BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasikan permasalah-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikan. Analisis dapat juga diartikan sebagai penelitian atas sistem yang telah ada dengan tujuan untuk merancang sistem yang baru atau diperbarui. Dalam proses pembuatan suatu sistem mutlak dilakukan penelitian dan penganalisaan tentang sistem yang akan dibangun.

3.1.1. Analisis Masalah

Analisis masalah adalah tahap penjabaran masalah yang ada sebelum aplikasi ini dibangun dan bertujuan untuk membantu pembangunan aplikasi ini. Analisis masalah yang ada meliputi hal-hal sebagai berikut:

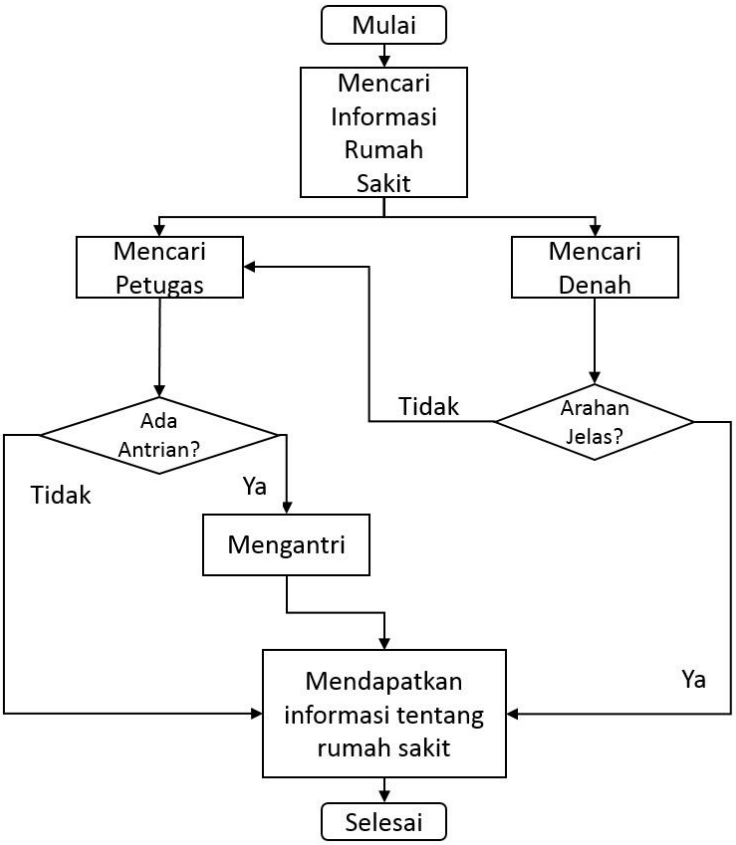
1. Pasien dan pengunjung tidak mengetahui apa saja fasilitas, layanan, prosedur, dan persyaratan administrasi yang ada di rumah sakit.
2. Terbatasnya jumlah media informasi dan petugas informasi, sehinga seringkali tidak dapat melayani pasien atau pengunjung dengan baik.
3. Sulit untuk menyampaikan informasi yang membutuhkan penjelasan secara rinci dan membutuhkan visualisasi dikarenakan keterbatasan media yang digunakan.
4. Media yang digunakan saat ini belum mampu menunjukan posisi ruangan atau fasilitas serta rute yang terpendek untuk menuju ruangan atau fasilitas tersebut.

3.1.2. Analisis Sistem yang Sedang Berjalan

Untuk prosedur pencarian informasi mengenai layanan yang ada di rumah sakit pasien atau pengunjung mencari petugas informasi, kemudian jika bagian

informasi sedang melayani pasien atau pengunjung lain maka harus mengantri, bagian informasi memberikan informasi yang diminta dan akan memberikan catatan atau dokumen pendukung jika diperlukan.

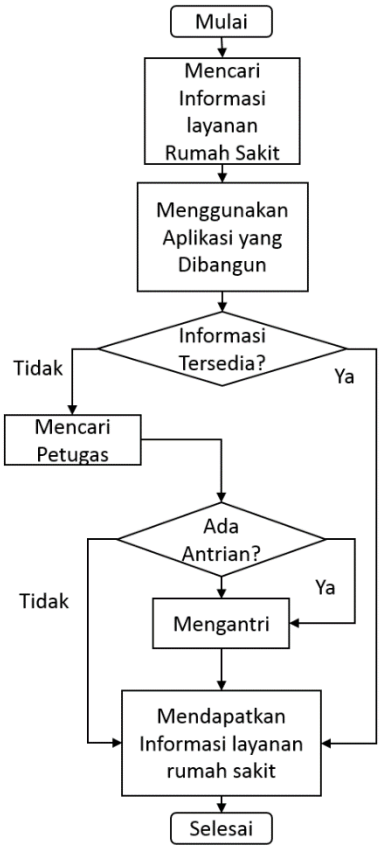
Untuk prosedur pencarian ruangan atau suatu fasilitas terdapat dua cara. Cara yang pertama adalah dengan bertanya pada petugas informasi, jika petugas informasi sedang melayani pasien atau pengunjung lain maka harus mengantri, petugas informasi memberikan informasi yang diminta dan akan memberikan catatan atau dokumen pendukung jika diperlukan. Cara yang kedua adalah dengan menggunakan denah yang ada di lokasi tertentu. Pasien atau pengunjung mencari denah, kemudian mengikuti arahan dari denah dan papan petunjuk arah, hal tersebut dilakukan hingga sampai ke ruangan atau fasilitas yang dituju. Jika terdapat hal yang kurang jelas pada arahan denah atau papan arah maka akan mendatangi petugas informasi, kemudian mengikuti prosedur cara pertama. Prosedur yang sedang berjalan dapat digambarkan ke dalam sebuah alur sebagai berikut:



Gambar 3.1 Prosedur Pada Sistem Yang Berjalan

3.1.3. Analisis Sistem yang Diusulkan

Dengan adanya aplikasi pada sistem yang akan dibangun pasien dan pengunjung memiliki pilihan untuk mendapatkan informasi rumah sakit dengan cara yang lebih cepat, karena tidak perlu melakukan pencarian petugas, denah, dan mengalami antrian. Prosedur pada sistem yang diusulkan dapat digambarkan kedalam alur sebagai berikut.



Gambar 3.2 Prosedur Pada Sistem Yang Diusulkan

3.1.4. Analisis Aplikasi Sejenis

Analisis aplikasi sejenis merupakan analisis yang akan membahas mengenai aplikasi yang menjadi acuan dalam pembangunan sebuah aplikasi. Dalam analisis aplikasi sejenis akan dibahas mengenai berbagai hal yang ada di dalam aplikasi sejenis diantaranya meliputi konten, cara menggunakan serta komponen-komponen apa saja yang ada dalam aplikasi tersebut sehingga dapat menjadi acuan bagi kebutuhan yang akan dibangun.

3.1.4.1. Aplikasi India *Green Reality (AR / VR)*

Aplikasi ini adalah aplikasi interaktif untuk *Real Estate* yang terdiri dari *augmented reality* dan *virtual reality*. *Augmented reality* menggunakan marker dan user dapat membuat custom marker sendiri atau menggunakan printout marker yang telah disediakan *developer*. Sedangkan untuk fitur *Virtual Reality* aplikasi ini menyediakan 3 jenis *view*, yaitu:

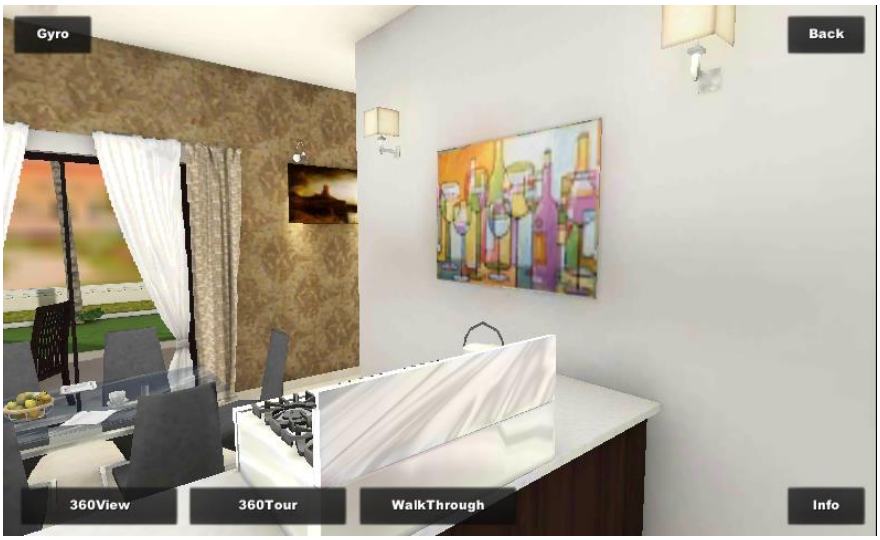
a. 360 *Views*



Gambar 3.3 360 Views

Jenis *view* ini digunakan untuk menampilkan *exterior* bangunan dengan *control pinch*, *drag*, and *touch* untuk pergerakan sudut pandang.

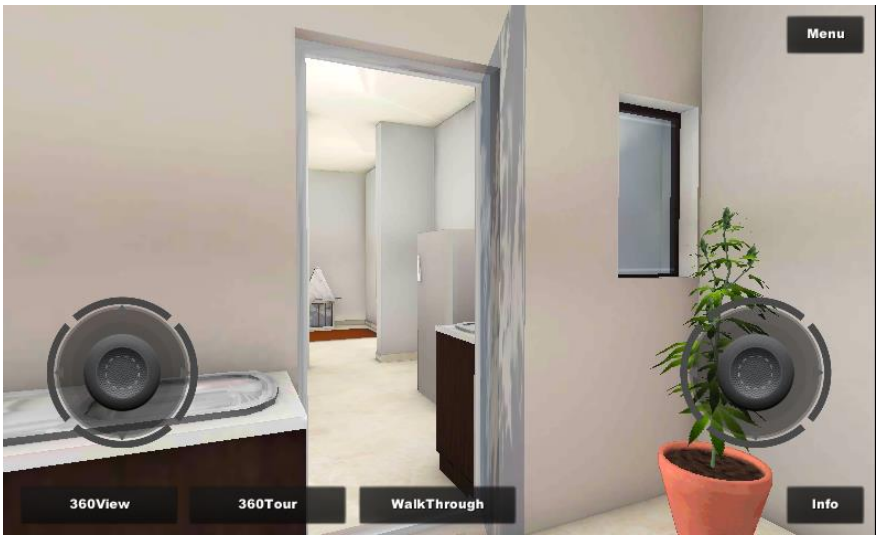
b. 360 *Degree Tours*



Gambar 3.4 360 *Degree Tours*

Jenis *view* yang menampilkan suatu ruangan dengan yang dapat dilihat sekelilingnya, namun jenis view ini tidak memungkinkan pengguna bergerak dari suatu posisi ke posisi lain nya.

c. *Real-Time Walkthrough* dengan *Virtual Joysticks*



Gambar 3.5 Real-Time Walkthrough dengan Virtual Joysticks

Jenis *view* yang memungkinakan pengguna bergerak mengelilingi suatu bangunan dan melihat sekeliling dengan menggunakan *virtual joystick.*

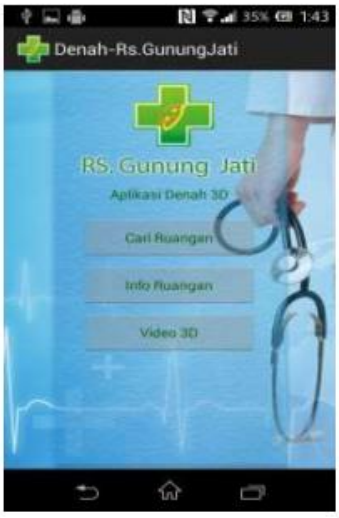
3.1.4.2. Aplikasi Denah 3D RSUD Gunung Jati

Aplikasi Denah 3D RSUD Gunung Jati merupakan suatu aplikasi yang digunakan untuk memberikan informasi tentang rute terpendek RSUD Gunung Jati di kotaCirebon. Grafik yang digunakan dalam aplikasi ini adalah 3 Dimensi(3D). Berikut merupakan gambaran tampilan Denah 3D RSUD Gunung Jati:



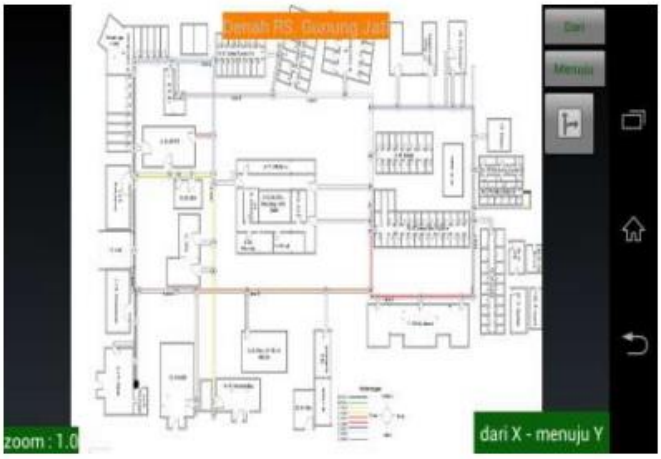
Gambar 3.6 Tampilan Menu Utama

Halaman utama aplikasi yang berisi pilihan untuk memulai aplikasi atau membaca petunjuk penggunaan.



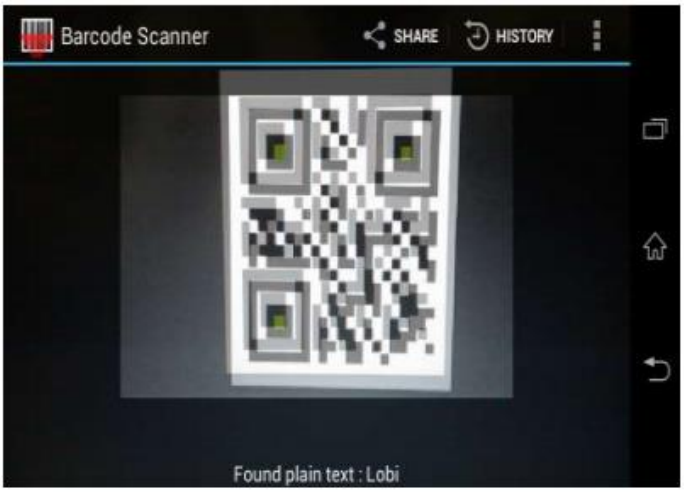
Gambar 3.7 Menu Denah 3D

Terdiri dari pilihan untuk mencari ruangan, info ruangan, dan *video* 3d.



Gambar 3.8 Tampilan Cari Ruang

Tampilan cari tuangan terdiri dari tombol menu dan cari



Gambar 3.9 Tampilan Scan QRCode Titik Keberadaan

Halaman *QRCode reader* untuk menentukan posisi awal.



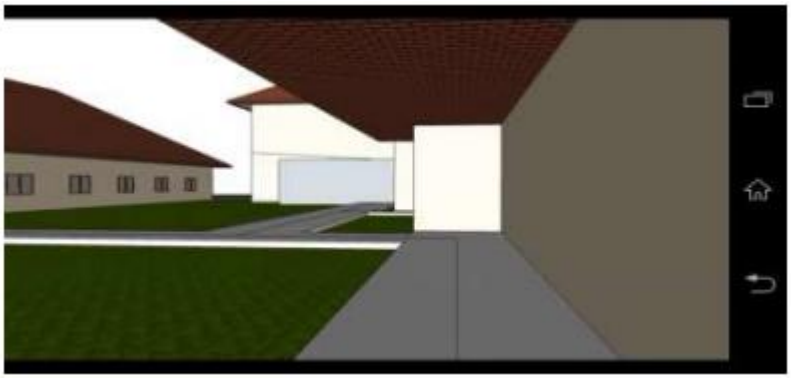
Gambar 3.10 Tampilan Input Ruang Tujuan

Memilih ruangan tujuan yang tersedia di menu.



Gambar 3.11 Tampilan Rute Terpendek

Tampilan rute terpendek berupa denah 2 dimensi, rute ditunjukan dengan garis berwarna.



Gambar 3.12 Tampilan Video Petunjuk Ruang

Tampilan video petunjuk ruang berupa tampilan 3d seperti gambar di atas.



Gambar 3.13 Tampilan Input Info Ruang

Untuk memilih info ruangan terlebih dahuklu dilakukan pemilihan ruangan.



Gambar 3.14 Tampilan Info Ruang

Tampilan info ruang berupa halaman keterangan seperti pada gambar di atas.



Gambar 3.15 Tampilan Video 3D RSUD Gunung Jati



Gambar 3.16 Tampilan Petunjuk Penggunaan

Petunjuk penggunaan berupa halaman berisi teks dan gambar seperti gambar di atas.

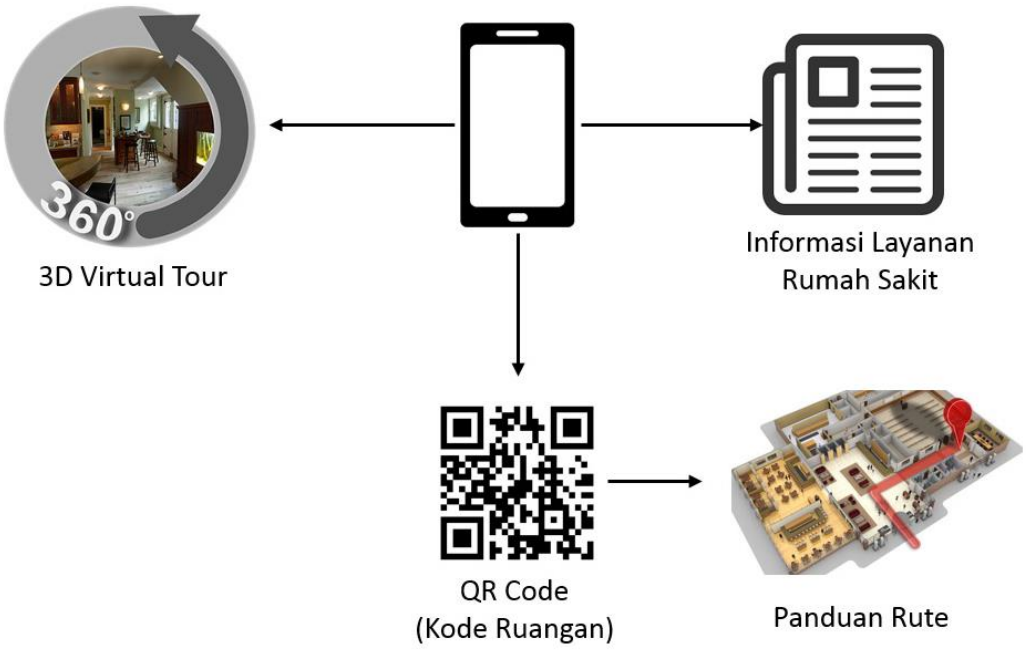
Tabel 3.1 Kesimpulan Analisis Aplikasi Sejenis

No	Nama Aplikasi	Basis Aplikasi	Grafis	Metode Pencarian	Ruang Lingkup	Konten
1	India Green Reality (AR / VR)	Mobile	3D	-	Real Estate India Green	Terdapat fitur virtual tour dengan berbagai jenis view
2	Denah 3D RSUD Gunung Jati	Mobile	3D	Algoritma Floyd Warsahall	Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon	Terdapat pilihan untuk mengetahui rute terpendek dan menampilkan petunjuk denah berupa video 3d
3	Panduan Layanan Pengunjung Berbasis Virtual Reality Di Rumah Sakit Mata Cicendo Bandung	Mobile	3D	Algoritma Dijkstra	Rumah Sakit Mata Cicendo Bandung	Menyediakan informasi layanan rumah sakit, *virtual tour* untuk memberikan visualisasi bangunan serta fasilitas rumah sakit dalam berbagai jenis view, dan denah interaktif untuk menunjukan rute terpendek menuju suatu tempat.

