BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis

Analisis bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem serta menentukan kebutuhan dari sistem yang dibangun.Analisis tersebut meliputi analisis masalah, analisis kebutuhan data dan analisis sistem.Analisis merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan, sehingga diperoleh solusi. Analisis merupakan tahapan yang paling penting, karena kesalahan dalam tahap ini akan menyebabkan kesalahan di tahap selanjutnya. Aplikasi yang dibuat merupakan program pemandu lokasi dalam bentuk denah tiga dimensi pada Rumah Sakit Gunung Jati Kota Cirebon. Objek yang dibuat merupakan ruangan ruangan pada rumah sakit sehingga pengguna dapat berinteraksi langsung dengan objek virtual dalam dunia nyata. Selain itu diberikan pula jalur tercepat untuk menuju ke ruangan yang dimaksud juga informasi mengenai ruangan tersebut.

3.1.1 Analisis Masalah

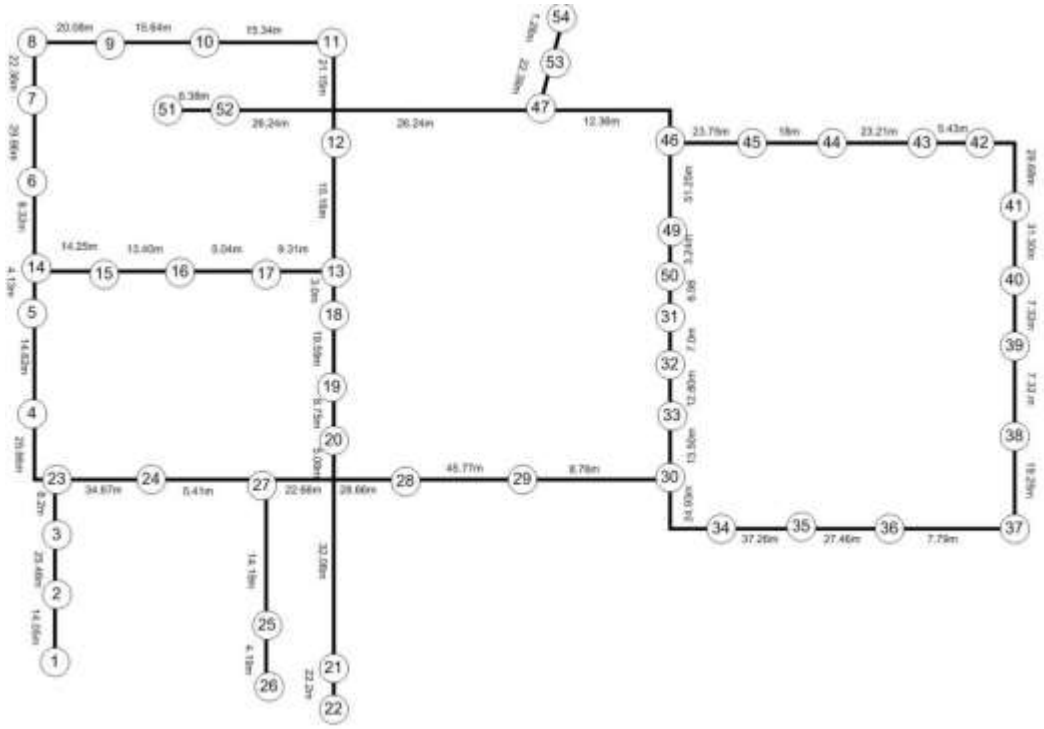
Setelah melakukan penelitian didapatkan beberapa masalah yaitu belum adanya sistem pemandu lokasi berbasis denah 3D untuk pengunjung rumah sakit mencari setiap ruangan yang ada. Dengan ±308 ruangan yang terdapat pada Rumah Sakit Umum Gunung Jati, serta ditambah informasi hanya di dapat melalui para pegawai rumah sakit dan petugas keamanan membuat beberapa pengunjung harus berputar-putar dalam menemukan ruangan yang mereka cari. Hal ini juga membuat pengunjung harus menempuh jarak yang cukup jauh untuk menuju keruangan, yang seharusnya dapat ditempuh dengan jarak yang lebih dekat.

3.1.2 Analisis Algoritma Floyd Warshall

Untuk mengetahui suatu metode yang akan digunakan dalam proses penelitian maka terlebih dahulu harus dilakukan analisis terhadap metode yang akan digunakan guna mendukung kesesuaian dalam hal implementasi terhadap masalah yang ada. Dalam hal ini penulis mencoba menganalisis algoritma *Floyd Warshall* yang dimana digunakan penulis sebagai metode untuk pencarian rute terpendek.

3.1.2.1 Penerapan Cara Kerja Algoritma Floyd Warshall

Misalkan seseorang akan melakukan pencarian ruang dari ruang 6 ke ruang 18. Orang tersebut mencoba untuk menerapkan algoritma *Floyd-Warshall* untuk mencari jalur terpendek dari ruang 6 ke ruang 18.



Gambar 3.1 Representasi keterhubungan antar ruang dalam *graf* berbobot

Sekarang, orang tersebut mencoba menerapkan algoritma *Floyd-Warshall* dengan pendekatan pemrograman dinamis maju (*forward*).

Basis

f1 (s) = *c*x1s (titik tujuan)

Rekurens

*f*k (s) = min *x*k {*cx*ks + *f*k-1(*x*k)} (jarak titik awal ditambah dengan titik tujuan)

x (titik awal)

$$f1\ (s) = cx1s$$

Tahap 1 :

f1 (s) = *c*x1s

Tabel 3.1 Matriks 1

S	**Solusi Optimum**	
	***f1* (s)**	***x1***
7	29.96	6
14	8.32	6

Pada tabel ini adalah dimana x1 (titik awal) = 6 dan S (titik yang berdekatan atau titik tujuan) = 7 dan 14, memiliki nilai jarak yaitu f1.

Tahap 2:

f2 (s) = min *s*2 {*c*x2s + *f*1(*x*2)}

Tabel 3.2 Matriks 2

***x*2**	***f2* (x2, s) = *c*x2s +*f*1(*x*2)**		**Solusi Optimum**	
S	7	14	***f 2* (s)**	***x*2**
8	53.32	∞	53.32	7
15	∞	22.57	22.57	14
5	∞	12.45	12.45	14

Pada tahap kedua didapatkan x2 (titik awal)= 7 dan 14, pada tahap ini titik 14 memiliki S lebih dari satu sehingga memperoleh jalur baru yaitu titik 15 dan 5.

Tahap 3:

f 3 (s) = min *s3* {*c*x3s + *f*2(*x*3)}

Tabel 3.3 Matriks 3

x3	***f 3* (*x3*, s) = *c*x3s + *f*2(*x*3)**			**Solusi Optimum**	
S	8	15	5	***f 3* (s)**	***x3***
9	73.40	∞	∞	73.40	8
16	∞	41.78	∞	41.78	15
4	∞	∞	27.07	27.07	5

Pada tahap 3 memiliki x3 = 8, 15dan 5, dan memiliki S = 9, 16, dan 4 dan f3 sebagai nilai jarak dari x3 menuju S.

Tahap 4:

f4(s) = min *s4* {*c*x4s + *f*3(*x*4)}

Tabel 3.4 Matriks 4

*x*4	***f 4* (*x*4, s) = *c*x4 + *f*3(*x*4)**			**Solusi Optimum**	
S	9	16	4	***f 4*(s)**	***x*4**
10	89.04	∞	∞	89.04	9
17	∞	46.82	∞	46.82	16
23	∞	∞	52.73	52.73	4

Pada tahap 4 memiliki x4 = 9, 16 dan 4, dan memiliki S = 10, 17, dan 23 dan f4 sebagai nilai jarak dari x4 menuju S.

Tahap 5:

$f5(s) = \min s5 \{cx5s + f4(x5)\}$

Tabel 3.5 Matriks 5

*x*5	***f 5*(*x5*, s) = *c*x5 + *f4*(*x5*)**			**Solusi Optimum**	
S	10	17	23	***f5*(s)**	**x5**
11	104.38	∞	∞	104.38	10
13	∞	56.13	∞	56.13	17
24	∞	∞	87.4	87.4	23
3	∞	∞	60.93	60.93	23

Pada tahap kelima didapatkan x5 (titik awal)= 10, 17, dan 23, pada tahap ini titik 23 memiliki S (titik yang berdekatan) lebih dari satu sehingga memperoleh jalur baru yaitu titik 24 dan 3.

Tahap 6:

$f6(s) = \min s6\{cx6s + f5(x6)\}$

Tabel 3.6 Matriks 6

*x*6	***f 6*(*x6*, s) = *c*x6 + *f5*(*x6*)**				**Solusi Optimum**	
S	11	13	24	3	***f6*(s)**	**x6**
12	125.53	15.16	∞	∞	125.53	11
52	151.77	∞	∞	∞	151.77	11
47	151.77	∞	∞	∞	151.77	11
18	∞	**59.13**	∞	∞	**59.13**	**13**
27	∞	∞	92.81	∞	92.81	24
2	∞	∞	∞	86.41	86.41	3

Pada tahap keenam didapatkan x5 (titik awal)= 11, 13, 24, dan 3, pada tahap ini titik 11 memiliki S (titik yang berdekatan) lebih dari satu sehingga memperoleh jalur baru yaitu

titik 12, 52, dan 47. Dan pencarian jalur terpendek dari ruang 6 menuju ruang 8 telah didapat dengan rute **6, 14, 15, 16, 17, 13, 18** dengan jarak tempuh 59.13m.

Tahap 7:

$f7(s) = \min s7\{c\text{x}7s + f6(x7)\}$

Tabel 3.7 Matriks 7

x7	***f 7*(*x7*, s) = *c*x7 + *f6*(*x7*)**					**Solusi Optimum**	
S	12	52	47	27	*2*	***f7*(s)**	**x7**
13	140.69	∞	∞	∞	∞	140.69	12
51	∞	158.33	∞	∞	∞	151.77	52
46	∞	∞	164.13	∞	∞	164.13	47
53	∞	∞	174.15	∞	∞	174.15	47
25	∞	∞	∞	107	∞	107	27
20	∞	∞	∞	120.55	∞	120.55	27
28	∞	∞	∞	141.13	∞	141.13	27
1	∞	∞	∞	∞	100.46	100.46	2

Pada tahap ketujuh didapatkan x7 (titik awal)= 12, 52, 47, 27 dan 2, pada tahap ini titik 47 dan 27 memiliki S (titik yang berdekatan) lebih dari satu sehingga memperoleh jalur baru yaitu titik 46, dan 53 untuk titik 47 dan 25, 20, 28 untuk titik 27.

Tahap 8:

$f8(s) = \min s8\{c\text{x}8s + f7(x8)\}$

Tabel 3.8 Matriks 8

*x*8	***f 8*(*x8*, s) = *c*x8 + *f7*(*x8*)**						**Solusi Optimum**	
S	13	46	53	25	20	28	***f8*(s)**	**x8**
18	**143.69**	∞	∞	∞	∞	∞	**140.69**	**13**
45	∞	187.88	∞	∞	∞	∞	187.88	46

49	∞	215.24	∞	∞	∞	∞	215.24	46
54	∞	∞	175.21	∞	∞	∞	175.21	53
26	∞	∞	∞	111.19	∞	∞	111.19	25
19	∞	∞	∞	∞	129.30	∞	129.30	20
29	∞	∞	∞	∞	∞	186.9	141.13	28

Pada tahap kedelapan didapatkan x8 (titik awal)= 13, 46, 53, 25, 20 dan 28, pada tahap ini titik 46 memiliki S (titik yang berdekatan) lebih dari satu sehingga memperoleh jalur baru yaitu titik 45, dan 49. Proses iterasi pada titik 1 dan 51 terhenti karena titik tersebut tidak mencapai tujuan yaitu titik 18. Dan didapat jalur terpendek dari ruang 6 menuju ruang 8 dengan rute 6,7,8,9,10,11,12,13,18 dengan jarak tempuh 143.69 m.

