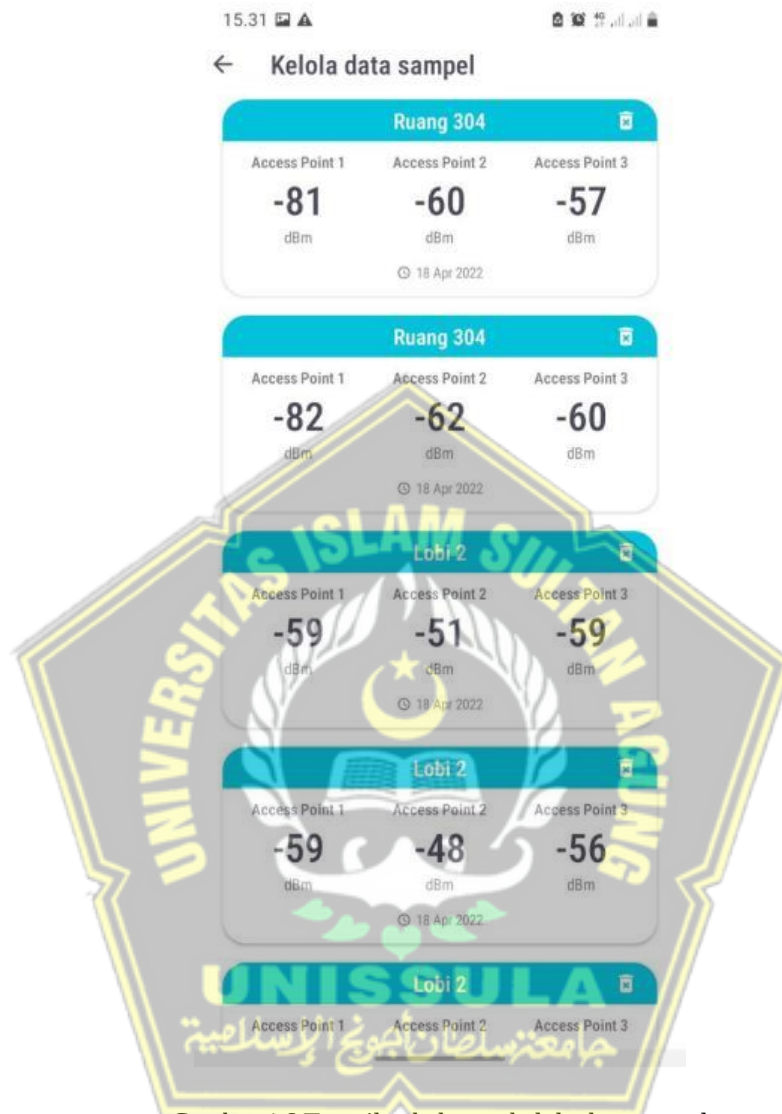
3. Tampilan halaman pengaturan sistem



Gambar 4.5 Tampilan halaman pengaturan sistem oleh admin

Pada halaman pengaturan sistem terdapat empat buah *card view* yang akan membawa admin ke halaman kelola data sampel, tambah data sampel, atur label SSID, dan atur nilai variabel k.

4. Tampilan halaman kelola data sampel



Gambar 4.6 Tampilan halaman kelola data sampel

Pada halaman kelola data sampel, akan menampilkan informasi data sampel yang tersimpan di dalam *database* data sampel *wi-fi*. Informasi yang tertera berupa label lokasi serta level kekuatan sinyal dari *transmitter*. Pada setiap *cardview* sampel terdapat pula tombol hapus untuk menghapus data sampel yang diinginkan.

5. Tampilan halaman tambah data sampel

15.31

← Tambah data sampel

Data level kekuatan sinyal wi-fi

Access point 1

Access point 2

Access point 3

Label lokasi

Scan jaringan wi-fi

Simpan data sampel

Gambar 4.7 Tampilan halaman tambah data sampel

Pada halaman tambah data sampel, terdapat 3 buah *form* level kekuatan sinyal *wi-fi* serta satu buah *form* pilihan label lokasi data sampel. Ketiga *form* level kekuatan sinyal *wi-fi* akan secara otomatis terisi ketika admin menekan tombol *scan jaringan wi-fi*. Jika *form* telah lengkap, tombol *simpan data sampel* akan *upload* data sampel ke *database*.