Dari tabel di atas dapat disimpulkan bahwa aplikasi yang dibangun menggabungkan konsep dan fitur - fitur dari aplikasi sejenis. Aplikasi yang dibangun mengambil konsep *virtual tour* yang digunakan aplikasi India Green Reality (AR / VR) dan mengambil konsep pencarian rute terpendek dari aplikasi Denah 3D RSUD Gunung Jati. Dan menambahkan fitur informasi layanan yang disediakan rumah sakit seperti dafatar laboratorium yang ada, alat diagnostik, kelengkapan fasilitas penunjang, serta informasi yang sering ditanyakan pasien atau pengunjung.

3.1.5. Analisis Arsitektur Sistem

Aplikasi yang akan dibangun merupakan aplikasi berbasis android yang memiliki 3 fungsional utama yaitu, virtual tour rumah sakit, panduan rute menuju ruangan tertentu, dan informasi layanan rumah sakit.



Gambar 3.17 Arsitektur Sistem

Aplikasi ini tidak menggunakan database, data seperti informasi layanan dan data *QR Code* ruangan bersifat statis. Kode ruangan hanya berada pada *script*. Dalam mencari panduan rute menuju suatu ruangan pengguna diharuskan menentukan posisi pengguna saat ini (titik awal) dan titik tujuan (titik akhir). Dalam menentukan posisi pengguna saat ini pengguna dapat melakukan scan *QR Code*

yang terdapat di bagian depan ruangan yang ada di rumah sakit. Jika proses scan *QR Code* tidak dimungkinkan pengguna tetap dapat menentukan posisi awal dengan memilih ruangan dalam bentuk *Combo Box*. Dalam menentukan rute terpendek menuju suatu ruangan tersebut digunakan algoritma Dijkstra, setelah hasil perhitungan selesai aplikasi akan menampilkan panduan rute dalam bentuk 3D dengan sudut pandang atas *(floor plan view)* dan menampilkan arahan atau rute dalam bentuk warna penunjuk seperti pada gambar di atas.

Untuk melakukan *virtual tour* pengguna menentukan kategori dan ruangan yang akan dipilih sebagai area *tour*. Sedangkan untuk melihat informasi layanan rumah sakit pengguna menentukan kategori dan konten yang diinginkan.

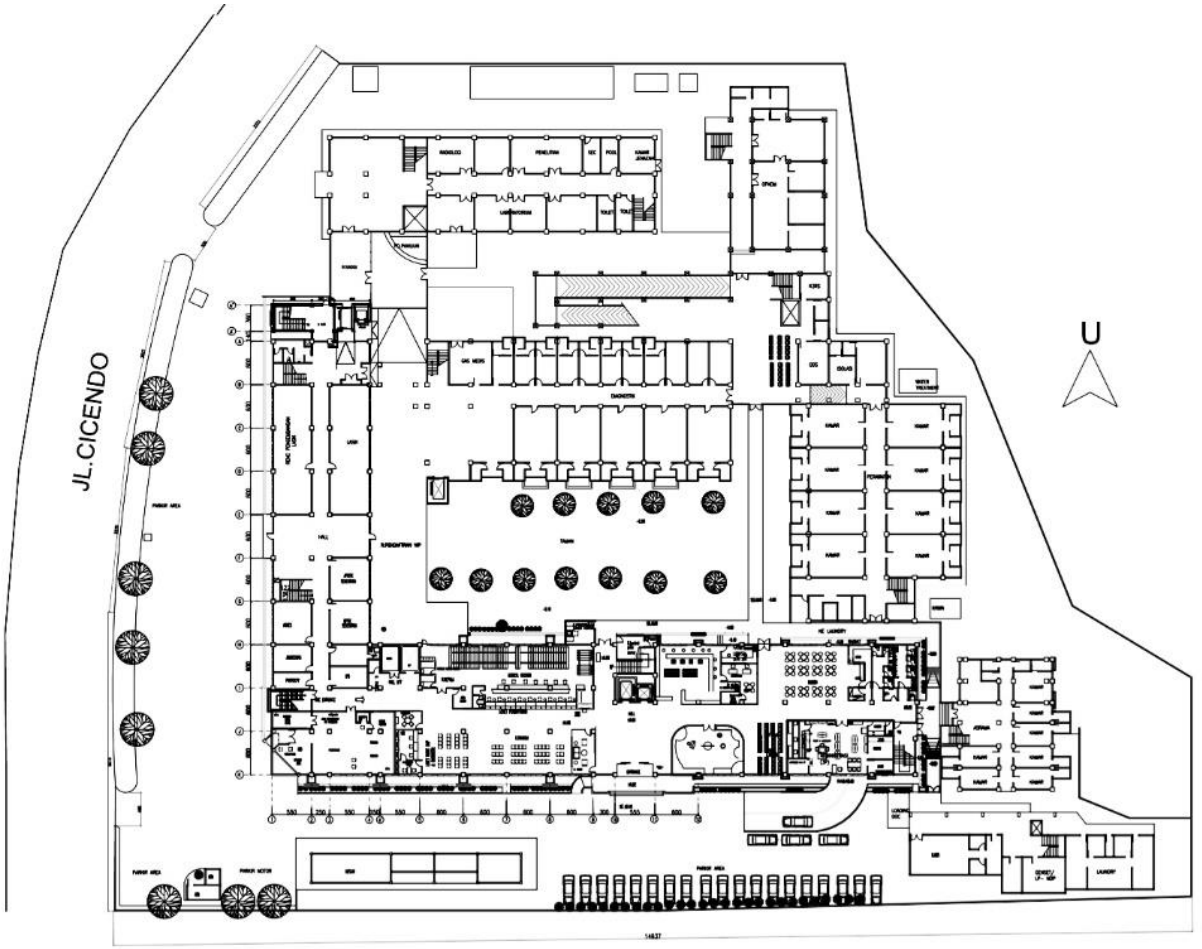
Untuk data *virtual tour*, informasi layanan rumah sakit, dan panduan rute ini bersifat statis sehingga tidak dibutuhkan penggunaan *database*, semua hanya menggunakan penyimpanan lokal dan tidak menggunakan koneksi internet saat penggunaannya.

3.1.5.1. Analisis *Virtual Reality*

Virtual reality yang akan dibangun adalah *Architectural Virtual Tour* berbasis android, *virtual tour* ini menggunakan model – model 3D yang dibentuk berdasarkan keadaan sesungguhnya di lapangan. Dalam pembangunan *Virtual Tour* dilakukan beberapa tahapan yaitu sebagai berikut:

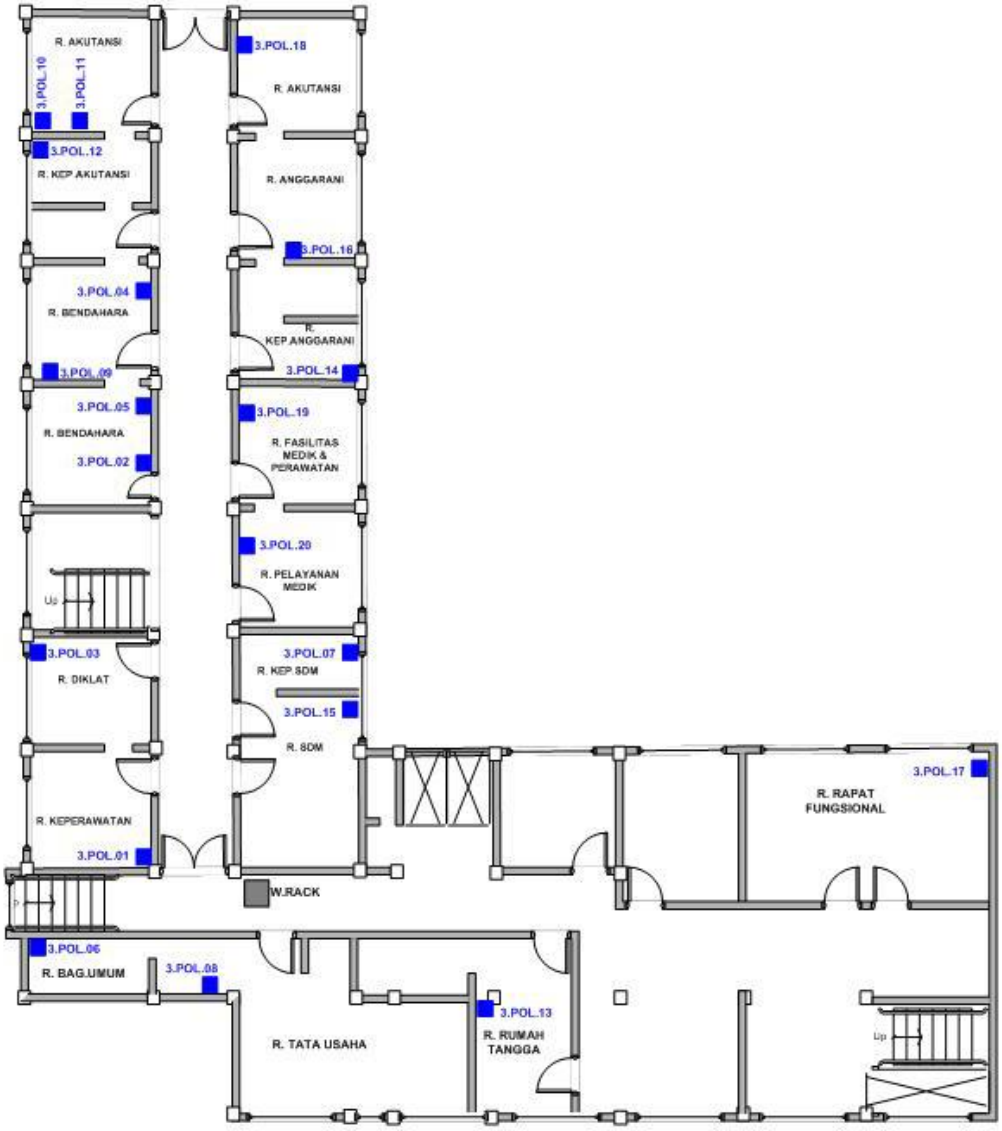
1. Pengumpulan data

Dalam tahap ini dilakukan pengumpulan data yang berupa denah rumah sakit (*floor plan*), foto bangunan dan fasilitas rumah sakit, serta pengambilan video keseluruhan rumah sakit untuk kemudian digunakan sebagai referensi pada proses *3D modelling.*



Gambar 3.18 Denah Umum Rumah Sakit Mata Cicendo bandung

Denah umum ini akan dijadikan referensi pemodelan 3D secara keseluruhan, setelah poemodelan pergedung atau perbagian selesai dilakukan.



Gambar 3.19 Denah Politeknik Lt.3 RSM Cicendo Bandung

Denah tiap bagian gedung rumah sakit seperti pada gambar di atas digunakan sebagai referensi pembangunan model yaitu tata letak, dinding, pintu, serta elemen lainnya.



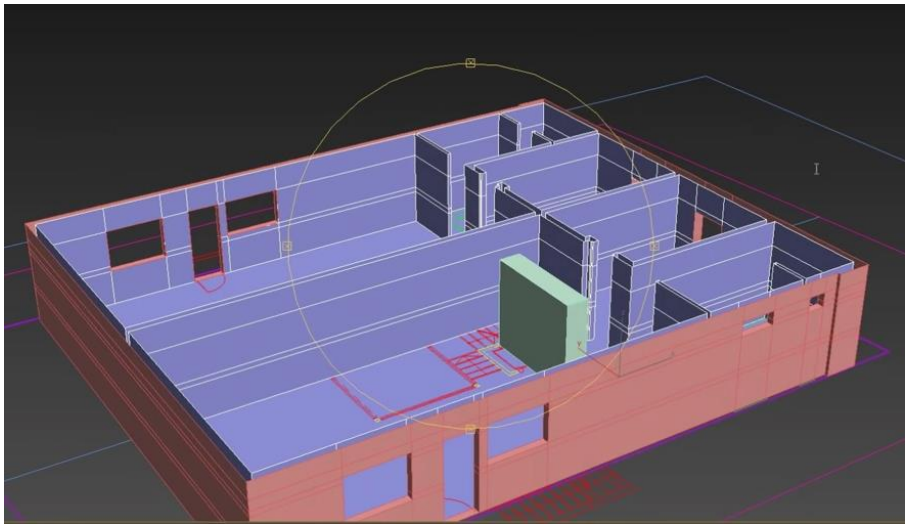
Gambar 3.20 Tampilan Depan Gedung Rumah Sakit

2. Pemodelan (*3D Modelling*)

Pada tahapan pemodelan ini data – data yang telah didapatkan dijadikan referensi pembuatan model 3D, pembuatan model menggunakan *tools* Autodesk 3d's Max Design 2015. Tahapan ini akan menghasilkan model – model 3D yang berupa gedung – gedung dan properti yang masih terpisah. Adapun langkah – langkah pemodelan 3d adalah sebagai berikut:

a) Modeling and Compositing Object

Pada tahap ini model dibuat berdasarkan *object* sesungguhnya di dunia nyata, dalam hal ini yang digunakan sebagai acuan denah, *floormap view*, dan foto objek. Berikut adalah contoh hasil *output* dari tahapan ini.



Gambar 3.21 Contoh *Object*

b) Texturing Object

Pada tahap ini dilakukan pemberian *texture* pada *object* seperti texture dinding, lantai, atap, kursi, dan properti lainnya. Berikut adalah contoh hasil *output* pada tahap ini.



Gambar 3.22 Contoh hasil T*exturing*

c) Export Object

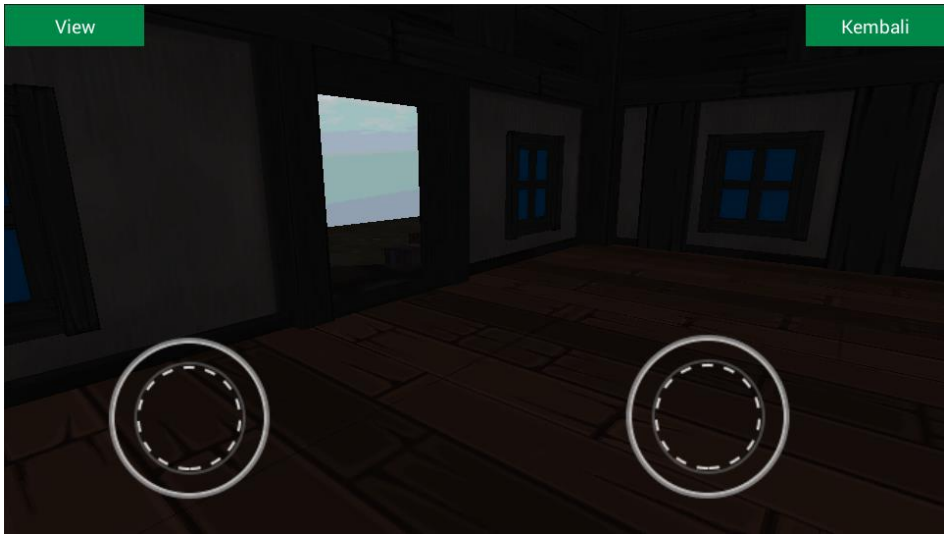
Pada tahap ini dilakukan *export object* ke dalam bentuk file .FBX. File ini dapat dimanipulasi lebih lanjut seperti penggabungan dan penyusunan objek, lalu dilakukan penambahan pencahayaan pada tahap berikutnya menggunakan *Lightmap* yang terdapat pada *tools* Unity 3D.

3. Penggabungan dan Penyusunan *Object*

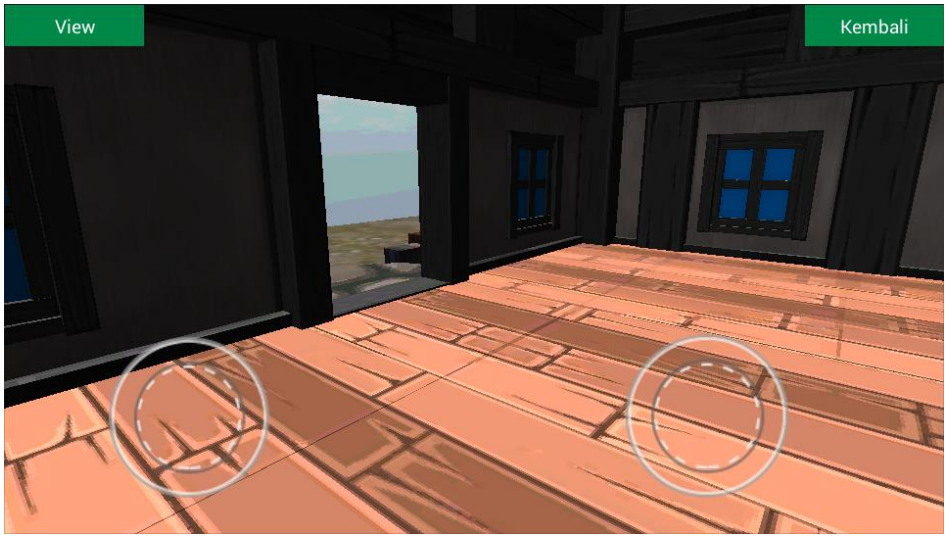
Langkah berikutnya adalah menggabungkan *object* 3D yang telah dibuat. Proses ini menggunakan *tools* Unity 3D, hasil *export* berupa file .FBX di import ke dalam Unity 3D, pada tahap ini *object – object* disusun sedemikian rupa sehingga menyerupai kondisi sesungguhnya di dunia nyata.