Tahap 9:

$f9(s) = \min s9\{c\text{x}9\text{s} + f8(x9)\}$

Tabel 3.9 Matriks 9

*x*9	***f* 9(*x9*, s) = *c*x9 + *f9*(*x9*)**				**Solusi Optimum**	
S	45	49	19	29	***f9*(s)**	**x9**
44	205.88	∞	∞	∞	205.88	45
50	∞	218.92	∞	∞	218.92	49
18	∞	∞	**148.89**	∞	**148.89**	**19**
30	∞	∞	∞	195.66	195.66	29

Pada tahap kesembilan didapatkan x8 (titik awal)= 45, 49, 19, dan 29, pada tahap ini. Proses iterasi pada titik 54 dan 26 terhenti karena titik tersebut tidak mencapai tujuan yaitu titik 18. Dan didapat jalur terpendek dari ruang 6 menuju ruang 8 dengan rute 6,14,5,4,23,24,27,20,19,18 dengan jarak tempuh 148.69m.

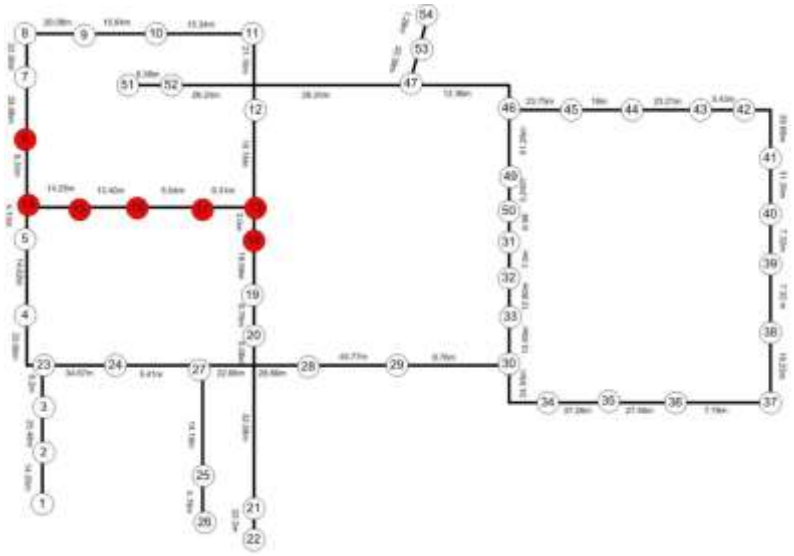
Setelah melakukan proses iterasi maka diperoleh 3 rute, maka dilakukanlah perbandingan hasil untuk memperoleh rute terpendek dari ruang 6 menuju ruang 18 :

Rute 1 : 6 + 14 + 15 + 16 + 17 + 13 + 18 = 59.13m

Rute 2 : 6 + 7 + 8 + 9 + 10 + 11 + 12 + 13 + 18 = 143.69m

Rute 3 : 6 + 14 + 5 + 4 + 23 + 24 + 27 + 20 + 19 + 18 = 148.69m

Dari hasil pencarian jalur terpendek dari ruang 6 ke ruang 18 menggunakan algoritma *Floyd-Warshall* (pemrograman dinamis), ditemukan bahwa jarak terpendek dari ruang 6 ke ruang 18 adalah **59.13**m dengan jalur (**6, 14, 15, 16, 17, 13, 18**). Setelah melakukan perbandingan maka didapat jalur *alternative* satu yaitu 6, 7, 8, 9, 10, 11, 12, 13, 18 dengan jarak tempuh 143.69m dan jalur *alternative* dua yaitu 6, 14, 5, 4, 23, 24, 27, 20, 19, 18 dengan jarak tempuh 148.69m.



Gambar 3.2 Representasi keterhubungan antar ruang setelah menerapkan algoritma *Floyd-Warshall* (sisi berwarna merah)

3.1.3 Analisis kebutuhan non-fungsional

Analisis *non-fungsional* dilakukan untuk menghasilkan spesifikasi kebutuhan *non-fungsional*. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan

untuk mengolah masukan menjadi keluaran, data yang akan ditangani sistem, jumlah pengguna dan kategori pengguna, serta kontrol terhadap sistem.

Pada analisis kebutuhan non fungsional ini dijelaskan analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras, dan analisis pengguna.

3.1.3.1 Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja pengguna aplikasi. Pengguna aplikasi ini sendiri yaitu para pengunjung Rumah Sakit Gunung Jati yang ber*platform* android sehingga dapat menggunakan aplikasi yang akan dibangun.

3.1.3.2 Analisis perangkat keras

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

Tabel 3.10 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	Layar	*TFT capacitive touchscreen, 16M colors*
5	Dimensi	128.7x 65 x 9.7 mm / 137 g

3.1.3.3 Analisis perangkat lunak

Untuk menjalankan aplikasi mobile Ngubek.com dibutuhkan perangkat lunak pendukung sehingga aplikasi yang dibangun dapat berjalan dengan baik. Adapun

perangkat lunak minimum yang dibutuhkan untuk menjalankan aplikasi yang dibangun adalah sistem operasi Android versi v4.3 (*Jelly Bean*).

Android *Jelly Bean* dipilih dikarenakan keunggulannya dalam grafis yang lebih tajam dibandingkan os android lainnya, dan didukung pula oleh *OpenGL* yang mampu menghasilkan refleksi tingggi, pencahayaan di tingkat yang lebih tinggi, peningkatan tekstur dan efek bayangan pada karakter 3D. Sehingga aplikasi Denah 3D ini mendapatkan grafik yang lebih baik dibandingkan menggunakan OS android yang ada sekarang.

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 Pencarian Rute Terpendek untuk Taksi

Aplikasi Pencarian Rute Terpendek untuk Taksi merupakan suatu aplikasi yang digunakan untuk mendapatkan jarak terdekat dengan waktu tercepat dengan rute antara Terminal Leuwi Panjang menuju Dipati Ukur menggunakan taksi. Grafik yang digunakan dalam aplikasi ini adalah 2 Dimensi (2D).

Berikut merupakan gambaran tampilan *output* Aplikasi Pencarian Rute Terpendek untuk Taksi.



Gambar 3.3 Output Total Jarak Terpendek



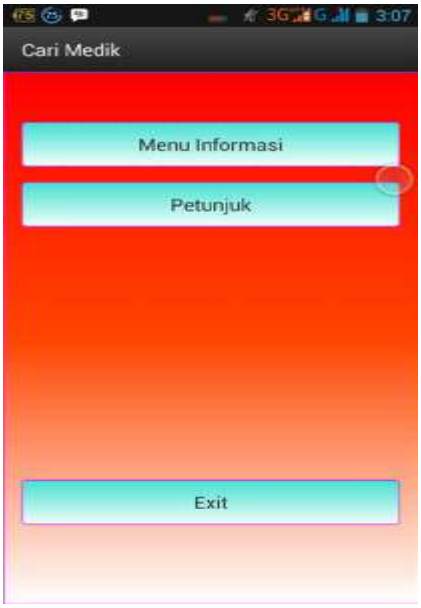
Gambar 3.4 *Output* Total Waktu Tercepat

3.1.4.2 Aplikasi Informasi RuteTerpendek Rumah Sakit, Rumah Bersalin dan Puskesmas di Palembang

Aplikasi Informasi Rute Terpendek Rumah Sakit, Rumah Bersalin dan Puskesmas di Palembang merupakan suatu aplikasi yang digunakan untuk memberikan informasi

tentang rute terpendekrumah sakit, rumah bersalin, dan puskesmas di kota Palembang. Grafik yang digunakan dalam aplikasi ini adalah 2 Dimensi (2D).

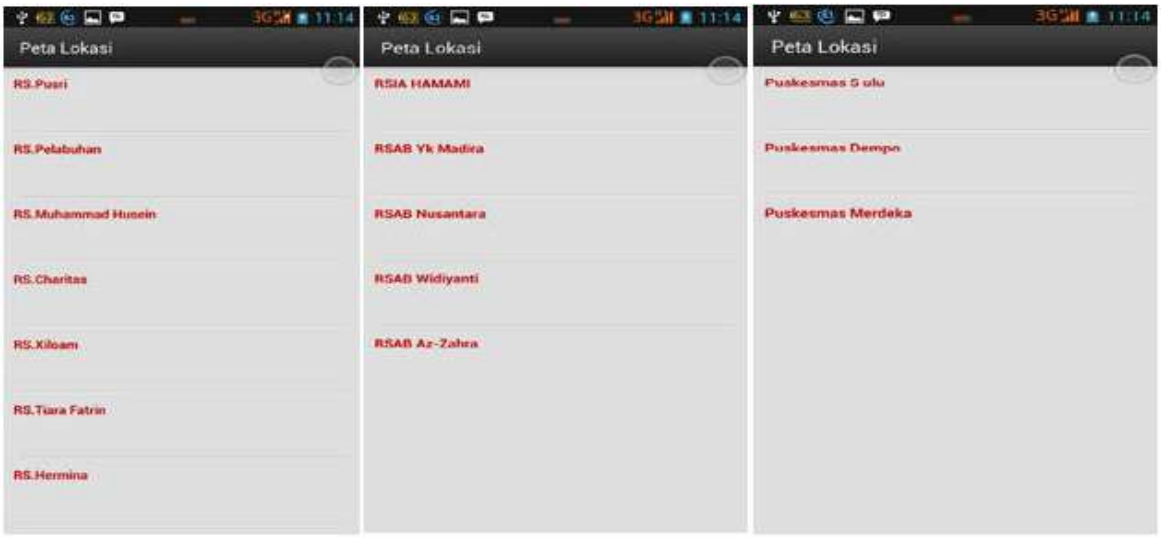
Berikut merupakan gambaran tampilan Aplikasi Informasi Rute Terpendek Rumah Sakit, Rumah Bersalin dan Puskesmas di Kota Palembang :



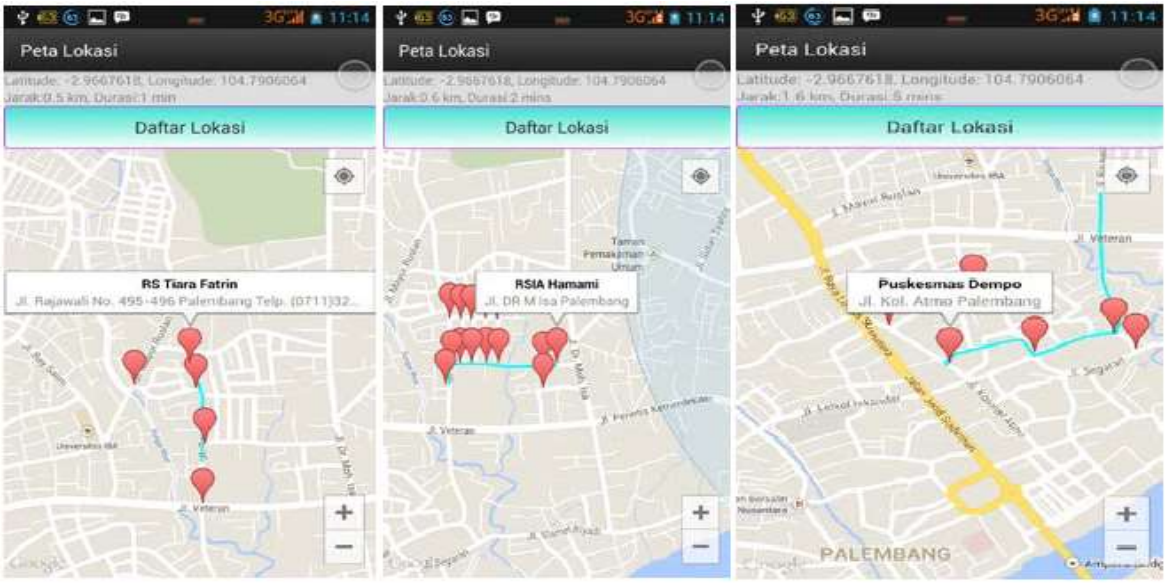
Gambar 3.5 *Form* Halaman Utama



Gambar 3.6 Tampilan Antarmuka Menu Cari Medik



Gambar 3.7 (a), (b) dan (c) Daftar Lokasi
Rumah Sakit, Rumah Bersalin dan Puskesmas



Gambar 3.8. (a), (b) dan (c) Peta Lokasi
Rumah Sakit, Rumah Bersalin dan Puskesmas



Gambar 3.9. Tampilan Antarmuka Menu Petunjuk



Gambar 3.10 Tampilan Map dari Titik Awal

3.1.4.3 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:



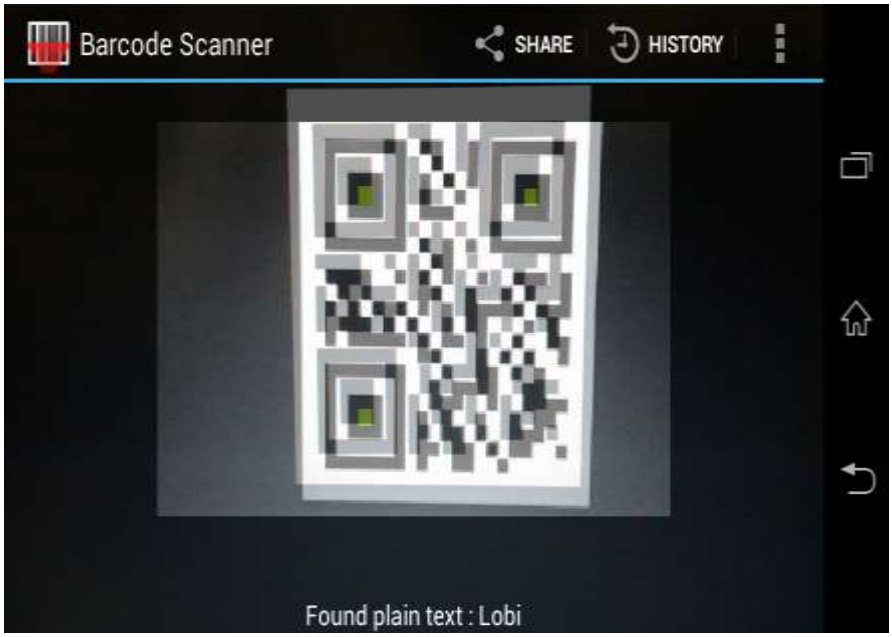
Gambar 3.11 Tampilan Main Menu



Gambar 3.12 Tampilan Aplikasi Denah 3D



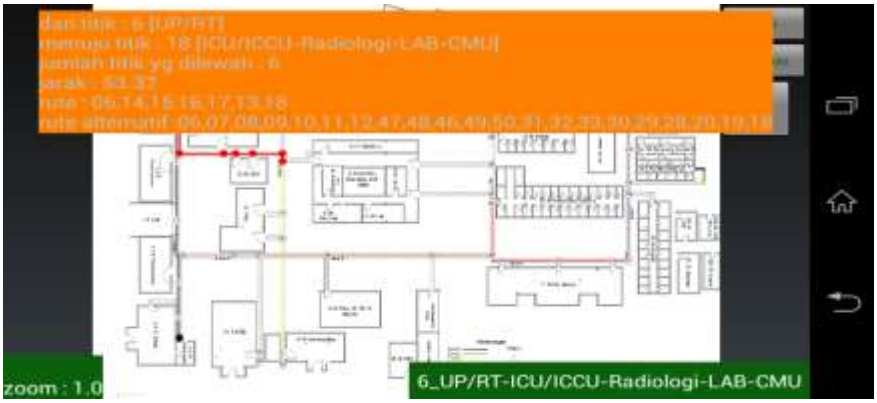
Gambar 3.13 Tampilan Cari Ruang



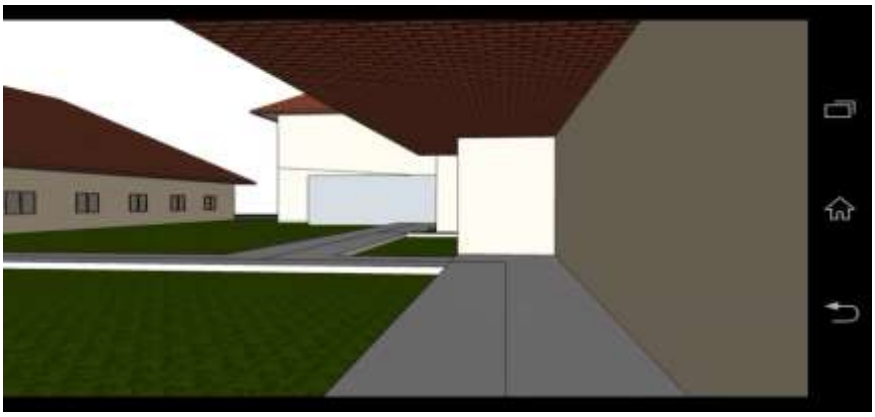
Gambar 3.14 Tampilan Scan Qrqode Titik Keberadaan



Gambar 3.15 Tampilan Input Ruang Tujuan



Gambar 3.16 Tampilan Info Rute Terpendek



Gambar 3.17 Tampilan Video Petunjuk Ruang



Gambar 3.18 Tampilan *Input* Info Ruang



Gambar 3.19 Tampilan Info Ruang



Gambar 3.20 Tampilan Video3D RSUD Gunung Jati



Gambar 3.21 Tampilan Petunjuk Penggunaan

Tabel 11 Kesimpulan Analisis Aplikasi Sejenis Terhadap Aplikasi yang Dibangun

No	Nama aplikasi	Basis aplikasi	Grafis	Metode pencarian	Ruang Lingkup	Konten
1	Aplikasi Pencarian Rute Terpendek untuk Taksi	*Website*	2 Dimensi	Algoritma *Floyd Warsahall*	Kota Bandung meliputi : Terminal Leuwi Panjang sampai Dipatiukur	Terdapat pilihan untuk mengetahui rute terpendek dan rute tercepat menuju tujuan.
2	Aplikasi Informasi Rute Terpendek Rumah Sakit, Rumah Bersalindan Puskesmas di Palembang	*Mobile*	2 Dimensi	Algoritma *Steepest Ascent Hill Climbing*	Kota Palembang	Terdapat pilihan untuk mengetahui informasi rumah sakit, rumah bersalin, puskesmas, dan mengetahui rute terpendek dalam bentuk peta.
3	Aplikasi Denah 3D di Rumah Sakit Umum Daerah Gunung Jati	*Mobile*	3 Dimensi	Algoritma *Floyd Warsahall*	Rumah Sakit Umum Daerah Gunung Jati Kota Cirebon	Terdapat pilihan untuk mengetahui rute terpendek dan menampilkan petunjuk denah dalam bentuk 3D

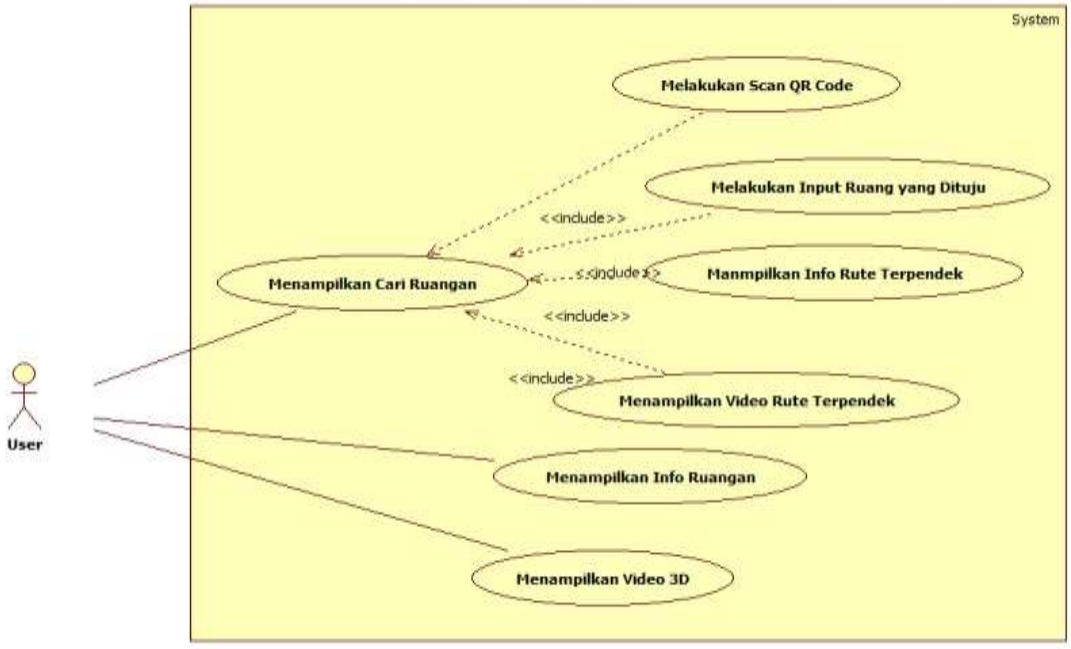
3.1.5 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional ini dimodelkan dengan menggunakan UML (*Unified Modeling Language*). Dimana tahap-tahap perancangan yang dilakukan dalam membangun aplikasi denah 3D Rumah Sakit Umum Daerah Gunung Jati dalam pencarian rute terpendek antara lain *Use Case Diagram, Class Diagram, Sequence Diagram*, dan *Activity Diagram*.

3.1.5.1 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 informasi yang akan dibuat.

Berikut ini adalah perancangan proses-proses yang terdapat pada aplikasi Denah 3D RSUD Gunung Jati *Use Case* Diagram yang dapat dilihat berikut ini:



Gambar 3.22 *Use Case Diagram*

3.1.5.1.1 Definisi Actor

Definisi actor untuk menjelaskan actor yang terdapat pada *Use Case* Diagram.Definisi actor dijelaskan pada tabel 3.5.

Tabel 3.12 Definisi *Actor*

No	Actor	*Deskripsi*
1	User	Orang yang menggunakan aplikasi

3.1.5.1.2 Definisi Use Case

Definisi *Use Case* menjelaskan fungsi use case yang terdapat pada *Use Case* Diagram. Definisi *Use Case* dijelaskan pada tabel 3.7.

Tabel 3.13 Definisi *Use Case*

No	*Use case*	*Deskripsi*
1	Cari Ruang	Proses untuk melakukan pencarian ruang, rute terdekat, dan menampilkan 3D .
2	Info Ruang	Proses dimana menampilkan informasi mengenai ruangan—ruangan RSUD Gunung Jati.
3	Video 3D	Proses menampilkan denah 3D rumah sakit.
4	Scan Qrqode	Proses untuk melakukan scan qrqode menentukan titik awal keberadaan.
5	Input Ruang Tujuan	Proses untuk menampilkan titik keberadaan tujuan atau menentukan ruangan yang dituju.
6	Video Rute Terpendek	Proses untuk menampilkan petunjuk ruangan denah 3D
7	Info Rute Terpendek	Proses untuk menampilkan informasi

		ruang yang dilewati pengguna dari titik awal menuju titik tujuan pada denah

3.1.5.2 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.

Tabel 3.14 Skenario Info Ruang

Nama *Use Case*	Info Ruang	
Nomor	1	
Aktor	User	
Kondisi Awal	User berada ditampilan menu aplikasi	
Reaksi Aktor		**Reaksi Sistem**
1. User memilih tombol menu Info Ruang 3. User memilih nama ruang yang diinginkan dan menekan tombol menu ya		2. Sistem menampilkan nama ruang 4. Sistem menampilkan informasi ruang berbentuk message
Kondisi Akhir	Aplikasi menampilkan Info Ruang	
Pengecualian	Aplikasi tidak berjalan baik	

Tabel 3.15 Skenario Cari Ruang

Nama *Use Case*	Cari ruang	
Nomor	2	
Aktor	User	
Kondisi Awal	User berada ditampilan menu aplikasi	
Reaksi Aktor		**Reaksi Sistem**
1. User memilih cari ruang 2. User memilih tombol cari ruang		3. Sistem menampilkan tampilan cari ruang
Kondisi Akhir	Aplikasi menampilkan Cari Ruang	
Pengecualian	Aplikasi tidak berjalan baik	

Tabel 3.16 Skenario Video 3D

Nama *Use Case*	Video 3D	
Nomor	3	
Aktor	User	
Kondisi Awal	User berada ditampilan menu aplikasi	
Reaksi Aktor		**Reaksi Sistem**
1. User memilih tombol Video 3D		

2. User menekan tombol video 3D	3. Sistem menampilkan video denah 3D
Kondisi Akhir	Aplikasi menampilkan Video 3D
Pengecualian	Aplikasi tidak berjalan baik

Tabel 3.17 Skenario Scan Qrqode

Nama *Use Case*	Scan Qrqode
Nomor	4
Aktor	User
Kondisi Awal	User berada ditampilan menu Cari Ruangan
Reaksi Aktor	**Reaksi Sistem**
1. User memilih tombol menu Scan Qrqode	2. Sistem menampilkan menu Scan Qrqode 3. Sistem melakukan inisialisasi kamera 4. Sistem melakukan scan qrqode

5. User melakukan scan qrqode	6. Sistem menampilkan keberadaan awal
Kondisi Akhir	Aplikasi menampilkan Scan Qrqode
Pengecualian	Aplikasi tidak berjalan baik