6. Tampilan halaman pengaturan SSID *access point*



Gambar 4.8 Tampilan halaman pengaturan SSID *access point*

Pada halaman *atur SSID access point*, terdapat 3 buah *form* yang berisi informasi SSID dari 3 *wi-fi access point* yang digunakan oleh sistem. Informasi tersebut dapat diubah oleh admin sesuai dengan kebutuhan sistem.

7. Tampilan halaman pengaturan nilai variabel k

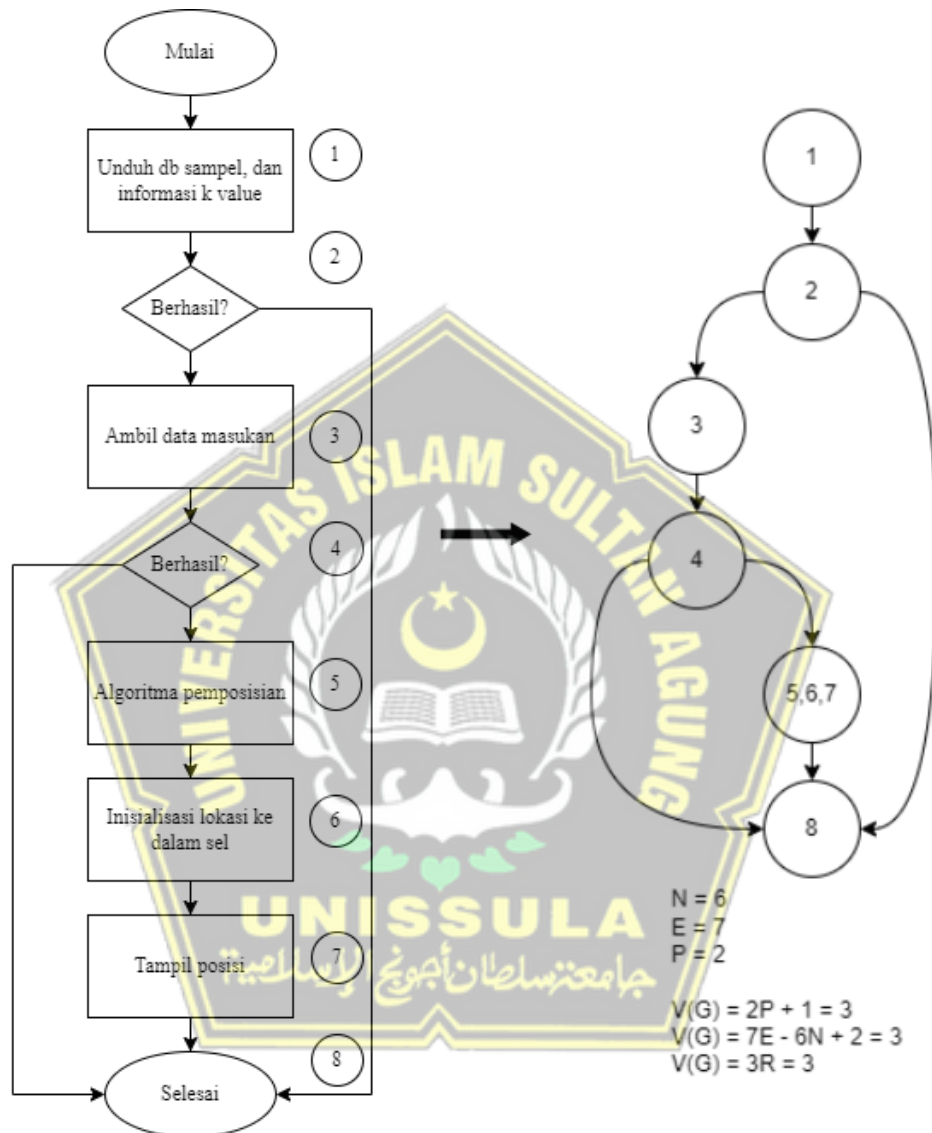


Gambar 4.9 Tampilan halaman pengaturan nilai variabel k

Pada halaman pengaturan nilai variabel k, terdapat sebuah *form* berisi informasi nilai variabel k pada algoritma *k-NN* yang digunakan pada sistem. Informasi tersebut dapat diubah oleh admin sesuai dengan kebutuhan sistem.