4. *Lightmapping*

Pada tahap ini dilakukan pemberian cahaya pada *scene*. Cahaya atau light merupakan bagian penting dari sebuah scene, cahaya menentukan warna dan suana lingkungan 3D. Dalam Unity 3D terdapat beberapa jenis light yang dapat digunakan yaitu *Point Lights, Spot Lights, Directional Lights, Area Lights*. Semua jenis *light* digunakan untuk membuat suasana serealistik mungkin. Berikut contoh *scene* yang belum diberi pencahayaan dan yang sudah diberi pencahayaan, jenis *lighting* yang digunakan adalah *Directional Light*.

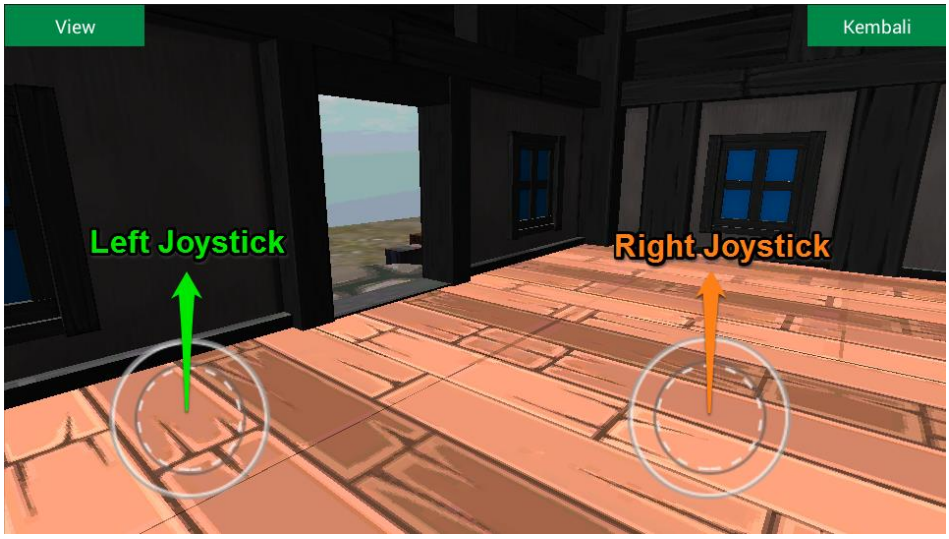


Gambar 3.23 Sebelum *Lightmapping*



Gambar 3.24 Sesudah *Lightmapping*

5. Pemberian C*ontroller*



Gambar 3.25 *Mobile Controller Rig*

Setelah *Scene* terbentuk, diperlukan sebuah controller untuk mengatur pergerakan *camera*, dalam hal ini adalah pengunjung dan pasien yang melakukan *virtual tour*. *Controller* berupa dua *virtual joystick* yang menyerupai *controller* pada *game console* Playstation.

a. Left Joystick

Berfungsi mengatur perpindahan posisi camera, dalam ini berperan sebagai pengatur jalan kaki pengunjung atau pasien yang melakukan

virtual tour. *Analog* ini mengatur dua *axis* yaitu *axis* x (horizontal) dan *axis* y (vertikal).



Gambar 3.26 Axis X dan Axis Y Pada *Left Joystick*

Berikut adalah *method* yang memberikan nilai posisi *camera* yang sudah disediakan oleh unity3D.

```
public  void OnDrag(PointerEventData data) {
        Vector3 newPos = Vector3.zero;
        if (useX) {
           int delta = (int) (data.position.x - startPos.x);
           newPos.x = delta;
        }
        if (useY)
        {
           int delta = (int)(data.position.y - startPos.y);
           newPos.y = delta;
        }
        transform.position = Vector3.ClampMagnitude( new
Vector3(newPos.x , newPos.y , newPos.z), MovementRange)+
startPos;
        UpdateVirtualAxes (transform.position); }
```

b. *Right Joystick*

Berfungsi mengatur *angle camera* dalam hal ini mengatur tatapan mata ke sekeliling lokasi atau *virtual tour scene.* Untuk membaut sudut pandang lebih natural dilakukan pembatasan sudut, adapun *method*

yang digunakan untuk membatasi sudut pandang agar sudut pandang lebih natural adalah sebagai berikut:

```
Vector2 mouseInput = new
Vector2(CrossPlatformInputManager.GetAxisRaw("HorizontalLoo
k"),

CrossPlatformInputManager.GetAxisRaw("VerticalLook"));

float camX = _camera.transform.localEulerAngles.x;

if((camX > 280 && camX <= 360) || (camX >= 0 && camX < 80)
|| (camX >= 80 && camX < 180 && mouseInput.y > 0) ||
  (camX > 180 && camX <= 280 && mouseInput.y < 0))
{
_camera.transform.localEulerAngles += new Vector3(-
mouseInput.y, _camera.transform.localEulerAngles.y,
_camera.transform.localEulerAngles.z);
}
transform.localEulerAngles += new Vector3(0, mouseInput.x, 0);
```

6. Pemberian *Collider*

Unity 3D memiliki berbagai jenis *collider* diantaranya *BoxCollider, SphereCollider, CapsuleCollider, MeshCollider, PhysicMaterial, Rigidbody.* Collider tersebut dapat digunakan sebagai *trigger* untuk menampilkan informasi mengenai suatu ruangan atau properti pada lingkungan *virtual tour.* Dalam hal ini jenis *collider* yang digunakan hanya jenis *BoxCollider* dan *CapsuleCollider. Collider* tersebut memiliki radius atau area yang dapat diatur sesuai kebutuhan. Berikut adalah contoh penggunaan *Box Collider* pada sebuah area pintu masuk sebagai *trigger* informasi atau keterangan mengenai ruangan atau area tersebut, informasi akan muncul setelah pengguna memasuki radius *collider* tersebut dan akan hilang setelah keluar dari radius *collider.*



Gambar 3.27 *Collider* Sebagai *Trigger*

7. Penanaman Informasi atau keterangan

Pada tahapan ini dilakukan penanaman informasi atau keterangan mengenai object – object pada virtual tour. Informasi yang ditampilkan pada proerti atau ruangan dapat berupa hal –hal berikut:

a. Nama gedung
b. No lantai
c. Nama Ruangan
d. Kegunaan ruangan (secara umum)
e. Jam operasional
f. Keterangan mengenai properti tertentu

Berikut adalah contoh tampilan ketika pengguna menghampiri suatu properti yang memiliki keterangan atau informasi.



Gambar 3.28 keterangan pada objek atau lokasi

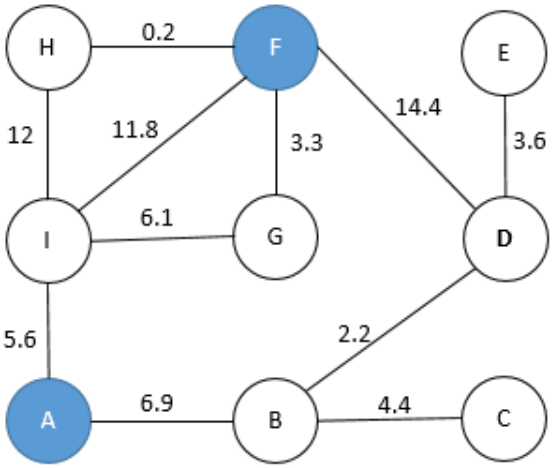
3.1.5.2. Analisis Algoritma Dijkstra

Berdasarkan hasil penelitian [15] mengenai perbandingan penghitung jalur terpendek yaitu algoritma dijkstra, algoritma floyd, dan algoritma two queues. Algoritma dijkstra lebih baik dari pada dua algoritma penghitungan jalur terpendek yang lainnya. Karena algoritma dijkstra mempunyai kecepatan eksekusi lebih baik, yaitu kurang dari 1 detik. Maka dari itu aplikasi ini menggunakan Algoritma Dijkstra dalam proses pencarian ruangan. Dalam hal ini yang menjadi node adalah ruangan – ruangan dan fasilitas yang ada di rumah sakit. Bobot ditentukan berdasarkan jarak objek (ruangan dan fasilitas pada model 3D yang ada pada aplikasi.

Tabel 3.2 Keterangan Node

Node	Nama Ruangan
A	Area Parkir
B	Informasi
C	Depo Farmasi Lt.1
D	Pendaftaran Reguler
E	IGD
F	Kasir Rawat Jalan dan Rawat Inap
G	Optik Lt.1
H	Kantin
I	Museum
J	Lobi Poliklinik
K	Depo Farmasi Lt.2
L	Penyakit Dalam
M	Katarak dan Bedah Refraktif
N	Elektrodiagnostik dan Terapi
O	Rekonstruksi, Okuloplasti dan Onkologi
P	Optik Lt.2
Q	Laboratorium
R	Konsultasi Gizi
S	Konsultasi Pasien
T	Infeksi - Imunologi
U	Refraksi
V	Glaukoma
W	Depo Farmasi Lt.3
X	Vitreoretina
Y	Neuro Oftalmologi
Z	Konsultasi Pasien
A2	Optik Lt.3
B2	Low Vision
C2	Lensa Kontak
D2	Optik Lt.4
E2	Ruang Rapat
F2	Depo Farmasi
G2	Strabismus dan Pediatrik Oftalmologi

Misalkan pengguna akan mencari rute dari ruangan A ke ruangan F, berikut adalah contoh penerapan algoritma Dijkstra untuk pencarian rute terpendek:



Gambar 30 Graph Ruangan di Lantai 1

Langkah – langkah untuk menentukan jarak terpendek dari A ke F dengan menggunakan algoritma Dijkstra adalah sebagai berikut:

1. Pada awalnya status dari node yang belum terpilih diinisialisasikan dengan "0" dan yang sudah terpilih diinisialisasikan dengan "1" dimulai dari node A.
2. Tentukan bobot dari node yang langsung berhubungan dengan node sumber yaitu node A, seperti node A ke node B = 6.9, node A ke node I = 5.6, dan untuk node C, D, E, F, G, H diinisialisasikan dengan "-" karena tidak ada lintasan yang menghubungkan secara langsung dengan node A.
3. Karena jarak di hitung dari node A, sehingga node A disebut sebagai *Predecessor* (node sumber).

Tabel 3.3 Hasil Iterasi Ke-1

Node	A	B	C	D	E	F	G	H	I
Status	1	0	0	0	0	0	0	0	0
Bobot	-	6.9	-	-	-	-	-	-	5.6
Predecessor	A	A	-	-	-	-	-	-	A

Node terpilih dari iterasi ke-1:

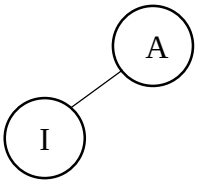


Gambar 3.29 Node Terpilih Pada Iterasi Ke-1

Dari Tabel 2 pilih node yang memiliki bobot yang paling kecil dan statusnya masih “0” yaitu node I. Untuk itu status node I menjadi “1” dan *predecessor*-nya tetap A, dan node lain *predecessor*-nya masih sama. Karena node I sudah terpilih, selanjutnya diperoleh node F dengan bobot 17.4, node G dengan bobot 11.7, node H dengan bobot 17.6. Sedangkan *predecessor* F, G, dan H adalah I.

Tabel 3.4 Hasil Iterasi Ke-2

Node	A	B	C	D	E	F	G	H	I
Status	1	0	0	0	0	0	0	0	1
Bobot	-	6.9	-	-	-	17.4	11.7	17.6	5.6
Predecessor	A	A	-	-	-	I	I	I	A

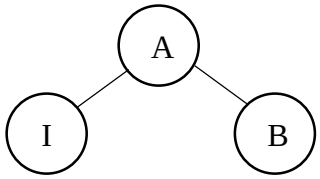


Gambar 3.30 Node Terpilih Pada Iterasi Ke-2

Dari Tabel 3 di dapatkan bahwa node B memiliki bobot yang paling kecil, sehingga statusnya akan berubah menjadi “1” dan *predecessor*-nya masih tetap A. Selanjutnya diperoleh node C dengan bobot 11.3 dan node D dengan bobot 9.1.

Tabel 3.5 Hasil Iterasi Ke-3

Node	A	B	C	D	E	F	G	H	I
Status	1	1	0	0	0	0	0	0	1
Bobot	-	6.9	11.3	9.1	-	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	-	I	I	I	A

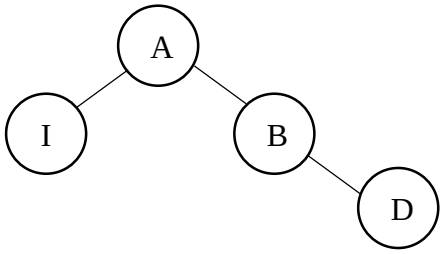


Gambar 3.31 Node Terpilih Pada Iterasi Ke-3

Dari tabel 4 didapat bahwa node yang memiliki bobot paling kecil dan statusnya masih ”0” adalah node D. Statusn node D akan berubah menjadi “1”. Selanjutnya didapat node E dengan bobot 12.7, sedangkan untuk node F tidak mengalami perubahan karena bobot yang baru 23.5 lebih besar dari bobot yang lama yaitu 17.4.

Tabel 3.6 Hasil Iterasi Ke-4

Node	A	B	C	D	E	F	G	H	I
Status	1	1	0	1	0	0	0	0	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A

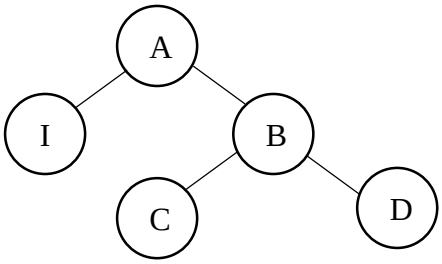


Gambar 3.32 Node Terpilih Pada Iterasi Ke-4

Dari Tabel 5 didapat bahwa node C memiliki bobot paling kecil sehingga statusnya akan berubah menjadi “1”.

Tabel 3.7 Hasil Iterasi Ke-5

Node	A	B	C	D	E	F	G	H	I
Status	1	1	1	1	0	0	0	0	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A

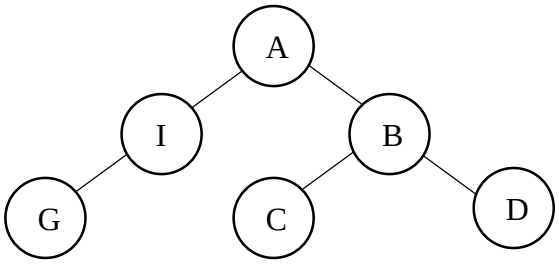


Gambar 3.33 Node Terpilih Pada Iterasi Ke-5

Dari Tabel 6 didapat bahwa node G memiliki bobot paling kecil sehingga statusnya akan berubah menjadi “1”.

Tabel 3.8 Hasil Iterasi Ke-6

Node	A	B	C	D	E	F	G	H	I
Status	1	1	1	1	0	0	1	0	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A

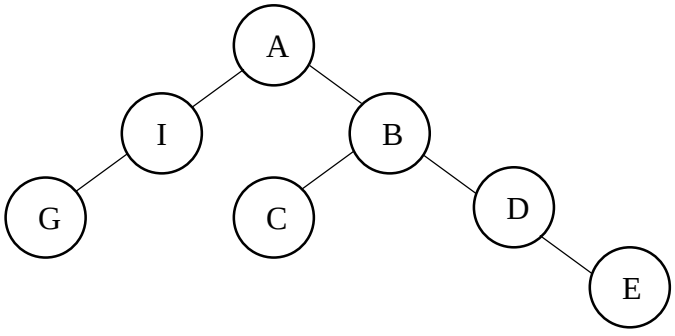


Gambar 3.34 Node Terpilih Pada Iterasi Ke-6

Dari Tabel 7 didapat bahwa node E memiliki bobot paling kecil sehingga statusnya akan berubah menjadi “1”.

Tabel 3.9 Hasil Iterasi Ke-7

Node	A	B	C	D	E	F	G	H	I
Status	1	1	1	1	1	0	1	0	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A



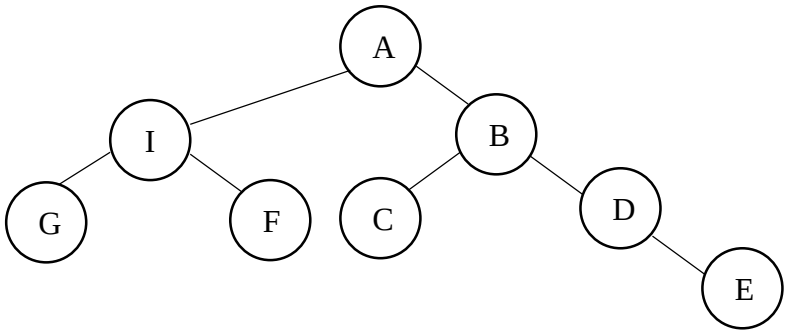
Gambar 3.35 Node Terpilih Pada Iterasi Ke-7

Dari Tabel 8 didapat bahwa node F memiliki bobot paling kecil sehingga statusnya akan berubah menjadi “1”. sedangkan untuk node H

tidak mengalami perubahan karena bobot yang baru 23.7 lebih besar dari bobot yang lama yaitu 17.6.

Tabel 3.10 Hasil Iterasi Ke-8

Node	A	B	C	D	E	F	G	H	I
Status	1	1	1	1	1	1	1	0	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A

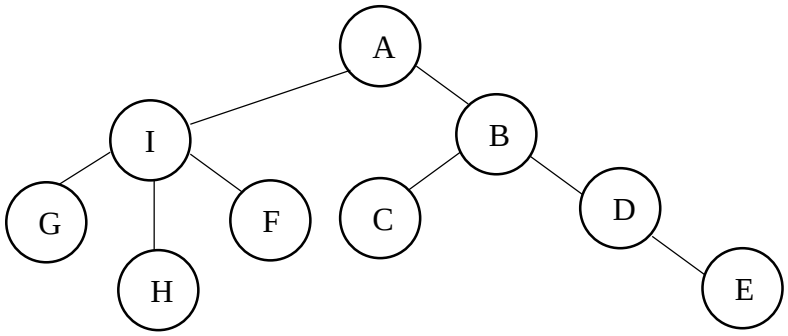


Gambar 3.36 Node Terpilih Pada Iterasi Ke-8

Dari Tabel 8 didapat bahwa node H memiliki bobot paling kecil sehingga statusnya akan berubah menjadi “1”.

