



TUGAS AKHIR - KI141502

PENGUNAAN QR CODE UNTUK MENUNJUKKAN POSISI DAN IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE TERPENDEK PADA NAVIGASI DALAM RUANGAN BERBASIS SISTEM OPERASI ANDROID

PRASETYO
NRP 5112 100 143

Dosen Pembimbing
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Abdul Munif, S.Kom., M.Sc.

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

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - KI141502

**PENGUNAAN QR CODE UNTUK
MENUNJUKKAN POSISI DAN IMPLEMENTASI
ALGORITMA DIJKSTRA DALAM PENCARIAN
RUTE TERPENDEK PADA NAVIGASI DALAM
RUANGAN BERBASIS SISTEM OPERASI
ANDROID**

PRASETYO
NRP 5112 100 143

Dosen Pembimbing
Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Abdul Munif, S.Kom., M.Sc.

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

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - KI141502

***QR CODE ASSISTED POSITIONING SYSTEM
WITH SHORTEST ROUTE
IMPLEMENTATION USING DIJKSTRA'S
ALGORITHM ON ANDROID OPERATING
SYSTEM***

PRASETYO
NRP 5112 100 143

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

Supervisor II
Abdul Munif, S.Kom., M.Sc.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

PENGUNAAN QR CODE UNTUK MENUNJUKKAN POSISI DAN IMPLEMENTASI ALGORITMA DIJKSTRA DALAM PENCARIAN RUTE TERPENDEK PADA NAVIGASI DALAM RUANGAN BERBASIS SISTEM OPERASI ANDROID

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Algoritma Pemrograman
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:

PRASETYO

NRP. 5112 100 143

Disetujui oleh Pembimbing Tugas Akhir

1. Dr.tech. Ir. R.V. Hari Ginard, M.Sc.
NIP: 19650518 199203 1 004 (Pembimbing 1)
2. Abdul Munif, S.Kom., M.Sc.
NIP: 19860823 201504 1 004 (Pembimbing 2)

**SURABAYA
JUNI, 2016**

[Halaman ini sengaja dikosongkan]

**PENGUNAAN QR CODE UNTUK MENUNJUKKAN
POSISI DAN IMPLEMENTASI ALGORITMA DIJKSTRA
DALAM PENCARIAN RUTE TERPENDEK PADA
NAVIGASI DALAM RUANGAN BERBASIS SISTEM
OPERASI ANDROID**

Nama Mahasiswa : Prasetyo
NRP : 5112 100 143
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Dr.tech. Ir. R.V. Hari Ginardi, M.Sc.
Dosen Pembimbing 2 : Abdul Munif, S.Kom., M.Sc.

ABSTRAK

GPS memiliki manfaat yang sangat besar dalam navigasi dan penentuan posisi, namun teknologi ini tetap memiliki kelemahan, yaitu akurasi yang sangat rendah saat pengguna berada dalam sebuah gedung bertingkat. GPS hanya dapat menunjukkan lokasi gedung tersebut, namun tidak dengan lokasi dari pengguna, dimana pengguna berada, pada ruangan mana, di lantai berapa. Hal ini dikarenakan GPS memiliki konsep 2D Localization dimana GPS tidak dapat menentukan ketinggian pengguna. Oleh karena itu, dikembangkan sebuah sistem yang lebih akurat untuk memberikan solusi untuk navigasi dalam gedung bertingkat yang dikembangkan dengan konsep 3D Indoor Localization.

Dengan menggunakan QR Code untuk menentukan posisi pengguna dalam suatu gedung bertingkat dan penggunaan algoritma Dijkstra dalam mencari rute terpendek ke suatu tujuan, maka Aplikasi ini menyediakan sebuah layanan navigasi dalam ruangan yang lebih baik dari penggunaan GPS. Untuk mengetahui posisinya, pengguna hanya perlu melakukan pemindaian terhadap QR Code yang terdapat pada sebuah ruang. Pengguna selanjutnya dapat memilih lokasi tujuan yang dikehendaki, dan aplikasi akan

mencari rute yang dapat ditempuh untuk sampai ke tujuan dengan jarak terpendek. Pengguna dapat memanfaatkan panel navigasi yang disediakan untuk membantu pengguna tiba di tujuan. Jika sewaktu-waktu pengguna ingin melakukan pembaruan posisi, maka pengguna hanya perlu menekan tombol berlogo QR dan melakukan proses pemindaian ulang QR.

Uji coba dilakukan menggunakan peta 3D dari Kampus Teknik Informatika ITS. Hasil yang didapat sesuai dengan yang diharapkan. Aplikasi dapat melakukan semua prosesnya dengan baik, dimulai dari proses pemindaian hingga proses penunjukan rute ke tujuan. Aplikasi juga dapat melakukan proses pengupdate-an posisi dengan baik. Aplikasi juga dapat dijalankan pada beberapa jenis smartphone yang berbeda dengan baik.

Kata kunci: Android, 3D Indoor Localization, QR Code, Dijkstra

**QR CODE ASSISTED POSITIONING SYSTEM
WITH SHORTEST ROUTE
IMPLEMENTATION USING DIJKSTRA'S
ALGORITHM ON ANDROID OPERATING
SYSTEM**

Student's Name : Prasetyo
Student's ID : 5112 100 143
Department : Informatics Engineering, FTIF-ITS
Supervisor I : Dr.tech. Ir. R. V. Hari Ginardi, M.Sc.
Supervisor II : Abdul Munif, S.Kom., M.Sc.

ABSTRACT

GPS has enormous benefits for navigation and positioning, but this technology still has a low accuracy when the user is in a multistory building. GPS can only indicate the location of the building, but not the location of the user, where the user is, in the room where, the floor of the room. This is because GPS has 2D concept Localization where GPS can not determine the height of the user. Therefore, a more accurate system to provide a solution for navigation in a multi-storey building is developed using 3D concept Indoor Localization.

By using the QR Code to determine the user's position in a high rise building and use Dijkstra's algorithm to find the shortest route to a destination, then this app provides a indoor navigation better than GPS. To determine its position, the user only needs to scan the QR Code that found at the door of the room. Then users can select the destination, and the application will find the route to the destination. Users can use the navigation panel that is provided to help users arrive at the destination. If user wants to update the

position, then the user just needs to press the logo button QR and the application will rescan the qr code.

The test is done using a 3D Map of the Campus Information Engineering of ITS. The results as expected, applications can perform all the process well, starting from the scanning process until the appointment of a route to the destination. Applications can also perform an updating process of the position well. Applications can also be run on several different types of smartphones well.

Keywords: Android, 3D Indoor Localization, QR Code, Dijkstra

KATA PENGANTAR

Puji syukur kepada Tuhan Yesus Kristus atas kasih dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Penggunaan QR Code untuk Menunjukkan Posisi dan Implementasi Algoritma Dijkstra dalam Pencarian Rute Terpendek pada Navigasi dalam Ruang berbasis Sistem Operasi Android”**.

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

1. Tuhan Yesus Kristus atas segala kasih dan hikmat yang telah diberikan kepada penulis selama ini.
2. Kedua orang tua penulis, Semy Lekal dan Nona Tan, yang tiada henti mencurahkan kasih sayang, perhatian, doa, dukungan, dan semangat kepada penulis selama ini.
3. Ketiga saudara kandung penulis, Tresia Lekal, Nafly Lekal, dan Stenly, yang telah memberikan dukungan dan doa kepada penulis dalam menuntut ilmu hingga detik ini.
4. Bapak Hari Ginardi dan Bapak Abdul Munif selaku dosen pembimbing yang telah memberikan bimbingan, motivasi, dan meluangkan waktu untuk membantu pengerjaan tugas akhir ini.
5. Bapak dan Ibu dosen Teknik Informatika ITS yang telah membina dan memberikan ilmu kepada penulis selama menempuh studi di Teknik Informatika ITS.
6. Sahabat dekat yang selalu memberikan dukungan, inspirasi dan semangat untuk berkarya: Anton, Andre, Ardi, Felix, Jesse, Kenny, Luis, Richard, Ryan, Vicky, William.

7. Keluarga besar KGC City Church Surabaya yang sudah memberikan doa, dukungan yang tiada henti, dan semangat untuk pantang menyerah.
8. Keluarga besar angkatan tercinta TC 2012 yang sudah menemani keseharian penulis di kampus perjuangan.
9. Serta pihak-pihak lain yang namanya tidak dapat penulis sebutkan satu-persatu.

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

Surabaya, Juni 2016

Prasetyo

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
DAFTAR KODE SUMBER.....	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	3
1.4 Tujuan dan Manfaat	3
1.5 Metodologi.....	4
1.6 Sistematika Penyusunan Laporan	5
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Indoor Positioning System</i>	7
2.2 QR Code	8
2.3 Android	10
2.4 Algoritma Dijkstra	12
BAB III PERANCANGAN PERANGKAT LUNAK.....	17
3.1 Deskripsi Umum Perangkat Lunak.....	17
3.1.1 Arsitektur Sistem.....	18
3.1.2 Kebutuhan Fungsional Perangkat Lunak.....	19
3.2 Perancangan	35

3.2.1	Perancangan Peta 3D.....	35
3.2.2	Perancangan Basis Data	40
3.2.3	Perancangan Antarmuka Aplikasi Perangkat Bergerak 44	
BAB IV	IMPLEMENTASI PERANGKAT LUNAK	51
4.1	Lingkungan Implementasi	51
4.1.1	Lingkungan Implementasi Perangkat Keras.....	51
4.1.2	Lingkungan Implementasi Perangkat Lunak.....	52
4.2	Implementasi Basis Data.....	52
4.2.1	Implementasi Tabel Map.....	52
4.2.2	Implementasi Tabel Vertex	52
4.2.3	Implementasi Tabel Edge.....	53
4.3	Implementasi Antarmuka.....	53
4.3.1	Antarmuka Halaman Utama.....	53
4.3.2	Antarmuka Halaman Pemindaian QR Code.....	54
4.3.3	Antarmuka Halaman untuk Memilih Tujuan dan Melihat Rute.....	55
4.3.4	Antarmuka Halaman About.....	57
4.3.5	Antarmuka Halaman Ketika Pengguna Telah Tiba di Tujuan.....	58
4.3.6	Antarmuka Halaman untuk <i>Update</i> Posisi Pengguna 59	
4.3.7	Antarmuka Halaman untuk Menampilkan Pesan <i>Error</i>	60
BAB V	UJI COBA DAN EVALUASI	61
5.1	Lingkungan Uji Coba.....	61
5.2	Dasar Pengujian	61

5.3	Pengujian Fungsionalitas	62
5.3.1	Pengujian Melakukan Pemindaian QR Code	62
5.3.2	Pengujian Menampilkan Lokasi Pengguna	64
5.3.3	Pengujian Menampilkan Daftar Lokasi Tujuan	65
5.3.4	Pengujian Memilih Lokasi Tujuan	67
5.3.5	Pengujian Menampilkan Rute ke Tempat Tujuan ...	69
5.3.6	Pengujian Melakukan Proses Navigasi	71
5.3.7	Pengujian Melakukan Pembaruan Posisi Pengguna	73
5.3.8	Pengujian Menampilkan Ringkasan Jalur yang Dilewati.....	75
5.3.9	Pengujian Pada <i>Smartphone</i>	77
5.4	Evaluasi Pengujian.....	77
5.4.1	Evaluasi Pengujian Fungsionalitas	77
5.4.2	Evaluasi Pengujian <i>Smartphone</i>	79
BAB VI KESIMPULAN DAN SARAN		81
6.1	Kesimpulan	81
6.2	Saran	82
DAFTAR PUSTAKA		83
LAMPIRAN A – IMPLEMENTASI FUNGSI PADA ANTARMUKA.....		85
LAMPIRAN B – KODE SUMBER.....		99
LAMPIRAN C – QR CODE.....		101
BIODATA PENULIS		117

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Struktur QR Code.....	9
Gambar 2.2 Versi QR Code.....	10
Gambar 3.1 Alur Jalannya Perangkat Lunak QR MAP.....	18
Gambar 3.2 Arsitektur Sistem.....	18
Gambar 3.3 <i>Use Case Diagram</i>	20
Gambar 3.4 Diagram Aktivitas UC-01.....	22
Gambar 3.5 Diagram Alir UC-01.....	23
Gambar 3.6 Diagram Aktivitas UC-02.....	24
Gambar 3.7 Diagram Alir UC-02.....	25
Gambar 3.8 Diagram Aktivitas UC-03.....	26
Gambar 3.9 Diagram Alir UC-03.....	27
Gambar 3.10 Diagram Aktivitas UC-04.....	28
Gambar 3.11 Diagram Alir UC-04.....	28
Gambar 3.12 Diagram Aktivitas UC-05.....	29
Gambar 3.13 Diagram Alir UC-05.....	30
Gambar 3.14 Diagram Aktivitas UC-06.....	31
Gambar 3.15 Diagram Alir UC-06.....	31
Gambar 3.16 Diagram Aktivitas UC-07.....	33
Gambar 3.17 Diagram Alir UC-07.....	33
Gambar 3.18 Diagram Aktivitas UC-08.....	34
Gambar 3.19 Diagram Alir UC-08.....	35
Gambar 3.20 Perancangan <i>vertex</i> lantai 1.....	37
Gambar 3.21 Perancangan <i>vertex</i> lantai 2.....	38
Gambar 3.22 Perancangan <i>vertex</i> lantai 3.....	39
Gambar 3.23 <i>Physical Data Model</i>	40
Gambar 3.24 <i>Conceptual Data Model</i>	41
Gambar 3.25 Perancangan Antarmuka Halaman Utama.....	44
Gambar 3.26 Perancangan Halaman Pemindaian QR Code.....	45
Gambar 3.27 Perancangan Halaman Memilih Tujuan dan Melihat Rute.....	46
Gambar 3.28 Perancangan Antarmuka Halaman About.....	47

Gambar 3.29 Perancangan Halaman Ketika Pengguna Telah Tiba di Tujuan	48
Gambar 3.30 Perancangan Halaman untuk <i>Update</i> Posisi Pengguna.....	49
Gambar 3.31 Perancangan Halaman untuk Menampilkan Pesan <i>Error</i>	50
Gambar 4.1 Implementasi Antarmuka Halaman Utama	54
Gambar 4.2 Implementasi Halaman Pemindaian QR Code.....	55
Gambar 4.3 (a) Implementasi Halaman untuk Memilih Tujuan, (b)Implementasi Halaman untuk Menampilkan Rute	56
Gambar 4.4 Implementasi Antarmuka Halaman About.....	57
Gambar 4.5 Implementasi Halaman Ketika Pengguna Telah Tiba di Tujuan	58
Gambar 4.6 Implementasi Halaman untuk Update Posisi Pengguna.....	59
Gambar 4.7 Antarmuka Halaman untuk Menampilkan Pesan <i>Error</i>	60
Gambar 5.1 Pengujian Melakukan Pemindaian QR Code	63
Gambar 5.2 Pengujian Menampilkan Lokasi Pengguna	65
Gambar 5.3 Pengujian Menampilkan Lokasi Tujuan	67
Gambar 5.4 Pengujian Memilih Lokasi Tujuan.....	69
Gambar 5.5 Menampilkan Rute ke Tempat Tujuan.....	71
Gambar 5.6 Pengujian Melakukan Proses Navigasi	73
Gambar 5.7 Pengujian Melakukan Pembaruan Posisi Pengguna	75
Gambar 5.8 Pengujian Menampilkan Jalur yang Dilewati	76

DAFTAR TABEL

Tabel 3.1 Deskripsi Tabel Penggunaan.....	20
Tabel 3.2 Rincian Alur Kasus Penggunaan UC-01.....	22
Tabel 3.3 Rincian Alur Kasus Penggunaan UC-02.....	24
Tabel 3.4 Rincian Alur Kasus Penggunaan UC-03.....	25
Tabel 3.5 Rincian Alur Kasus Penggunaan UC-04.....	27
Tabel 3.6 Rincian Alur Kasus Penggunaan UC-05.....	29
Tabel 3.7 Rincian Alur Kasus Penggunaan UC-06.....	30
Tabel 3.8 Rincian Alur Kasus Penggunaan UC-07.....	32
Tabel 3.9 Rincian Alur Kasus Penggunaan UC-08.....	34
Tabel 3.10 Keterangan Simbol pada Perancangan <i>Vertex</i>	36
Tabel 3.11 Atribut Tabel Map	41
Tabel 3.12 Atribut Tabel Vertex	42
Tabel 3.13 Atribut Tabel Edge.....	43
Tabel 5.1 Spesifikasi Lingkungan Pengujian Perangkat Lunak..	61
Tabel 5.2 Skenario Pengujian Melakukan Pemindaian QR Code	62
Tabel 5.3 Skenario Pengujian Menampilkan Lokasi Pengguna..	64
Tabel 5.4 Skenario Pengujian Menampilkan Lokasi Tujuan	66
Tabel 5.5 Skenario Pengujian Memilih Lokasi Tujuan	68
Tabel 5.6 Skenario Menampilkan Rute ke Tempat Tujuan	70
Tabel 5.7 Skenario Melakukan Proses Navigasi.....	72
Tabel 5.8 Skenario Melakukan Pembaruan Posisi Pengguna	74
Tabel 5.9 Skenario Menampilkan Ringkasan Jalur yang Dilewati	75
Tabel 5.10 Perangkat <i>Smartphone</i> Pengujian	77
Tabel C.1 Daftar QR Code yang Digunakan	101

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 2.1 <i>Pseudocode</i> Algoritma Dijkstra	14
Kode Sumber 2.2 <i>Pseudocode</i> untuk Mendapatkan Jalur Terpendek dari Algoritma Dijkstra	15
Kode Sumber A.1 Implementasi Algoritma Dijkstra.....	86
Kode Sumber A.2 Implementasi <i>Query</i> ke Basis Data	89
Kode Sumber A.3 Implementasi Fungsi untuk Menampilkan Posisi Pengguna	89
Kode Sumber A.4 Implementasi Fungsi untuk Menampilkan Garis Penunjuk Rute yang Dilalui	90
Kode Sumber A.5 Implementasi Fungsi Pembacaan QR Code ..	93
Kode Sumber A.6 Implementasi Fungsi untuk <i>Update</i> Posisi Kamera.....	96
Kode Sumber A.7 Implementasi Fungsi untuk Panel Navigasi ..	98
Kode Sumber B.1 Implementasi Tabel Map.....	99
Kode Sumber B.2 Implementasi Tabel Vertex	99
Kode Sumber B.3 Implementasi Tabel Edge.....	100

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sejak dimulainya revolusi industri di Britania Raya pada pertengahan abad ke-18, semua negara di dunia berlomba-lomba untuk saling memamerkan kehebatannya dalam setiap bidang kehidupan, tidak terlupeut bidang konstruksi. Gedung-gedung berukuran besar mulai dibangun di hampir seluruh belahan dunia. Dengan didukung dengan perkembangan ilmu pengetahuan yang pesat, gedung yang dibangun tidak hanya berukuran besar namun juga memiliki tinggi yang menakjubkan. Keterbatasan lahan yang ada menyebabkan banyak gedung dibangun dengan beberapa fungsi sekaligus seperti perkantoran, pusat perbelanjaan, pusat hiburan, tempat tinggal, dan lain sebagainya. Oleh karena itu, gedung dibangun dengan memiliki banyak tingkat dan dibagi menjadi bagian-bagian tertentu sesuai dengan fungsinya masing-masing. Dalam suatu gedung bertingkat, dibutuhkan pedoman lokasi yang dapat digunakan oleh pengunjung untuk mengetahui letak lokasi yang akan dikunjungi. Untuk membantu pengunjung mengetahui lokasi, umumnya oleh pihak pengembang gedung dibuatkan sebuah peta statik dan menggunakan kode ruang untuk membedakan ruangan yang ada dalam gedung tersebut. Namun, terkadang pengunjung membutuhkan waktu yang lama untuk mencari lokasi dari peta statik ini.