4.2. Pengujian Sistem dengan Basis path Testing (White Box)

4.2.1. Pengujian pada fungsi pemposisian



Gambar 4.11 Perhitungan *cyclomatic complexity* pada *flowchart* pemposisian

Pada perhitungan *cyclomatic complexity* di atas, dapat diketahui bahwa terdapat 3 buah jalur independen pada fungsi pemposisian. Rincian jalurnya adalah sebagai berikut :

- Jalur 1 = 1-2-8
- Jalur 2 = 1-2-3-4-8

- Jalur 3 = 1-2-3-4-5-6-7-8

Berikut ini merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

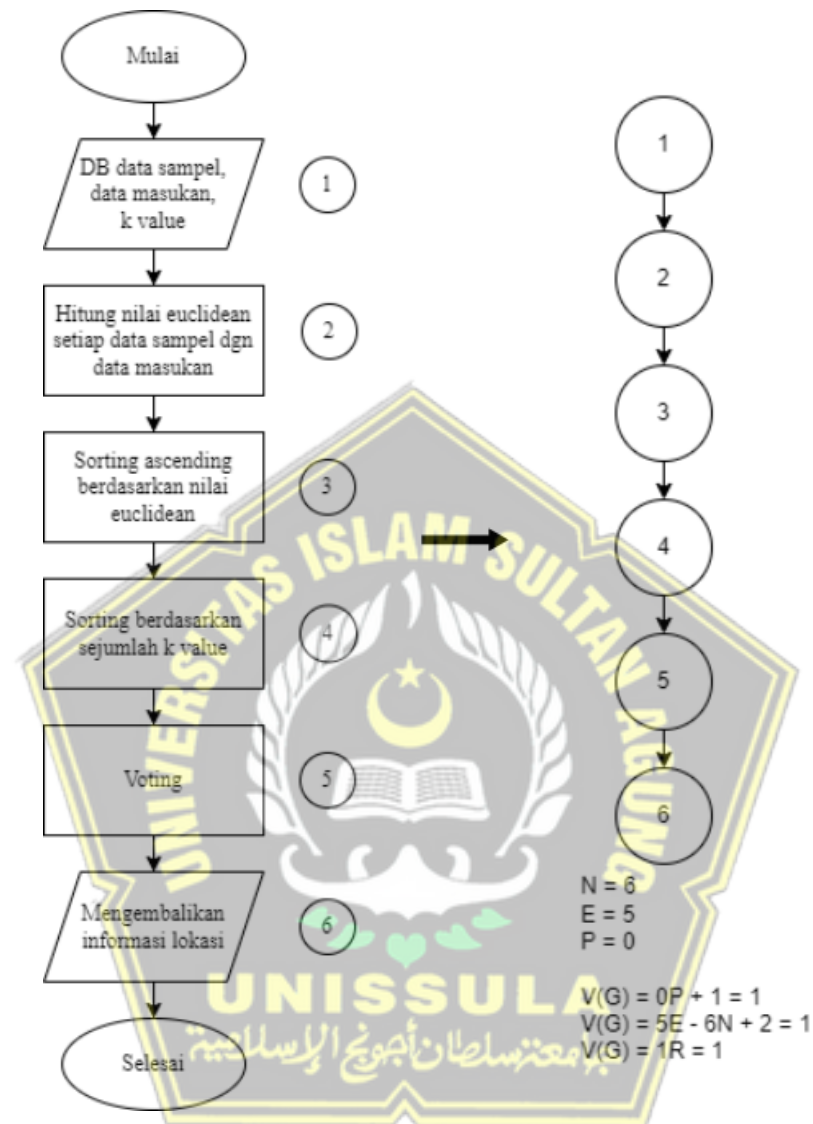
Tabel 4.2 *Test case basis path testing* pada fungsi pemposisian

Jalur 1	Test case	1. Menghapus permission internet pada program 2. Menonaktifkan koneksi internet pada device android
	Expected result	1. Muncul notifikasi tidak dapat mengunduh data sampel 2. Tidak dapat melakukan pemposisian
	Output	1. Muncul notifikasi tidak dapat mengunduh data sampel 2. Tidak dapat melakukan pemposisian
	Validation	<i>Valid</i>
Jalur 2	Test case	1. Mengubah kode program agar tidak melakukan scanning jaringan wi-fi 2. Menghapus permission wi-fi pada program 3. Menonaktifkan layanan wi-fi pada device android
	Expected result	1. Muncul notifikasi data RSS tidak lengkap 2. Tidak dapat melakukan pemposisian
	Output	1. Muncul notifikasi data RSS tidak lengkap 2. Tidak dapat melakukan pemposisian
	Validation	<i>Valid</i>
Jalur 3	Test case	1. Permission dan layanan wi-fi pada device android aktif 2. Database data sampel dapat diunduh 3. Data masukan RSS <i>Valid</i>
	Expected result	1. Berhasil mengambil data masukan 2. Pemposisian berhasil dan muncul lokasi pengguna