Tabel 3.11 Hasil Iterasi Ke-9

Node	A	B	C	D	E	F	G	H	I
Status	1	1	1	1	1	1	1	1	1
Bobot	-	6.9	11.3	9.1	12.7	17.4	11.7	17.6	5.6
Predecessor	A	A	B	B	D	I	I	I	A



Gambar 3.37 Node Terpilih Pada Iterasi Ke-9

Iterasi akan berhenti karena semua node sudah terpilih. Sehingga akan menghasilkan rute terpendek dari node A ke setiap node yang ada. Untuk melihat jalur mana yang terpilih dapat ditelusuri dari *predecessor*-nya, sehingga akan didapat:

Tabel 3.12 Iterasi Keseluruhan

Sumber → Tujuan	**Rute**	Bobot
A → B	A-B	6.9
A → C	A-B-C	11.3
A → D	A-B-D	9.1
A → E	A-B-D-E	12.7
A → F	A-I-F	17.4
A → G	A-I-G	11.7
A → H	A-I-H	17.6
A → I	A-I	5.6

Dalam contoh kasus pencarian rute terpendek dari ruangan A menuju ruangan F maka solusinya adalah rute A-I-F.

3.1.5.3. Analisis *QR Code*

QR Code digunakan sebagai salah satu cara untuk mendapatkan posisi awal dalam fungsional pencarian rute terpendek menuju suatu ruangan. Pada sistem yang akan dibangun *QR Code* digunakan untuk menyimpan data nama ruangan. Berikut adalah tabel data dan panjang karakter yang digunakan:

Tabel 3.13 Panjang Karakter Nama Ruangan

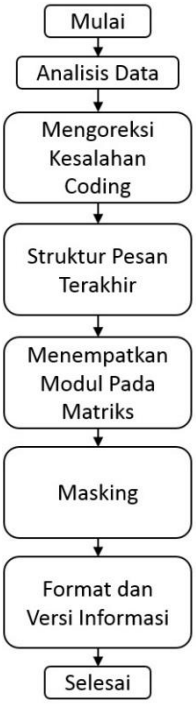
Atribut Data Ruangan	**Panjang Karakter Maksimal**
Nama Ruangan	52

3.1.5.3.1. Proses Encode

Proses pembuatan *QR Code* dilakukan diluar sistem yang dibangun, proses *encoding* menggunakan aplikasi *Encode/Decode* berbasis *desktop*. Berikut adalah tampilan aplikasi pada saat proses *encode.*

Untuk membuat sebuah *QR Code*, ada beberapa proses yang harus dilakukan. Karena jenis masukan untuk membuat *QR Code* beragam, ada beberapa hal yang harus diperhatikan seperti konversi dari masukan tersebut ke dalam biner

sehingga dapat diproses untuk dijadikan titik hitam dan titik putih pada *QR Code* yang akan dibuat. Tidak hanya informasi yang telah dikonversi, *QR Code* juga harus memiliki pola-pola standar yang dapat memudahkan proses pembacaan. Proses-proses secara umum yang harus dilalui untuk membangkitkan *QR Code* dari sebuah teks dapat dilihat pada diagram alir berikut.



Gambar 3.38 Diagram alir pembangkitan *QR Code*

Data yang digunakan berupa string, jenis encoding mode yang digunakan adalah *Byte* dengan *Correction Level* = M yang mampu Recovers 15% of data. Adapun versi *QR Code* yang digunakan adalah versi 1, kombinasi ini dipilih karena dengan *correction level* M, versi *QR Code* 3 dan *mode encoding byte* panjang *text* atau karakter yang dapat di *encode* adalah 14 dimana panjang *string* atau text yang akan di *encode* hanya 10 karakter, ini merupakan pilihan yang paling optimal karena paling sedikit menyisakan ruang yang tidak terpakai. Berikut adalah tabel detail *QR Code* yang digunakan.

Tabel 3.14 Spesifikasi *QR Code*

Data	String
Encoding	Byte
Version	3
Correction Level	M

Berikut adalah contoh hasil *encode* dari kode ruangan dengan nama ruangan terpanjang yaitu Strabismus Pediatrik Oftalmologi.



Gambar 3.39 *QR Code* Kode Ruangan Strabismus Pediatrik Oftalmologi

QR Code tersebut akan diguanakan sebagai alat bantu untuk menentukan posisi awal pada saat pencarian rute terpendek menuju suatu ruangan atau fasilitas di rumah sakit. *QR Code* tersebut akan ditempel pada bagian depan ruangan baik pada pintu atau properti tertentu.

3.1.5.3.2. Proses Decode

Berikut adalah contoh proses *decode* dari sebuah nama ruangan RRIBGNVC01.



Pengguna melakukan *scan QR Code* yang terdapat pada bagaian depan suatu ruangan atau properti, setelah *QR Code* berhasil terbaca dengan baik

pengguna akan mendapatkan kode ruangan yang kemudian digunakan untuk melanjutkan proses mencari rute terpendek menuju suatu ruangan, dalam contoh diatas kode yang didapat adalah RRIBGNVC01, sesuai dengan aturan pengkodean kode ruangan maka arti dari kode tersebut adalah ruang rawat inap dengan nama ruangan Bougenville terletak pada gedung C dan berada di lantai 1.

3.1.6. Analisis Kebutuhan Non-Fungsional

3.1.6.1. Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna aplikasi. Pengguna aplikasi ini sendiri yaitu para pengunjung Rumah Sakit Mata Cicendo Bandung yang menggunakan *Smartphone* atau *Computer Tablet* dengan *platform* android.

3.1.6.2. Analisis Perangkat Keras

Spesifikasi minimal perangkat keras yang dapat digunakan untuk menjalankan aplikasi adalah sebagai berikut :

Tabel 3.15 Spesifikasi Minimal Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Processor*	Qualcomm MSM8230 Snapdragon Dual-core 1GHzGPU
2	Memori	RAM: 1 GB
3	OS	Android OS, v4.1 (Jelly Bean)
4	Resolusi Layar	800x480

3.1.6.3. Analisis Perangkat Lunak

Perangkat lunak yang dibutuhkan untuk proses pembangunan aplikasi Panduan Layanan Pengunjung Berbasis Virtual Reality Di Rumah Sakit Mata Cicendo Bandung adalah sebagai berikut:

Tabel 3.16 Spesifikasi Perangkat Lunak

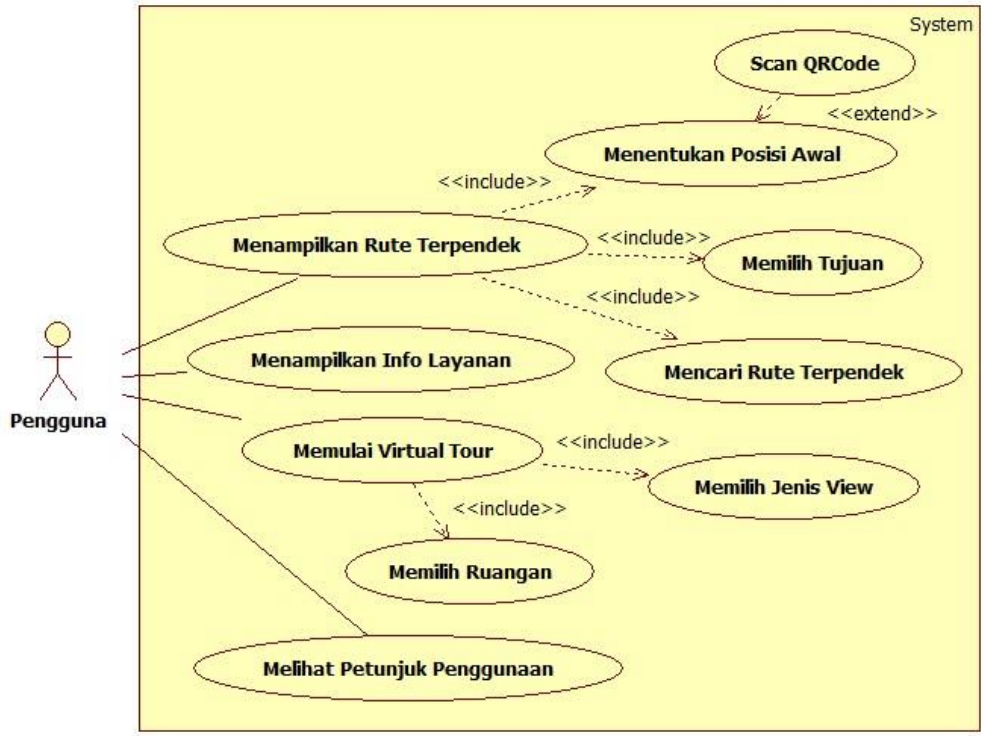
No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi Windows	Versi 8.1
2	IDE Mono Develop	Versi 4.01
3	Game Engine Unity	Versi 4.6
4	Android SDK	Android 4.22 (API 17)
5	Autodesk 3DS Max	2015

3.1.7. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Diagram yang digunakan yaitu *Use Case Diagram, Class Diagram, Sequence Diagram, dan Activity Diagram.*

3.1.7.1. *Use Case Diagram*

Diagram *Use Case* merupakan pemodelan untuk kelakuan (*behaviour*) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Berikut ini adalah perancangan proses-proses yang terdapat pada aplikasi Panduan Layanan Pengunjung Berbasis *Virtual Reality* di Rumah Sakit Mata Cicendo Bandung:



Gambar 3.40 *Use Case Diagram*

3.1.7.1.1. Definisi Aktor

Definisi aktor untuk menjelaskan aktor yang terdapat pada *Use Case Diagram*. Definisi aktor dijelaskan pada tabel berikut:

Tabel 3.17 Definisi Aktor

No	Aktor	Deskripsi
1	Pengguna	Pasien atau pengunjung yang menggunakan aplikasi

3.1.7.1.2. Definisi *Use Case*

Definisi *Use Case* menjelaskan fungsi *use case* yang terdapat pada *Use Case Diagram*. Definisi Use Case dijelaskan pada tabel berikut:

Tabel 3.18 Definisi *Use Case*

No	Use Case	Deskripsi
1	Menampilkan Rute Terpendek	Berfungsi untuk menampilkan rute terpendek menuju suatu ruangan
2	Menampilkan Info Layanan	Berfungsi untuk menampilkan informasi seputar layanan yang disediakan rumah sakit
3	Memulai *Virtual Tour*	Berfungsi untuk memulai *virtual tour*
4	Melihat Petunjuk Penggunaan	Berfungsi menampilkan petunjuk penggunaan aplikasi
5	Menentukan Posisi Awal	Berfungsi untuk melakukan *Scan QR Code* untuk menentukan titik awal keberadaan (posisi saat ini).
6	*Scan QR Code*	Berfungsi untuk menentukan titik awal keberadaan (posisi saat ini).
7	Memilih Tujuan	Berfungsi memilih ruangan tujuan
8	Mencari Rute Terpendek	Berfungsi mencari rute terpendek (perhitungan)
9	Memilih Jenis *View*	Berfungsi memilih jenis *view camera* untuk *virtual tour*
10	Memilih Ruangan	Berfungsi memilih ruangan yang akan digunakan untuk *virtual tour*

3.1.7.2. *Use Case* Skenario

Dalam Skenario Use Case ini menggambarkan alur penggunaan sistem dimana setiap skenario digambarkan dari sudut pandang aktor, seseorang, atau piranti yang berinteraksi dengan perangkat lunak dalam berbagai cara.

1. Menampilkan Rute Terpendek

Proses ini bertujuan untuk menampilkan hasil pencarian rute terpendek untuk menuju suatu ruangan atau fasilitas.

Tabel 3.19 Skenario Menampilkan Rute Terpendek

Use Case Name	Menampilkan Rute Terpendek	
Goals	Menampilkan rute terpendek menuju suatu ruangan atau fasilitas	
Preconditions	Telah menentukan posisi awal, memilih tujuan, dan mencari rute terpendek	
Successful End Condition	Berhasil menampilkan rute terpendek	
Failed End Condition	Rute tidak muncul	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih menu cari ruangan
	2 **Include: Menentukan Posisi Awal**	Menentukan posisi awal (posisi pengguna saat ini)
	3 **Include: Memilih Tujuan**	Menentukan tujuan (posisi yang akan dituju oleh pengguna)
	4 **Include: Mencari Rute Terpendek**	Mencari rute terpendek (algoritma dijkstra)
	5	Menampilkan rute terpendek

2. Menampilkan Info Layanan

Proses ini bertujuan untuk menampilkan informasi layanan rumah sakit seperti informasi layanan medis, informasi fasilitas, prosedur, dll.

Tabel 3.20 Skenario Menampilkan Info Layanan

Use Case Name	Menampilkan Info Layanan	
Goals	Menampilkan informasi layanan rumah sakit	
Preconditions	-	
Successful End Condition	Berhasil menampilkan informasi layanan	
Failed End Condition	Gagal menampilkan informasi layanan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***

	1	Memilih menu info layanan
	2	Memilih kategori info layanan
	3	Memilih info layanan
	4	Menampilkan info layanan

3. Memulai *Virtual Tour*

Proses ini bertujuan untuk menjalankan atau memulai *virtual tour* seuai ruangan atau area yang telah dipilih.

Tabel 3.21 Skenario *Virtual Tour*

Use Case Name	Memulai *Virtual Tour*	
Goals	Menampilkan *virtual tour*	
Preconditions	-	
Successful End Condition	Berhasil menampilkan *virtual tour*	
Failed End Condition	*Virtual Tour* tidak dapat dilakukan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih menu *virtual tour*
	2 **Include: Memilih ruangan**	Memilih ruangan atau area *virtual tour*
	3 **Include: Memilih Jenis View**	Memilih jenis view
	4	Menampilkan virtual tour

4. Melihat Petunjuk Penggunaan

Proses ini bertujuan untuk menampilkan petunjuk penggunaan aplikasi.

Tabel 3.22 Skenario Melihat Petunjuk Penggunaan

Use Case Name	Melihat Petunjuk Penggunaan
Goals	Menampilkan petunjuk penggunaan aplikasi
Preconditions	-

Successful End Condition	Berhasil menampilkan petunjuk penggunaan	
Failed End Condition	Gagal menampilkan petunjuk penggunaan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih menu petunjuk penggunaan
	2	Menampilkan petunjuk penggunaan

5. Menentukan Posisi Awal

Proses ini bertujuan untuk menentukan posisi awal dalam rangka pencarian rute terpendek menuju suatu ruangan atau fasilitas.

Tabel 3.23 Skenario Menentukan Posisi Awal

Use Case Name	Menentukan Posisi Awal	
Goals	Menentukan posisi awal (posisi pengguna saat ini)	
Preconditions	-	
Successful End Condition	Berhasil menentukan posisi awal	
Failed End Condition	Posisi awal gagal ditentukan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih menu posisi awal
	2	Memilih pilihan posisi awal yang tersedia
	3 **Exclude:** **Scan QR Code**	*Scan QR Code* untuk menentukan posisi awal

6. Scan QR Code

Proses ini bertujuan untuk melakukan pembacaan terhadap *QR Code* yang bertujuan untuk menentukan posisi awal dalam pencarian rute terpendek.

Tabel 3.24 Skenario *Scan QR Code*

Use Case Name	*Scan QR Code*
Goals	Melakukan proses *scan* terhadap *QR Code* untuk menentukan posisi awal

Preconditions	-	
Successful End Condition	Berhasil menentukan posisi awal	
Failed End Condition	Hasil scan tidak dapat memberikan posisi awal	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih *button Scan QR Code*
	2	*Scan QR Code*
Extension	***Steps***	***Branching Action***
	2.1	Hasil scan *QR Code* memberikan *string* yang tidak teradaftar

7. Memilih Tujuan

Proses ini bertujuan untuk menentukan tujuan dalam proses pencarian rute terpendek menuju suatu ruangan atau fasilitas.

Tabel 3.25 Skenario Memilih Tujuan

Use Case Name	Memilih Tujuan	
Goals	Menentukan posisi akhir (tujuan) untuk pencarian rute terpendek	
Preconditions	-	
Successful End Condition	Berhasil menentukan tujuan	
Failed End Condition	Tujuan gagal ditentukan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih menu tujuan
	2	Memilih tujuan yang tersedia

8. Mencari Rute Terpendek

Proses ini bertujuan untuk melakukan pencarian rute (perhitungan) menggunakan algoritma dijkstra.

Tabel 3.26 Skenario Mencari Rute Terpendek

Use Case Name	Mencari Rute Terpendek
Goals	Melakukan perhitungan untuk menentukan rute

Preconditions	Posisi awal dan tujuan sudah ditentukan	
Successful End Condition	Berhasil mendapatkan rute	
Failed End Condition	Rute tidak dapat ditemukan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih *button* Cari Rute
	2	Pencarian rute (perhitungan)

9. Memilih Jenis View

Proses ini bertujuan untuk menentukan jenis *view* untuk *virtual tour*.