Untuk mengatasi masalah ini, maka dibuatlah sebuah perangkat lunak berupa navigasi dalam ruangan atau gedung. Selama ini, perangkat lunak navigasi selalu identik dengan GPS (*Global Positioning System*) yang menggunakan satelit sebagai alat bantu penunjuk posisi. Namun, beberapa tahun belakangan ini, mulai dibuat navigasi untuk gedung atau dikenal dengan sebutan IPS (*Indoor Positioning System*). Sama seperti GPS, IPS juga memiliki dua masalah utama yaitu bagaimana mengetahui posisi saat ini dan jalur manakah yang harus ditempuh untuk sampai ke

tujuan. IPS yang sudah pernah dibuat adalah IPS yang menggunakan sinyal WiFi untuk mengetahui posisi pengguna dalam ruangan. Pada tugas akhir ini, akan diambil pendekatan yang lain, yaitu dengan menggunakan QR Code untuk mengetahui posisi saat ini. QR Code (*Quick Response Code*) merupakan kode matriks dua dimensi yang dapat menyimpan informasi secara horizontal dan vertikal. Agar dapat membaca QR Code diperlukan sebuah pemindai QR. Telepon genggam jaman sekarang sudah memiliki kemampuan untuk memindai QR Code. QR Code ini akan dimasukkan informasi yang dibutuhkan, kemudian ditempatkan pada posisi yang diinginkan. QR Code yang telah dimasukkan informasi inilah yang nantinya akan digunakan sebagai penunjuk posisi sekarang.

Dalam tugas akhir ini, perangkat lunak yang dibuat, diharapkan dapat menunjukkan posisi pengguna saat ini pada peta yang digunakan berdasarkan hasil pemindaian QR Code. Perangkat lunak juga diharapkan dapat menampilkan informasi dari QR Code yang dipindai dan menampilkan daftar tujuan yang tersedia. Setelah itu, perangkat lunak yang dibuat juga diharapkan dapat menentukan jalur yang harus ditempuh berdasarkan algoritma Dijkstra dan dapat menampilkan jalur tersebut pada peta agar pengguna dapat tiba di tujuan.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana mengetahui posisi pengguna saat ini ketika berada pada suatu gedung atau bangunan bertingkat dengan menggunakan QR Code?
2. Bagaimana caranya untuk sampai ke suatu ruang atau tempat tertentu dari posisi saat ini pada suatu gedung

bertingkat dengan menggunakan bantuan dari QR Code dan algoritma Dijkstra?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Perangkat lunak hanya dapat menampilkan posisi awal, rute yang akan dilalui dan tujuan.
2. Rute yang ditampilkan hanya berupa rute terpendek.
3. Peta bangunan yang digunakan merupakan peta dari gedung Teknik Informatika ITS.
4. Pemindai QR dibuat dengan memanfaatkan kamera pada telepon genggam dan pustaka yang telah tersedia.

1.4 Tujuan dan Manfaat

Tujuan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Membuat perangkat lunak berbasis Android yang dapat menunjukkan posisi pengguna saat ini, menampilkan daftar tujuan yang tersedia dan menampilkan rute yang terpendek ke tempat tujuan dalam suatu gedung.

Manfaat yang diharapkan dari dibuatnya perangkat lunak ini adalah agar dapat membantu masyarakat awam dalam menentukan posisinya dan rute yang harus ditempuh agar dapat tiba di tempat tujuan yang dikehendaki ketika berada dalam suatu gedung bertingkat.

1.5 Metodologi

Ada beberapa tahap dalam proses pengerjaan tugas akhir ini, yakni sebagai berikut:

a. Studi Literatur

Merupakan tahap pengumpulan informasi yang dibutuhkan dalam perancangan dan implementasi perangkat lunak yang akan dibangun. Literatur yang dibutuhkan antara lain sebagai berikut:

- a. Pembuatan peta menggunakan SketchUp Pro.
- b. Pembuatan perangkat lunak untuk Android menggunakan Unity.
- c. *Export* peta ke Unity.
- d. Pembuatan pemindai QR menggunakan Unity.

b. Analisis dan Perancangan Perangkat Lunak

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

1. Rancangan sistem basis data.
2. Rancangan perangkat lunak pemindai QR Code.
3. Rancangan perangkat lunak **QR MAP** sebagai aplikasi yang digunakan untuk pencarian rute terpendek pada gedung bertingkat.

c. **Implementasi**

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

1. Implementasi rancangan basis data.
2. Implementasi rancangan aplikasi perangkat bergerak berbasis Android untuk menentukan lokasi pengguna.
3. Implementasi rancangan aplikasi pemindai QR Code.
4. Implementasi penggunaan metode Dijkstra pada konsep *3D Indoor Localization* untuk menentukan rute yang harus dilalui.

d. **Pengujian dan Evaluasi**

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak yang telah dibuat berdasarkan skenario yang telah ditentukan, yaitu:

1. Melakukan pengujian dan evaluasi terhadap pemindai QR Code yang telah dibuat. Pada pengujian ini, perangkat lunak dapat memindai QR Code secara tepat dan dapat menampilkan posisi pengguna berdasarkan hasil pemindaian.
2. Perangkat lunak harus dapat menampilkan daftar tujuan yang ada pada suatu gedung.
3. Perangkat lunak dapat menampilkan rute terpendek yang harus dilalui untuk dapat mencapai tujuan yang telah dipilih.

e. **Penyusunan Buku Tugas Akhir**

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan tentang dasar teori dan metode yang digunakan dalam pembuatan tugas akhir, rancangan, proses yang dilakukan dan hasil yang diperoleh selama pengerjaan perangkat lunak.

1.6 **Sistematika Penyusunan Laporan**

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak

yang telah dibuat. Sistematika penulisan buku tugas akhir ini secara garis besar adalah sebagai berikut:

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

BAB II

TINJAUAN PUSTAKA

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

2.1 *Indoor Positioning System*

Indoor Positioning System (IPS) merupakan sistem navigasi yang digunakan untuk mengetahui lokasi dari suatu objek dalam sebuah bangunan [1] [2] [3]. IPS memiliki keterkaitan yang erat dengan *Indoor Localization*, dimana keduanya sama-sama digunakan untuk menentukan posisi suatu objek dalam sebuah gedung. Yang membedakan keduanya adalah *Indoor Positioning System* menentukan koordinat global dari sebuah lokasi (contoh: garis bujur dan garis lintang), sedangkan *Indoor Localization* menentukan koordinat relatif (contoh: Laboratorium Alpro, Ruang IF-102) [4]. IPS pada dasarnya sama dengan GPS, hanya saja tujuan dan media yang digunakan untuk menentukan posisi berbeda. GPS menggunakan bantuan satelit dengan gelombang *microwave*, gelombang ini akan dipantulkan dan diserap oleh atap, tembok, maupun objek lainnya. Sehingga GPS tidak dapat digunakan dalam ruangan. Untuk menangani permasalahan ini, maka dibuatlah IPS. IPS sendiri dapat menggunakan beberapa media untuk mengetahui posisi seseorang, seperti gelombang radio, medan magnet, sinyal akustik, atau informasi yang diperoleh dari sensor yang ada pada perangkat *mobile*. Sampai saat ini, IPS dapat digunakan untuk beberapa hal, yaitu:

1. Navigasi dalam sebuah gedung, kantor, dan fasilitas industri.
2. *Location Based Services*.

3. Kepentingan evakuasi.
4. Penyimpanan barang.
5. Informasi audio visual.

Teknologi yang paling umum digunakan dalam menentukan posisi pada IPS adalah teknologi berbasis nirkabel. Teknologi-teknologi ini memanfaatkan gelombang yang dipancarkan dalam menentukan posisi. Teknologi-teknologi nirkabel yang pernah diterapkan adalah:

1. Wi-Fi, merupakan teknologi terbaik dalam menentukan posisi. Hal ini dikarenakan pertukaran data yang stabil dan daerah jangkauan sinyal yang luas (sekitar 30-40 meter) dan stabil dibandingkan dengan teknologi yang lainnya.
2. *Radio Frequency Identification* (RFID), menggunakan *tag* dan *reader* yang memungkinkan adanya pertukaran data yang cepat, namun tidak dapat digunakan untuk menentukan posisi.
3. *Ultra Wide Band* (UWB), menawarkan pertukaran data yang cepat, hemat energi, dan dalam jumlah yang besar. Dapat menentukan posisi dengan tingkat presisi yang tinggi, sekitar 15 cm, namun teknologi ini terlampau mahal.

2.2 QR Code

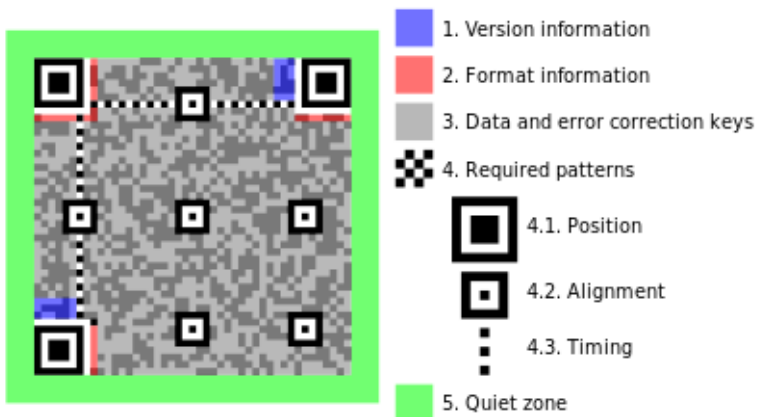
QR Code merupakan suatu jenis kode matriks dua dimensi yang dikembangkan oleh salah satu perusahaan Jepang yaitu, Denso Wave (divisi dari DENSO CORP.) yang merilisnya pada tahun 1994¹. QR dari kata QR Code merupakan singkatan dari *Quick Response*, hal ini dikarenakan QR Code diharapkan dapat dibaca secara cepat oleh pemindai QR. Pada awalnya QR Code digunakan pada industri mobil untuk mengefisienkan pekerjaan

¹ <http://www.qrcode.com/en/history/>

mulai dari produksi sampai kepada pembelian dan penerbitan nota transaksi. Pada tahun 2002, penggunaan QR Code meluas di seluruh wilayah Jepang. Penggunaan QR Code yang terus meningkat ini, mengakibatkan mulai bermunculan telepon genggam yang memiliki fitur pembacaan QR Code. Selain dapat menyimpan informasi, QR Code juga dapat digunakan untuk menyimpan logo perusahaan, klip video, ataupun foto. QR Code memiliki beberapa standar yang digunakan dalam menyembunyikan data yang ada di dalamnya, yaitu:

1. AIM (*Association for Automatic Identification and Mobility*) International, mulai berlaku pada Oktober 1997.
2. JIS (*Japanese Industrial Standards*) X 0510, muncul pada January 1999.
3. ISO/IEC 18004:2000, pada Juni 2000.
4. ISO/IEC 18004:2006, pada 1 September 2006.

Struktur dari QR Code dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur QR Code

Perkembangan QR Code yang pesat menyebabkan banyaknya sistem operasi untuk perangkat mobile untuk menanamkan kemampuan membaca QR Code. QR Code pada saat ini digunakan untuk menampung data berupa URL sebuah *website*, toko virtual, kode pembayaran suatu tagihan, *login* ke sebuah *website*, digunakan untuk kepentingan pemakaman, dan untuk proses enkripsi dengan menggunakan algoritma DES. QR Code menggunakan kata sandi dengan panjang 8 bit, dan menggunakan algoritma Reed-Solomon untuk memperbaiki kesalahan yang ada. QR Code dapat digolongkan ke dalam 40 versi sesuai dengan dimensi matriks yang digunakan, beberapa contoh dari versi QR Code dapat dilihat pada Gambar 2.2.



Gambar 2.2 Versi QR Code

Penjelasan dari Gambar 2.2 adalah sebagai berikut:

1. QR Code versi 1, tersusun atas matriks 21×21 .
2. QR Code versi 2, tersusun atas matriks 25×25 .
3. QR Code versi 3, tersusun atas matriks 29×29 .
4. QR Code versi 4, tersusun atas matriks 33×33 .
5. QR Code versi 10, tersusun atas matriks 57×57 .

2.3 Android

Android merupakan sebuah sistem operasi berbasis Linux, yang dirancang untuk perangkat seluler layar sentuh seperti *smartphone* dan komputer *tablet* dengan sifat *open source*. Oleh

Google, kode Android dirilis di bawah Lisensi Apache. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi yang berbiaya rendah, dapat di-*custom*, dan ringan untuk perangkat berteknologi tinggi. Akibatnya, Android juga dikembangkan untuk televisi, konsol permainan, kamera digital, dan perangkat elektronik lainnya. Android memiliki beberapa fitur², antara lain:

1. Antarmuka. Pengguna dapat melakukan manipulasi secara langsung pada antarmuka Android, seperti menggesek, mengetuk, dan mencubit. Beberapa perangkat lunak menggunakan bantuan perangkat keras internal seperti akselerometer, giroskop, dan sensor proksimitas untuk merespon tindakan pengguna.
2. Perangkat lunak. Melalui Google Play, Android memberikan kemudahan bagi penggunanya untuk memilih, mengunduh, dan memperbarui perangkat lunak dari pihak ketiga. Perangkat lunak Android dikembangkan dengan bahasa pemrograman Java dan menggunakan kit SDK, dimana SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk *debugger*, perpustakaan perangkat lunak emulator *handset* berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh IDE Android Studio.
3. Pengelolaan memori. Android dirancang untuk mengelola RAM secara efisien untuk menjaga konsumsi daya minimal. Ketika sebuah perangkat lunak Android tidak lagi digunakan, sistem secara otomatis akan menangguhkannya ke dalam memori, secara teknis perangkat lunak tersebut masih terbuka, namun tidak akan mengkonsumsi sumber daya, dan akan diam

² [https://id.wikipedia.org/wiki/Android_\(sistem_operasi\)](https://id.wikipedia.org/wiki/Android_(sistem_operasi))

sampai perangkat lunak yang dimaksud digunakan kembali.

Mulai dari awalnya pembuatan Android sampai sekarang ini, terdapat beberapa versi Android yang pernah dirilis. Berikut merupakan versi Android yang masih tergolong baru dan banyak digunakan di seluruh dunia:

1. **Android 5.+ / Lollipop**

Lollipop pertama kali dirilis pada 25 Juni 2014. Lollipop mengalami perubahan tak hanya pada segi tampilan, performa juga menjadi fokus dan perhatian utama dari Google. Google mengklaim bahwa Android versi ini mampu memberikan kinerja baterai dan manajemen *multitasking* yang lebih bagus.

2. **Android 6.0 / Marshmallow**

Merupakan Android versi termutakhir yang dirilis oleh Google. Android versi ini pertama kali digunakan pada telepon genggam Nexus besutan Google. Marshmallow pertama kali diperkenalkan pada tanggal 28 Mei 2015, namun baru dimunculkan ke publik pada tanggal 17 Agustus 2015. Android ini difokuskan pada perbaikan inkremental dan penambahan fitur lainnya seperti dukungan asli untuk pengenalan sidik jari, dukungan untuk USB Type-C.

2.4 **Algoritma Dijkstra**

Algoritma Dijkstra merupakan algoritma untuk menemukan jalur terpendek antar *node* dalam sebuah *graph* yang tidak mempunyai arah [5]. Algoritma ini dinamakan sesuai dengan penemunya Edsger W. Dijkstra. Edsger Wybe Dijkstra merupakan seorang ilmuwan komputer berkebangsaan Belanda. Algoritma ini

dipublikasikan pada tahun 1959³. Dalam pengaplikasian algoritma Dijkstra dalam suatu permasalahan diperlukan beberapa syarat berikut:

1. *Graph* yang digunakan dapat berupa *graph* yang memiliki arah maupun yang tidak memiliki arah.
2. Semua *edge* pada *graph* harus memiliki bobot *non-negative*.
3. *Graph* harus terhubung.

Cara kerja algoritma ini adalah sebagai berikut:

1. Beri bobot pada setiap *edge*. Lalu pilih *node* awal dan *set* nilai 0 pada *node* awal dan nilai tak terhingga pada *node* lainnya.
2. *Set* semua *node* belum dikunjungi dan *set node* sekarang sebagai *node* keberangkatan.
3. Dari *node* sekarang, hitung semua jarak dari *node* sekarang ke tetangganya yang belum dikunjungi. Jika jaraknya lebih kecil, hapus data lama dan simpan data baru dengan jarak yang baru.
4. Setelah selesai disimpan, tandai *node* tersebut sebagai *node* yang sudah dikunjungi. *Node* yang sudah dikunjungi tidak akan dicek kembali.
5. *Set node* belum dikunjungi dengan jarak terkecil sebagai *node* sekarang, dan ulangi langkah nomor 3.

Kode Sumber 2.1 merupakan *pseudocode* dari algoritma Dijkstra. Untuk mendapatkan jalur terpendek antara titik asal ke tujuan dapat dilihat pada Kode Sumber 2.2.