	Output	1. Berhasil mengambil data masukan 2. Pemposisian berhasil dan muncul lokasi pengguna
	Validation	<i>Valid</i>



4.2.2. Pengujian *whitebox* pada algoritma pemposisian *k-NN*



Gambar 4.12 Perhitungan *cyclomatic complexity* pada algoritma pemposisian *k-NN*

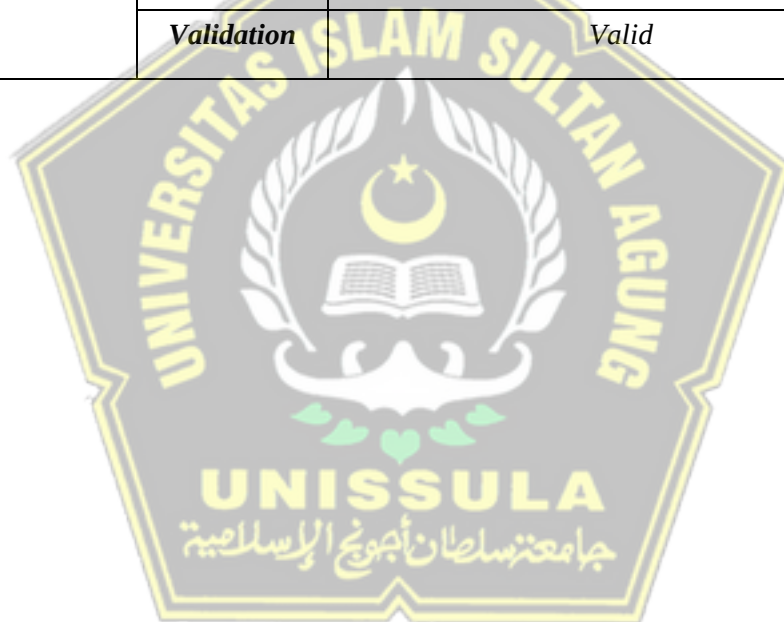
Pada perhitungan *cyclomatic complexity* di atas, dapat diketahui hanya terdapat 1 buah jalur independen pada algoritma pemposisian *k-NN*. Rincian jalurnya adalah sebagai berikut.