Tabel 3.27 Skenario Memilih Jenis *View*

Use Case Name	Memilih Jenis *View*	
Goals	Menentukan jenis *view virtual tour*	
Preconditions	Ruangan atau area *virtual tour* sudah ditentukan	
Successful End Condition	Berhasil menentukan jenis *view*	
Failed End Condition	Jenis *view* belum ditentukan	
Primary Actors	Pengguna	
Main Flow	***Steps***	***Actions***
	1	Memilih *button* jenis *view*

10. Memilih Ruangan

Proses ini bertujuan untuk memilih ruangan atau area yang akan dijadikan sebagai ruangan atau area *virtual tour.*

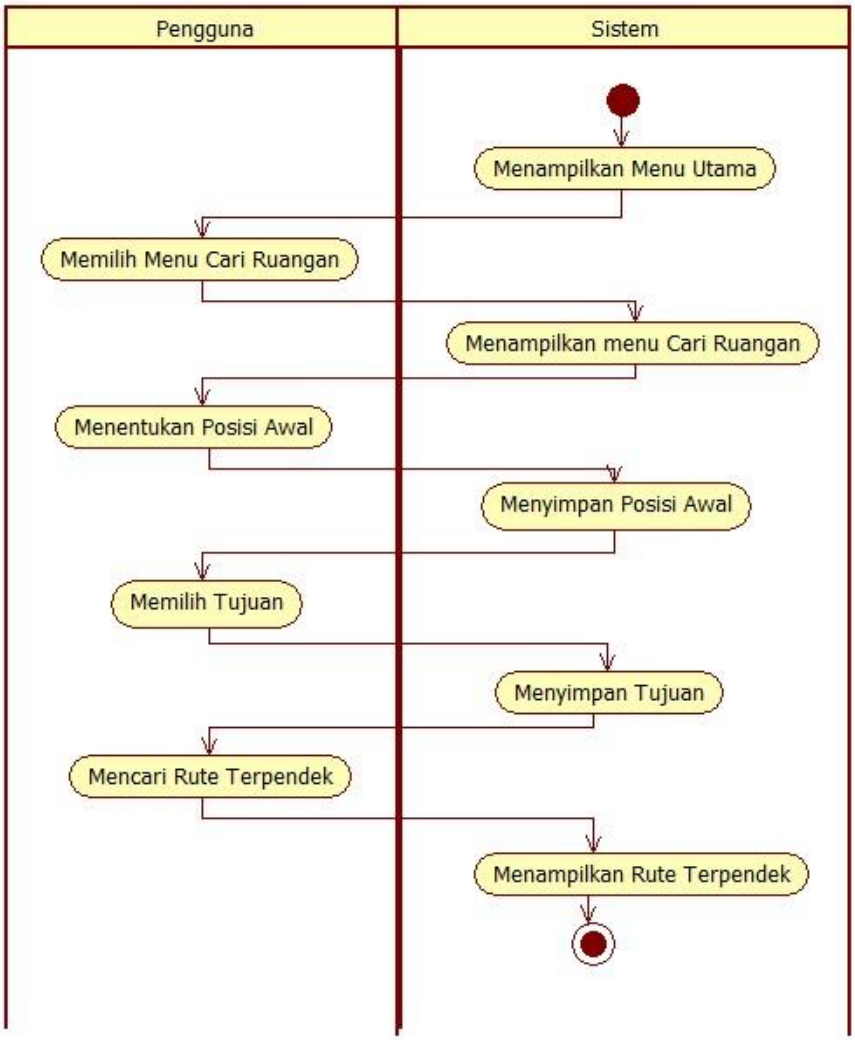
Tabel 3.28 Skenario Memilih Ruangan

Use Case Name	Memilih Ruangan
Goals	Menentukan ruangan atau area untuk *virtual tour*
Preconditions	-
Successful End Condition	Ruangan atau area *virtual tour* sudah ditentukan
Failed End Condition	Gagal menentukan ruangan atau area *virtual tour*
Primary Actors	Pengguna

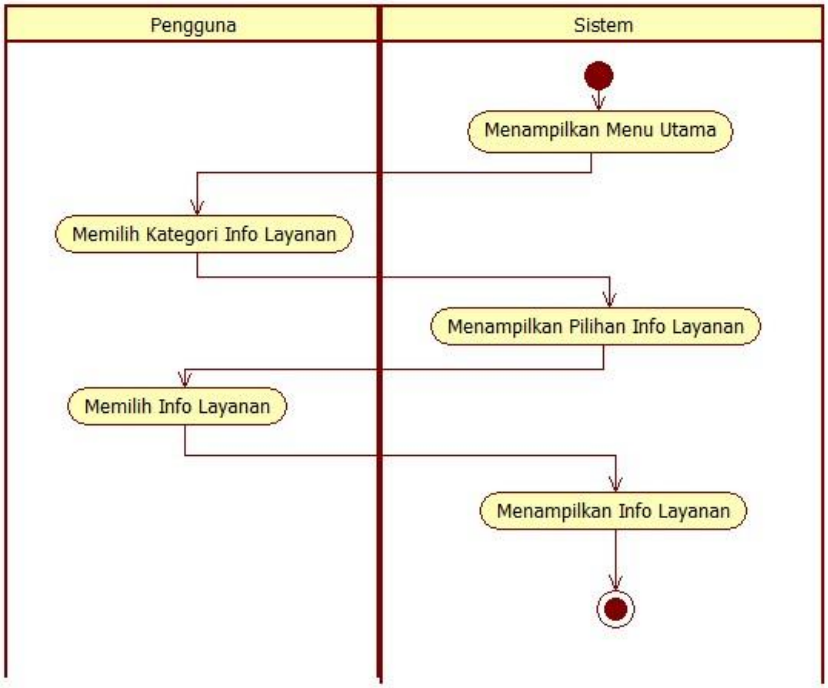
Main Flow	***Steps***	***Actions***
	1	Memilih gedung
	2	Memilih Ruangan atau area

3.1.7.3. *Activity Diagram*

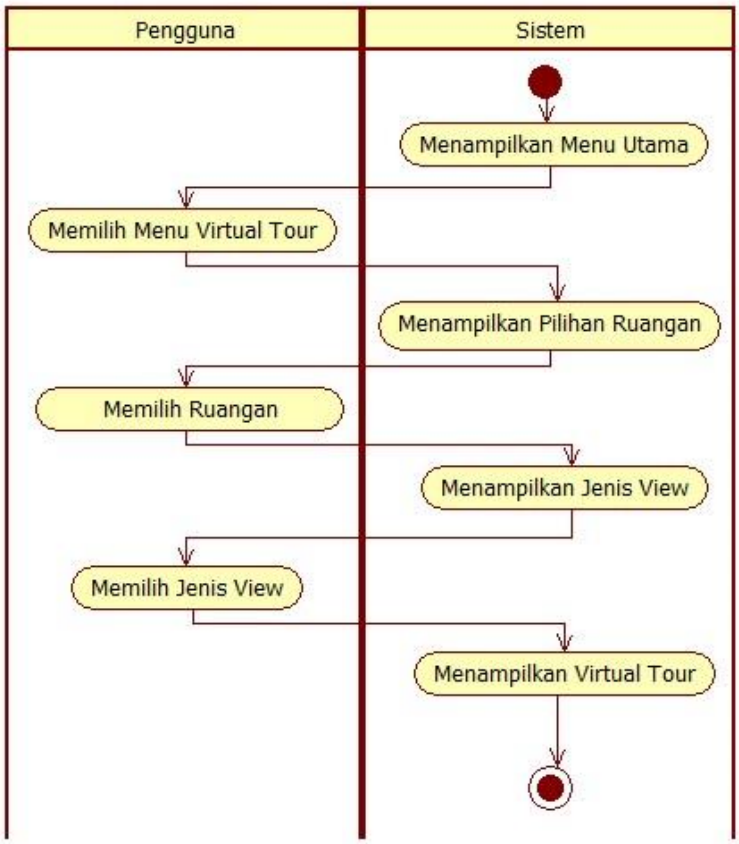
Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis atau sebuah sistem dan urutan aktivitas dalam sebuah proses.



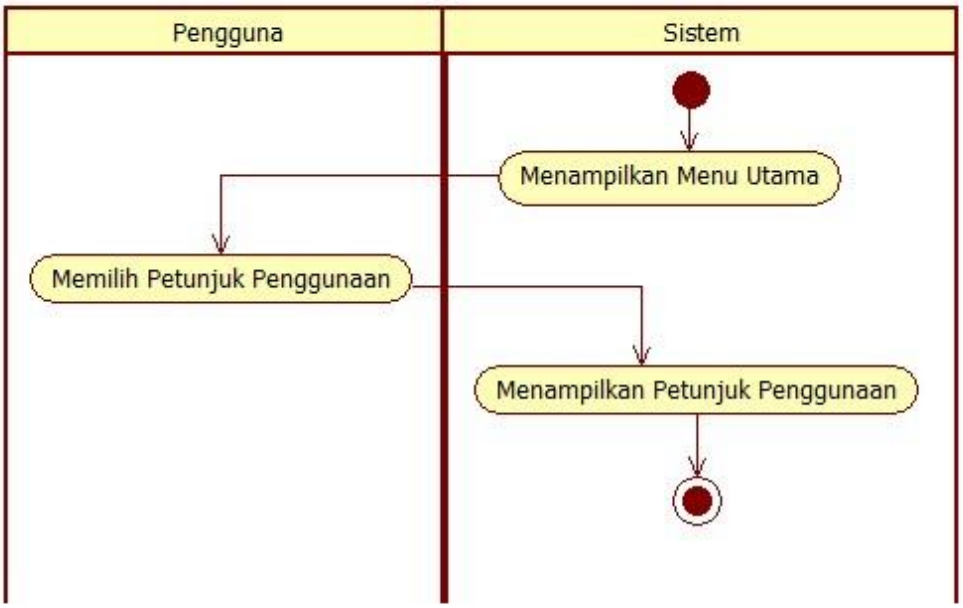
Gambar 3.41 *Activity Diagram* Menampilkan Rute Terpendek



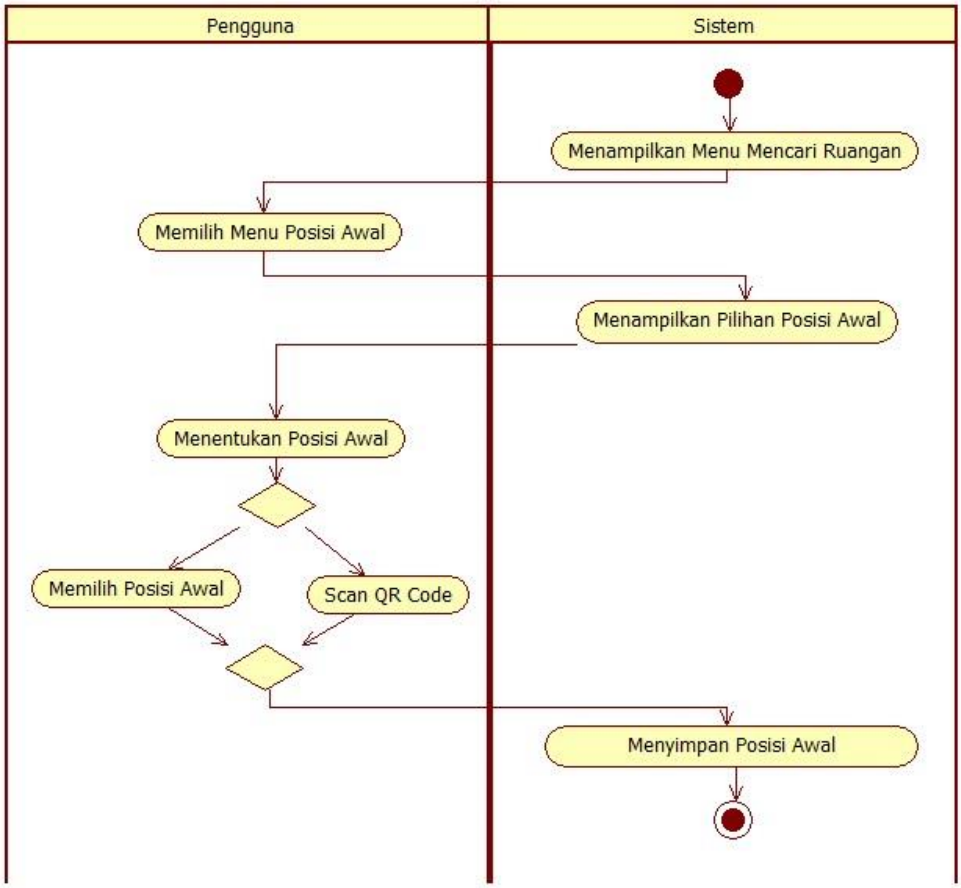
Gambar 3.42 *Activity Diagram* Menampilkan Info Layanan



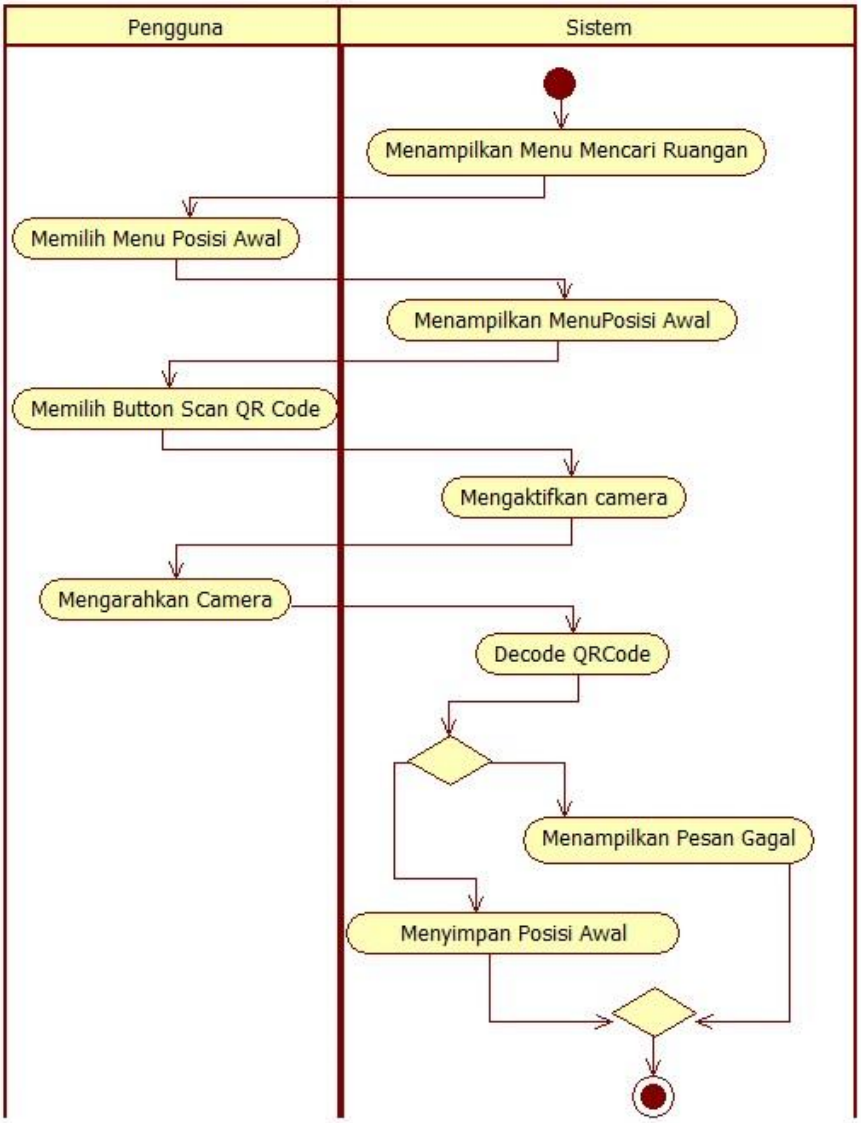
Gambar 3.43 *Activity Diagram* Memulai *Virtual Tour*



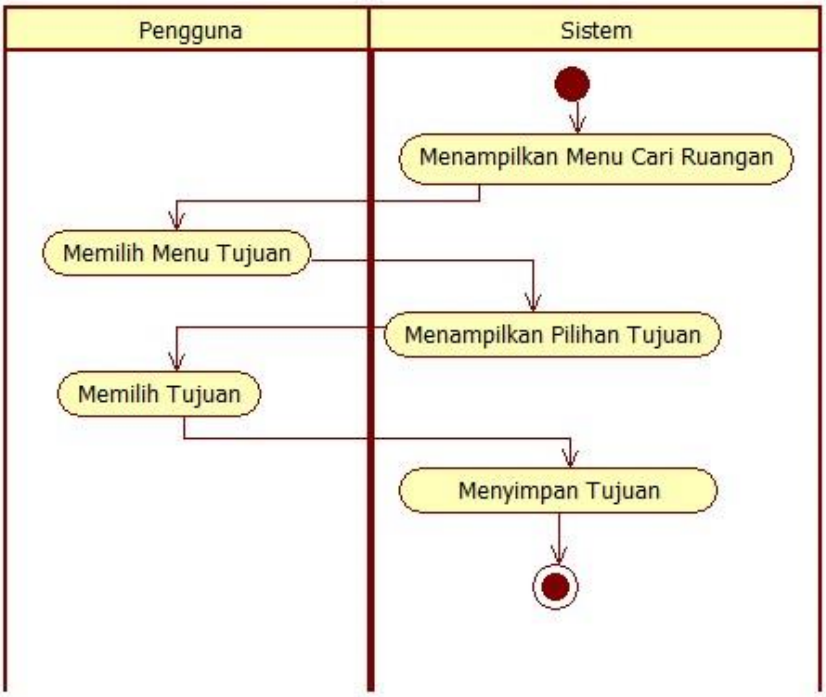
Gambar 3.44 *Activity Diagram* Melihat Petunjuk Penggunaan



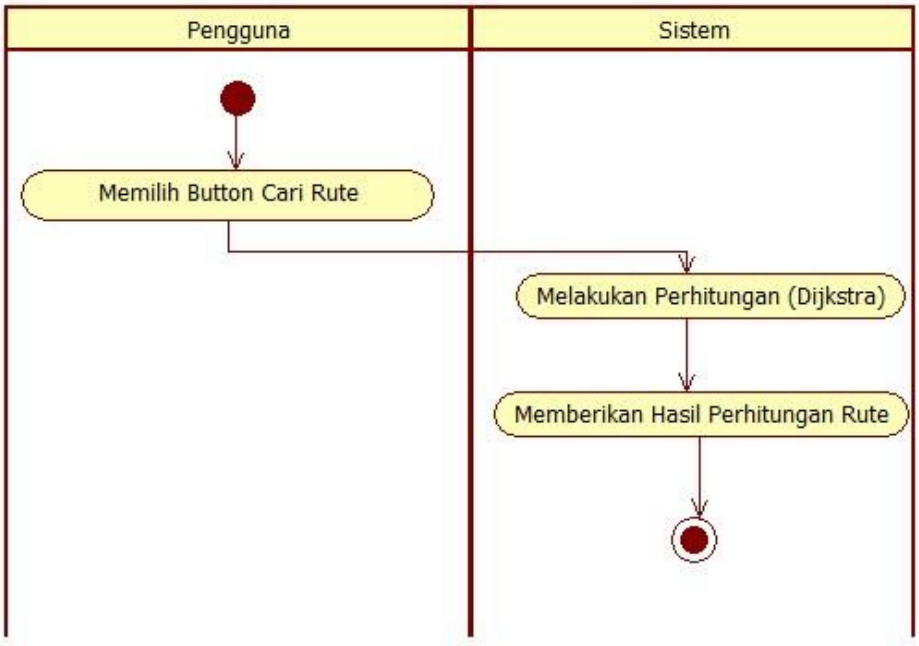
Gambar 3.45 *Acitivity Diagram* Menentukan Posisi Awal