Tabel 3.18 Skenario Input Ruang Tujuan

Nama *Use Case*	Input Ruang Tujuan
Nomor	5
Aktor	User
Kondisi Awal	User berada ditampilan menu cari ruangan
Reaksi Aktor	**Reaksi Sistem**
1. User memilih tombol menu input ruang tujuan	2. Sistem menampilkan tampilaninput ruang tujuan
3. User memilih tombol menuju	4. Sistem menampilkan nama ruang tujuan
5. User memilih ruang yang dituju	6. Sistem menampilkan titik tujuan
Kondisi Akhir	Aplikasi menampilkan Input Ruang Tujuan
Pengecualian	Aplikasi tidak berjalan baik

Tabel 3.19 Skenario Info Rute Terpendek

Nama *Use Case*	Info Rute Terpendek	
Nomor	6	
Aktor	User	
Kondisi Awal	User berada ditampilan menu cari ruangan	
Reaksi Aktor		**Reaksi Sistem**
User memilih menu Info Rute Terpendek 2. User menekan tombol Info Rute Terpendek		3. Sistem menampilkan rute yang ditempuh dari titik awal menuju ruangan yang dituju pada denah
Kondisi Akhir	Aplikasi menampilkan Info Rute Terpendek	
Pengecualian	Aplikasi tidak berjalan baik	

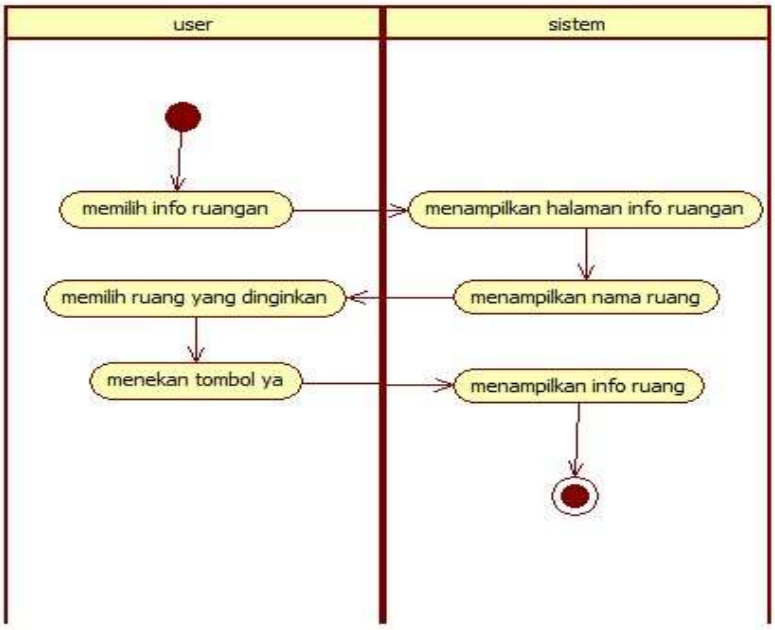
Tabel 3.20 Skenario Video Rute Terpendek

Nama *Use Case*	Video Rute Terpendek	
Nomor	7	
Aktor	User	
Kondisi Awal	User berada ditampilan menu cari ruangan	
Reaksi Aktor		**Reaksi Sistem**

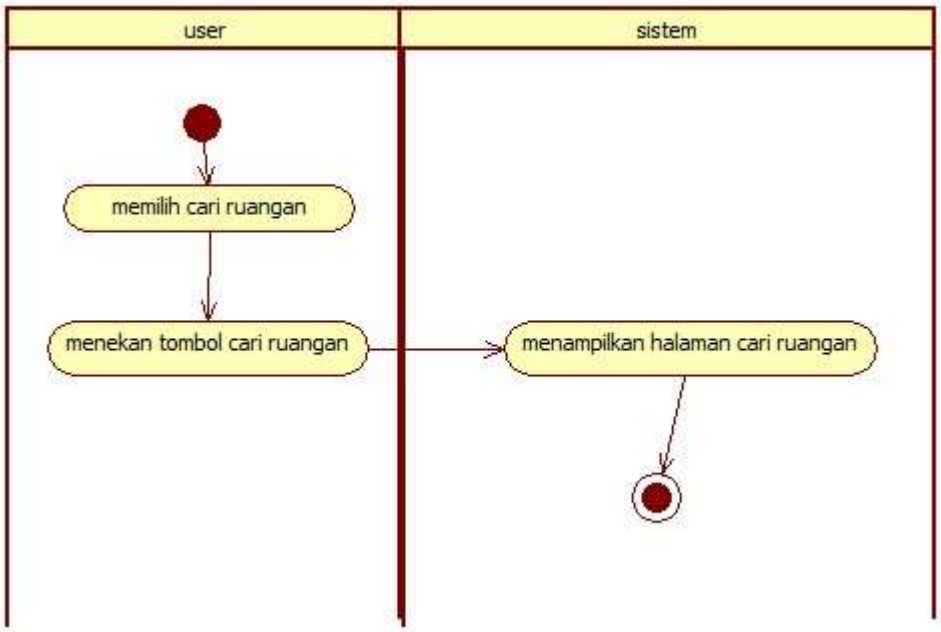
1. User memilih menu video rute terpendek 2. User menekan tombol video rute terpendek	3. Sistem menampilkan video penunjuk arah menuju ruangan yang dituju
Kondisi Akhir	Aplikasi menampilkan video rutte terpentek
Pengecualian	Aplikasi tidak berjalan baik

3.1.5.3 Activity Diagram

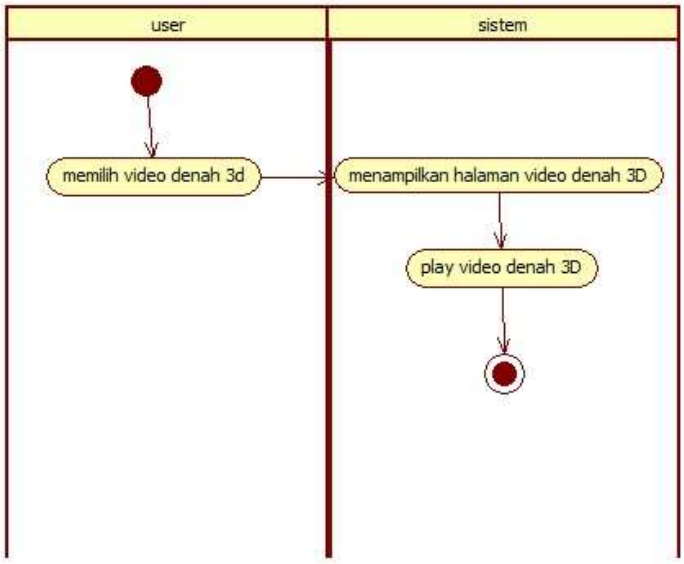
Activity diagram memiliki pengertian yaitu lebih fokus kepada menggambarkan proses bisnis atau sebuah sistem dan urutan aktivitas dalam sebuah proses.



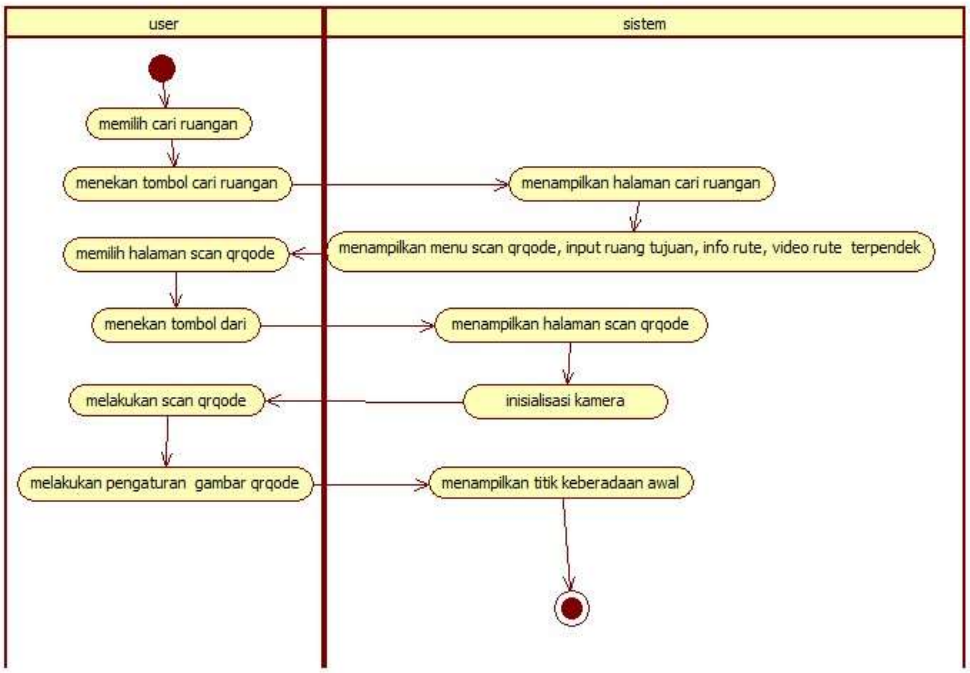
Gambar 3.23 *Activity Diagram* Info Ruang



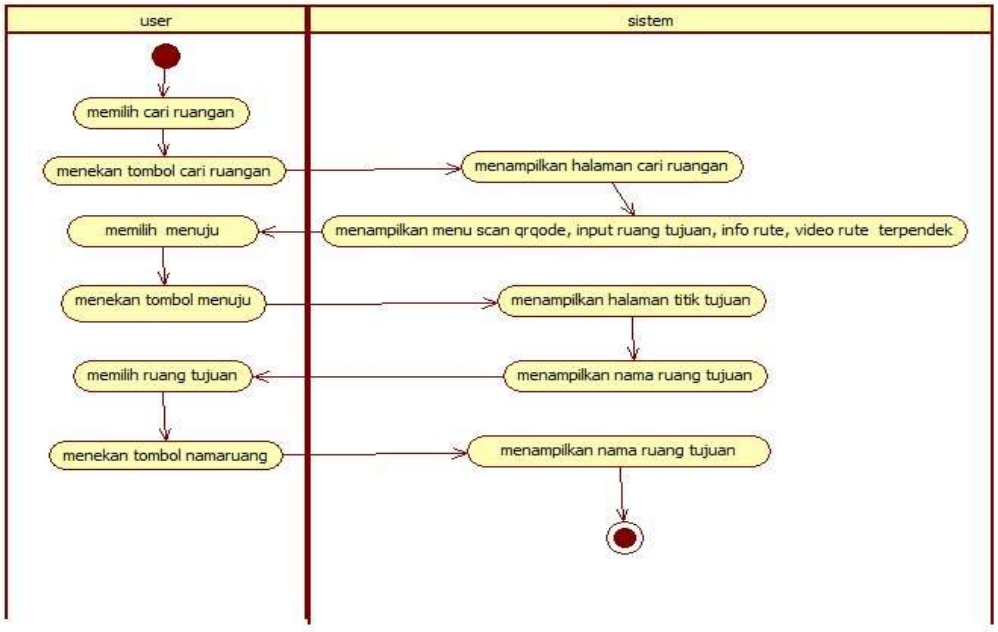
Gambar 3.24 *Activity Diagram* Cari Ruang