-Jalur 1 = 1-2-3-4-5-6-7-8

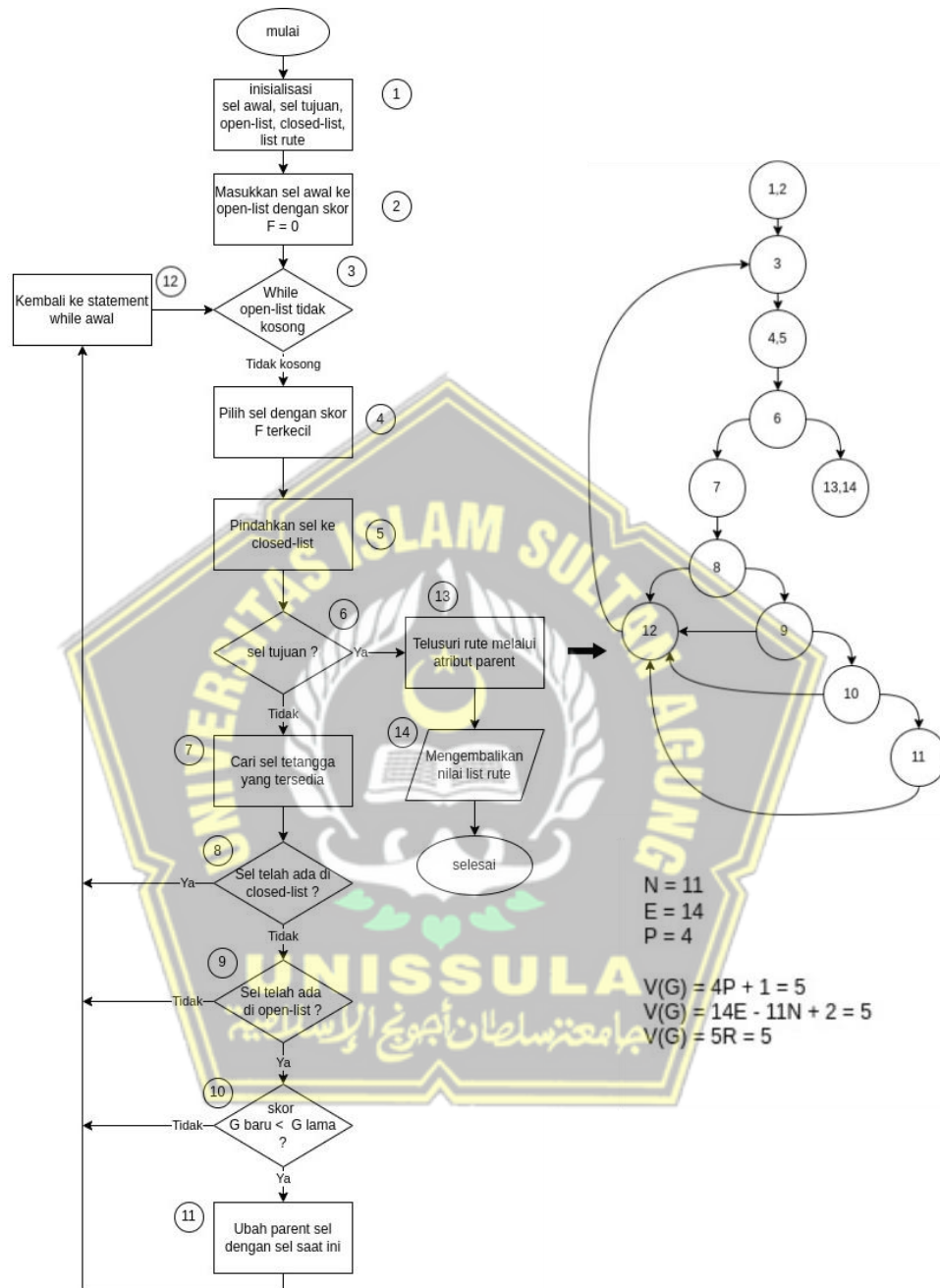
Berikut ini merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

Tabel 4.3 *Test case basis path testing* pada algoritma *k-NN*

Jalur 1	<i>Test case</i>	1. Diberikan data masukan RSS yang <i>Valid</i> sebagai <i>input</i> 2. <i>Database</i> data sampel tersedia
	<i>Expected result</i>	Berhasil menentukan posisi pengguna
	<i>Output</i>	Berhasil menentukan posisi pengguna
	<i>Validation</i>	<i>Valid</i>



4.2.3. Pengujian *whitebox* pada algoritma *pathfinding A**



Gambar 4.13 Perhitungan *cyclomatic complexity* pada algoritma *pathfinding A**

Pada perhitungan *cyclomatic complexity* di atas, diketahui bahwa terdapat 5 buah jalur independen pada algoritma *pathfinding A**. Rincian jalurnya adalah sebagai berikut.

- Jalur 1 = 1-2-3-4-5-6-13-14
- Jalur 2 = 1-2-3-4-5-6-7-8-12-3-4-5-6-13-14
- Jalur 3 = 1-2-3-4-5-6-7-8-9-12-3-4-5-6-13-14
- Jalur 4 = 1-2-3-4-5-6-7-8-9-10-12-3-4-5-6-13-14
- Jalur 5 = 1-2-3-4-5-6-7-8-9-10-11-12-3-4-5-6-13-14

Pada tabel 4.4 berikut ini merupakan *test case* yang diberikan ke program agar setiap jalur independen di atas dapat diketahui *output*-nya.