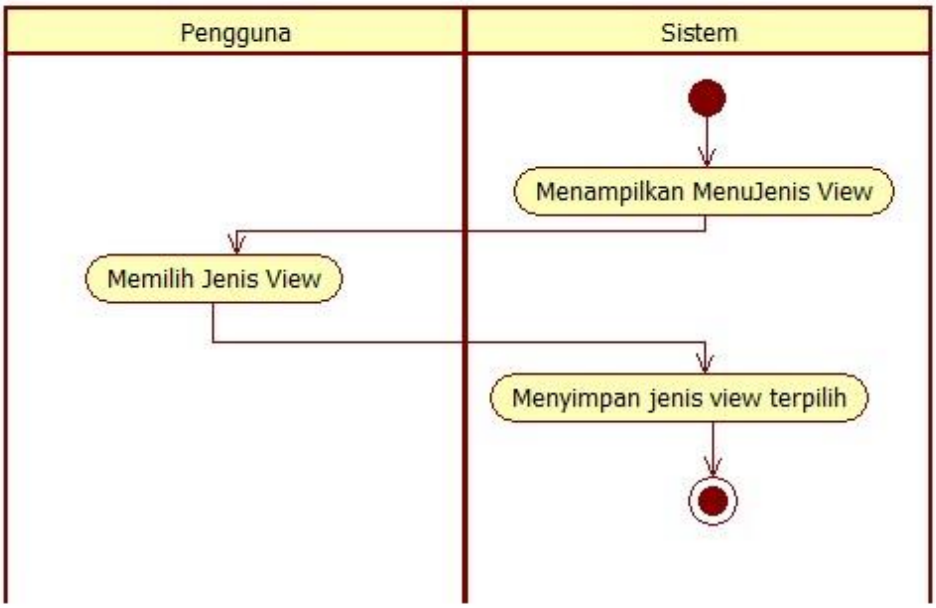
Gambar 3.46 ***Activity Diagram Scan QR Code***



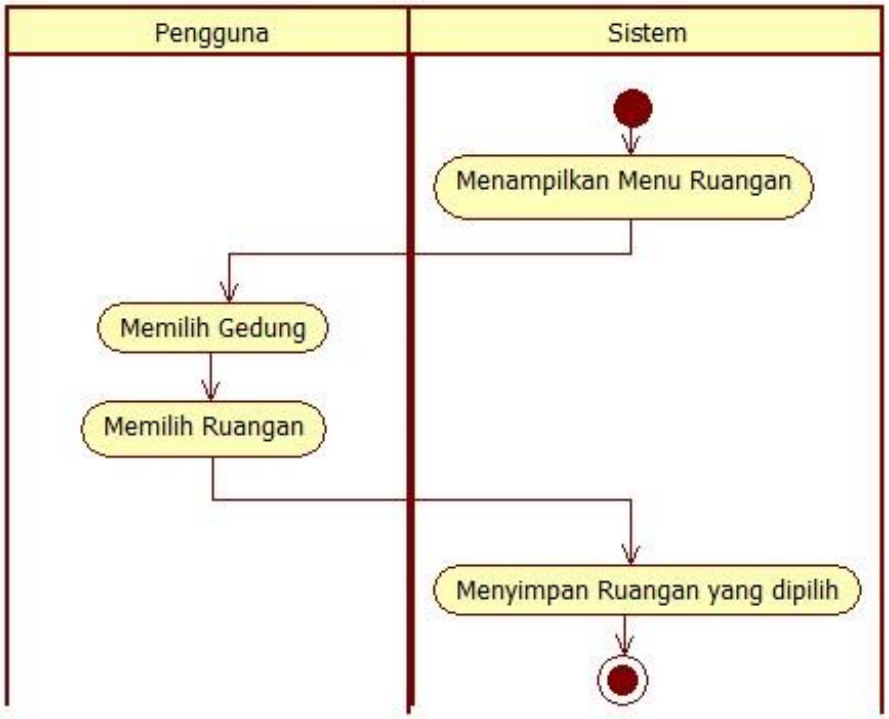
Gambar 3.47 *Activity Diagram* Memilih Tujuan



Gambar 3.48 *Activity Diagram* Mencari Rute Terpendek



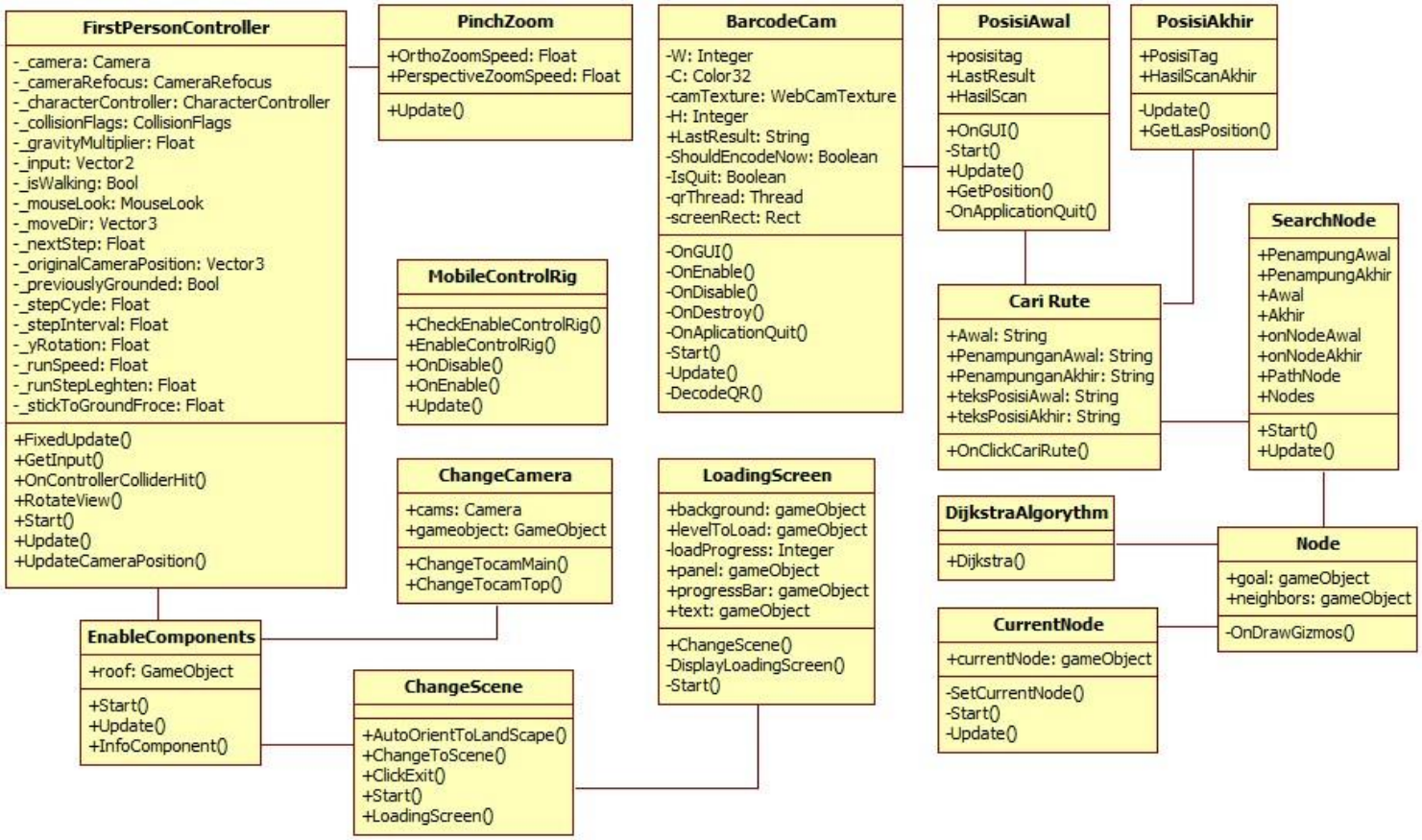
Gambar 3.49 *Activity Diagram* Memilih Jenis view



Gambar 3.50 *Activity Diagram* Memilih Ruangan

3.1.7.4. *Class Diagram*

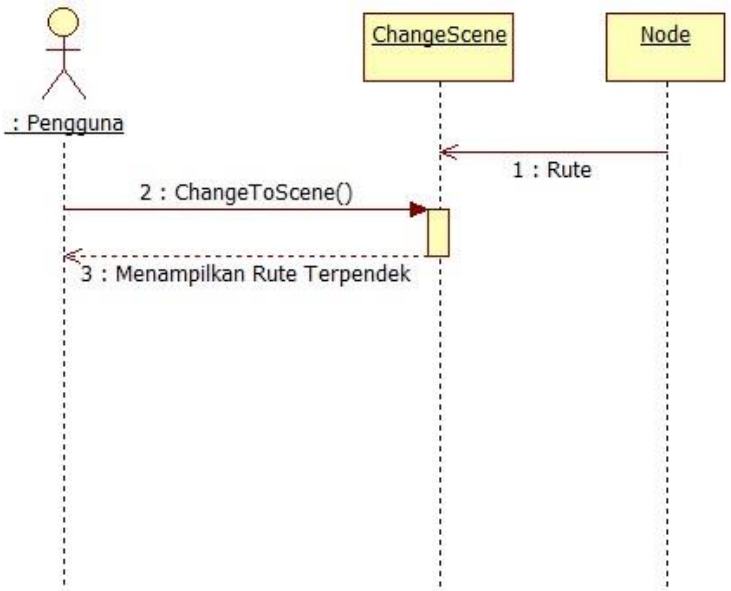
Class Diagram menggambarkan tipe-tipe objek dalam sistem dan berbagai jenis hubungan atau relasi statis yang ada diantara mereka. Diagram ini memberikan gambaran umum dari sistem. Seperti tipe-tipe dari objek dengan menunjukan kelasnya dan relationship yang diantara mereka, serat penjelasan detail tiap-tiap kelas ke dalam model suatu sistem. Class diagram bersifat statik (tidak berubah) yang akan menunjukan apa itu interaksi tapi tidak menjelaskan apa yang terjadi ketika mereka melakukan interaksi.



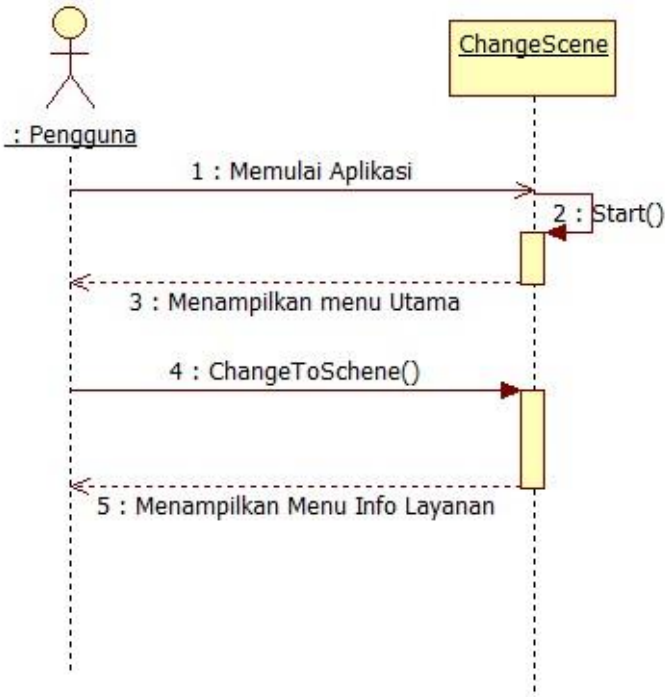
Gambar 3.51 ***Class Diagram***

3.1.7.5. *Sequence Diagram*

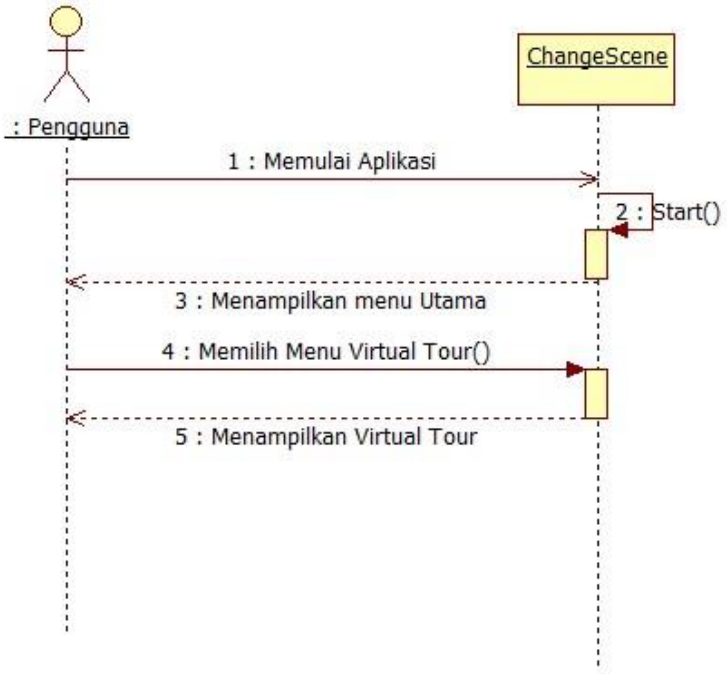
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu.



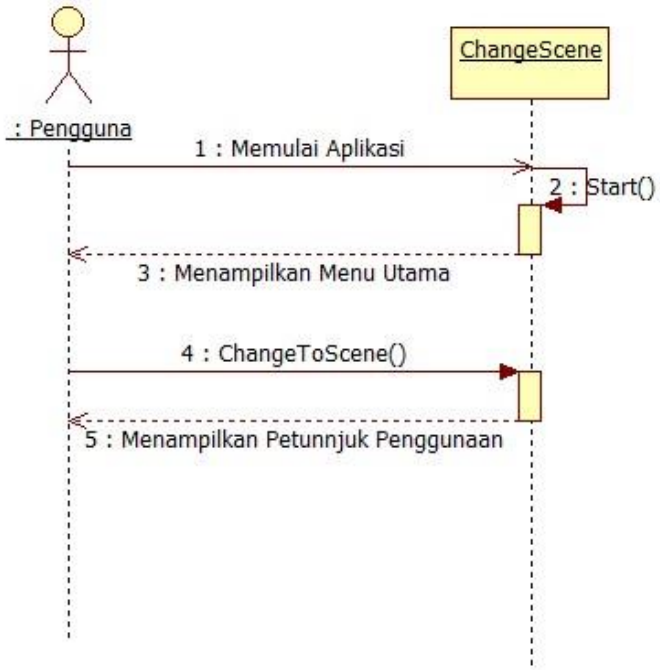
Gambar 3.52 *Sequence Diagram* Menampilkan Rute Terpendek



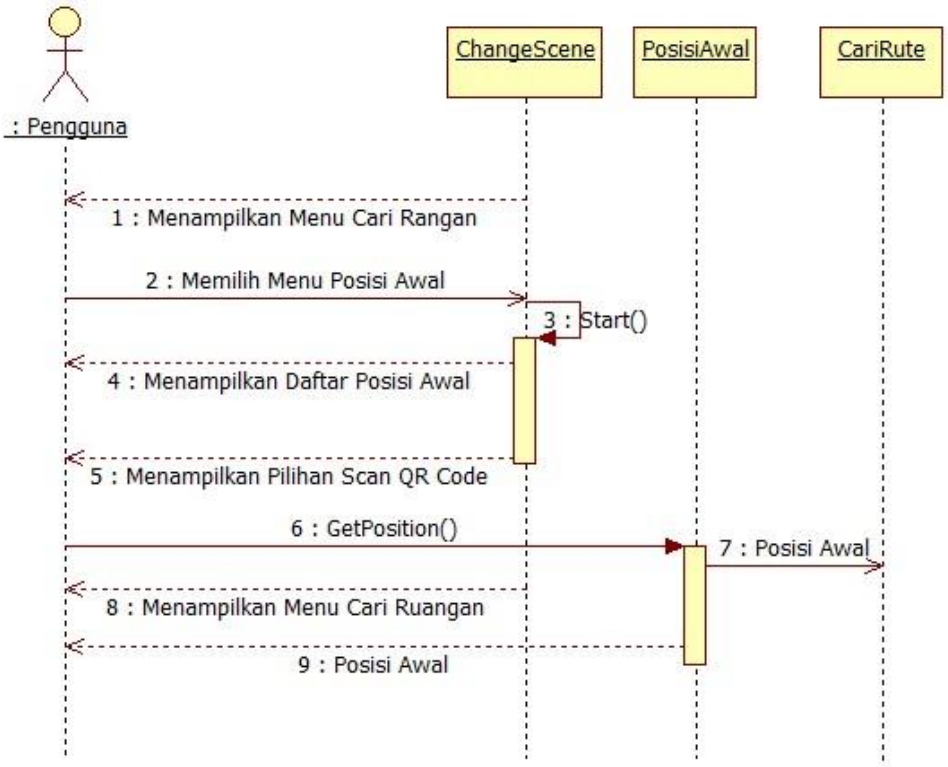
Gambar 3.53 *Sequence Diagram* Menampilkan Info Layanan



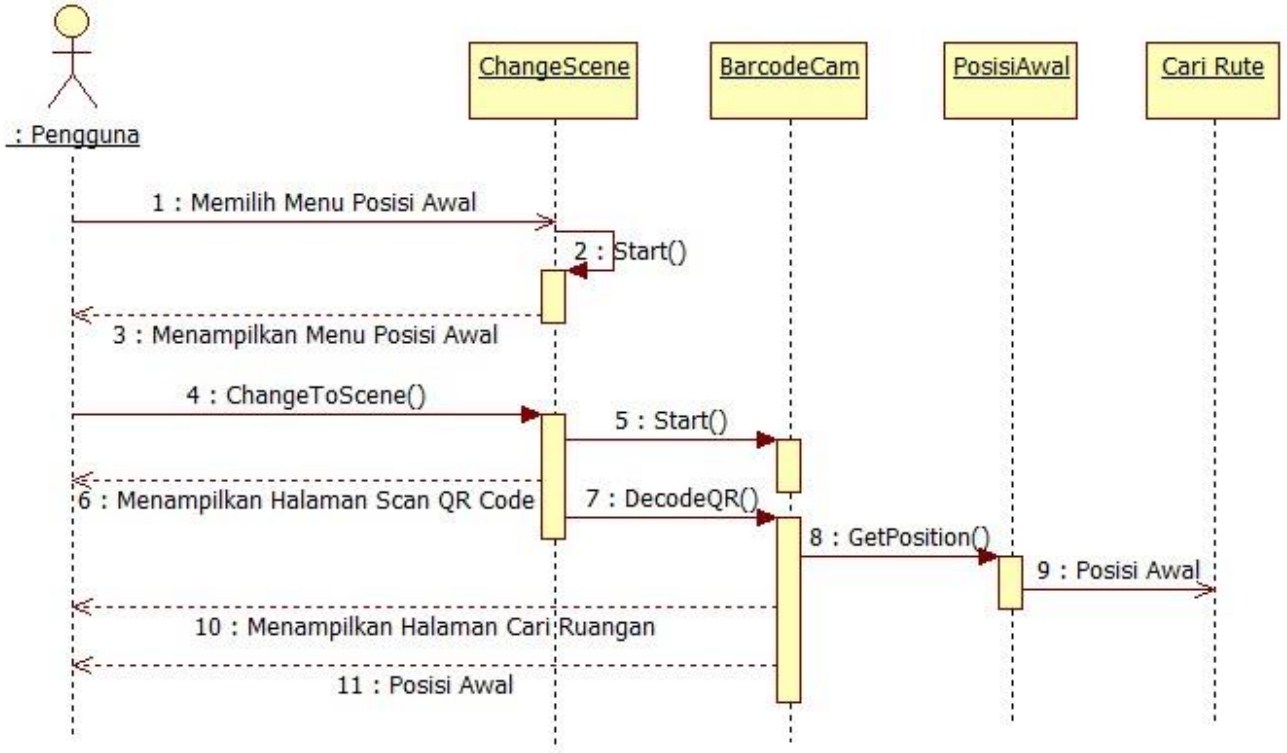
Gambar 3.54 *Sequence Diagram* Memulai *Virtual Tour*