³ https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

```

1 function Dijkstra(Graph, source):
2
3   create vertex set Q
4
5   for each vertex v in Graph:           // Inisialisasi
6     dist[v] ← INFINITY                 // Jarak dari titik awal ke
tujuan v belum diketahui
7     prev[v] ← UNDEFINED // Node asal dari titik awal
8     add v to Q                         // Semua node belum
dikunjungi
9
10    dist[source] ← 0                    // Jarak dari node ke dirinya
sendiri
11
12    while Q is not empty:
13      u ← vertex in Q with min dist[u] // Titik awal akan
dipilih lebih dulu
14      remove u from Q
15
16      for each neighbor v of u:       // Dimana node v
belum dikunjungi
17        alt ← dist[u] + length(u, v)
18        if alt < dist[v]:           // Jarak paling pendek
ke v telah diketahui
19          dist[v] ← alt
20          prev[v] ← u
21
22    return dist[], prev[]

```

Kode Sumber 2.1 Pseudocode Algoritma Dijkstra

```
1 S ← empty sequence
2 u ← target
3 while prev[u] is defined:           // Buat sebuah stack untuk
menampung jalur terpendek
4   insert u at the beginning of S     // Masukkan vertex ke
dalam stack
5   u ← prev[u]                       // Titik awal dijadikan tujuan
6   insert u at the beginning of S     // Masukkan titik awal ke
stack
```

Kode Sumber 2.2 Pseudocode untuk Mendapatkan Jalur Terpendek dari Algoritma Dijkstra

[Halaman ini sengaja dikosongkan]

BAB III

PERANCANGAN PERANGKAT LUNAK

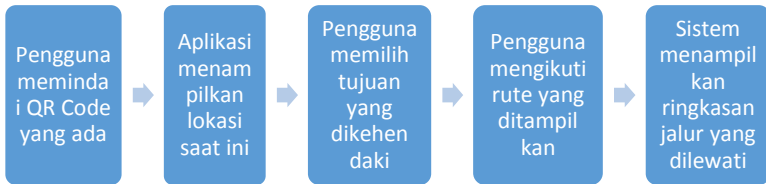
Perancangan merupakan bagian penting dari pembuatan suatu perangkat lunak yang berupa perencanaan-perencanaan secara teknis aplikasi yang dibuat. Sehingga bab ini secara khusus akan menjelaskan perancangan sistem yang dibuat dalam tugas akhir ini. Berawal dari deskripsi umum aplikasi hingga perancangan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

3.1 Deskripsi Umum Perangkat Lunak

Perangkat lunak yang akan dibuat pada tugas akhir ini adalah suatu perangkat lunak berbasis Android yang menerapkan konsep *Indoor Positioning System*. Perangkat lunak memiliki kemampuan untuk melakukan pemindaian QR Code. QR Code yang digunakan akan dibuat menggunakan *QR Code Generator*. QR Code nantinya akan menyimpan data berupa:

1. ID Map: merupakan penomoran dari peta gedung yang akan digunakan.
2. ID Ruang: merupakan penomoran dari ruangan-ruangan yang terdapat pada suatu gedung.

Hasil pemindaian QR Code akan digunakan untuk menentukan peta mana yang akan digunakan dan dimana posisi pengguna sekarang. Setelah ditentukan peta yang akan digunakan, perangkat lunak akan menampilkan daftar ruangan yang terdapat pada peta yang dipilih. Setelah pengguna memilih tujuan yang dikehendaki, maka perangkat lunak akan mencari rute terpendek dari posisi pengguna sekarang ke tujuan yang dipilih dengan menggunakan algoritma Dijkstra. Rute yang didapat kemudian akan digambar pada peta. Gambar 3.1 merupakan alur jalannya perangkat lunak yang dibuat.



Gambar 3.1 Alur Jalannya Perangkat Lunak QR MAP

Kelemahan dari perangkat lunak ini adalah perangkat lunak tidak dapat meng-*update* posisi pengguna secara *real time*. Dengan kata lain, pengguna hanya mengetahui posisinya ketika melakukan pemindaian QR Code. Untuk sampai pada tujuan, pengguna dapat mengikuti rute yang ditampilkan oleh perangkat lunak. Untuk mengatasi kelemahan dari perangkat lunak ini, disediakan sebuah panel navigasi yang terdiri atas maju, mundur, putar kanan, dan putar kiri, dimana panel navigasi ini mempermudah pengguna untuk mengetahui posisinya ketika mengikuti petunjuk rute ke tempat tujuan. Perangkat lunak ini menggunakan SQLite sebagai lokasi penyimpanan data pada Android.

3.1.1 Arsitektur Sistem

Gambar 3.2 adalah arsitektur dari perangkat lunak yang dibuat:



Gambar 3.2 Arsitektur Sistem

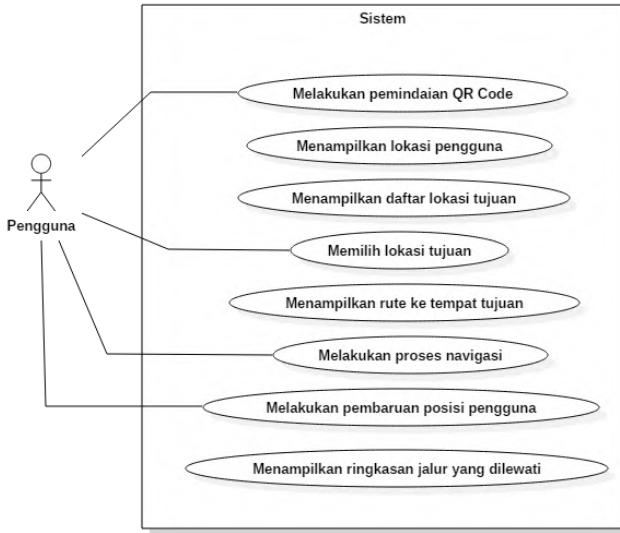
Berdasarkan perancangan arsitektur sistem pada Gambar 3.2, perangkat lunak mendapatkan informasi berupa peta yang digunakan dan posisi pengguna dari QR Code. Data yang digunakan untuk mendapatkan data-data yang diperlukan pada basis data lokal. Perangkat lunak kemudian mengolah data hasil pemindaian dan data tujuan pilihan pengguna menggunakan algoritma Dijkstra untuk mendapatkan rute terpendek, kemudian menampilkannya untuk pengguna.

3.1.2 Kebutuhan Fungsional Perangkat Lunak

Aplikasi berbasis perangkat bergerak Android digunakan untuk menemukan rute terpendek dari lokasi pengguna sekarang ke tempat tujuan. Berikut ini adalah daftar kebutuhan fungsional dari perangkat lunak berbasis Android:

1. Melakukan pemindaian QR Code.
2. Menampilkan lokasi pengguna.
3. Menampilkan daftar lokasi tujuan.
4. Memilih lokasi tujuan.
5. Menampilkan rute ke tempat tujuan.
6. Melakukan proses navigasi.
7. Melakukan pembaruan posisi pengguna.
8. Menampilkan ringkasan jalur yang dilewati.

Use Case Diagram dari aplikasi ini dapat dilihat pada Gambar 3.3. Penjelasan lengkap mengenai kasus penggunaan pada perangkat lunak ini berada pada Tabel 3.1.



Gambar 3.3 Use Case Diagram

Tabel 3.1 Deskripsi Tabel Penggunaan

No	Kode	Nama	Keterangan
1	UC-01	Melakukan pemindaian QR Code	Pengguna dapat melakukan pemindaian QR Code
2	UC-02	Menampilkan lokasi pengguna	Sistem dapat menampilkan lokasi pengguna berdasarkan pemindaian QR Code
3	UC-03	Menampilkan daftar lokasi tujuan	Sistem dapat menampilkan daftar lokasi tujuan berdasarkan pemindaian QR Code

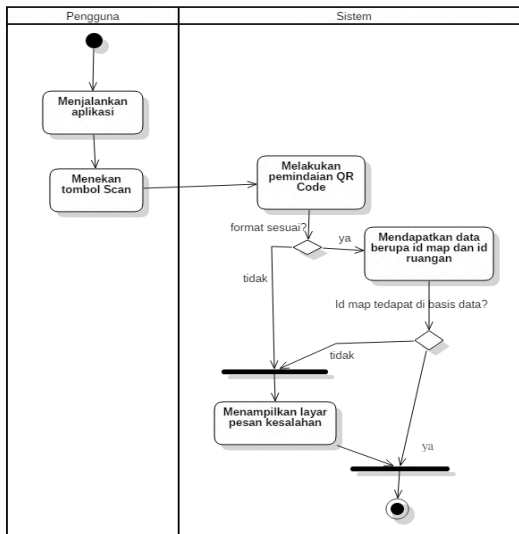
4	UC-04	Memilih lokasi tujuan	Pengguna dapat memilih lokasi tujuan yang ditampilkan
5	UC-05	Menampilkan rute ke tempat tujuan	Sistem dapat menampilkan rute yang harus ditempuh dari asal ke tempat tujuan
6	UC-06	Melakukan proses navigasi	Pengguna dapat melakukan proses navigasi dari panel yang disediakan
7	UC-07	Melakukan pembaruan posisi pengguna	Pengguna dapat melakukan pembaruan posisi dengan melakukan pemindaian ulang
8	UC-08	Menampilkan ringkasan jalur yang dilewati	Sistem dapat menampilkan nama jalur yang dilalui dan total jarak yang ditempuh

3.1.2.1 Deskripsi Kasus Penggunaan UC-01

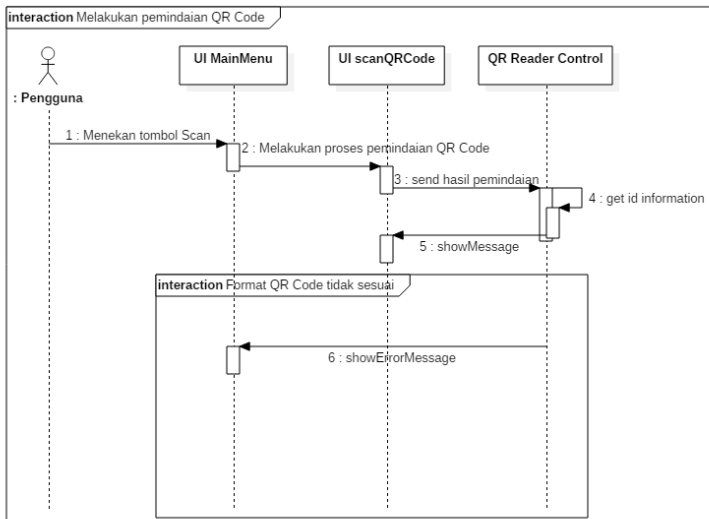
Kasus penggunaan UC-01 merupakan kasus dimana pengguna dapat melakukan pemindaian QR Code untuk mengetahui lokasi pengguna. Rincian alur kasus UC-01 dijelaskan pada Tabel 3.2 dan diagram aktivitas ditunjukkan pada Gambar 3.4. Sedangkan diagram alir kasus penggunaan UC-01 ditunjukkan pada Gambar 3.5.

Tabel 3.2 Rincian Alur Kasus Penggunaan UC-01

Melakukan Pemindaian Qr Code	
Nama Use Case	Melakukan Pemindaian Qr Code
Nomor	UC-01
Aktor	Pengguna
Kondisi Awal	QR Code belum dipindai
Kondisi Akhir	Mendapatkan id map dan id ruangan
Alur Normal	1. Pengguna memindai QR Code 2. Sistem mendapatkan data berupa id map dan id ruangan
Alur Ketika Format QR Code Tidak Sesuai	2. Sistem menampilkan pesan error kalau format QR Code tidak sesuai
Alur Ketika Peta Tidak Ditemukan di Basis Data	2. Sistem menampilkan pesan <i>error</i> kalau peta tidak ditemukan pada basis data



Gambar 3.4 Diagram Aktivitas UC-01



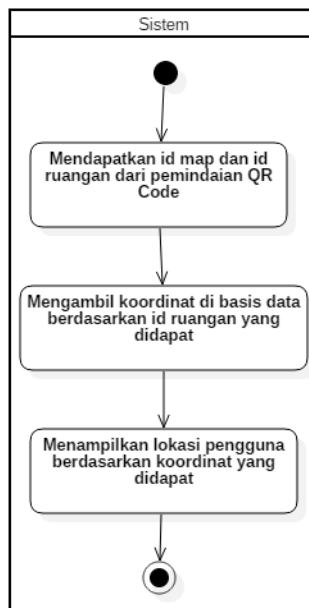
Gambar 3.5 Diagram Alir UC-01

3.1.2.2 Deskripsi Kasus Penggunaan UC-02

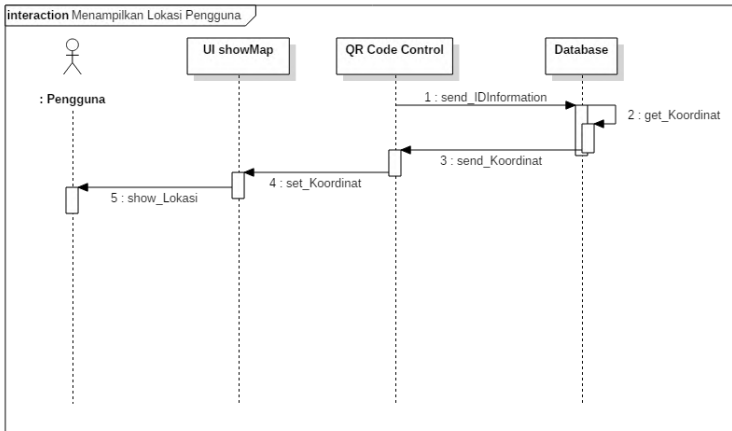
Kasus penggunaan UC-02 merupakan kasus dimana sistem menampilkan lokasi pengguna berdasarkan hasil pemindaian QR Code. Rincian alur kasus UC-02 dijelaskan pada Tabel 3.3 dan diagram aktivitas ditunjukkan pada Gambar 3.6. Sedangkan diagram alir kasus penggunaan UC-02 ditunjukkan pada Gambar 3.7.

Tabel 3.3 Rincian Alur Kasus Penggunaan UC-02

Nama Use Case		Menampilkan Lokasi Pengguna
Nomor	UC-02	
Aktor	Sistem	
Kondisi Awal	Lokasi pengguna belum diketahui	
Kondisi Akhir	Lokasi pengguna ditampilkan	
Alur Normal	<ol style="list-style-type: none"> 1. Sistem mendapatkan id map dan id ruangan dari pemindaian QR Code 2. Sistem mengambil koordinat berdasarkan id ruangan yang didapat 3. Sistem menampilkan lokasi pengguna berdasarkan koordinat dari basis data 	



Gambar 3.6 Diagram Aktivitas UC-02



Gambar 3.7 Diagram Alir UC-02

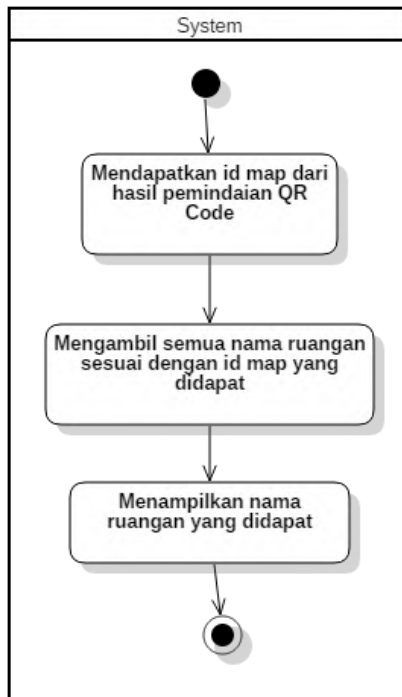
3.1.2.3 Deskripsi Kasus Penggunaan UC-03

Kasus penggunaan UC-03 merupakan kasus dimana sistem menampilkan daftar lokasi tujuan sesuai dengan id map yang didapat. Rincian alur kasus UC-03 dijelaskan pada Tabel 3.4 dan diagram aktivitas ditunjukkan pada Gambar 3.8. Sedangkan diagram alir kasus penggunaan UC-03 ditunjukkan pada Gambar 3.9.

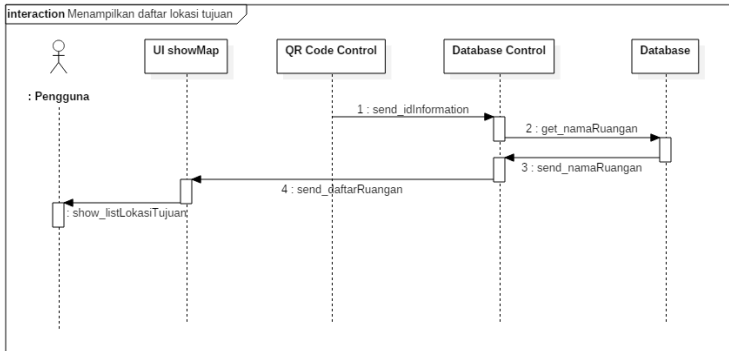
Tabel 3.4 Rincian Alur Kasus Penggunaan UC-03

Nama Use Case	Menampilkan Daftar Lokasi Tujuan
Nomor	UC-03
Aktor	Sistem
Kondisi Awal	Daftar lokasi belum ditampilkan
Kondisi Akhir	Daftar lokasi tujuan ditampilkan
Alur Normal	1. Sistem mendapatkan id map dari hasil pemindaian QR Code.

	2. Sistem mengambil semua data ruangan sesuai dengan id map yang didapat dari basis data.
	3. Sistem menampilkan data ruangan yang didapat.



Gambar 3.8 Diagram Aktivitas UC-03



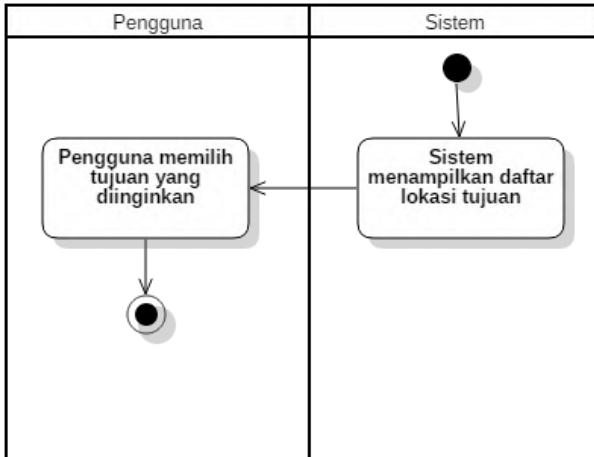
Gambar 3.9 Diagram Alir UC-03

3.1.2.4 Deskripsi Kasus Penggunaan UC-04

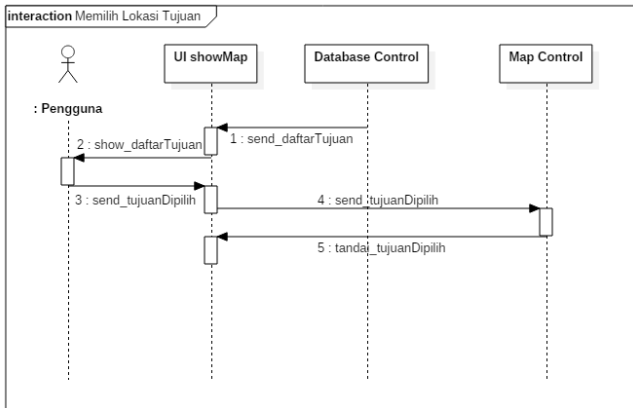
Kasus penggunaan UC-04 merupakan kasus dimana pengguna memilih lokasi tujuan yang diinginkan. Rincian alur kasus UC-04 dijelaskan pada Tabel 3.5 dan diagram aktivitas ditunjukkan pada Gambar 3.10. Sedangkan diagram alir kasus penggunaan UC-04 ditunjukkan pada Gambar 3.11.