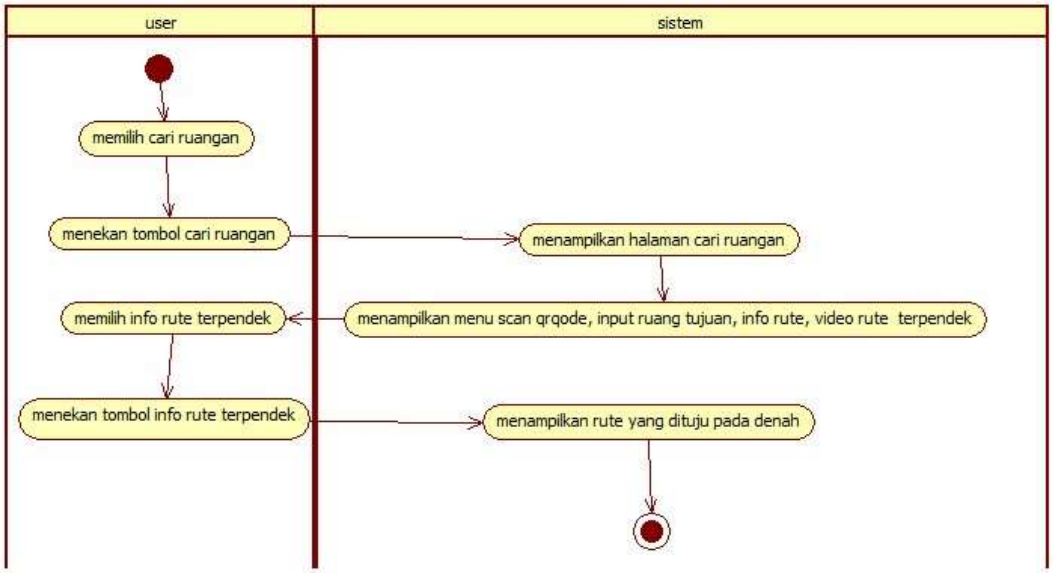
Gambar 3.25 *Activity Diagram Video 3D*



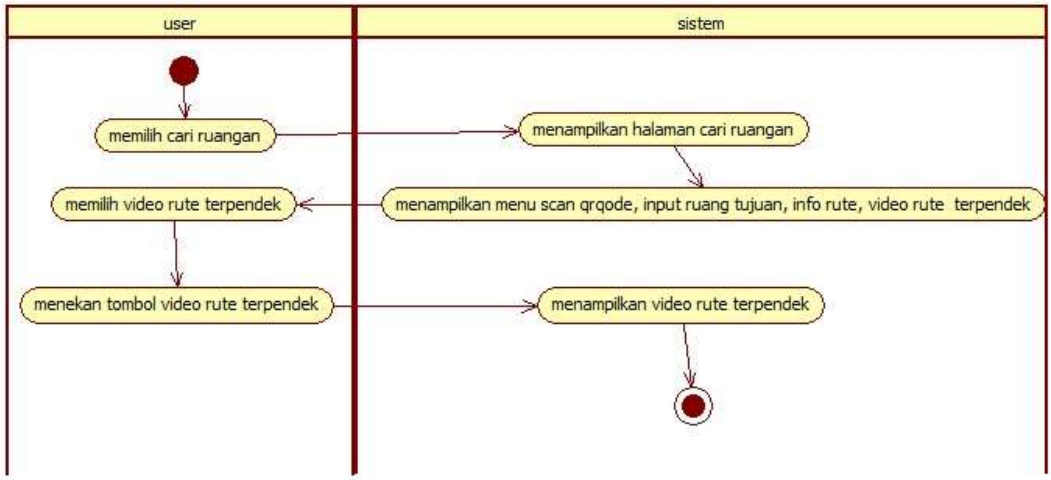
Gambar 3.26 *Activity Diagram* Scan Qrqode



Gambar 3.27 *Activity Diagram* Input Ruang Tujuan



Gambar 3.28 *Activity Diagram* Info Rute Terpendek

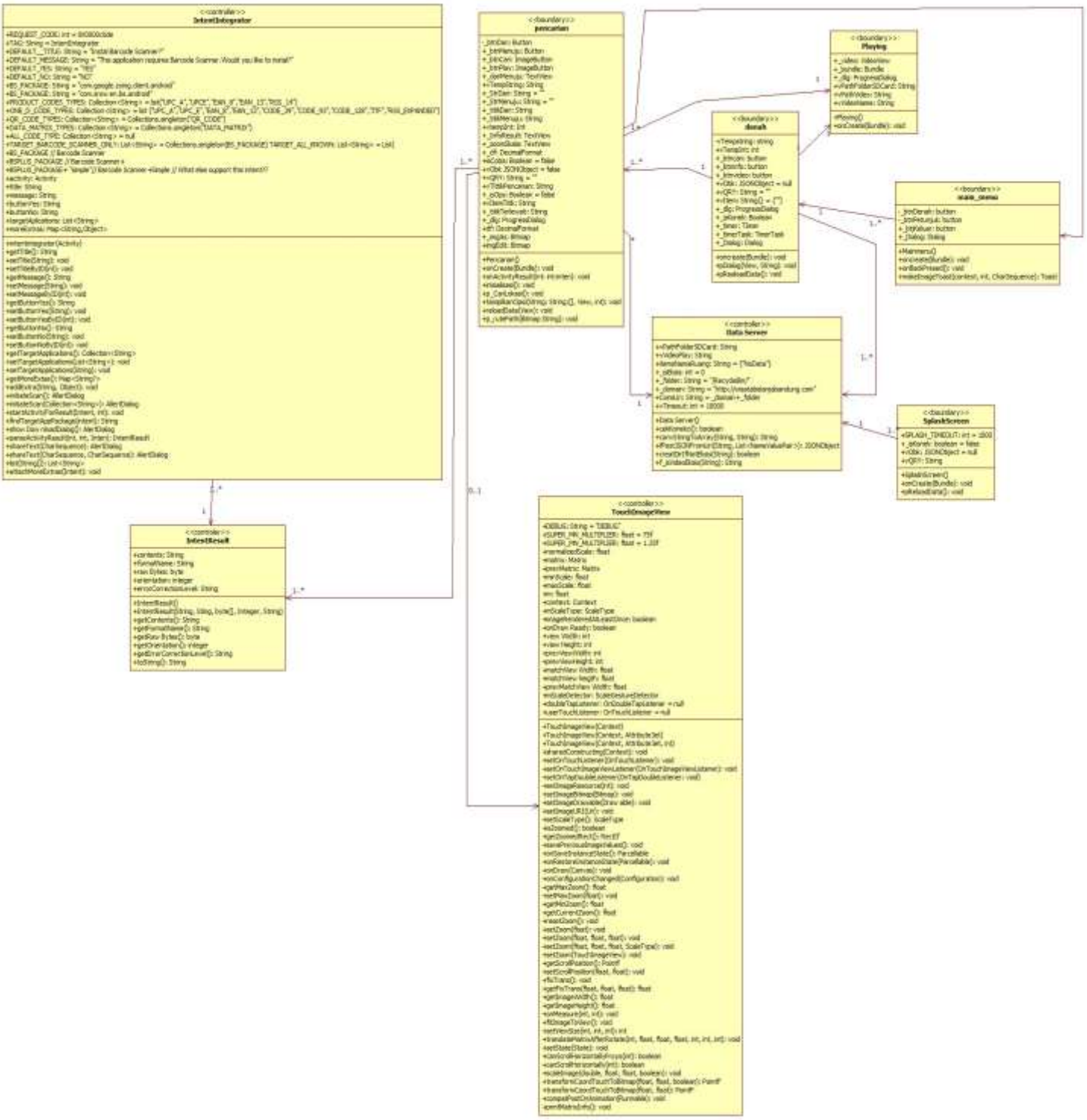


Gambar 3.29 *Activity Diagram* video rute terpendek

3.1.5.4 Class Diagram

Class Diagram menggambarkan tipe-tipe objek dalam system 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.



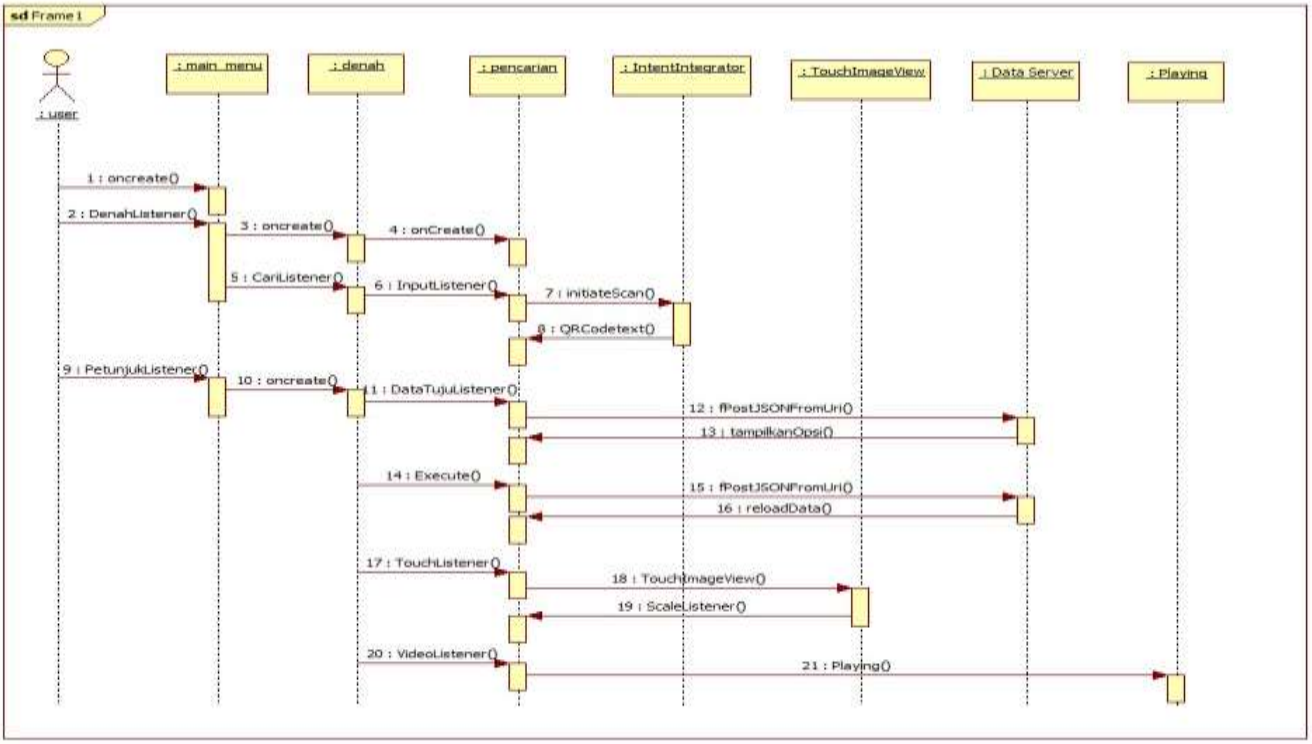
Gambar 3.30 *Class Diagram* Aplikasi Denah 3D RSUD Gunung Jati

Tabel 3.21 *Class Diagram*

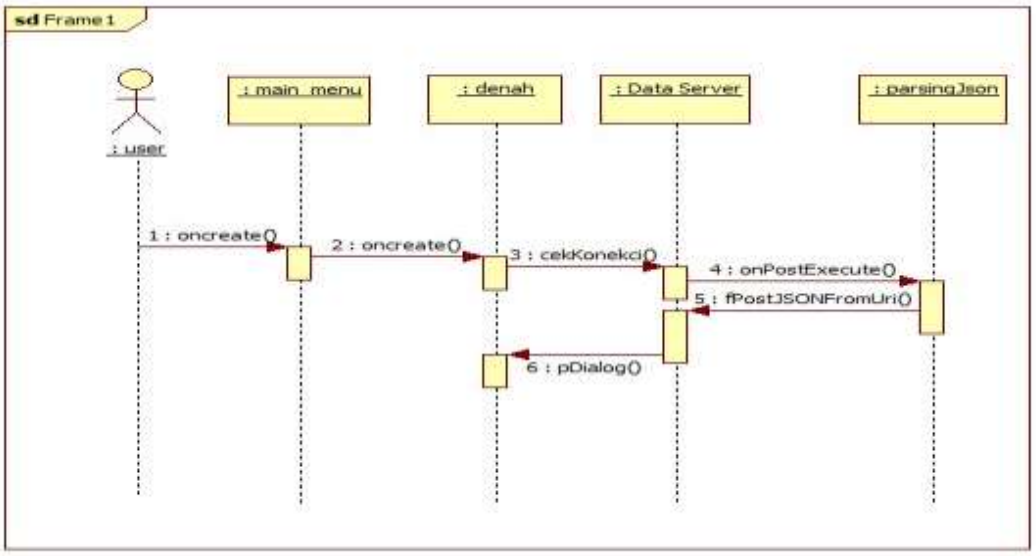
Nama *Class*	Jenis *Class*	Fungsionalitas *Class*
Main menu	*Boundary*	*Class* untuk mengatur tampilan menu
Denah	*Boundary*	*Class* untuk mengatur 3D dan peta
Pencarian	*Boundary*	*Class* untuk mencari rute
IntenIntegrator	*Controler*	*Class* untuk menjalankan fungsi kamera
IntentResult	*Controler*	*Class* menampilkan hasil kamera
Data server	*Controler*	Kumpulan *Class* yang berfungsi sebagai library method-method data atau parsing
SplashScreen	*Boundary*	*Class* untuk memvalidasi hasil dari database
Playing	*Boundary*	*Class* untuk menampilkan video
TouchImageView	*Controler*	*Class* yang mengatur tampilan resolusi gambar