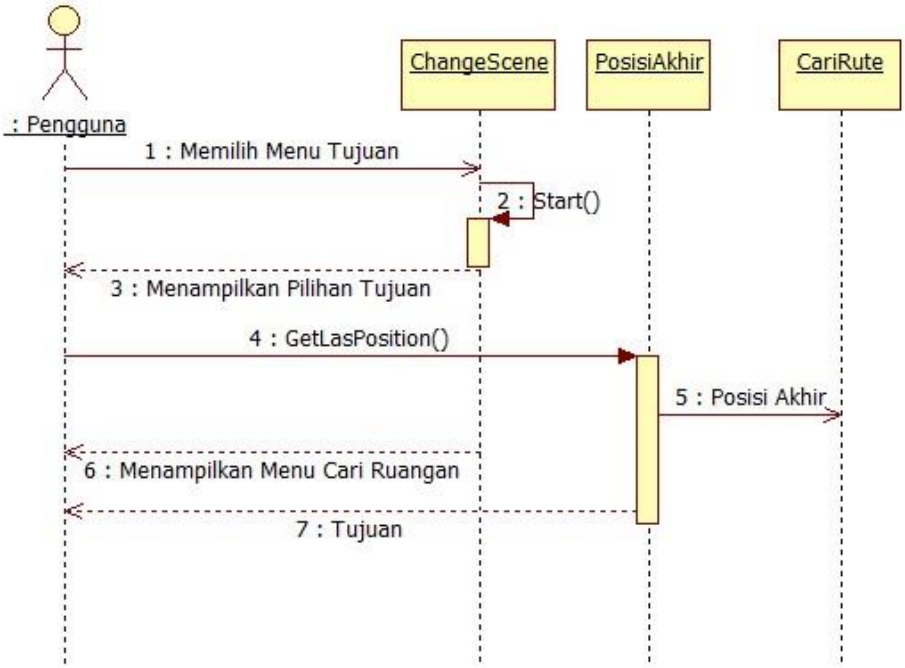
Gambar 3.55 *Sequence Diagram* Melihat Petunjuk Penggunaan



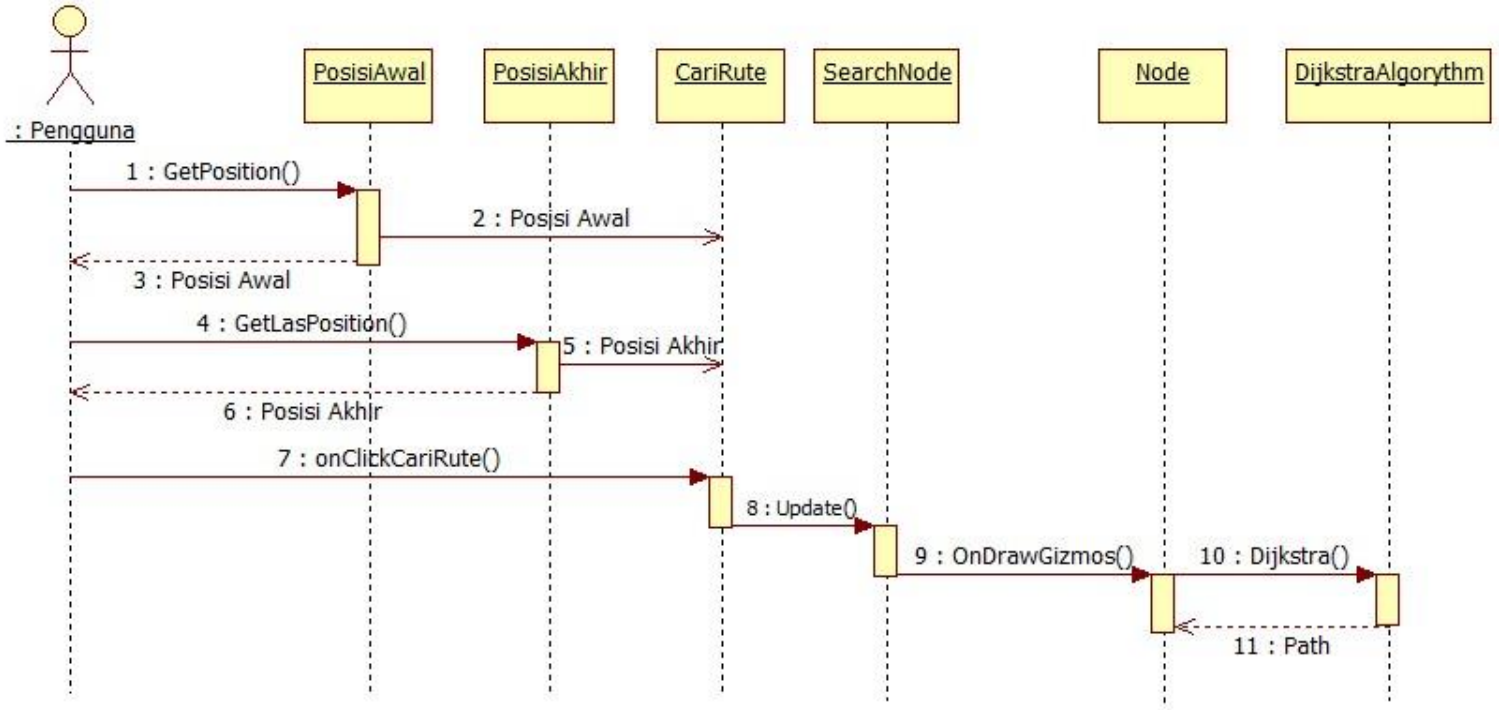
Gambar 3.56 *Sequence Diagram* Menentukan Posisi Awal



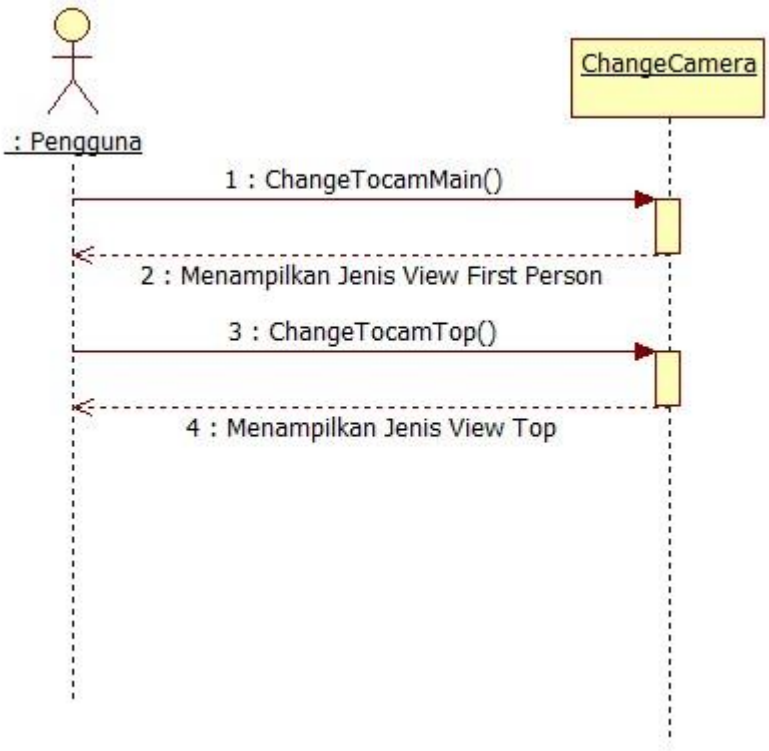
Gambar 3.57 ***Sequence Diagram Scan QR Code***



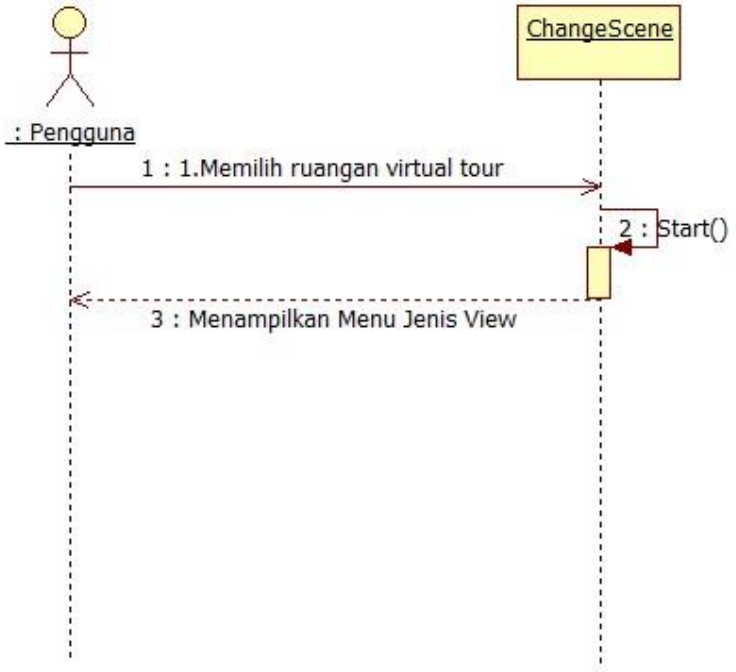
Gambar 3.58 *Sequence Diagram* Memilih Tujuan



Gambar 3.59 S*equence Diagram* Mencari Rute Terpendek



Gambar 3.60 *Sequence Diagram* Memilih Jenis View



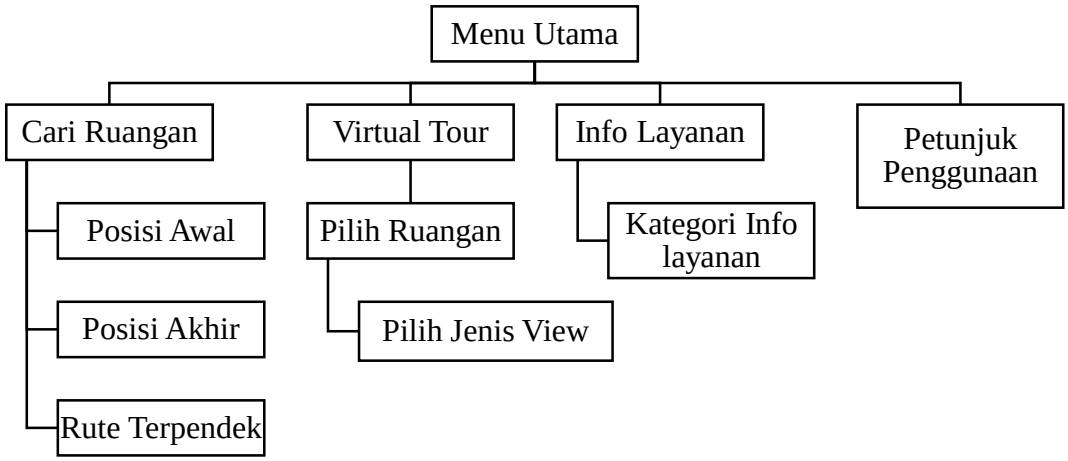
Gambar 3.61 *Sequence Diagram* Memilih Ruangan

3.1.8. Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan menu, perancangan antarmuka, dan jaringan semantik.

3.1.8.1. Perancangan Struktur Menu

Perancangan struktur menu merupakan gambaran alur pemakaian aplikasi. Perancangan struktur menu dari aplikasi ini adalah sebagai berikut:

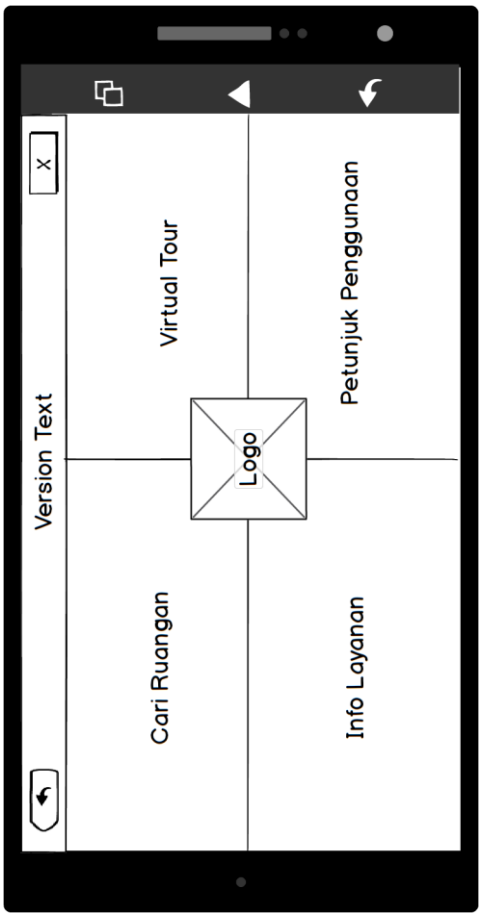


Gambar 3.62 Perancangan Struktur Menu

3.1.8.3. Perancangan Antarmuka

Perancangan antarmuka dilakukan untuk mempermudah dalam mengimplementasikan sistem yang akan dibangun. Berikut ini adalah perancangan antar muka pada aplikasi Panduan Layanan Pengunjung Berbasis Virtual Reality di Rumah Sakit Mata Cicendo Bandung:

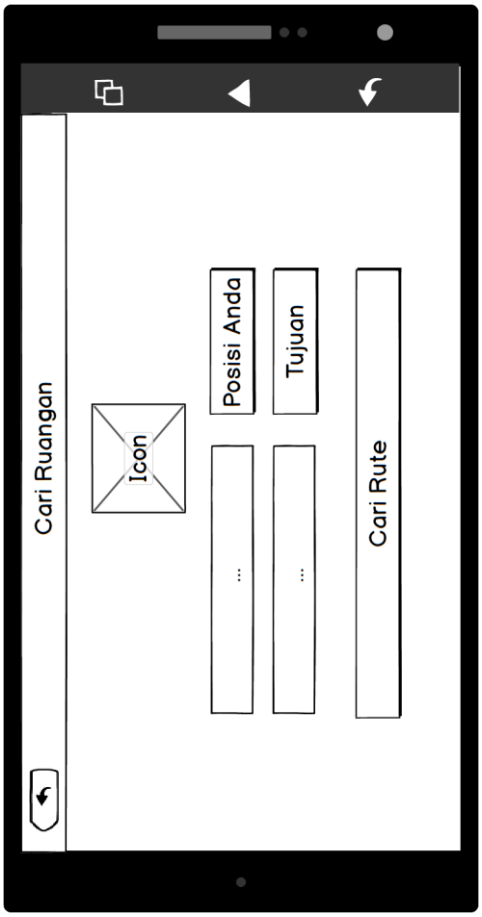
1. Perancangan Antarmuka Menu Utama

<table>
<tr><td colspan="2">T01</td></tr>
<tr><td>
</td><td>• Klik button Cari Ruangan untuk menuju T02
• Klik button Virtual Tour untuk menuju T07
• Klik button Info Layanan untuk menampilkan info layanan menuju T06
• Klik button x untuk menutup aplikasi</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

Gambar 3.63 Perancangan Antarmuka Menu Utama

2. Perancangan Antarmuka Menu Cari Ruangan

T02

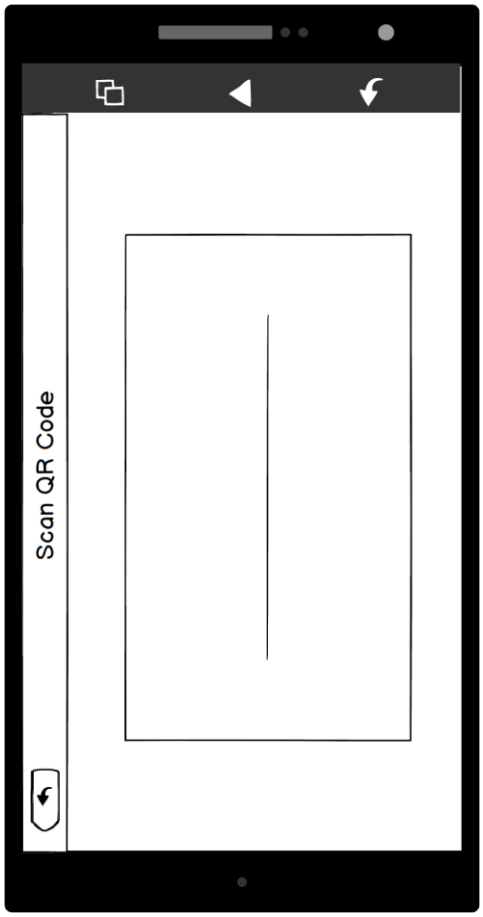


- Klik b*utton* Posisi Anda untuk menentukan posisi awal menuju ke halaman Scan QR Code T03
- Klik *button* Tujuan untuk menentukan ruangan tujuan di T04
- Klik *button* Cari Rute untuk menuju ke tampilan hasil rute terpendek pada T05
- Klik *button* kembali untuk menuju T01