Tabel 3.5 Rincian Alur Kasus Penggunaan UC-04

Nama Use Case	Memilih Lokasi Tujuan
Nomor	UC-04
Aktor	Pengguna
Kondisi Awal	Tujuan belum dipilih
Kondisi Akhir	Tujuan dipilih
Alur Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan daftar lokasi tujuan 2. Pengguna memilih lokasi tujuan yang diinginkan 3. Sistem menandai lokasi tujuan yang dipilih



Gambar 3.10 Diagram Aktivitas UC-04



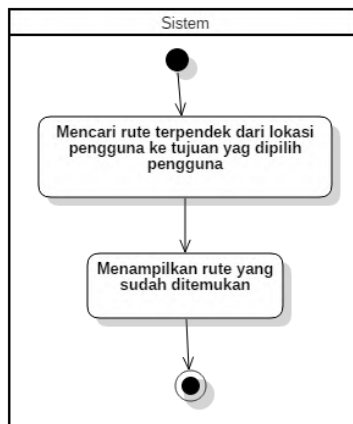
Gambar 3.11 Diagram Alir UC-04

3.1.2.5 Deskripsi Kasus Penggunaan UC-05

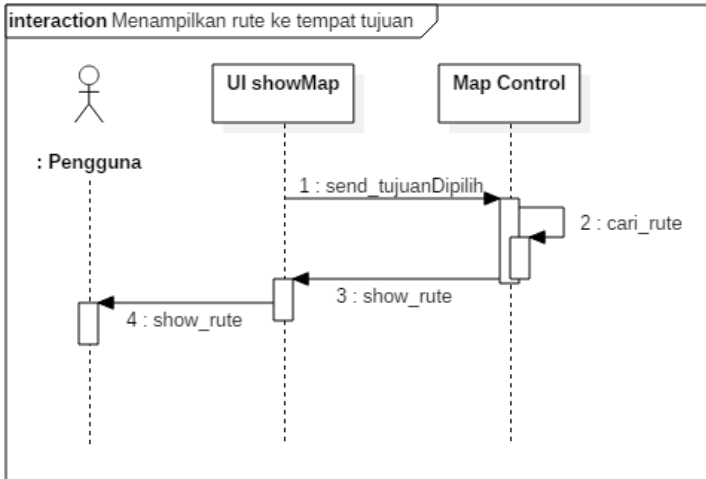
Kasus penggunaan UC-05 merupakan kasus dimana sistem menampilkan rute dari tempat asal ke tempat tujuan. Rincian alur kasus UC-05 dijelaskan pada Tabel 3.6 dan diagram aktivitas ditunjukkan pada Gambar 3.12. Sedangkan diagram alir kasus penggunaan UC-05 ditunjukkan pada Gambar 3.13.

Tabel 3.6 Rincian Alur Kasus Penggunaan UC-05

Nama Use Case	Menampilkan Rute ke Tempat Tujuan
Nomor	UC-05
Aktor	Sistem
Kondisi Awal	Rute belum dibuat
Kondisi Akhir	Rute sudah dibuat dan ditampilkan
Alur Normal	<ol style="list-style-type: none"> 1. Sistem mencari rute terpendek berdasarkan lokasi pengguna dan tujuan yang dipilih oleh pengguna 2. Sistem menampilkan rute yang sudah didapat



Gambar 3.12 Diagram Aktivitas UC-05



Gambar 3.13 Diagram Alir UC-05

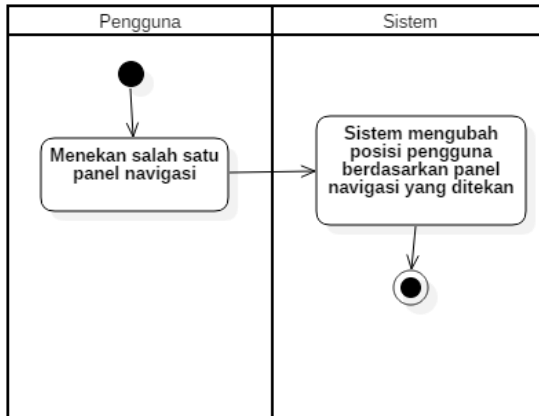
3.1.2.6 Deskripsi Kasus Penggunaan UC-06

Kasus penggunaan UC-06 merupakan kasus dimana pengguna dapat melakukan proses navigasi. Rincian alur kasus UC-06 dijelaskan pada Tabel 3.7 dan diagram aktivitas ditunjukkan pada Gambar 3.14. Sedangkan diagram alir kasus penggunaan UC-06 ditunjukkan pada Gambar 3.15.

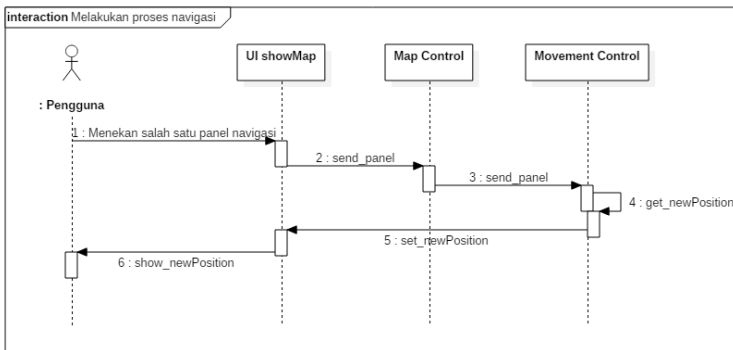
Tabel 3.7 Rincian Alur Kasus Penggunaan UC-06

Nama Use Case	Melakukan Proses Navigasi
Nomor	UC-06
Aktor	Pengguna
Kondisi Awal	Posisi pengguna tidak berubah
Kondisi Akhir	Posisi pengguna berubah
Alur Normal	1. Pengguna menekan salah satu panel navigasi

2. Sistem mengubah posisi pengguna sesuai dengan panel navigasi yang ditekan pengguna



Gambar 3.14 Diagram Aktivitas UC-06



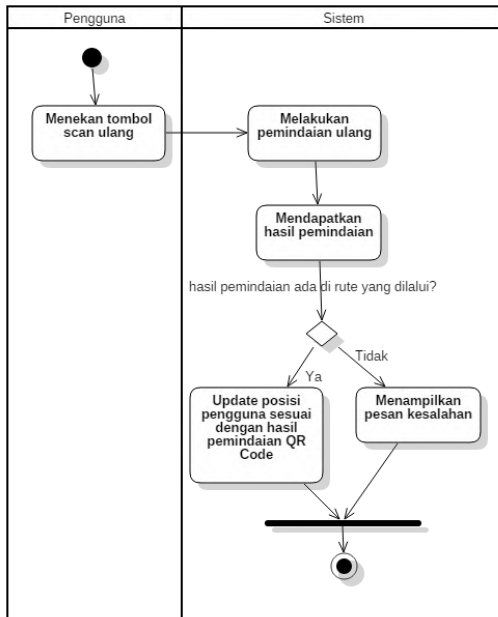
Gambar 3.15 Diagram Alir UC-06

3.1.2.7 Deskripsi Kasus Penggunaan UC-07

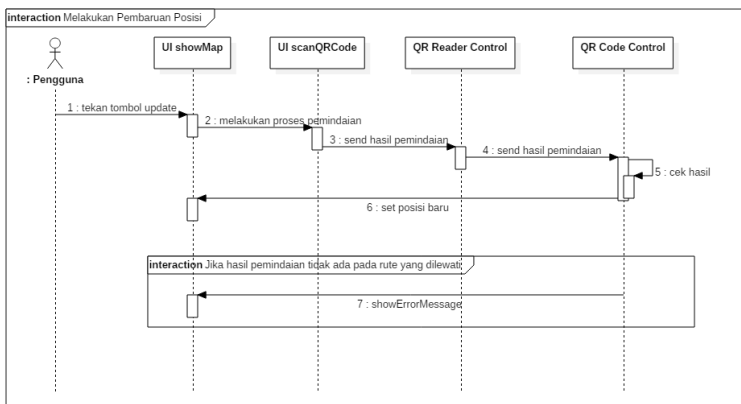
Kasus penggunaan UC-07 merupakan kasus dimana pengguna dapat melakukan pembaruan posisinya dengan cara melakukan pemindaian QR Code. Rincian alur kasus UC-07 dijelaskan pada Tabel 3.8 dan diagram aktivitas ditunjukkan pada Gambar 3.16. Sedangkan diagram alir kasus penggunaan UC-07 ditunjukkan pada Gambar 3.17.

Tabel 3.8 Rincian Alur Kasus Penggunaan UC-07

Nama Use Case		Melakukan Pembaruan Posisi Pengguna
Nomor	UC-07	
Aktor	Pengguna	
Kondisi Awal	Posisi pengguna belum berubah	
Kondisi Akhir	Posisi pengguna berubah	
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna menekan tombol bergambar QR Code. 2. Sistem melakukan pemindaian ulang QR Code. 3. Sistem melakukan pembaruan posisi pengguna berdasarkan hasil pemindaian. 	
Alur Ketika Hasil Pemindaian QR Code tidak ada di rute yang dilewati	<ol style="list-style-type: none"> 3. Sistem menampilkan pesan kalau hasil pemindaian tidak ditemukan pada rute yang dilewati. 	



Gambar 3.16 Diagram Aktivitas UC-07



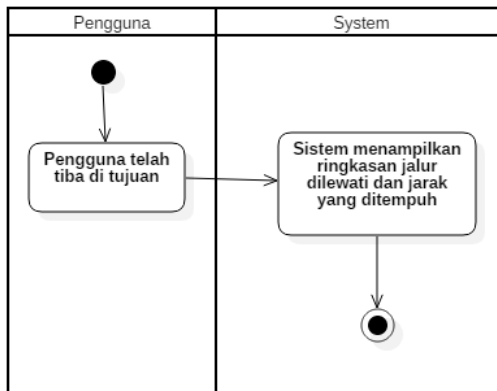
Gambar 3.17 Diagram Alir UC-07

3.1.2.8 Deskripsi Kasus Penggunaan UC-08

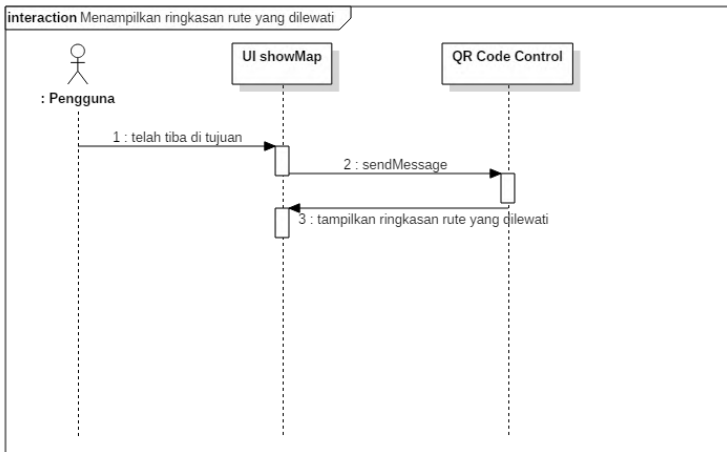
Kasus penggunaan UC-08 merupakan kasus dimana sistem dapat menampilkan daftar rute yang dilalui oleh pengguna untuk dapat tiba di tempat tujuan beserta total jarak yang ditempuh oleh pengguna. Rincian alur kasus UC-08 dijelaskan pada Tabel 3.9 dan diagram aktivitas ditunjukkan pada Gambar 3.18. Sedangkan diagram alir kasus penggunaan UC-08 ditunjukkan pada Gambar 3.19.

Tabel 3.9 Rincian Alur Kasus Penggunaan UC-08

Nama Use Case		Menampilkan Ringkasan Rute yang Dilewati
Nomor	UC-08	
Aktor	Sistem	
Kondisi Awal	Ringkasan rute belum ditampilkan	
Kondisi Akhir	Ringkasan rute ditampilkan	
Alur Normal	<ol style="list-style-type: none"> 1. Pengguna telah tiba di tujuan. 2. Sistem menampilkan ringkasan rute yang dilewati dan jarak yang ditempuh. 	



Gambar 3.18 Diagram Aktivitas UC-08



Gambar 3.19 Diagram Alir UC-08

3.2 Perancangan

Subbab berikut membahas tentang perancangan dari perangkat lunak **QR MAP**. Subbab berikut membahas rancangan konsep peta 3D, rancangan antarmuka, rancangan proses dan rancangan basis data. Pembahasan lebih detail akan dibahas berikut ini.

3.2.1 Perancangan Peta 3D




Pada subbab perancangan konsep peta 3D akan dipaparkan mengenai bagaimana pembuatan peta 3D yang akan diimplementasikan. Peta 3D yang dibuat merupakan peta dari gedung Teknik Informatika ITS, dimana pembuatannya menggunakan kaskas bantu Google SketchUp Pro 2016 dengan skala 1:2 dari ukuran yang sebenarnya. Hal yang akan dibahas pada rancangan konsep peta 3D adalah perancangan *vertex* yang

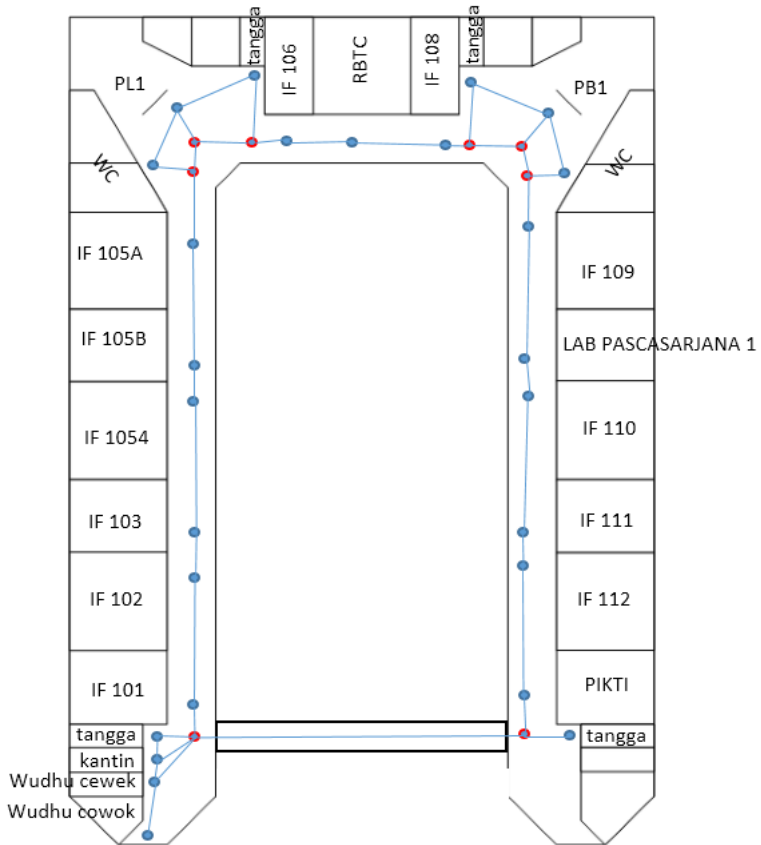
digunakan. Berikut penjelasan detail mengenai perancangan di atas.

3.2.1.1 Perancangan *Vertex*

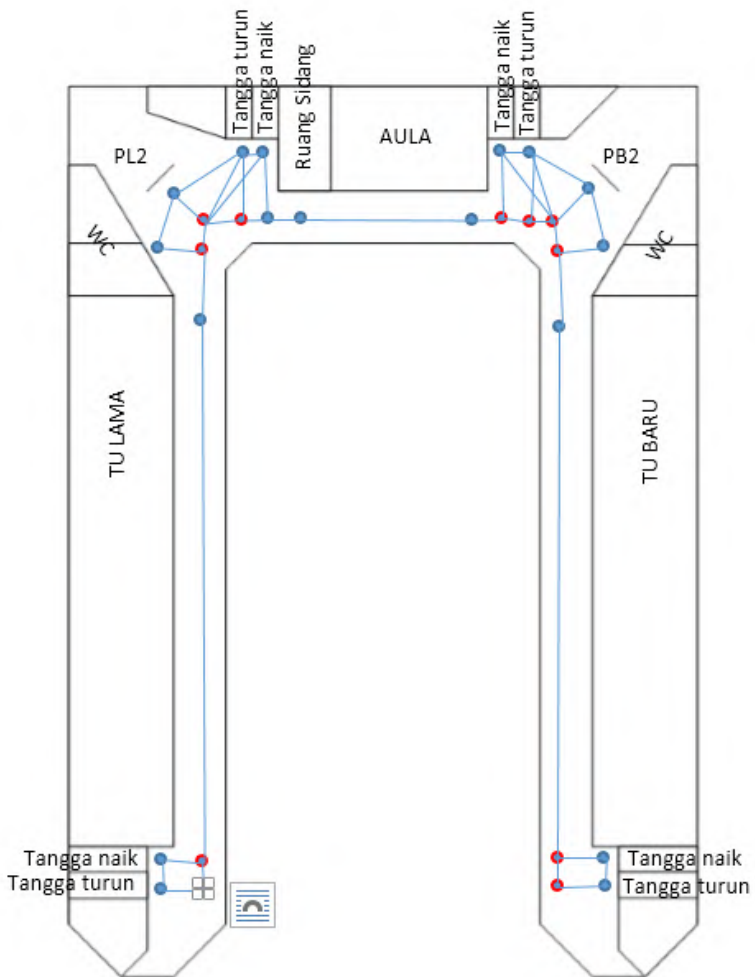
Perancangan *vertex* adalah hal yang penting dalam pembuatan perangkat lunak ini. Perancangan *vertex* ini dibuat berdasarkan algoritma Dijkstra, dimana algoritma Dijkstra membutuhkan *vertex*, *edge*, dan *cost*. Tiap ruangan yang ada pada suatu gedung akan diwakilkan dengan sebuah *vertex* pada peta 3D ini. Jalan yang menghubungkan antara ruangan yang satu dengan ruangan yang lain akan diwakilkan oleh sebuah *edge*. Sementara *cost* didapat dengan cara menghitung jarak antara *vertex* yang satu dengan *vertex* yang lainnya menggunakan *Euclidean Distance*. Melalui perancangan *vertex* inilah sebuah rute dapat ditentukan. Gambar 3.20 merupakan perancangan *vertex* untuk lantai 1, Gambar 3.21 merupakan perancangan *vertex* untuk lantai 2, dan Gambar 3.22 merupakan perancangan *vertex* untuk lantai 3. Keterangan dari tiap simbol yang digunakan pada perancangan *vertex* dijelaskan pada Tabel 3.10.