Tabel 4.4 *Test case basis path testing* pada algoritma *pathfinding A**

Jalur 1	Test case	Diberikan data sel awal dan sel tujuan yang sama
	Expected result	Muncul notifikasi posisi pengguna saat ini dan lokasi tujuan sama
	Output	Muncul notifikasi posisi pengguna saat ini dan lokasi tujuan sama
	Validation	Valid
Jalur 2	Test case	Diberikan data sel awal dan sel tujuan yang bersebelahan
	Expected result	Rute ditemukan dan Valid
	Output	Rute ditemukan dan Valid
	Validation	Valid
Jalur 3	Test case	Diberikan data sel awal dan tujuan yang disesuaikan
	Expected result	Rute ditemukan dan Valid
	Output	Rute ditemukan dan Valid

	Validation	<i>Valid</i>
Jalur 4	Test case	Diberikan data sel awal dan tujuan yang disesuaikan
	Expected result	Rute ditemukan dan <i>Valid</i>
	Output	Rute ditemukan dan <i>Valid</i>
	Validation	<i>Valid</i>
Jalur 5	Test case	Diberikan data sel awal dan tujuan yang disesuaikan
	Expected result	Rute ditemukan dan <i>Valid</i>
	Output	Rute ditemukan dan <i>Valid</i>
	Validation	<i>Valid</i>

4.3. Pengujian Akurasi Pemposisian dengan Nilai Variabel “k” Berbeda

Untuk mengetahui tingkat efektifitas kinerja dari algoritma *k*-NN dalam penentuan posisi pada sistem, maka perlu dilakukan pengujian dengan penggunaan variasi dari nilai variabel *k*. Hasil dari pengujiannya adalah sebagai berikut.

Tabel 4.1 Pengujian akurasi pemposisian dengan nilai variabel *k*=3

Variabel k = 3		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Tidak Konsisten
9	Lobi2	Tidak Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten

12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Tidak Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Keterangan :

- Konsisten = Hasil pengujian benar dan hasil konsisten setelah beberapa kali pengujian
- Tidak Konsisten = Hasil pengujian tidak konsisten setelah beberapa kali pengujian

Hasil pengujian dengan nilai variabel $k=3$ menghasilkan beberapa hasil yang tidak konsisten yaitu lokasi Lobi 1, Lobi 2, Koridor 3, dan Koridor 4. Diketahui bahwa keempat posisinya saling berdekatan serta tidak terhalang oleh dinding satu sama lain, sehingga ketiga lokasi ini memiliki beberapa data sampel *wi-fi* yang datanya mirip. Berbeda dengan lokasi lainnya yang terlingkupi oleh dinding dan mengakibatkan variasi data yang sangat berbeda di setiap lokasinya.

Tabel 4.2 Pengujian akurasi pemposisian dengan nilai variable $k=4$

Variabel $k = 4$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten

6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Tidak Konsisten
9	Lobi2	Tidak Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Hasil pengujian akurasi pemposisian dengan nilai variabel $k=4$, masih menghasilkan beberapa output yang tidak konsisten. Ada 3 lokasi yang hasil pengujianjnya masih tidak konsisten yaitu Lobi 1, Lobi 2, dan Koridor 4. Ketiga lokasi tersebut kondisinya sama seperti pengujian sebelumnya yaitu posisinya yang saling berdekatan serta tidak terhalang oleh dinding.

Tabel 4.3 Pengujian akurasi pemposisian dengan nilai variabel $k=5$

Variabel $k = 5$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Tidak Konsisten

9	Lobi2	Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten
13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Tidak Konsisten
16	Koridor5	Konsisten

Hasil pengujian dengan nilai variabel $k = 5$ masih menghasilkan beberapa lokasi yang hasil pemposisiannya tidak konsisten. Beberapa lokasi tersebut yaitu Lobi 1 dan Koridor 4. Kedua lokasi ini masih berlokasi sama seperti pengujian sebelumnya, serta saling berdekatan dan tanpa ada sekat dinding.

Tabel 4.4 Pengujian akurasi pemposisian dengan nilai variabel $k=6$

Variabel $k = 6$		
No	Lokasi	Hasil Pengujian
1	Ruang 302	Konsisten
2	Ruang 303	Konsisten
3	Ruang 304	Konsisten
4	Ruang 305	Konsisten
5	Ruang 306	Konsisten
6	Ruang 307	Konsisten
7	Aula	Konsisten
8	Lobi1	Konsisten
9	Lobi2	Konsisten
10	Lobi3	Konsisten
11	Toilet	Konsisten
12	Koridor1	Konsisten

13	Koridor2	Konsisten
14	Koridor3	Konsisten
15	Koridor4	Konsisten
16	Koridor5	Konsisten

Hasil pengujian pemposisian dengan nilai variabel $k = 6$ sangat baik. Semua lokasi menghasilkan *output* yang tepat dan konsisten. Dengan hasil dari rangkaian pengujian pemposisian dengan variasi nilai variabel k di atas, maka dapat diambil kesimpulan bahwa menggunakan nilai variabel k yang cenderung besar akan menghasilkan hasil pemposisian yang semakin baik atau konsisten.



BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil pengkajian mulai dari Bab I hingga Bab IV, maka penulis dapat mengambil langkah kesimpulan antara lain :

1. Pada penelitian ini peneliti dapat membangun prototipe sistem navigasi dalam ruangan dengan mengimplementasikan metode *wi-fi fingerprint positioning* dengan algoritma *k-NN* untuk pemposisian, dan algoritma *pathfinding A** untuk pencarian rute dengan hasil sistem dapat melakukan penentuan posisi dan melakukan pencarian rute dengan baik.
2. Dari hasil pengujian kinerja penentuan posisi dengan variasi nilai variabel *k* yaitu *k=3*, *k=4*, *k=5*, dan *k=6*, didapatkan nilai variabel *k=6* yang memiliki hasil penentuan posisi yang konsisten. Serta dapat disimpulkan bahwa penggunaan nilai variabel *k* yang cenderung besar menghasilkan *output* yang lebih konsisten.
3. Dengan adanya sistem ini dapat memudahkan pengunjung baru suatu public area dalam menelusuri lokasi-lokasi yang berada dalam lingkup public area khususnya pada area *indoor*.

5.2. Saran

Saran yang sekiranya peneliti dapat berikan untuk membantu penelitian selanjutnya dalam membangun sistem navigasi dalam ruangan berbasis *wi-fi fingerprint positioning* adalah penggunaan algoritma yang lebih mutakhir dari algoritma *k-NN* yang memungkinkan sistem mendapatkan akurasi pemposisian yang baik meskipun dengan pengambilan data sampel yang minimal.

DAFTAR PUSTAKA

- Bai, Brian Yuntian, Suqin Wu, Guenther Retscher, dkk. 2014. “A new method for improving Wi-Fi based indoor positioning accuracy”. *Journal of Location Based Services*. 8(3). 135-147.
- Cahyadi, M.A., Bambang, M. Arif, dkk. 2017. “Perbandingan Algoritma A*, Dijkstra dan Floyd Warshall Untuk Menentukan Jalur Terpendek Pada Permainan *Bacteria Defense*”. *Teknik Informatika*. STIMIK GI MDP. Palembang.
- Deng, Z.-L., Yu, Y.-P., Xu, L.-M. 2013. “*Wireless Location and Navigation in Indoor and Outdoor*”. Beijing University of Posts and Telecommunications Press. China.
- Lester, Patrick. 2005. “*A* Pathfinding for Beginners*”. Musthafa, A.R., 2016. “Sistem Navigasi Indoor Menggunakan Sinyal Wi-Fi dan Kompas Digital Berbasis Integrasi dengan Smartphone untuk Studi Kasus pada Gedung Bertingkat”. *Jurnal Teknik ITS*. 5(2). A448-A452.
- Pasinggi, Eko Suropto, Srivan Paleleng, Ferayanti Boas Gallaran. 2018. “*Literature Riview: Arsitektur Sistem Penentuan Posisi di Dalam Ruang*”. *Journal Dynamic Saint*. 3(2). 672-683
- Permana, D.Y., Handojo, A. and Andjarwirawan, J., 2013. “Aplikasi Indoor Positioning System Menggunakan Android dan Wireless Local Area Network Dengan Metode Fuzzy Logic Indoor Positioning System”. *Jurnal Infra*. 1(2). 13-18.
- Suyanto. 2018. “*Machine Learning Tingkat Dasar dan Lanjut*”. Informatika. BI-OBSES. Bandung Xia, S., Liu, Y., Yuan, G., Zhu, M. and Wang, Z., 2017. “*Indoor Fingerprint Positioning Based on Wi-Fi: An Overview*”. *ISPRS International Journal of Geo-Information*, 6(5).
- Xiao Cui, Hao Shi. 2011 “*A*-based Pathfinding in Modern Computer Games*”. *IJCSNS International Journal of Computer Science and Network Security*. 11(1). 125-130.