Resolusi : *Relative*
Orientasi : *Landscape*

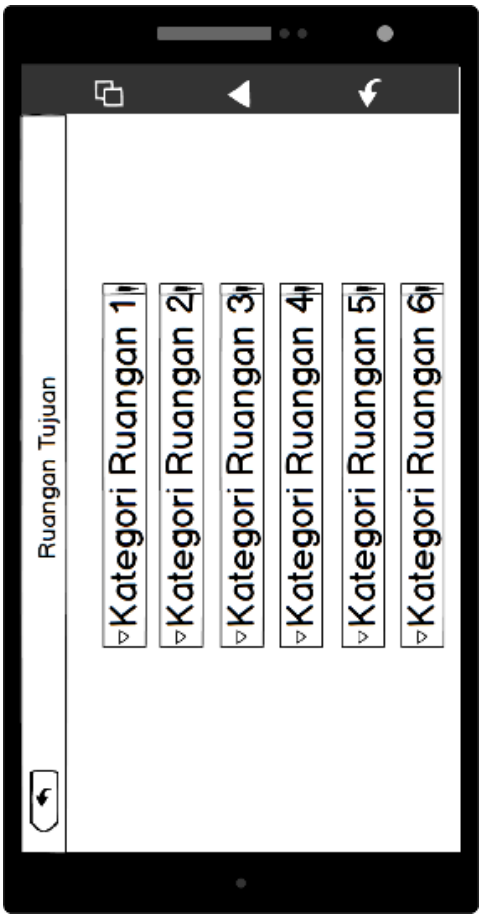
Gambar 3.64 Perancangan Antarmuka Menu Cari Ruangan

3. Perancangan Antarmuka Scan QR Code

<table>
<tr><td colspan="2">T03</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T01</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

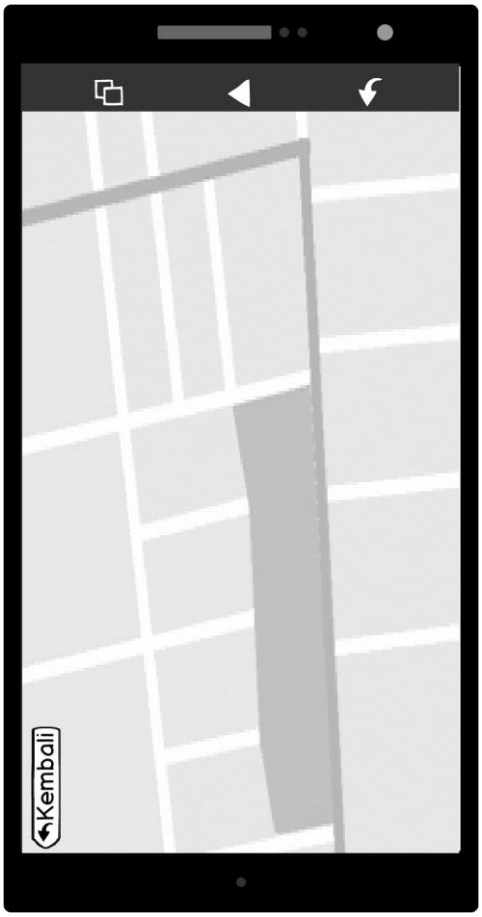
Gambar 3.65 Perancangan Antarmuka Scan QR Code

4. Perancangan Antarmuka Menu Ruang Yang Dituju

T04	
	• Teks Panduan adalah slot text untuk keterangan mengenai langkah yang harus dilakukan pada saat berada I antarmuka ini • Klik *button* kembali untuk menuju T02 • Pilih ComboBox Kategori untuk menampilkan pilihan ruangan dalam kategori tersebut
Resolusi : *Relative* Orientasi : *Landscape*	

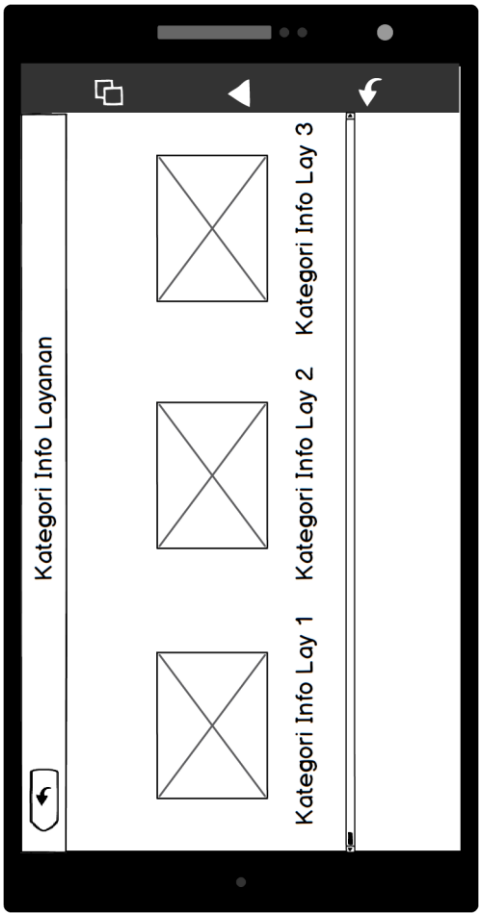
Gambar 3.66 Perancangan Antarmuka Menu Ruang Yang Dituju

5. Perancangan Antarmuka Rute Terpendek

<table>
<tr><td colspan="2">T05</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T02</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

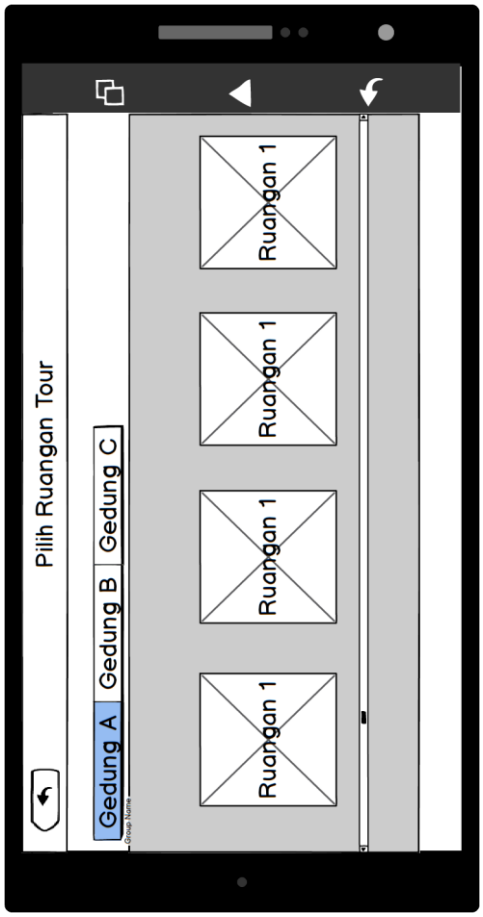
Gambar 3.67 Perancangan Antarmuka Rute Terpendek

6. Perancangan Antarmuka Kategori Info Layanan

<table>
<tr><td colspan="2">T06</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T01
• Pilih button berupa image kategori info layanan untuk memilih layanan informasi menuju T11</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Potrait, Landscape</td></tr>
</table>

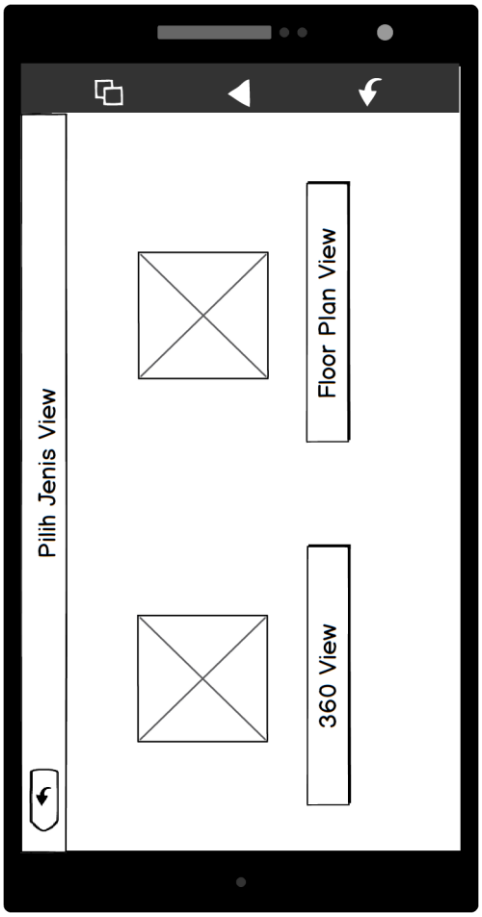
Gambar 3.68 Perancangan Antarmuka Kategori Info Layanan

7. Perancangan Antarmuka Menu Pilih Ruangan Virtual Tour

<table>
<tr><td colspan="2">T07</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T01
• Pilih Kategori Gedung untuk memilih kategori gedung
• Pilih gambar yang berfungsi sebagai button untuk menuju tampilan virtual tour T09</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Potrait</td></tr>
</table>

Gambar 3.69 Perancangan Antarmuka Menu Pilih Ruangan Virtual Tour

8. Perancangan Antarmuka Pilih Jenis View

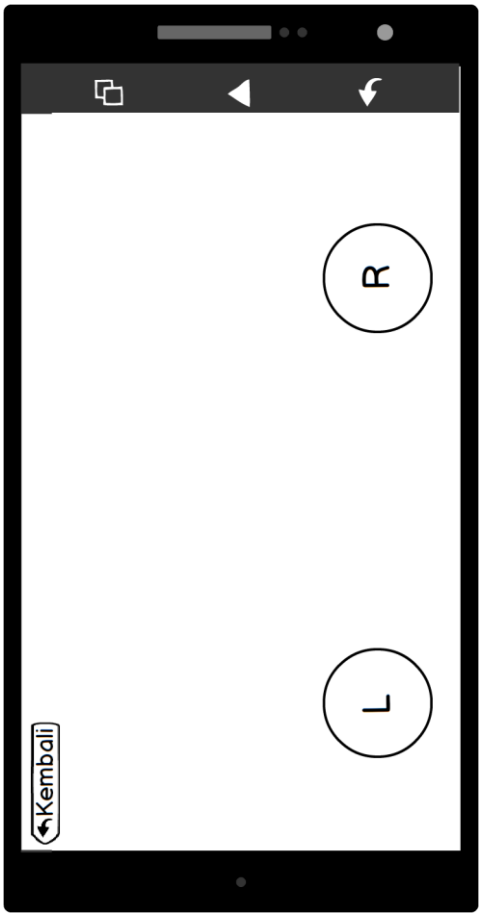
<table>
<tr><td colspan="2">T08</td></tr>
<tr><td>
</td><td>

Klik button kembali untuk menuju T07
Klik button 360 View untuk menuju T09
Klik button Floor Plan View untuk menuju T09

</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Potrait</td></tr>
</table>

Gambar 3.70 Perancangan Antarmuka Pilih Jenis View

9. Perancangan Antarmuka Virtual Tour

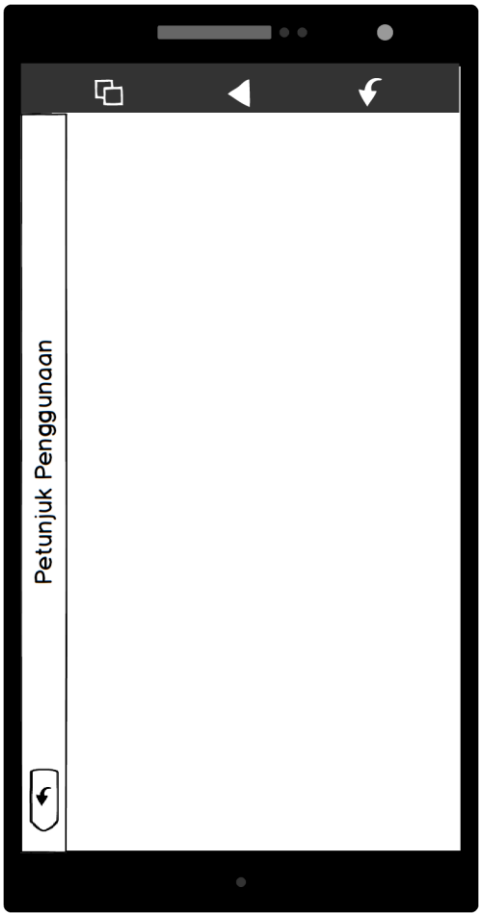
<table>
<tr><td colspan="2">T09</td></tr>
<tr><td>
</td><td>

Klik button kembali untuk menuju T08
L adalah controller untuk movement
R adalah controller untuk mengatur sudut pandang

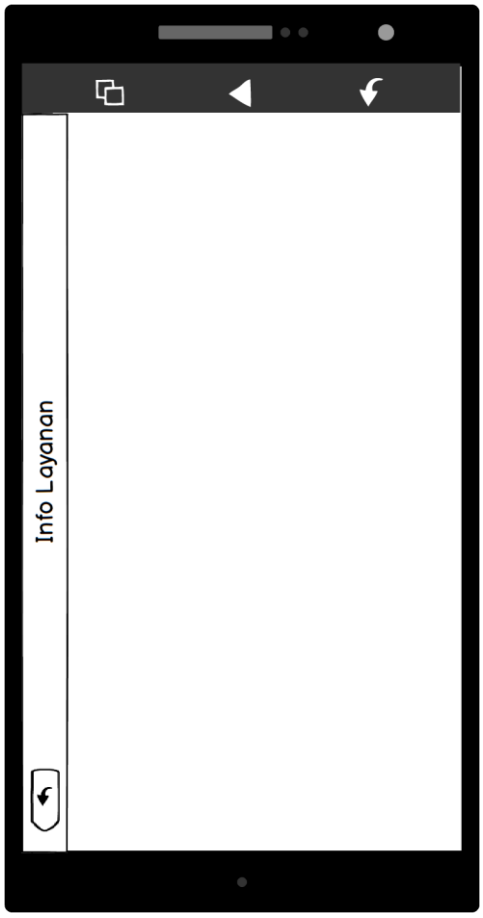
</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

Gambar 3.71 Perancangan Antarmuka Virtual Tour

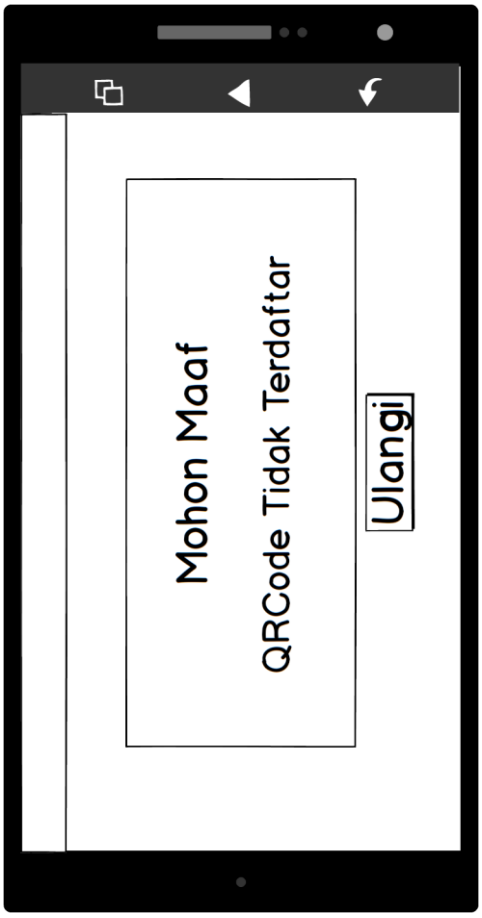
Tabel 3.29 Perancangan Antarmuka Petunjuk Penggunaan

<table>
<tr><td colspan="2">T10</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T01
• Blank space adalah area untuk konten petunjuk penggunaan</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

Tabel 3.30 Perancangan Antarmuka Info Layanan

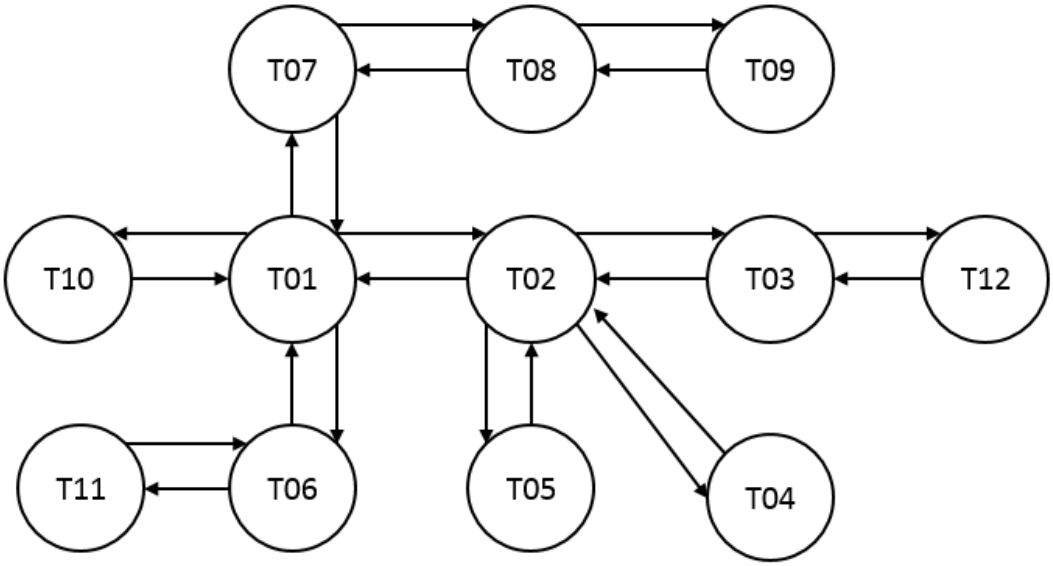
<table>
<tr><td colspan="2">T11</td></tr>
<tr><td>
</td><td>• Klik button kembali untuk menuju T06
• Blank space adalah area untuk konten info layanan</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

Tabel 3.31 Perancangan Antarmuka Pesan Salah

<table>
<tr><td colspan="2">T12</td></tr>
<tr><td>
</td><td>• Klik button ulangi untuk menuju T03
• Blank space adalah area untuk konten info layanan</td></tr>
<tr><td colspan="2">Resolusi : Relative
Orientasi : Landscape</td></tr>
</table>

3.1.8.4. Jaringan Semantik

Jaringan semantik merupakan gambaran tentang relasi dari masing-masing halaman yang bisa diakses oleh pengguna. Jaringan semantik ini, digunakan sebagai panduan penggunaan menu-menu dalam sebuah aplikasi. Adapaun jaringan semantik aplikasi ini adalah sebagai berikut:



Gambar 3.72 Jaringan Semantik