Tabel 3.10 Keterangan Simbol pada Perancangan *Vertex*

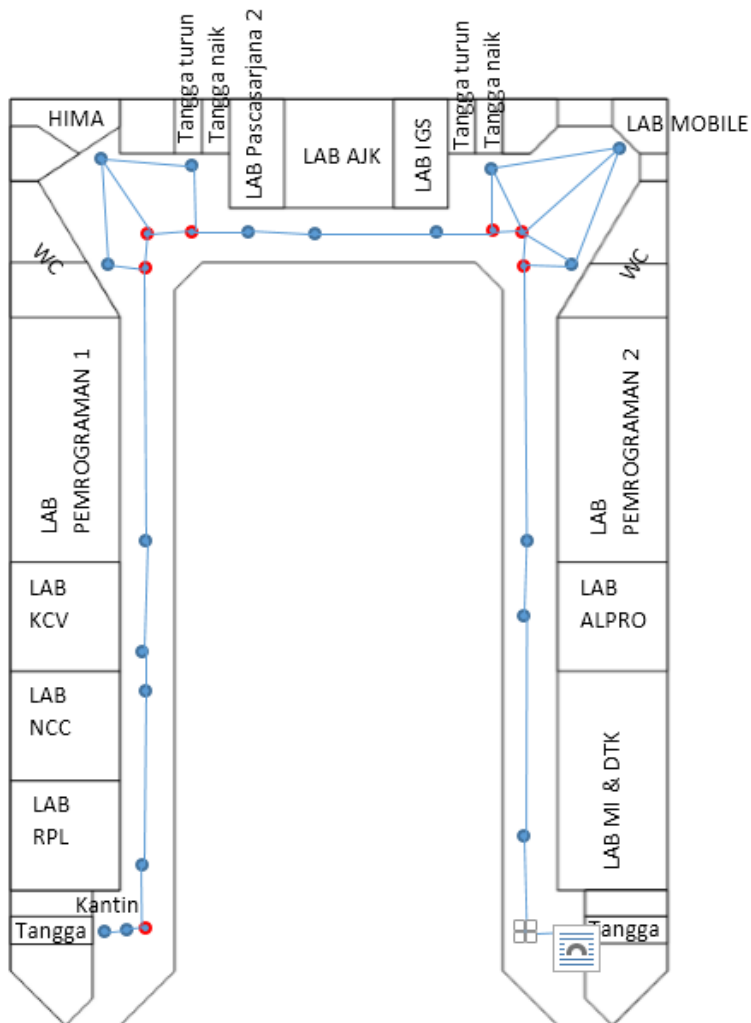
	<i>Vertex</i> yang mewakili ruangan
	<i>Vertex</i> yang digunakan sebagai bantuan penghubung
	<i>Edge</i>



Gambar 3.20 Perancangan vertex lantai 1



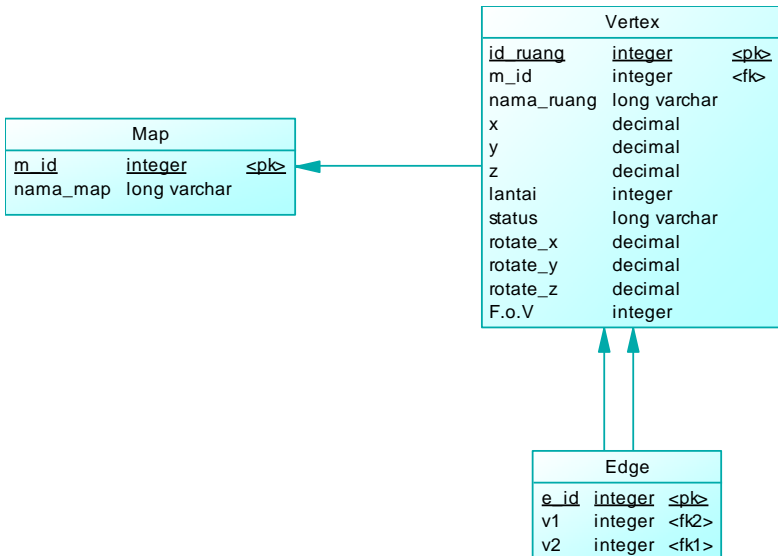
Gambar 3.21 Perancangan *vertex* lantai 2



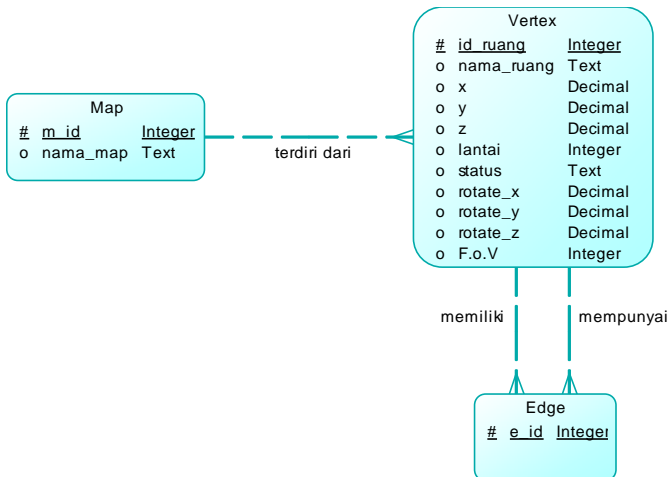
Gambar 3.22 Perancangan *vertex* lantai 3

3.2.2 Perancangan Basis Data

Pada subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada perangkat lunak **QR MAP**. Basis data pada sistem yang ini menggunakan sistem manajemen basis data relasional SQLite. SQLite digunakan untuk menyimpan data *peta*, *edge*, dan *vertex*. *Physical Data Model* (PDM) dan *Conceptual Data Model* (CDM) dari basis data sistem ini dapat dilihat pada Gambar 3.23 dan Gambar 3.24.



Gambar 3.23 *Physical Data Model*



Gambar 3.24 Conceptual Data Model

3.2.2.1 Rancangan Tabel Map

Tabel Map digunakan untuk menyimpan data peta yang digunakan dalam studi kasus. Tabel Map menyimpan data yang akan digunakan untuk menentukan peta mana yang digunakan. Tabel ini memiliki relasi dengan tabel lainnya yakni:

1. Tabel Vertex

Hubungan dengan tabel ini adalah menyimpan data ruangan yang ada pada peta. Ini berguna agar mengetahui ruangan apa saja yang ada pada suatu gedung bertingkat.

Detail Tabel Map dijelaskan pada Tabel 3.11.

Tabel 3.11 Atribut Tabel Map

Nama Kolom	Type Data	Keterangan
m_id	<i>Integer</i>	<i>Primary key</i> dari Tabel Map
nama_map	<i>Text</i>	Nama gedung

3.2.2.2 Rancangan Tabel Vertex

Tabel Vertex digunakan untuk menyimpan data ruangan-ruangan yang termasuk dalam studi kasus. Tabel Vertex menyimpan data ruangan berupa koordinat ruangan, status, dan gedung dimana ruangan tersebut berada. Tabel ini memiliki relasi ke tabel lainnya yaitu:

1. Tabel Map
Tabel ini menyimpan data gedung dimana sebuah ruangan berada.
2. Tabel Edge
Tabel ini menyimpan semua *edge* yang berhubungan dengan suatu *vertex*.

Detail Tabel Vertex dijelaskan pada Tabel 3.12.

Tabel 3.12 Atribut Tabel Vertex

Nama Kolom	Tipe Data	Keterangan
id_ruang	<i>Integer</i>	<i>Primary key</i> dari Tabel Vertex
m_id	<i>Integer</i>	<i>Foreign key</i> dari Tabel Map
nama_ruang	<i>Text</i>	Nama dari ruangan
x	<i>Real</i>	Koordinat sumbu x dari ruangan
y	<i>Real</i>	Koordinat sumbu y dari ruangan
z	<i>Real</i>	Koordinat sumbu z dari ruangan
lantai	<i>Integer</i>	Lantai dimana ruangan tersebut berada

status	<i>Text</i>	Status untuk menampilkan ruangan atau tidak
rotate_x	<i>Real</i>	Rotasi sumbu x dari ruangan
rotate_y	<i>Real</i>	Rotasi sumbu y dari ruangan
rotate_z	<i>Real</i>	Rotasi sumbu z dari ruangan
F.o.V	<i>Integer</i>	Koefisien perspektif untuk kamera

3.2.2.3 Rancangan Tabel Edge

Tabel Edge digunakan untuk menyimpan semua jalur dari dan ke suatu ruang pada sebuah gedung. Tabel lokasi menyimpan data berupa *vertex* asal dan *vertex* tujuan beserta dengan *cost* antar *vertex* tersebut. Tabel ini memiliki relasi ke tabel lainnya yaitu:

1. Tabel Vertex

Tabel ini menyimpan informasi dari *vertex* yang ada di Tabel Edge.

Detail Tabel Edge dijelaskan pada Tabel 3.13.

Tabel 3.13 Atribut Tabel Edge

Nama Kolom	 Tipe Data	Keterangan
e_id	<i>Integer</i>	<i>Primary key</i> dari Tabel Edge
v1	<i>Integer</i>	<i>Vertex</i> asal
v2	<i>Integer</i>	<i>Vertex</i> tujuan
cost	<i>Real</i>	Jarak antar <i>vertex</i>

3.2.3 Perancangan Antarmuka Aplikasi Perangkat Bergerak

Pada subbab ini akan dibahas secara mendetail dari rancangan antarmuka perangkat lunak **QR MAP**.

3.2.3.1 Antarmuka Halaman Utama

Gambar 3.25 merupakan gambar rancangan antarmuka halaman utama. Halaman ini baru ketika perangkat lunak pertama kali dijalankan. Pada halaman utama ini, pengguna dapat membuka semua halaman yang ada pada perangkat lunak.



Gambar 3.25 Perancangan Antarmuka Halaman Utama

Komponen-Komponen yang ada pada halaman utama adalah sebagai berikut:

1. **textTitle** merupakan nama perangkat lunak.
2. **textHow** merupakan petunjuk penggunaan perangkat lunak.
3. **Image** merupakan logo perangkat lunak.
4. **textScan** untuk melakukan proses pemindaian QR Code.

5. **textAbout** untuk melihat informasi perangkat lunak.
6. **textQuit** untuk keluar dari aplikasi

3.2.3.2 Antarmuka Halaman Pemindaian QR Code

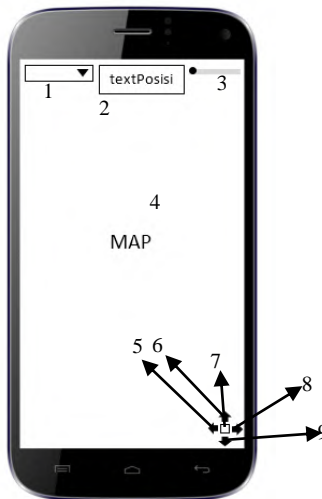
Gambar 3.26 merupakan gambar rancangan antarmuka halaman untuk melakukan pemindaian QR Code. Pada halaman ini, perangkat lunak akan melakukan pemindaian QR Code, ketika informasi sudah diperoleh, akan berpindah ke halaman selanjutnya. Pemindaian akan dilakukan terus menerus sampai informasi diperoleh. Pada antarmuka ini hanya terdapat satu komponen yaitu **RawImage** untuk menampilkan pemindaian QR Code.



**Gambar 3.26 Perancangan
Halaman Pemindaian QR Code**

3.2.3.3 Antarmuka Halaman untuk Memilih Tujuan dan Melihat Rute

Gambar 3.27 merupakan gambar rancangan antarmuka halaman untuk memilih tujuan dan melihat rute yang harus ditempuh. Pada antarmuka ini akan tersedia panel navigasi dan *slider* yang berfungsi mengatur kecepatan navigasi. Selain itu juga, pengguna dapat melakukan proses pemindaian QR Code untuk meng-*update* posisinya sekarang.



Gambar 3.27 Perancangan Halaman Memilih Tujuan dan Melihat Rute

Berikut penjelasan masing-masing nomor yang tertera pada Gambar 3.27.

1. Berupa *dropdown list* yang digunakan untuk menampilkan daftar tujuan yang tersedia.
2. **textPosisi** untuk menampilkan posisi pengguna.

3. Berupa *slider* untuk mengatur kecepatan perpindahan. Tersedia 3 tingkat kecepatan yang tersedia yaitu biasa, cepat, dan sangat cepat.
4. Berupa peta 3D dari bangunan.
5. Berupa *button* untuk melakukan navigasi ke arah kiri.
6. Berupa *button* untuk melakukan navigasi ke arah depan.
7. Berupa *button* untuk melakukan proses pemindaian posisi sekarang.
8. Berupa *button* untuk melakukan navigasi ke arah kanan.
9. Berupa *button* untuk melakukan navigasi ke arah belakang.

3.2.3.4 Antarmuka Halaman About

Gambar 3.28 merupakan gambar rancangan antarmuka halaman about. Halaman ini berisi informasi dari pembuat dan tujuan dibuatnya perangkat lunak ini beserta dosen yang membimbing selama pembuatan perangkat lunak ini. Pada antarmuka ini terdapat satu komponen yaitu **textInformasi** untuk menampilkan informasi dari perangkat lunak.



**Gambar 3.28 Perancangan
Antarmuka Halaman About**

3.2.3.5 Antarmuka Halaman Ketika Pengguna Telah Tiba di Tujuan

Gambar 3.29 merupakan rancangan dari antarmuka ketika pengguna telah tiba di tempat tujuan. Pada antarmuka ini akan terdapat daftar ruangan yang dilalui pengguna beserta jarak yang ditempuh oleh pengguna.



Gambar 3.29 Perancangan Halaman Ketika Pengguna Telah Tiba di Tujuan

Berikut ini merupakan penjelasan dari komponen-komponen pada Gambar 3.29:

1. **textDone** merupakan *text* yang menerangkan bahwa pengguna telah tiba di lokasi tujuan.

2. **textRoute** merupakan daftar jalur yang dilalui oleh pengguna untuk tiba di tempat tujuan dan juga berisi jarak yang ditempuh oleh pengguna.
3. **buttonOK** untuk kembali ke halaman utama.

3.2.3.6 Antarmuka Halaman untuk *Update Posisi Pengguna*

Gambar 3.30 merupakan rancangan dari antarmuka dimana pengguna ingin melakukan pembaruan posisi. Pada halaman ini pengguna dimungkinkan untuk melakukan pemindaian QR Code. Pada antarmuka ini hanya terdapat satu komponen yaitu **RawImage** untuk menampilkan pemindaian QR Code.



Gambar 3.30 Perancangan Halaman untuk *Update Posisi Pengguna*

3.2.3.7 Antarmuka Halaman untuk Menampilkan Pesan *Error*

Gambar 3.31 merupakan rancangan dari antarmuka untuk menampilkan pesan *error* ketika hasil pemindaian QR Code tidak sesuai dengan syarat yang ditentukan. Pesan *error* yang ditampilkan tergantung pada syarat yang tidak dipenuhi.



Gambar 3.31 Perancangan Halaman untuk Menampilkan Pesan *Error*

Berikut ini merupakan penjelasan dari komponen pada Gambar 3.31:

1. **textError** untuk menampilkan pesan *error* dari hasil pemindaian QR Code.
2. **buttonBack** untuk kembali ke halaman utama.

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

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

4.1 Lingkungan Implementasi

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

4.1.1 Lingkungan Implementasi Perangkat Keras

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

- Laptop Sony VPCSA35GG
 - Windows 7 Pro 64-bit,
 - Prosesor Intel® Core™ i7-2640M CPU @ 2.8GHz, dan
 - RAM 6.00 GB.
- Perangkat *smartphone* Samsung Galaxy Note 4 SM-N910H
 - Sistem Operasi : Android v5.0.1 (Lollipop),
 - CPU : Quad-core 1.9 GHz Cortex-A57,
 - Memori internal : 32.00 GB, dan
 - RAM : 3.00 GB.

4.1.2 Lingkungan Implementasi Perangkat Lunak

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

- Microsoft Windows 7 Pro sebagai sistem operasi,
- Unity 5.3 untuk mengimplementasikan aplikasi perangkat bergerak,
- SQLite sebagai basis data lokal,
- Power Designer untuk merancang basis data,
- StarUML untuk merancang diagram perencanaan perangkat lunak, dan
- Google SketchUp 2016 untuk merancang peta 3D.

4.2 Implementasi Basis Data

Pada subbab ini akan dibahas implementasi dari rancangan basis data yang telah dibahas pada Bab 3. Implementasi struktur basis data merupakan implementasi dari sintaks SQL yang digunakan untuk membangun tabel-tabel yang dibutuhkan pada perangkat lunak **QR MAP**.

4.2.1 Implementasi Tabel Map

Tabel Map digunakan untuk menyimpan data gedung yang terdapat pada perangkat lunak. Tabel ini digunakan untuk mendapatkan data ruang yang ada pada suatu gedung. Implementasi dari Tabel Map dapat dilihat pada Lampiran Kode Sumber B.1.

4.2.2 Implementasi Tabel Vertex

Tabel Vertex digunakan untuk menyimpan data lokasi dari tiap ruang yang berada pada suatu gedung. Selain itu, tabel ini juga berisi beberapa titik yang menjadi penghubung antar titik. Implementasi dari Tabel Vertex dapat dilihat pada Lampiran Kode Sumber B.2.

4.2.3 Implementasi Tabel Edge

Tabel Edge digunakan untuk menyimpan data dari dua buah ruangan yang berhubungan. Tabel ini digunakan dalam proses Dijkstra. Implementasi dari Tabel Edge dapat dilihat pada Lampiran Kode Sumber B.3.