3.1.5.5 Sequence Diagram

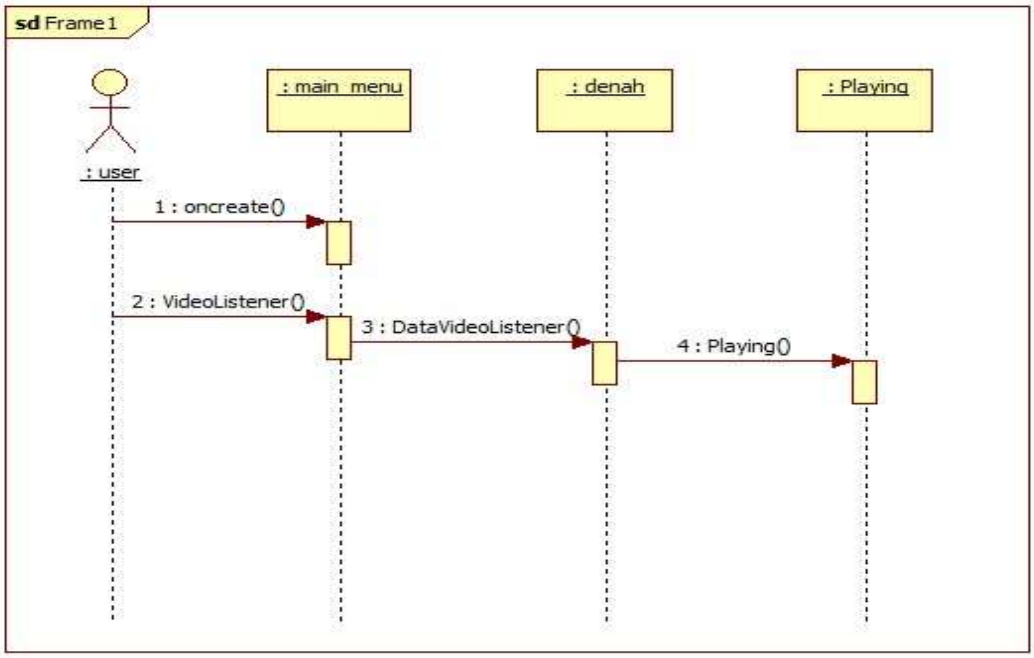
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi diantara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu.



Gambar 3.31 *Sequence Diagram* Cari Ruangan



Gambar 3.36 *Sequence Diagram* Info Ruangan

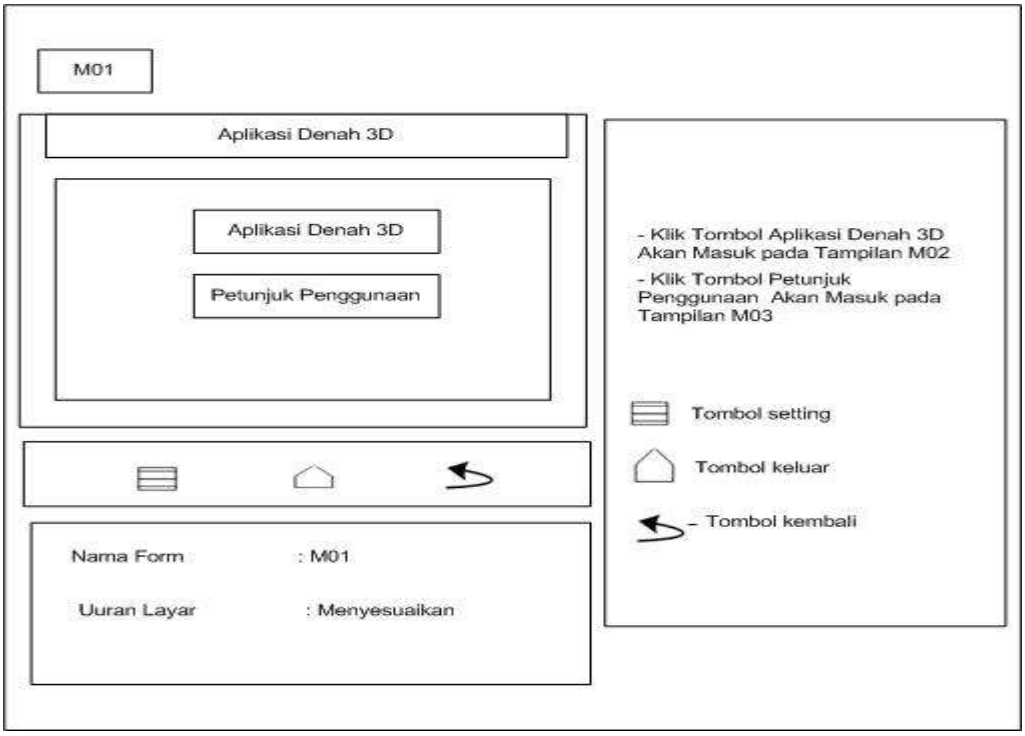


Gambar 3.37 *Sequence Diagram* Video 3D

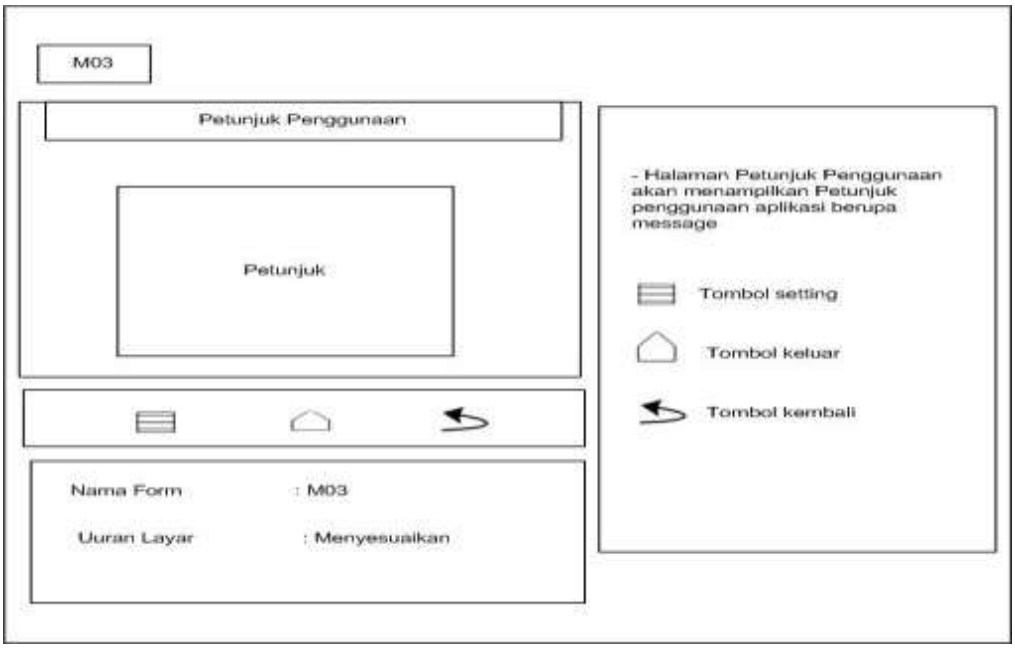
3.1.5 Perancangan

3.1.5.1 Perancangan Antarmuka

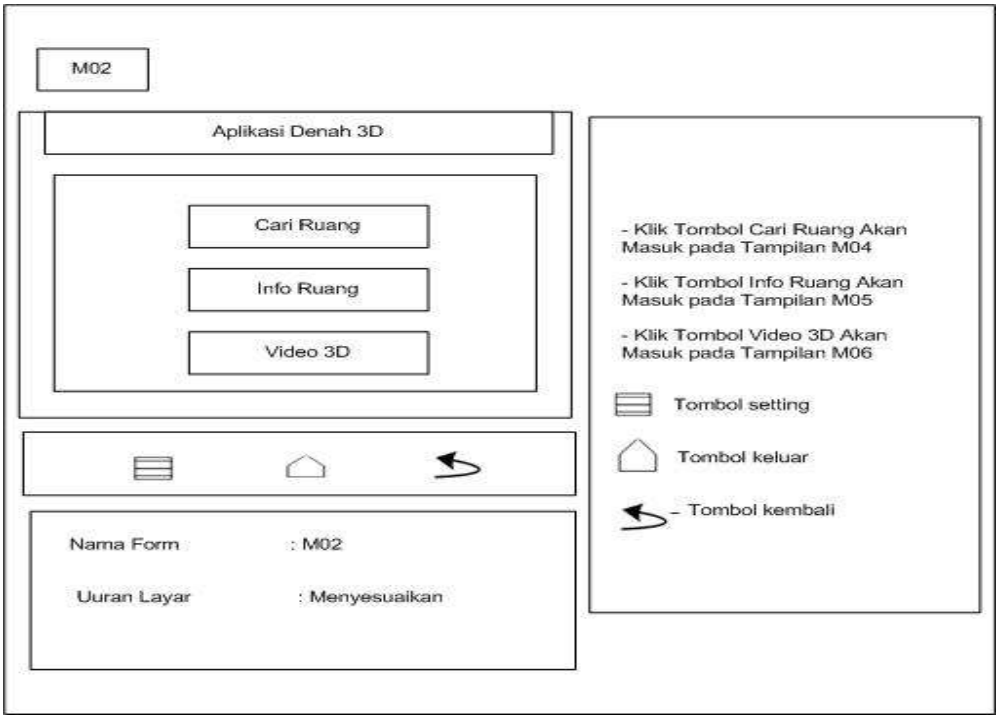
Desain yang dibuat berupa aplikasi berbasis mobile yang mudah untuk digunakan oleh pengguna, serta informasi yang dihasilkan aplikasi tersebut dapat dimengerti oleh pengguna.Perancangan antarmuka bertujuan untuk memberikan gambaran aplikasi yang dibuat. Objek tiga dimensi merupakan objek yang akan ditampilkan.



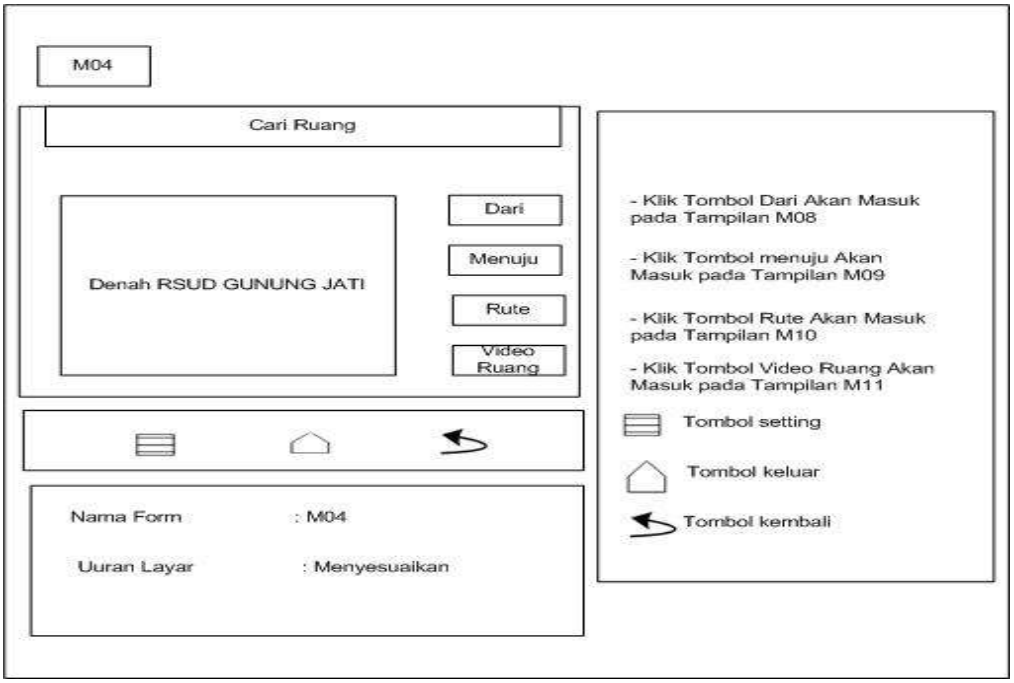
Gambar 3.38 Antar Muka Menu Aplikasi Denah 3D