4.3 Implementasi Antarmuka

Pada subbab ini akan dibahas implementasi dari antarmuka perangkat lunak berdasarkan rancangan antarmuka yang telah dibahas pada Bab 3. Implementasi antarmuka yang akan dibahas adalah antarmuka aplikasi perangkat bergerak. Implementasi dari fungsi yang terdapat pada antarmuka aplikasi ini, seperti implementasi algoritma Dijkstra, pembacaan QR Code, fungsi untuk melakukan proses navigasi, fungsi pembaruan posisi, dan *query* ke basis data terdapat pada Lampiran A.

4.3.1 Antarmuka Halaman Utama

Halaman Utama merupakan halaman pertama yang akan ditampilkan ketika aplikasi pertama kali dijalankan. Pada halaman ini terdapat 3 (tiga) menu yang dapat dipilih oleh pengguna, yaitu *Scan* untuk melakukan pemindaian QR Code untuk pertama kali, *About* untuk melihat informasi dari aplikasi, dan *Quit* untuk keluar dari aplikasi. Implementasi dari halaman utama dapat dilihat pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka Halaman Utama

4.3.2 Antarmuka Halaman Pemindaian QR Code

Halaman pemindaian QR Code merupakan antarmuka yang akan tampil ketika menu Scan dipilih pada halaman Utama. Pada halaman ini aplikasi akan melakukan pemindaian terhadap QR Code, setelah berhasil melakukan pemindaian, aplikasi akan berpindah ke halaman berikutnya. Implementasi dari halaman ini dapat dilihat pada Gambar 4.2. Untuk QR Code yang digunakan, dapat dilihat pada Lampiran C.

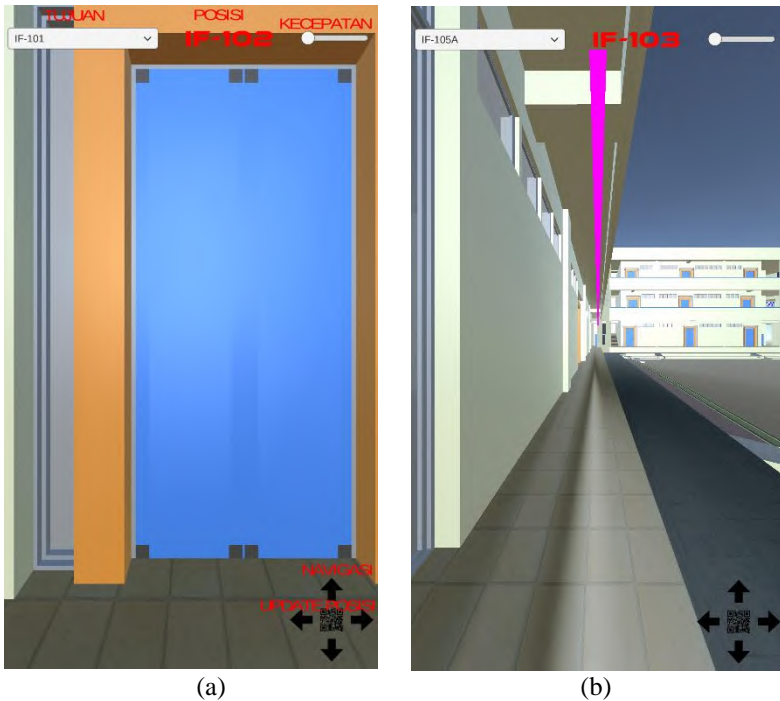


Gambar 4.2 Implementasi Halaman Pemindaian QR Code

4.3.3 Antarmuka Halaman untuk Memilih Tujuan dan Melihat Rute

Halaman ini akan ditampilkan ketika aplikasi berhasil melakukan pemindaian QR Code. Ruang yang ditampilkan pada halaman ini sesuai dengan hasil pemindaian QR Code pada halaman pemindaian QR Code. Pada halaman ini, pengguna dapat memilih tujuan yang dikehendaki pada *dropdown list* yang tersedia, saat tujuan sudah dipilih, maka sistem akan mencari rute terpendek menggunakan algoritma Dijkstra dan akan menampilkan sebuah garis yang menjadi panduan untuk dapat tiba di tujuan. Pengguna juga dapat melakukan proses navigasi seperti berjalan ke depan, mundur, lihat kiri dan kanan, dan juga pengguna dapat mengatur kecepatan navigasi pada *panel slider* yang tersedia, dimana terdapat 3 tingkat kecepatan. Pada halaman ini akan ditampilkan informasi berupa ruangan yang dilewati oleh

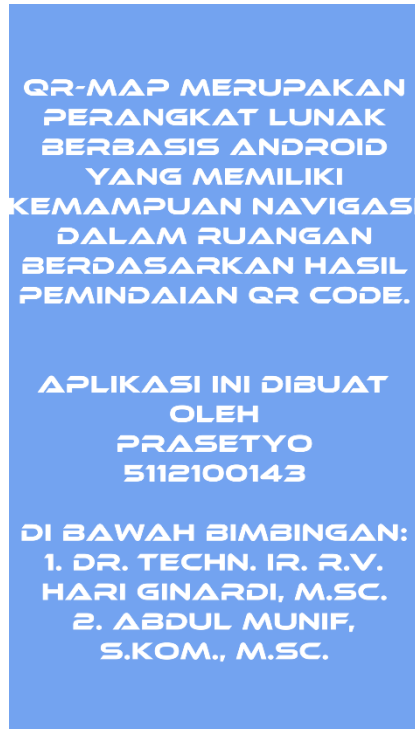
pengguna. Ketika pengguna ini melakukan pemindaian ulang untuk mengecek posisinya sekarang, disediakan sebuah *button* yang terletak di antara 4 panel navigasi. Implementasi dari halaman ini ditunjukkan pada Gambar 4.3.



Gambar 4.3 (a) Implementasi Halaman untuk Memilih Tujuan, (b) Implementasi Halaman untuk Menampilkan Rute

4.3.4 Antarmuka Halaman About

Halaman ini akan ditampilkan ketika pengguna memilih menu About pada halaman Utama. Pada halaman ini terdapat informasi singkat mengenai aplikasi. Implementasi dari halaman ini dapat dilihat pada Gambar 4.4.



**Gambar 4.4 Implementasi
Antarmuka Halaman About**

4.3.5 Antarmuka Halaman Ketika Pengguna Telah Tiba di Tujuan

Halaman ini akan muncul ketika pengguna melakukan proses navigasi hingga tiba di tujuan yang dipilih. Pada halaman ini terdapat informasi mengenai ruang atau titik mana yang dilalui oleh pengguna hingga sampai di tujuan dan juga ditampilkan jarak yang telah ditempuh pengguna dalam satuan sentimeter (cm). Pada halaman ini juga tersedia sebuah *button* untuk berpindah ke halaman utama. Implementasi dari halaman ini dapat dilihat pada Gambar 4.5.



Gambar 4.5 Implementasi Halaman Ketika Pengguna Telah Tiba di Tujuan

4.3.6 Antarmuka Halaman untuk *Update Posisi Pengguna*

Halaman ini akan ditampilkan ketika pengguna ingin melakukan proses pembaruan posisi untuk mendapatkan posisi dari pengguna yang lebih akurat. Untuk menampilkan halaman ini, pengguna dapat menekan *button* yang ada di antara 4 panel navigasi. Ketika aplikasi selesai melakukan pemindaian QR Code, maka posisi pengguna akan secara otomatis dipindah berdasarkan posisi hasil pemindaian yang dilakukan. Implementasi dari halaman ini dapat dilihat pada Gambar 4.6.



**Gambar 4.6 Implementasi
Halaman untuk Update Posisi
Pengguna**

4.3.7 Antarmuka Halaman untuk Menampilkan Pesan *Error*

Halaman ini akan ditampilkan ketika hasil pemindaian QR Code tidak sesuai dengan syarat yang telah ditentukan. Syarat yang digunakan yaitu hasil pemindaian QR Code harus sesuai dengan format yang digunakan dan hasil pemindaian dapat ditemukan pada lokasi penyimpanan data. Ketika *button* yang terdapat pada halaman ini ditekan, maka aplikasi akan menampilkan halaman Utama. Implementasi dari halaman ini dapat dilihat pada Gambar 4.7.

**MAP TIDAK
DITEMUKAN**

BACK

**Gambar 4.7 Antarmuka Halaman
untuk Menampilkan Pesan *Error***

BAB V

UJI COBA DAN EVALUASI

Bab ini membahas tentang uji coba dan evaluasi dari aplikasi yang dibuat pada tugas akhir ini. Pengujian dilakukan pada fungsionalitas perangkat lunak dan pengujian pemasangan perangkat lunak pada beberapa perangkat *smartphone*. Pembahasan tentang uji coba dan evaluasi meliputi lingkungan uji coba, dasar pengujian, pengujian fungsionalitas, pengujian perangkat lunak pada beberapa perangkat *smartphone*, dan evaluasi pengujian.

5.1 Lingkungan Uji Coba

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

Tabel 5.1 Spesifikasi Lingkungan Pengujian Perangkat Lunak

Nama <i>Smartphone</i>	Samsung Note 4
Sistem Operasi	Android OS, v5.0.1
Prosesor	Quad-core 1.9 GHz Cortex-A57
RAM	3 GB

5.2 Dasar Pengujian

Pengujian yang dilakukan berupa pengujian fungsionalitas dan pengujian jalannya perangkat lunak pada beberapa perangkat *smartphone*. Pengujian fungsionalitas dilakukan dengan model *black box* untuk masing-masing fungsionalitas dari aplikasi ini. Pengujian ini dilakukan untuk menguji apakah fungsionalitas yang

diidentifikasi pada tahap kebutuhan sudah diimplementasikan dengan baik dan bekerja seperti yang diharapkan. Pengujian jalannya perangkat lunak pada beberapa *smartphone* dilakukan untuk mengetahui apakah perangkat lunak berjalan dengan baik pada semua *smartphone* yang diuji ataukah tidak.

5.3 Pengujian Fungsionalitas

Pada subbab ini dibahas tentang pengujian yang dilakukan pada aplikasi **QR MAP**. Pengujian dilakukan oleh pengguna dengan tujuan untuk mengetahui lokasi keberadaan pengguna pada suatu gedung. Pengujian ini dilakukan secara sistematis sebagai tolak ukur keberhasilan sistem. Validasi pengujian ini ditentukan dengan informasi yang ditampilkan pada aplikasi.

5.3.1 Pengujian Melakukan Pemindaian QR Code

Berikut ini merupakan pengujian untuk melakukan pemindaian QR Code. Pada tahap pengujian ini, aplikasi akan terus melakukan pemindaian hingga QR Code selesai dibaca, kemudian hasil pemindaian akan ditampilkan. Skenario pengujian fungsionalitas ini terdapat pada Tabel 5.2. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.1.

Tabel 5.2 Skenario Pengujian Melakukan Pemindaian QR Code

Nomor	PF-01
Nama	Melakukan Pemindaian QR Code
Use Case	UC-001
Tujuan	Mendapatkan posisi pengguna berdasarkan hasil pemindaian QR Code.
Kondisi Awal	Belum mendapatkan id map dan id ruangan
Skenario	Pengguna memindai QR Code
Masukan	-

Keluaran yang diharapkan	Ruangan yang ditampilkan sesuai dengan hasil pembacaan QR Code
Hasil Pengujian	Berhasil



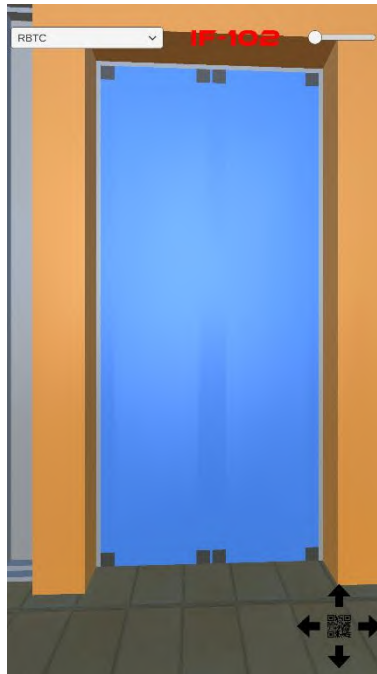
**Gambar 5.1 Pengujian
Melakukan Pemindaian QR Code**

5.3.2 Pengujian Menampilkan Lokasi Pengguna

Berikut ini merupakan pembahasan untuk pengujian menampilkan lokasi pengguna. Pada pengujian ini, aplikasi harus dapat menunjukkan lokasi pengguna berdasarkan hasil pemindaian QR Code. Skenario untuk pengujian menampilkan lokasi pengguna dapat dilihat pada Tabel 5.3. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.2.

Tabel 5.3 Skenario Pengujian Menampilkan Lokasi Pengguna

Nomor	PF-02
Nama	Menampilkan Lokasi Pengguna
Use Case	UC-002
Tujuan	Menunjukkan lokasi pengguna pada suatu gedung.
Kondisi Awal	Lokasi pengguna belum diketahui
Skenario	<ol style="list-style-type: none"> 1. Sistem mendapatkan id map dan id ruangan dari hasil pemindaian. 2. Sistem mengambil koordinat ruangan berdasarkan id ruangan. 3. Sistem menampilkan lokasi pengguna.
Masukan	-
Keluaran yang diharapkan	Lokasi pengguna ditampilkan
Hasil Pengujian	Berhasil



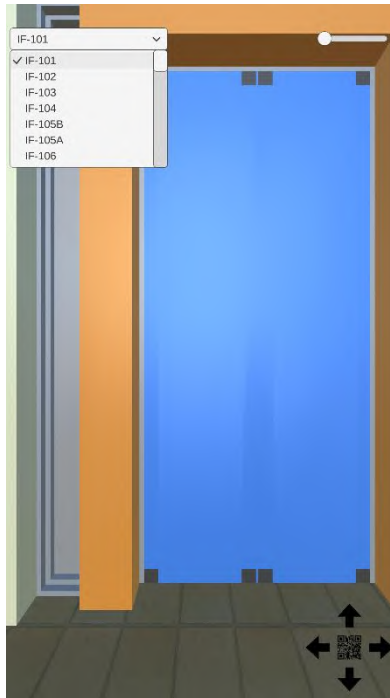
Gambar 5.2 Pengujian Menampilkan Lokasi Pengguna

5.3.3 Pengujian Menampilkan Daftar Lokasi Tujuan

Berikut ini merupakan pembahasan untuk menampilkan daftar lokasi tujuan. Pada pengujian ini, aplikasi mampu menampilkan daftar lokasi tujuan berdasarkan id map yang didapat dari hasil pemindaian. Skenario untuk pengujian menampilkan daftar lokasi tujuan dapat dilihat pada Tabel 5.4. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.3.

Tabel 5.4 Skenario Pengujian Menampilkan Lokasi Tujuan

Nomor	PF-03
Nama	Menampilkan Daftar Lokasi Tujuan
Use Case	UC-003
Tujuan	Menampilkan daftar lokasi tujuan yang tersedia pada suatu gedung
Kondisi Awal	Daftar Lokasi tujuan belum ditampilkan
Skenario	<ol style="list-style-type: none"> 1. Sistem mendapatkan id map dari hasil pemindaian. 2. Sistem menampilkan daftar lokasi tujuan berdasarkan id map yang didapat.
Masukan	-
Keluaran yang diharapkan	Daftar lokasi tujuan ditampilkan
Hasil Pengujian	Berhasil



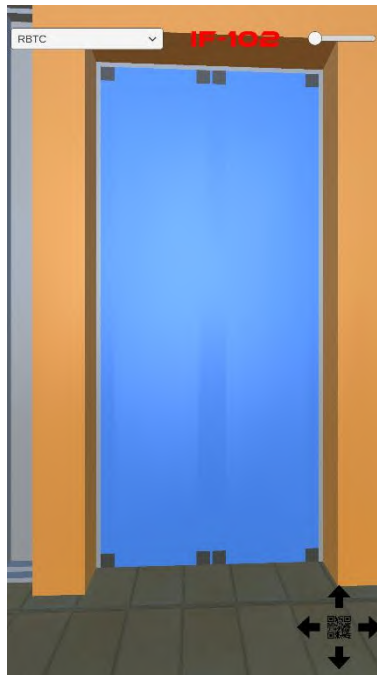
Gambar 5.3 Pengujian Menampilkan Lokasi Tujuan

5.3.4 Pengujian Memilih Lokasi Tujuan

Berikut ini merupakan pembahasan untuk pengujian memilih lokasi tujuan. Pada pengujian ini, aplikasi harus dapat menyimpan lokasi tujuan yang dipilih oleh pengguna. Skenario untuk pengujian memilih lokasi tujuan dapat dilihat pada Tabel 5.5. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.4.

Tabel 5.5 Skenario Pengujian Memilih Lokasi Tujuan

Nomor	PF-04
Nama	Memilih Lokasi Tujuan
Use Case	UC-004
Tujuan	Memilih lokasi tujuan yang diinginkan
Kondisi Awal	Lokasi tujuan belum dipilih
Skenario	<ol style="list-style-type: none">1. Pengguna memilih lokasi tujuan yang diinginkan.2. Sistem menyimpan lokasi tujuan yang dipilih pengguna.
Masukan	-
Keluaran yang diharapkan	Lokasi tujuan sudah dipilih
Hasil Pengujian	Berhasil



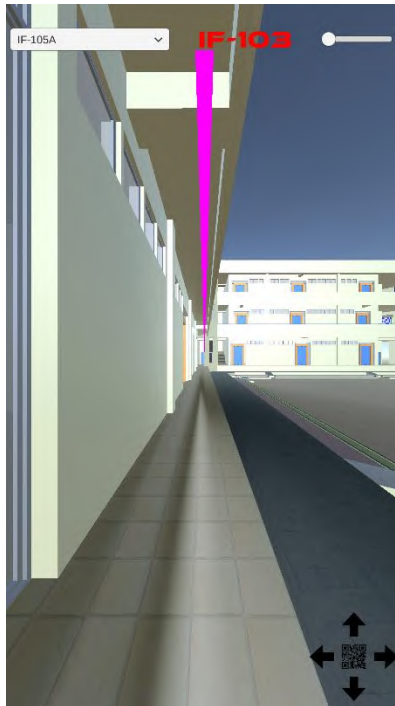
Gambar 5.4 Pengujian Memilih Lokasi Tujuan

5.3.5 Pengujian Menampilkan Rute ke Tempat Tujuan

Berikut ini merupakan pembahasan untuk pengujian menampilkan rute ke tempat tujuan. Pada pengujian kali ini, aplikasi harus dapat menghitung jarak terpendek dari posisi pengguna ke lokasi tujuan pengguna menggunakan algoritma Dijkstra. Aplikasi juga harus mampu menampilkan garis yang menjadi panduan untuk dapat tiba di tujuan. Skenario untuk pengujian menampilkan rute ke tempat tujuan dapat dilihat pada Tabel 5.6. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.5.

Tabel 5.6 Skenario Menampilkan Rute ke Tempat Tujuan

Nomor	PF-05
Nama	Menampilkan Rute ke Tempat Tujuan
Use Case	UC-005
Tujuan	Menampilkan rute ke tempat tujuan
Kondisi Awal	Rute belum ditampilkan
Skenario	<ol style="list-style-type: none">1. Sistem mencari jarak terpendek dari posisi pengguna ke tujuan yang dipilih.2. Sistem membuat rute berdasarkan jarak terpendek.
Masukan	Lokasi tujuan
Keluaran yang diharapkan	Rute ditampilkan
Hasil Pengujian	Berhasil



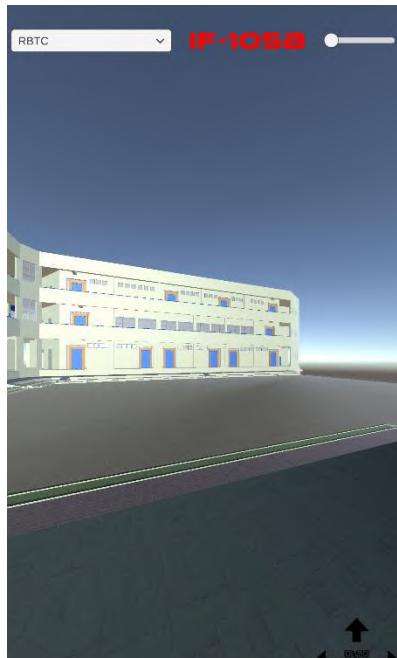
Gambar 5.5 Menampilkan Rute ke Tempat Tujuan

5.3.6 Pengujian Melakukan Proses Navigasi

Berikut ini merupakan pembahasan untuk pengujian melakukan proses navigasi. Pada pengujian ini, dilakukan uji coba untuk maju, mundur, putar kiri, dan putar kanan. Skenario untuk pengujian melakukan proses navigasi dapat dilihat pada Tabel 5.7. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.6.

Tabel 5.7 Skenario Melakukan Proses Navigasi

Nomor	PF-06
Nama	Melakukan Proses Navigasi
Use Case	UC-006
Tujuan	Berpindah posisi
Kondisi Awal	Posisi tidak berubah
Skenario	<ol style="list-style-type: none"> 1. Pengguna menekan salah satu <i>button</i> navigasi. 2. Sistem mengubah posisi sesuai dengan <i>button</i> yang ditekan.
Masukan	<i>Button</i> yang dipilih
Keluaran yang diharapkan	Posisi berubah sesuai <i>button</i> yang diklik
Hasil Pengujian	Berhasil



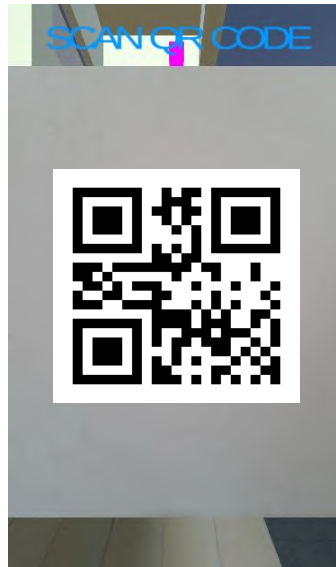
Gambar 5.6 Pengujian Melakukan Proses Navigasi

5.3.7 Pengujian Melakukan Pembaruan Posisi Pengguna

Berikut ini merupakan pembahasan untuk pengujian melakukan pembaruan posisi pengguna. Pada pengujian ini, dilakukan uji coba dimana aplikasi mampu melakukan pembaruan posisi pengguna berdasarkan hasil pemindaian QR Code. Skenario untuk pengujian melakukan pembaruan posisi pengguna dapat dilihat pada Tabel 5.8. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.7.

Tabel 5.8 Skenario Melakukan Pembaruan Posisi Pengguna

Nomor	PF-07
Nama	Melakukan Pembaruan Posisi Pengguna
Use Case	UC-007
Tujuan	Memperbarui posisi pengguna
Kondisi Awal	Posisi belum diperbarui
Skenario	<ol style="list-style-type: none"> 1. Pengguna menekan tombol <i>update</i>. 2. Sistem melakukan pemindaian QR Code. 3. Sistem melakukan pembaruan posisi berdasarkan hasil pemindaian.
Masukan	Tombol <i>update</i> yang berada di antara 4 panel navigasi ditekan
Keluaran yang diharapkan	Posisi berubah sesuai hasil pemindaian
Hasil Pengujian	Berhasil



**Gambar 5.7 Pengujian
Melakukan Pembaruan
Posisi Pengguna**

5.3.8 Pengujian Menampilkan Ringkasan Jalur yang Dilewati

Berikut ini merupakan pembahasan untuk pengujian menampilkan ringkasan jalur yang dilewati. Skenario untuk pengujian menampilkan ringkasan jalur yang dilewati dapat dilihat pada Tabel 5.9. Hasil pengujian fungsionalitas ini dapat dilihat pada Gambar 5.8.

Tabel 5.9 Skenario Menampilkan Ringkasan Jalur yang Dilewati

Nomor	PF-08
Nama	Menampilkan Ringkasan Jalur yang Dilewati

Use Case	UC-008
Tujuan	Menampilkan jalur yang dilewati pengguna dan jarak yang ditempuh
Kondisi Awal	Ringkasan jalur belum ditampilkan
Skenario	<ol style="list-style-type: none"> 1. Pengguna tiba di tujuan. 2. Sistem menampilkan ringkasan jalur yang dilewati pengguna dan jarak yang ditempuh.
Masukan	-
Keluaran yang diharapkan	Ringkasan jalur sudah ditampilkan
Hasil Pengujian	Berhasil



Gambar 5.8 Pengujian Menampilkan Jalur yang Dilewati

5.3.9 Pengujian Pada *Smartphone*

Pengujian ini dilakukan untuk melihat perilaku aplikasi ketika dijalankan pada beberapa perangkat bergerak berbasis Android yang berbeda. Pengujian ini juga dilakukan untuk melihat apakah struktur dari aplikasi berubah ketika dijalankan pada beberapa perangkat yang berbeda. Perangkat *smartphone* utama yang digunakan adalah Samsung Note 4. Untuk pengujian ini, akan ada beberapa *smartphone* yang digunakan. Perangkat *smartphone* yang digunakan pengujian ini dapat dilihat pada Tabel 5.10.

Tabel 5.10 Perangkat *Smartphone* Pengujian

Nama <i>Smartphone</i>	Sistem Operasi	Prosesor	RAM	Hasil
Samsung Note 4	Android OS, v5.0.1	Quad-core 1.9 GHz Cortex-A57	3 GB	Berhasil
Lenovo P70	Android OS, v4.4.4	Octa-core 1.7 GHz	2 GB	Berhasil
Xiaomi Red Mi 2 Prime	Android OS, v5.1.1	Quad-core 1.2 GHz	2 GB	Berhasil

5.4 Evaluasi Pengujian

Berdasarkan hasil pengujian fungsionalitas dan pengujian pada *smartphone*, maka dapat dilakukan evaluasi sebagai berikut.

5.4.1 Evaluasi Pengujian Fungsionalitas

Pengujian fungsionalitas yang telah dilakukan memberikan hasil yang sesuai dengan skenario yang telah direncanakan.

Evaluasi pengujian pada masing-masing fungsionalitas dijelaskan sebagai berikut.

1. Pengujian melakukan pemindaian QR Code telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-01 yang memberikan informasi bahwa proses melakukan pemindaian QR Code telah berjalan dengan baik.
2. Pengujian menampilkan lokasi pengguna telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-02 yang memberikan informasi bahwa proses menampilkan lokasi pengguna telah berjalan dengan baik.
3. Pengujian menampilkan daftar lokasi tujuan telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-03 yang memberikan informasi bahwa proses menampilkan daftar lokasi tujuan telah berjalan dengan baik.
4. Pengujian memilih lokasi tujuan telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-04 yang memberikan informasi bahwa proses memilih lokasi tujuan telah berjalan dengan baik.
5. Pengujian menampilkan rute ke tempat tujuan telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-05 yang memberikan informasi bahwa proses menampilkan rute ke tempat tujuan telah berjalan dengan baik.
6. Pengujian melakukan proses navigasi telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-06 yang memberikan informasi bahwa

proses melakukan proses navigasi telah berjalan dengan baik.

7. Pengujian melakukan pembaruan posisi pengguna telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-07 yang memberikan informasi bahwa proses melakukan pembaruan posisi pengguna telah berjalan dengan baik
8. Pengujian menampilkan ringkasan jalur yang dilewati telah sesuai dengan yang diharapkan. Kondisi ini ditunjukkan pada pengujian UC-08 yang memberikan informasi bahwa proses menampilkan ringkasan jalur yang dilewati telah berjalan dengan baik.

5.4.2 Evaluasi Pengujian *Smartphone*

Untuk mengevaluasi perilaku sistem pada perangkat *smartphone*, telah dilakukan pengujian pada beberapa *smartphone*. Seperti yang terlihat pada Tabel 5.10 di atas, versi sistem operasi paling rendah yang digunakan adalah Android v.4.4.4 (KitKat) dan yang paling tinggi adalah Android v.5.1.1 (Lollipop), hal ini menunjukkan bahwa aplikasi telah berjalan pada lingkungan yang diharapkan yaitu minimum SDK versi 4.4.4. Ketika dilakukan pengujian pada 3 buah *smartphone* di atas, ditemukan bahwa sistem berjalan sesuai dengan yang diharapkan, dimana semua fungsi pada aplikasi berjalan dengan baik.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A – IMPLEMENTASI FUNGSI PADA ANTARMUKA

```
//algoritma Dijkstra - start
public void Dijkstra()
{
    int ori = 0;
    int dest = 0;
    int pos;
    double min;

    string destination = dropTujuan.captionText.text;
    for (int i = 100; i < 300; i++)
    {
        if (node_id [i] == origin)
        {
            ori = i;
        }

        if (node [i] == destination)
        {
            dest = i;

            if (dest == ori)
                SceneManager.LoadScene ("MapMenu");
        }
    }
    PlayerPrefs.SetInt ("PosisiAkhir", dest);
    for (int i = 0; i < 300; i++)
    {
        step [i] = 0;
        stack [i] = 0;
        visit [i] = 0;
        jarak [i] = 10000.0;
        asal [i] = -1;
    }

    pos = ori;
    jarak [ori] = 0;

    //Dijkstra
    while (pos != dest)
    {
```

```

for (int i = 100; i < 300; i++)
{
    if (edge [pos, i] + jarak [pos] < jarak [i])
    {
        jarak [i] = edge [pos, i] + jarak [pos];
        asal [i] = pos;
    }
}

visit [pos] = 1;
min = 10000.0;
for(int i=100; i<300; i++)
{
    if(jarak[i] < min && visit[i] == 0)
    {
        min = jarak[i];
        pos = i;
    }
}

if (min == 10000.0) {
    Debug.Log ("Rute tidak tersedia!");
    break;
} else {
    PlayerPrefs.SetFloat ("JarakMin", (float)min);
}
}

k=0;
step[0] = ori;
while(asal[pos] != -1)
{
    step[k] = pos;
    k++;

    pos = asal[pos];
}
//algoritma Dijkstra - end

```

Kode Sumber A.1 Implementasi Algoritma Dijkstra

```

void openDB()
{
    //akses database pada android
    string filepath = Application.persistentDataPath + "/" + db;
    if (!File.Exists (filepath))
    {
        WWW loadDb = new WWW ("jar:file://" + Application.dataPath + "!/assets/" + db
    );
        while (!loadDb.isDone) {
            };
            File.WriteAllBytes (filepath, loadDb.bytes);
        }

        conn = "URI=file:" + filepath;

//akses database pada unity
    //conn = "URI=file:" + Application.dataPath + "/StreamingAssets/" + db;

    sqlcon = new SqliteConnection (conn);
    sqlcon.Open ();

    dbcmd = sqlcon.CreateCommand();
    sqlQuery = "SELECT * FROM Edge";
    dbcmd.CommandText = sqlQuery;
    reader = dbcmd.ExecuteReader ();

    while (reader.Read ())
    {
        int v1 = reader.GetInt32 (1);
        int v2 = reader.GetInt32 (2);
        double w = reader.GetDouble (3);

        edge [v1, v2] = w;
        edge [v2, v1] = w;
    }
    reader.Close ();

    sqlQuery = "SELECT * FROM Vertex";
    dbcmd.CommandText = sqlQuery;
    reader = dbcmd.ExecuteReader ();

    while (reader.Read ())
    {
        int idruang = reader.GetInt32 (0);
        string ruang = reader.GetString (2);
        double x = reader.GetDouble (3);
    }
}

```

```

double y = reader.GetDouble (4);
double z = reader.GetDouble (5);
int lantai = reader.GetInt32 (6);
string status = reader.GetString (7);
double rotatex = reader.GetDouble (8);
double rotatey = reader.GetDouble (9);
double rotatez = reader.GetDouble (10);
int fov = reader.GetInt32 (11);

```

```

node_id [idruang] = idruang;
node [idruang] = ruang;
node_x [idruang] = x;
node_y [idruang] = y;
node_z [idruang] = z;
node_lantai [idruang] = lantai;
node_status [idruang] = status;
node_rotatex [idruang] = rotatex;
node_rotatey [idruang] = rotatey;
node_rotatez [idruang] = rotatez;
node_fov [idruang] = fov;
}
reader.Close ();

```

```

sqlQuery = "SELECT nama_ruang FROM Vertex where m_id = '"+PlayerPrefs.GetInt(
("NamaMap")+"'";

```

```

dbcmd.CommandText = sqlQuery;
reader = dbcmd.ExecuteReader ();

```

```

string nama = "";
bool isRead = false;
while (reader.Read ())

```

```

{
    isRead = true;
    nama = reader.GetString (0);

```

```

    dropTujuan.options.Add (new Dropdown.OptionData (nama));
}

```

```

reader.Close ();
reader = null;
dbcmd.Dispose ();
dbcmd = null;
sqlcon.Close ();
sqlcon = null;

```

```

if (lisRead)
{
    Debug.Log ("MAP BELUM TERSEDIA");
    PlayerPrefs.SetInt ("statusError", 3);
    SceneManager.LoadScene ("ErrorMenu");

    dropTujuan.captionText.text = "";
}
else
{
    dropTujuan.value = 0;
    dropTujuan.captionText.text = dropTujuan.options.ToArray () [0].text;
}
}

```

Kode Sumber A.2 Implementasi *Query* ke Basis Data

```

void posisiCameraAwal()
{
    if (statusawal == 1)//pertama kali dijalankan
    {
        transformCameraMain = cameraMain.GetComponent<Transform> ();
        transformCameraMain.position = new Vector3 ((float)node_x [origin], (float)node_
y [origin], (float)node_z [origin]);
        transformCameraMain.rotation = Quaternion.Euler (new Vector3 ((float)node_rotat
ex [origin], (float)node_rotatey [origin], (float)node_rotatez [origin]));
        cameraMain.fieldOfView = node_fov [origin];
        txt_position.text = node[origin];}
    else if (statusawal == 2)//scan kedua dan seterusnya
    {
        int cameraNow = 0;
        if (ok == 2)
        {
            cameraNow = PlayerPrefs.GetInt ("PosisiSekarang");

            transformCameraMain = cameraMain.GetComponent<Transform> ();
            transformCameraMain.position = new Vector3 ((float)node_x [cameraNow], (flo
at)node_y [cameraNow], (float)node_z [cameraNow]);
            txt_position.text = node[cameraNow]; }
        PlayerPrefs.SetInt ("StatusAwal", 1);    }}

```

Kode Sumber A.3 Implementasi Fungsi untuk Menampilkan Posisi Pengguna

```

void drawLine(int i, int ori, int dest)
{
    float x_ori = (float)node_x [ori];
    float y_ori = (float)node_y [ori];
    float z_ori = (float)node_z [ori];

    float x_dest = (float)node_x [dest];
    float y_dest = (float)node_y [dest];
    float z_dest = (float)node_z [dest];

    line [i] = new GameObject ();
    lineRender [i] = line [i].AddComponent<LineRenderer> ();
    lineRender [i].SetWidth ((float)0.03, (float)0.03);
    lineRender [i].SetPosition (0, new Vector3 (x_ori, y_ori+0.5f, z_ori));
    lineRender [i].SetPosition (1, new Vector3 (x_dest, y_dest+0.5f, z_dest));
}
}

```

Kode Sumber A.4 Implementasi Fungsi untuk Menampilkan Garis Penunjuk Rute yang Dilalui

```

public class QRMenu : MonoBehaviour
{
    public RawImage scanQR;
    public Text hasilQR;

    private WebCamTexture camTexture;
    private Thread qrThread;

    private Color32[] c;
    private int W, H;

    private string qrcode;
    private bool run = true;

    public Quaternion baseRotation;

    private int cekFormatQR;

    void OnEnable()
    {
        Debug.Log ("Camera play");
        if (camTexture != null)
        {

```

```

        camTexture.Play ();
        H = camTexture.height;
        W = camTexture.width;
    }
}

void OnDisable()
{
    if (camTexture != null)
    {
        camTexture.Pause ();
    }
}

void OnDestroy()
{
    qrThread.Abort ();
    camTexture.Stop ();
}

// Use this for initialization
void Start ()
{
    camTexture = new WebCamTexture ();
    scanQR.texture = camTexture;
    scanQR.material.mainTexture = camTexture;

    baseRotation = transform.rotation;

    qrcode = "";
    OnEnable ();

    c = camTexture.GetPixels32 ();//

    qrThread = new Thread (DecodeQR);
    qrThread.Start ();
}

// Update is called once per frame
void Update ()
{
    c = camTexture.GetPixels32 ();
    transform.rotation = baseRotation * Quaternion.AngleAxis (camTexture.videoRotationAngle, Vector3.right);

    check();

    if (Input.GetKeyDown (KeyCode.Escape))