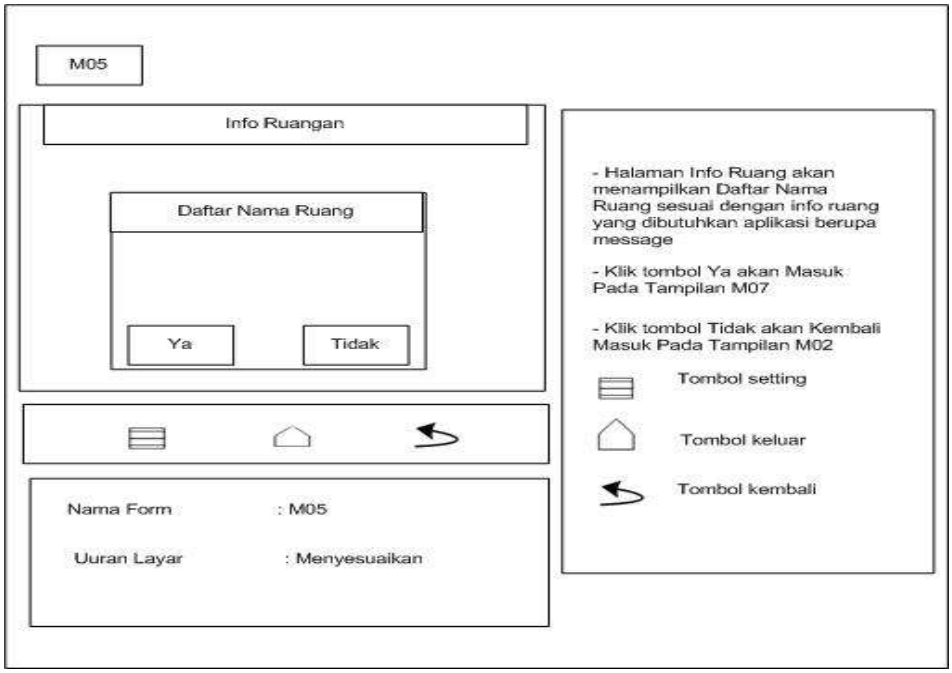
Gambar 3.39 Antar Muka Menu Cara Penggunaan



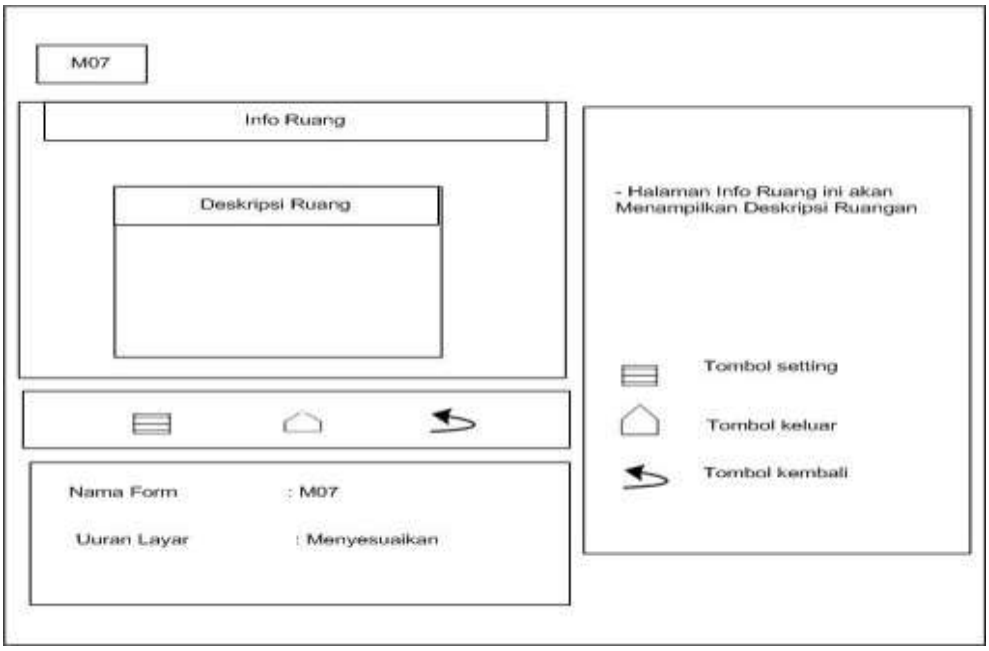
Gambar 3.40 Antar Muka Menu Aplikasi Denah 3D



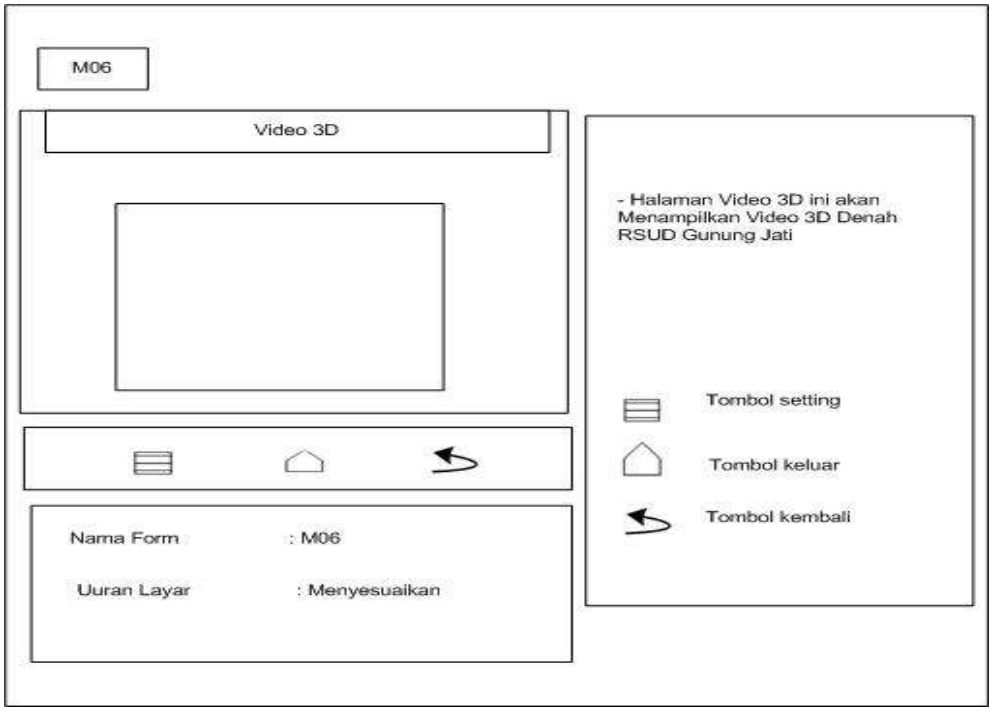
Gambar 3.41 Antar Muka Menu Cari Ruangan



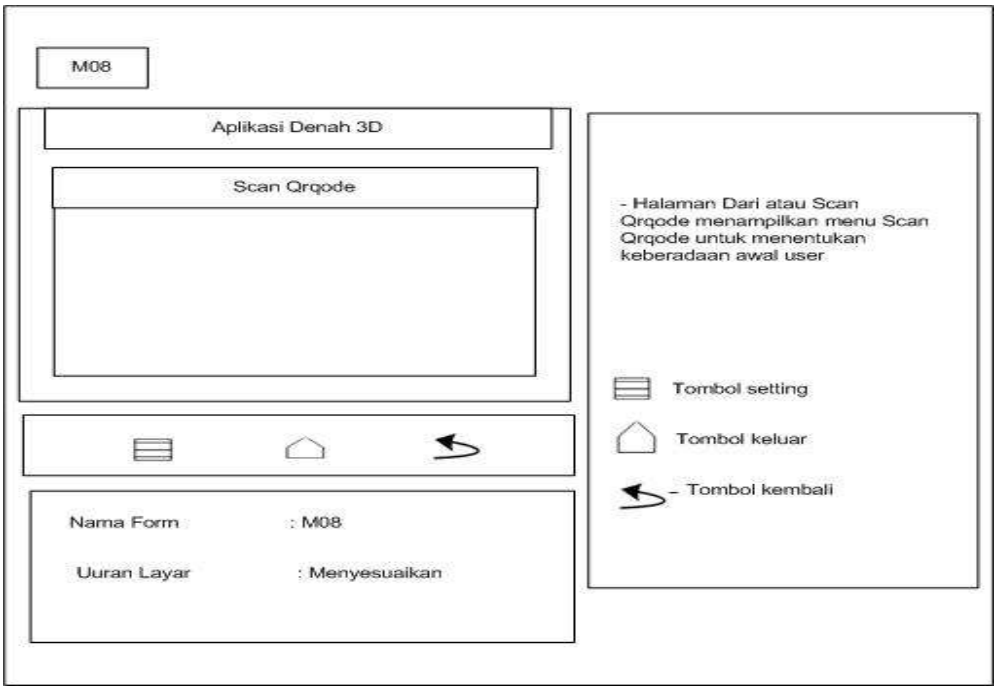
Gambar 3.42 Antar Muka Menu Info Ruangan



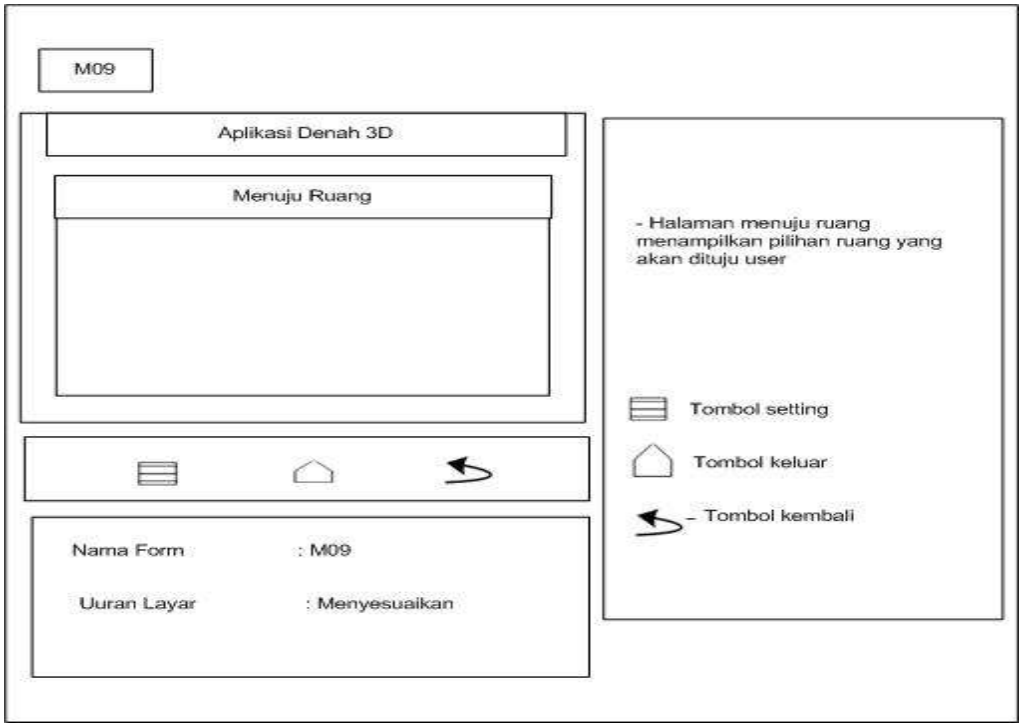
Gambar 3.43 Antar Muka Menu Info Ruangan