```

```

    {
        SceneManager.LoadScene ("GameMenu");
    }
}

void DecodeQR()
{
    while (run)
    {
        try{
            BarcodeReader qrReader = new BarcodeReader();
            var hasilQR_temp = qrReader.Decode(c, W, H);

            if(hasilQR_temp != null)
            {
                qrcode = hasilQR_temp.Text;
                run = false;
            }
        }
        catch
        {
            //Debug.Log ("ERROR QRMenu");
            break;
        }
    }
}

void check()
{
    try
    {
        if (!run)
        {
            int map = 0;
            int posisi_awal = 0;
            string mapcopy = "";
            string posisicopy = "";
            //hasilQR.text = qrcode;

            int lengthDataQR = qrcode.Length;
            for (int i = 0; i < lengthDataQR; i++)
            {
                if (qrcode [i + 1] == '-')
                {
                    mapcopy = qrcode.Substring (0, i + 1);
                    posisicopy = qrcode.Substring(i+2, lengthDataQR-(i+2));
                    cekFormatQR += 1;
                    break;
                }
            }
        }
    }
}

```



```

    }

    if(cekFormatQR == 0)
    {
        PlayerPrefs.SetInt("statusError", 2);
        SceneManager.LoadScene("ErrorMenu");
    }

    if(int.TryParse(mapcopy, out map))
    {
        if(int.TryParse(posisicopy, out posisi_awal))
        {
            PlayerPrefs.SetInt("NamaMap", map);
            PlayerPrefs.SetInt("PosisiAwal", posisi_awal);

            OnDisable ();
            OnDestroy ();

            scanQR.texture = null ;
            scanQR.material.mainTexture = null;

            SceneManager.LoadScene ("MapMenu");
        }
        else
        {
            PlayerPrefs.SetInt("statusError", 2);
            SceneManager.LoadScene("ErrorMenu");
        }
    }
    else
    {
        PlayerPrefs.SetInt("statusError", 2);
        SceneManager.LoadScene("ErrorMenu");
    }
}
}
catch (System.Exception ex)
{
    hasilQR.text = "Error"+ex.ToString();
    PlayerPrefs.SetInt("statusError", 2);
    SceneManager.LoadScene("ErrorMenu");
}
}
}

```

Kode Sumber A.5 Implementasi Fungsi Pembacaan QR Code

```

void setPosition(int stepke)
{
    t = 0;
    for (int i = 1; i <= stepke; i++)
    {
        float tStep = Mathf.Sqrt ( (float) ((node_x1[step1[i-1]]-
node_x1[step1[i]])*(node_x1[step1[i-1]]-node_x1[step1[i]])
+ (node_y1[step1[i-1]]-node_y1[step1[i]])*(node_y1[step1[i-1]]-
node_y1[step1[i]])
+ (node_z1[step1[i-1]]-node_z1[step1[i]])*(node_z1[step1[i-1]]-
node_z1[step1[i]]));

        t += tStep;
    }
}

void setPosition()
{
    if (t < 0) t = 0;
    float tTotalStep = 0;
    jarakMin = PlayerPrefs.GetFloat("JarakMin");
    int i;
    for (i = 1; i <= jumlahStep; i++)
    {
        float tStep = Mathf.Sqrt ( (float) ((node_x1[step1[i-1]]-
node_x1[step1[i]])*(node_x1[step1[i-1]]-node_x1[step1[i]])
+ (node_y1[step1[i-1]]-node_y1[step1[i]])*(node_y1[step1[i-1]]-
node_y1[step1[i]])
+ (node_z1[step1[i-1]]-node_z1[step1[i]])*(node_z1[step1[i-1]]-
node_z1[step1[i]]));

        if (t < tTotalStep + tStep)
        {
            float tSkrng = t - tTotalStep;
            float nodeX = (float) (node_x1 [step1 [i - 1]] + tSkrng / tStep * (node_x1 [step1 [i]
] - node_x1 [step1 [i - 1]]));
            float nodeY = (float) (node_y1 [step1 [i - 1]] + tSkrng / tStep * (node_y1 [step1 [i]
] - node_y1 [step1 [i - 1]]));
            float nodeZ = (float) (node_z1 [step1 [i - 1]] + tSkrng / tStep * (node_z1 [step1 [i]
] - node_z1 [step1 [i - 1]]));

            findAngle (step1 [i - 1], step1 [i]);
            //Debug.Log (t.ToString()+" "+angle_y.ToString()+" "+angle_x.ToString());
            transformCameraMovement.rotation = Quaternion.Euler (new Vector3 ((float)-
angle_x, (float)angle_y, 0));
            transformCameraMovement.position = new Vector3 (nodex, nodey, nodez);

            if (node_status1 [step1 [i - 1]] == "aktif") {

```

```

        txt_pos.text = node1 [step1 [i - 1]];
    }

    break;
}

tTotalStep += tStep;
}

if (i > jumlahStep)
{
    if (node_status1 [step1 [i - 1]] == "aktif") {
        txt_pos.text = node1 [step1 [i - 1]];
    }
    t = tTotalStep;

    txt_tiba.text = "JALUR YANG DILALUI: \n\n";
    for (int j = 0; j <= jumlahStep; j++)
    {
        txt_tiba.text = txt_tiba.text + node1[step1 [j]] + "\n";
    }

    txt_tiba.text = txt_tiba.text + "Jarak yang ditempuh: " + jarakMin;
    canvasUtama.enabled = false;
    canvasDone.enabled = true;
}
}

void findAngle(int or, int de)
{
    double gradien_x = 0.0;
    double gradien_y = 0.0;

    double or_x = node_x1 [or];
    double or_y = node_y1 [or];
    double or_z = node_z1 [or];

    double dest_x = node_x1 [de];
    double dest_y = node_y1 [de];
    double dest_z = node_z1 [de];

    //gradien X
    if (Math.Sqrt((dest_z - or_z)*(dest_z - or_z)+(dest_x - or_x)*(dest_x - or_x)) == 0)
        gradien_x = -1000.0;
    else
        gradien_x = ((dest_y - or_y) / Math.Sqrt((dest_z - or_z)*(dest_z - or_z)+(dest_x -
or_x)*(dest_x - or_x)) );

    //gradien Y

```

```

if (dest_z == or_z)
    gradien_y = -1000.0;
else
    gradien_y = ((dest_x - or_x) / (dest_z - or_z));

if (gradien_x == -1000)
    angle_x = Mathf.PI / 2;
else
    angle_x = Mathf.Atan ((float) gradien_x);

if (gradien_y == -1000)
    angle_y = Mathf.PI / 2;
else
    angle_y = Mathf.Atan ((float) gradien_y);

double R_x = dest_x - or_x;
double R_y = dest_y - or_y;
double R_z = dest_z - or_z;

if (angle_y < 0)
    angle_y += Mathf.PI;

if (R_x > 0 || (R_x == 0 && R_z >= 0))
    angle_y = angle_y;
else
    angle_y = angle_y + Mathf.PI;

angle_x = angle_x / Mathf.PI * 180;
angle_y = angle_y / Mathf.PI * 180;
}

```

Kode Sumber A.6 Implementasi Fungsi untuk *Update* Posisi Kamera

```

void Start ()
{
    if (name == "btn_up")
        btn = 1;
    else if (name == "btn_down")
        btn = 2;
    else if (name == "btn_right")
        btn = 3;
    else if (name == "btn_left")
        btn = 4;
}

```

```
public void OnMouseDown()
{
    int[] tempstr = new int[2];
    tempstr [0] = 1;
    tempstr [1] = btn;

    cameraNavArrow.SendMessage ("movement", tempstr);
}

public void OnMouseUp()
{
    int[] tempstr = new int[2];
    tempstr [0] = 0;
    tempstr [1] = btn;

    cameraNavArrow.SendMessage ("movement", tempstr);
}

void Update ()
{
    if (cek1)
    {
        cek1 = false;
        t = 0;
    }

    if (cekScan == 1)
    {
        c = camTexture.GetPixels32 ();
        check ();
    }

    if (statusjalan == 0)
    {
        //t = 0;
    }
    else if (statusjalan == 1)
    {
        if (arahjalan == 1)
        {
            t += 0.5f + sliderAcc.value;
            setPosition ();
        }
        else if (arahjalan == 2)
```

```
{
    t -= 0.5f + sliderAcc.value;
    setPosition ();
}
else if (arahjalan == 3)
{
    transformCameraMovement.Rotate (0, (10 + sliderAcc.value) * Time.deltaTime
* sliderAcc.value, 0, Space.Self);
}
else if (arahjalan == 4)
{
    transformCameraMovement.Rotate (0, (-
(10 + sliderAcc.value)) * Time.deltaTime * sliderAcc.value, 0, Space.Self);
}
}
}
```

Kode Sumber A.7 Implementasi Fungsi untuk Panel Navigasi

LAMPIRAN B – KODE SUMBER

```
CREATE TABLE `Map` (  
    `m_id` INTEGER,  
    `nama_map` TEXT,  
    PRIMARY KEY(m_id)  
)
```

Kode Sumber B.1 Implementasi Tabel Map

```
CREATE TABLE "Vertex" (  
    `id_ruang` INTEGER,  
    `m_id` INTEGER,  
    `nama_ruang` TEXT,  
    `x` REAL,  
    `y` REAL,  
    `z` REAL,  
    `lantai` INTEGER,  
    `status` TEXT,  
    `rotate_x` REAL,  
    `rotate_y` REAL,  
    `rotate_z` REAL,  
    `F.o.V` INTEGER,  
    PRIMARY KEY(id_ruang),  
    FOREIGN KEY(`m_id`) REFERENCES `Map`  
)
```

Kode Sumber B.2 Implementasi Tabel Vertex

```
CREATE TABLE "Edge" (  
    `e_id` INTEGER NOT NULL PRIMARY KEY  
    AUTOINCREMENT UNIQUE,  
    `v1` INTEGER,  
    `v2` INTEGER,  
    `cost` REAL,
```





```
FOREIGN KEY(`v1`) REFERENCES `Ruangan`,  
FOREIGN KEY(`v2`) REFERENCES `Ruangan`  
)
```





Kode Sumber B.3 Implementasi Tabel Edge





LAMPIRAN C – QR CODE

Tabel C.1 Daftar QR Code yang Digunakan



No	Nama Ruang	Nama Gedung	QR Code
1	IF-101	Teknik Informatika ITS	
2	IF-102	Teknik Informatika ITS	
3	IF-103	Teknik Informatika ITS	
4	IF-104	Teknik Informatika ITS	





5	IF-105B	Teknik Informatika ITS	
6	IF-105A	Teknik Informatika ITS	
7	IF-106	Teknik Informatika ITS	
8	RBTC	Teknik Informatika ITS	





9	IF-108	Teknik Informatika ITS	
10	IF-109	Teknik Informatika ITS	
11	Lab. Pascasarjan a 1	Teknik Informatika ITS	
12	IF-110	Teknik Informatika ITS	

13	IF-111	Teknik Informatika ITS	
14	IF-112	Teknik Informatika ITS	
15	PIKTI	Teknik Informatika ITS	
16	Wudhu Cowok	Teknik Informatika ITS	

17	Wudhu Cewek	Teknik Informatika ITS	
18	Kantin 1	Teknik Informatika ITS	
19	Tangga Kantin 1	Teknik Informatika ITS	
20	WC Plasa Lama 1	Teknik Informatika ITS	





21	Plasa Lama 1	Teknik Informatika ITS	
22	Tangga Plasa Lama 1	Teknik Informatika ITS	
23	Tangga Plasa Baru 1	Teknik Informatika ITS	
24	Plasa Baru 1	Teknik Informatika ITS	

25	WC Plasa Baru 1	Teknik Informatika ITS	
26	Tangga PIKTI 1	Teknik Informatika ITS	
27	Ruang Dosen Plasa Lama	Teknik Informatika ITS	
28	Ruang Sidang	Teknik Informatika ITS	

29	Aula Informatika	Teknik Informatika ITS	
30	Ruang Dosen Plasa Baru	Teknik Informatika ITS	
31	Tangga Kantin 2	Teknik Informatika ITS	
32	WC Dosen Plasa Lama	Teknik Informatika ITS	




33	Plasa Lama 2	Teknik Informatika ITS	
34	Tangga Plasa Lama 2	Teknik Informatika ITS	
35	Tangga Plasa Baru 2	Teknik Informatika ITS	
36	Plasa Baru 2	Teknik Informatika ITS	

37	WC Dosen Plasa Baru	Teknik Informatika ITS	
38	Tangga PIKTI 2	Teknik Informatika ITS	
39	Lab. RPL	Teknik Informatika ITS	
40	Lab. KBJ	Teknik Informatika ITS	

41	Lab. KCV	Teknik Informatika ITS	
42	Lab. Pemrograman 1	Teknik Informatika ITS	
43	Lab. Pascasarjana 2	Teknik Informatika ITS	
44	Lab. AJK	Teknik Informatika ITS	

45	Lab. IGS	Teknik Informatika ITS	
46	Lab. Mobile	Teknik Informatika ITS	
47	Lab. Pemrograman 2	Teknik Informatika ITS	
48	Lab. ALPRO	Teknik Informatika ITS	

49	Lab. MI-DTK	Teknik Informatika ITS	
50	Tangga Kantin 3	Teknik Informatika ITS	
51	Kantin 2	Teknik Informatika ITS	
52	WC Plasa Lama 3	Teknik Informatika ITS	

53	Ruang HIMA	Teknik Informatika ITS	
54	Tangga Plasa Lama 3	Teknik Informatika ITS	
55	Tangga Plasa Baru 3	Teknik Informatika ITS	
56	WC Plasa Baru 3	Teknik Informatika ITS	

57	Tangga PIKTI 3	Teknik Informatika ITS	
----	-------------------	---------------------------	---

[Halaman ini sengaja dikosongkan]

BAB VI

KESIMPULAN DAN SARAN

Pada bab ini dijelaskan tentang kesimpulan yang didapat dalam pengerjaan tugas akhir ini dan saran yang dapat diberikan untuk pengembangan lebih lanjut dari sistem *Indoor Positioning* pada perangkat berbasis Android.

6.1 Kesimpulan

Dalam proses pengerjaan tugas akhir ini dari tahap pendahuluan, kajian pustaka, analisa, perancangan, implementasi dan pengujian sistem *Indoor Positioning* diperoleh kesimpulan sebagai berikut.

1. Aplikasi perangkat bergerak **QR MAP** dapat menentukan posisi pengguna dalam sebuah gedung bertingkat.
2. Aplikasi perangkat bergerak **QR MAP** dapat menunjukkan jalur yang dilalui oleh pengguna untuk tiba di lokasi tujuan dalam suatu gedung bertingkat berdasarkan perhitungan menggunakan algoritma Dijkstra.
3. Penggunaan peta yang tidak dibuat di Unity agak merepotkan terutama ketika akan diberikan *collider*, karena harus diberikan *collider* per bagian.
4. Lokasi penyimpanan basis data pada Unity dan Android berbeda, sehingga cara aksesnya pun tidak sama.
5. Aplikasi perangkat bergerak **QR MAP** dapat memberikan informasi berupa rute yang dilalui dan jarak yang ditempuh untuk dapat tiba di tujuan.
6. Aplikasi perangkat bergerak **QR MAP** tidak dapat menunjukkan posisi pengguna secara *real-time*.
7. Aplikasi perangkat bergerak **QR MAP** tidak dapat menyesuaikan tampilan dengan arah hadap pengguna.

6.2 Saran

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

1. Peta 3D yang akan digunakan sebaiknya dibuat pada Unity. Walaupun dibuat menggunakan perkakas bantu lain, dianjurkan untuk memberi grup sesuai bidang, untuk memudahkan proses *collider*.
2. Memperbaiki desain peta 3D yang sudah dibuat, dikarenakan ada beberapa bagian dimana rute yang dilalui terhalang oleh tembok.
3. Menambahkan jumlah gedung yang dapat diuji coba.
4. Memperbaiki desain tampilan sehingga lebih sederhana namun elegan.
5. Menambah fitur dimana aplikasi dapat menampilkan posisi pengguna secara *real-time*.
6. Menambah fitur dimana aplikasi dapat menyesuaikan tampilan sesuai dengan arah hadap pengguna menggunakan bantuan sensor giroskop dan kompas yang terdapat pada *smartphone*.

DAFTAR PUSTAKA

- [1] J. T. Sospedra, R. Montoliu, S. Trilles, O. Belmonte dan J. Huerta, "Comprehensive Analysis of Distance and Similarity Measures for Wi-Fi Fingerprinting Indoor Positioning Systems," *Expert Systems With Applications*, 2015.
- [2] M. F. Ghanianto, Implementasi Indoor Localization Menggunakan Sinyal WiFi dan Clustering Filtered K-Nearest Neighbors untuk Pelacakan Keberadaan Seseorang dan Evaluasi Akurasi Pelacakan di Kampus Teknik Informatika ITS, Surabaya, 2015.
- [3] N. Y. Arrifa, Implementasi Indoor Localization Menggunakan Sinyal Wi-Fi dan Decision Tree untuk Pelacakan Keberadaan Seseorang di Kampus Teknik Informatika ITS, Surabaya, 2014.
- [4] K. A. Nastiti, Sistem Layanan Pencatatan Kehadiran Mahasiswa Berbasis Mobile dengan Mengimplementasikan Indoor Localization Menggunakan Sinyal Wi-Fi dan Decision Tree di Kampus Teknik Informatika ITS, Surabaya, 2015.
- [5] C. Jingrong, K. Lei dan C. Xiaochuan, "An Approximation Algorithm for the Minimum Vertex Cover," *Procedia Engineering*, vol. 137, pp. 180-185, 2016.
- [6] H. Qi, X. Lu dan L. Lu, "A Localization Algorithm for Disorted or Rotated QR Code," dalam *Computing, Communication and Networking Technologies (ICCCNT), 2014 International Conference on*, Hefei, 2014.

- [7] Y. Zhuang, Z. Syed, J. Georgy dan N. El-Sheimy, "Autonomous Smartphone-based WiFi Position System by Using Access Points Localization and Crowdsourcing," *Pervasive and Mobile Computing*, vol. 18, pp. 118-136, 2015.

BIODATA PENULIS



Penulis, Prasetyo lahir di Kota Ambon pada 2 Maret 1995 dan dibesarkan di Kota Ambon. Penulis adalah anak pertama dari empat bersaudara. Penulis menempuh pendidikan formal di SD Xaverius A1 Ambon (2000-2006), SMP Negeri 6 Ambon (2006-2009), SMA Negeri 1 Ambon (2009-2012) dan S1 Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh

Nopember, Surabaya (2012-2016).

Selama kuliah di Teknik Informatika ITS, penulis memiliki ketertarikan pada bidang pemrograman Android. Penulis dapat dihubungi melalui alamat surel danielprasetyo5@gmail.com.