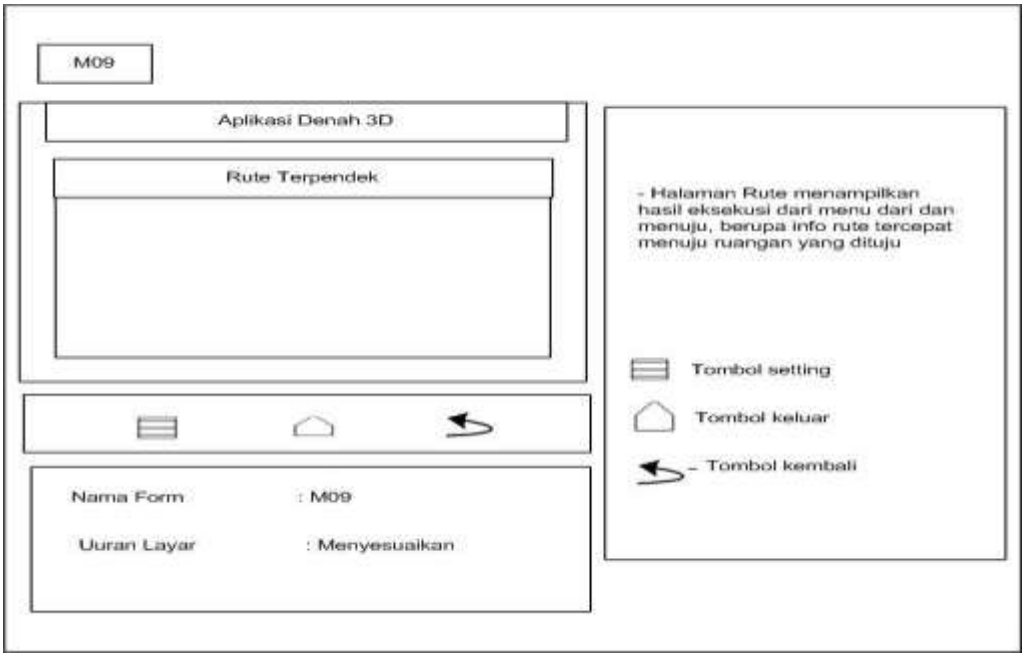
Gambar 3.44 Antar Muka Menu Video 3D



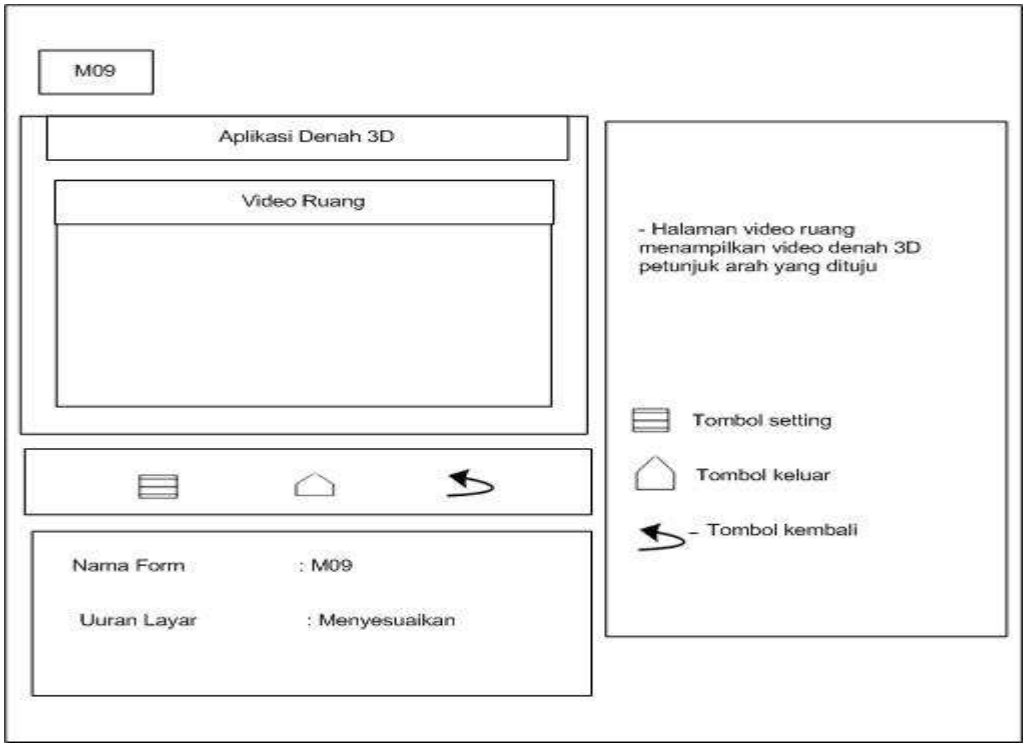
Gambar 3.45 Antar Muka Menu Scan Qrqode



Gambar 3.46 Antar Muka Menu Input Ruang Tujuan



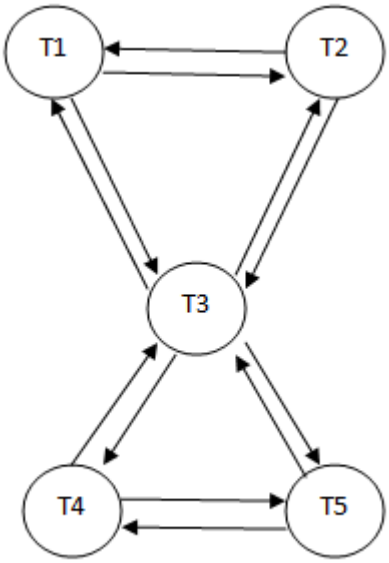
Gambar 3.47 Antar Muka Menu Info Rute Terpendek



Gambar 3.48 Antar Muka Menu Video Rute Terpendek

3.1.6 Jaringan Semantik

Jaringan semantik adalah diagram yang menggambarkan aliran-aliran menu dan pesan dalam sebuah program. Berikut ini adalah jaringan semantik pada sistem yang dibangun.



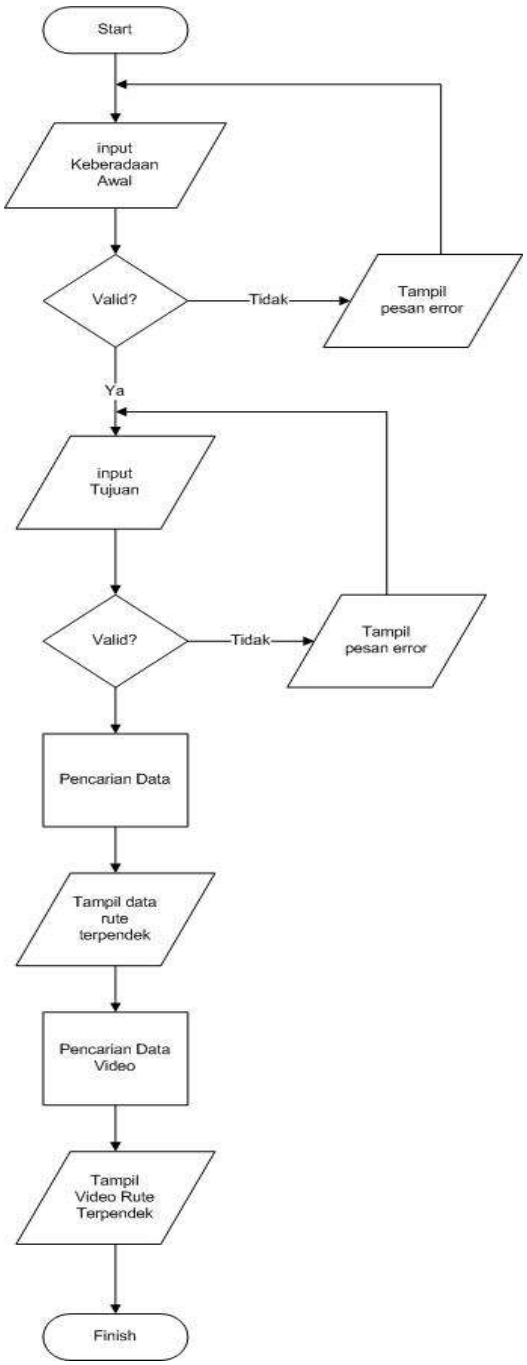
Gambar 3.49 Jaringan Semantik Aplikasi Denah 3D

3.1.5 Perancangan *Method*

Perancangan komponen *method* merupakan perancangan yang dibuat setelah perancangan antarmuka dan jaringan semantik. Perancangan ini berfungsi untuk mendeskripsikan *method-method* yang berada di dalam aplikasi Denah 3D Rumah Sakit Umum Daerah Gunung Jati. Adapun method-method yang terdapat dalam aplikasi Denah 3D Rumah Sakit Umum Daerah Gunung Jati yang dibangun adalah sebagai berikut :

3.1.7.1 *Method* Cari Ruang

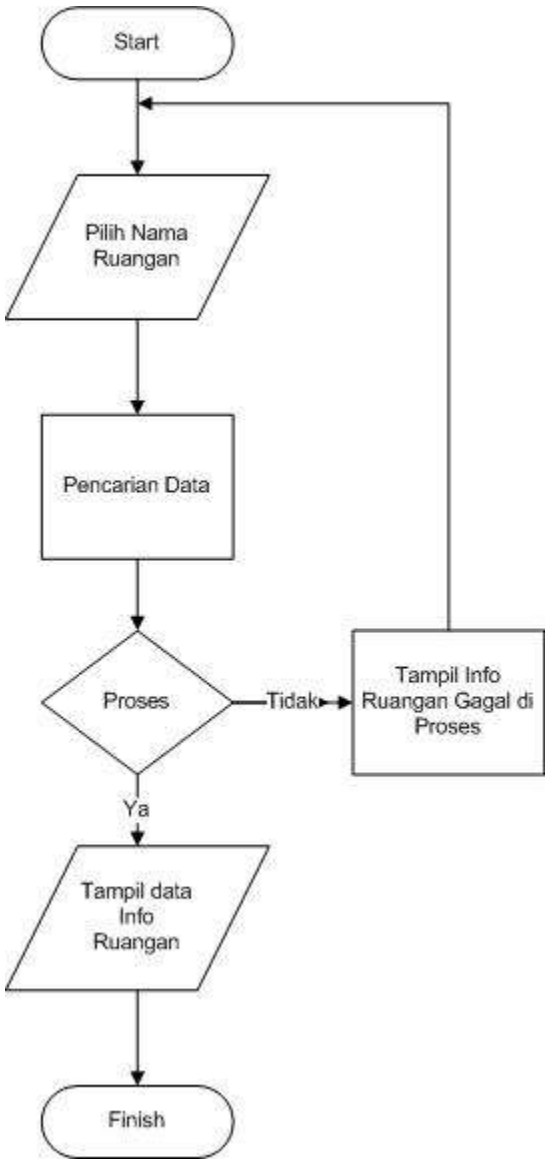
Method Cari Ruang akan dipanggil pada saat user melakukan proses cari ruang, seperti terlihat pada gambar 3.46.



Gambar 3.46 *Flow Chart* Cari Ruang

3.1.7.2 *Method* Info Ruang

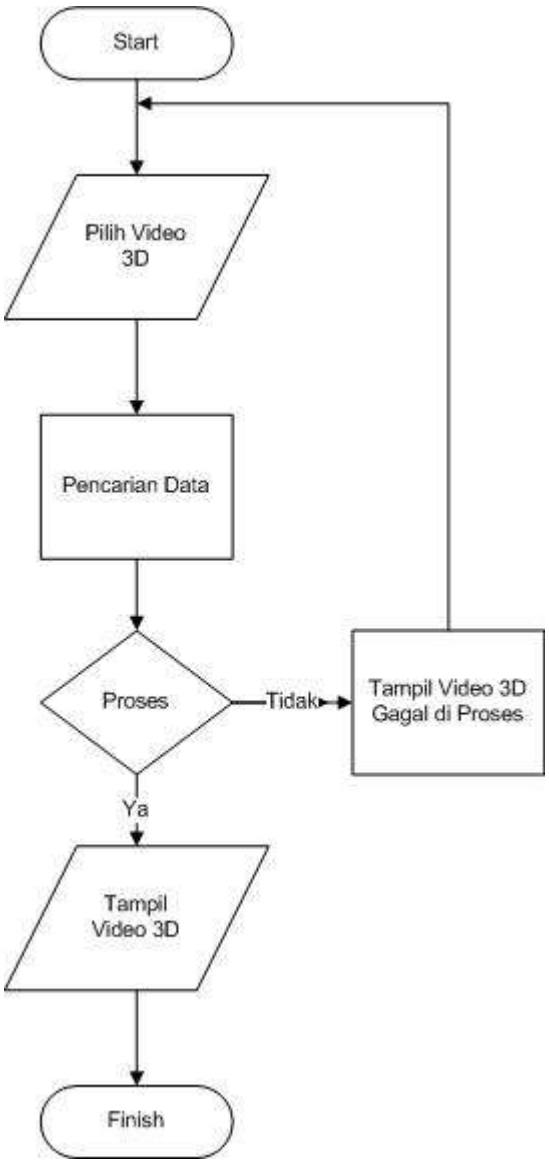
Method info ruang akan dipanggil pada saat user melakukan proses mencari info ruangan, seperti terlihat pada gambar 3.47.



Gambar 3.47 *Flow chart* Info Ruang

3.1.7.3 *Method* Video 3D

Method video 3d akan dipanggil pada saat user melakukan proses menampilkan video 3d seperti terlihat pada gambar 3.48.



Gambar 3.48 *Flow Chart* Video 3D