BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem (*systems analysis*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahanpermasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikanperbaikannya.

3.1.1 Analisis Masalah

Setelah melakukan penelitian berupa data seperti gambar yang diperoleh, jarak ruangan ke ruangan dan menanyakan langsung ke pegawai didapatkan masalah yaitu belum adanya sistem pemandu lokasi untuk mahasiswa baru mencari setiap ruangan yang ada. Proses pencarian ruangan saat ini masih mengandalkan informasi atau keterangan yang tertera pada dinding kampus atau bertanya kepada petugas, pegawai kampus dan mahasiswa. Sehingga mahasiswa kesana kemari mencari ruangan yang dicari dan mengakibatkan pencarian ruangan menjadi lama.

Selain itu informasi ruangan yang didapat belum tentu benar, mengakibatkan mahasiswa baru salah dalam mencari ruangan. Adapun calon mahasiswa baru dan orangtuanya yang ingin melihat keadaan sekitar kampus harus datang ke kampus terlebih dahulu. Karena belum adanya gambaran sekitar kampus dalam bentuk gambar 360°. Sehingga menyulitkan bagi mereka yang berada jauh dari kampus UNIKOM atau di luar kota bandung.

Berdasarkan permasalahan diatas dibangunlah sebuah aplikasi pemandu lokasi Unviersitas Komputer Indonesia (UNIKOM) berbasis web, dimana aplikasi ini akan memberikan panduan kepada pengguna dalam mencari setiap ruangan yang ada di UNIKOM dalam bentuk gambar 360° dan bangunan 3D. Selain itu diberikan pula jalur terpendek untuk menuju ke ruangan yang dimaksud juga

informasi mengenai ruangan tersebut. Dimana algoritma penghitung jalur terpendek yaitu algoritma *dijkstra*, algoritma *floyd*, dan algoritma *two queues*. Dimana 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.

3.1.2 Analisis Virtual Tour 360

Aplikasi pemandu lokasi di UNIKOM ini bertujuan mempermudah pengguna dalam memperoleh informasi tentang UNIKOM. Pada aplikasi ini terdapat titik-titik yang merepresentasikan lokasi di UNIKOM. Penambahan Virtual Tour 360 dilakukan pada titik tersebut. Pada setiap titik lokasi diambil foto-foto yang mewakili keadaan sekitar yang kemudian dijadikan gambar panorama. Menurut Vinod et al. (2007) gambar panorama menciptakan foto realistis, tiga dimensional, dan dapat dinavigasi. Terdapat beberapa jenis gambar panorama, antara lain planar atau flat, cylinder, sphere, dan cube. Planar adalah gambar panorama yang dilihat tanpa perspektif tertentu atau datar. Cylinder adalah gambar panorama silindris yang dibuat untuk diilihat kurva di sekeliling bagian dalam silinder. Sphere adalah tipe gambar panorama dengan format equirectangular yang menggambarkan dengan pasti keadaan gambar axis horizontal secara 360 derajat dan axis vertikal secara 180 derajat. Cube adalah tipe gambar panorama berbentuk kubus yang bagian atas dan bawahnya dapat dilihat. Gambar panorama tipe kubus ini memiliki enam gambar di permukaannya (Brosz & Samavati 2010). Jenis gambar panorama yang dilakukan pada aplikasi ini ialah cylinder. Contoh gambar cylinder sebagai berikut:



Gambar 3.1 cylinder halaman depan unikom

Ada tiga pendekatan yang bisa digunakan untuk menghasilkan panorama virtual tour, yakni fotografi stitching, berbasis video, dan memanfaatkan perangkat

lunak khusus. Pedekatan pertama umumnya lebih disukai karena cenderung lebih praktis dalam implementasinya. Pada pedekatan ini, teknik pengambilan gambar panorama menggunakan kamera meiliki peran penting, agar gambar bisa disusun dengan baik, maka pengambilan gambar dilakukan secara berpotongan (overlap). Contoh pengambilan gambar diperlihatkan pada gambar 3.23.



Gambar 3.1 pengambilan gambar dilakukan overlap

Pengambilan gambar dengan teknik overlap ini dilakukan berulang sesuai kebutuhan. Apabila ingin menyahikan panorama dengan sudut pandang hinhha 360, maka pemotretan dilakukan penuh secara menyeluruh. Langkah selanjutnya, potongan-potongan gambar overlap tadi akan di-stitch menjadi sebuah gambar utuh yang merepresentasikan semua sudut panorama yang dimana menggunakan aplikasi photoshop menggabungkannya.

Pada tahap ini dilakukan analisis kebutuhan dalam pembuatan Virtual Tour 360 berupa sejumlah foto yang dapat menjadi gambar panorama, software yang digunakan untuk penggabungkan foto-foto tersebut menjadi gambar panorama dan menentukan perangkat pembuatan Virtual Tour 360 itu sendiri. Virtual Tour 360 yang telah dirancang kemudian digabungkan dengan pemandu lokasi di UNIKOM. Penggabungannya dilakukan dengan cara menambahkan beberapa fungsi pada ActionScript, sedangkan pembuatan Virtual Tour 360 itu sendiri menggunakan Javascript berupa jQuery plugin. JQuery merupakan suatu library Javascript yang

menekankan bagaimana interaksi antara Javascript dan HTML. Pada perkembangannya, jQuery tidak sekedar sebagai library Javascript, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak developer web yang menggunakannya. JQuery memiliki slogan "Write less, do more" yang maksudnya adalah kesederhanaan dalam penulisan kode, tetapi dengan hasil yang lebih banyak (Chaffer & Karl 2007). Pada tahap ini juga dilakukan penentuan koordinat hotspot, derajat perputaran dan posisi layar pada pada setiap tempatnya. Koordinat hotspot disesuaikan dengan arah tempat yang akan dituju. Contoh gambarnya sebagai berikut:



Hostspot yang berwarna biru aqua berbentuk persegi memiliki 4 buah titik yaitu x,x1,y,y1. Dalam gambar mendapatkan sebuah angka menjadi 3600,3800,240,340. Sedangkan posisi layar yang berwarna hijau berbentuk persegi yang memiliki titik tengah yang dimana titik tengah yaitu x dan titik y sudah langsung pertengahan gambar. Dalam gambar mendapatkan sebuah angka menjadi 2500,250. Tetapi didalam aplikasi titik y tidak ditampilkan karena sudah langsung dapat.

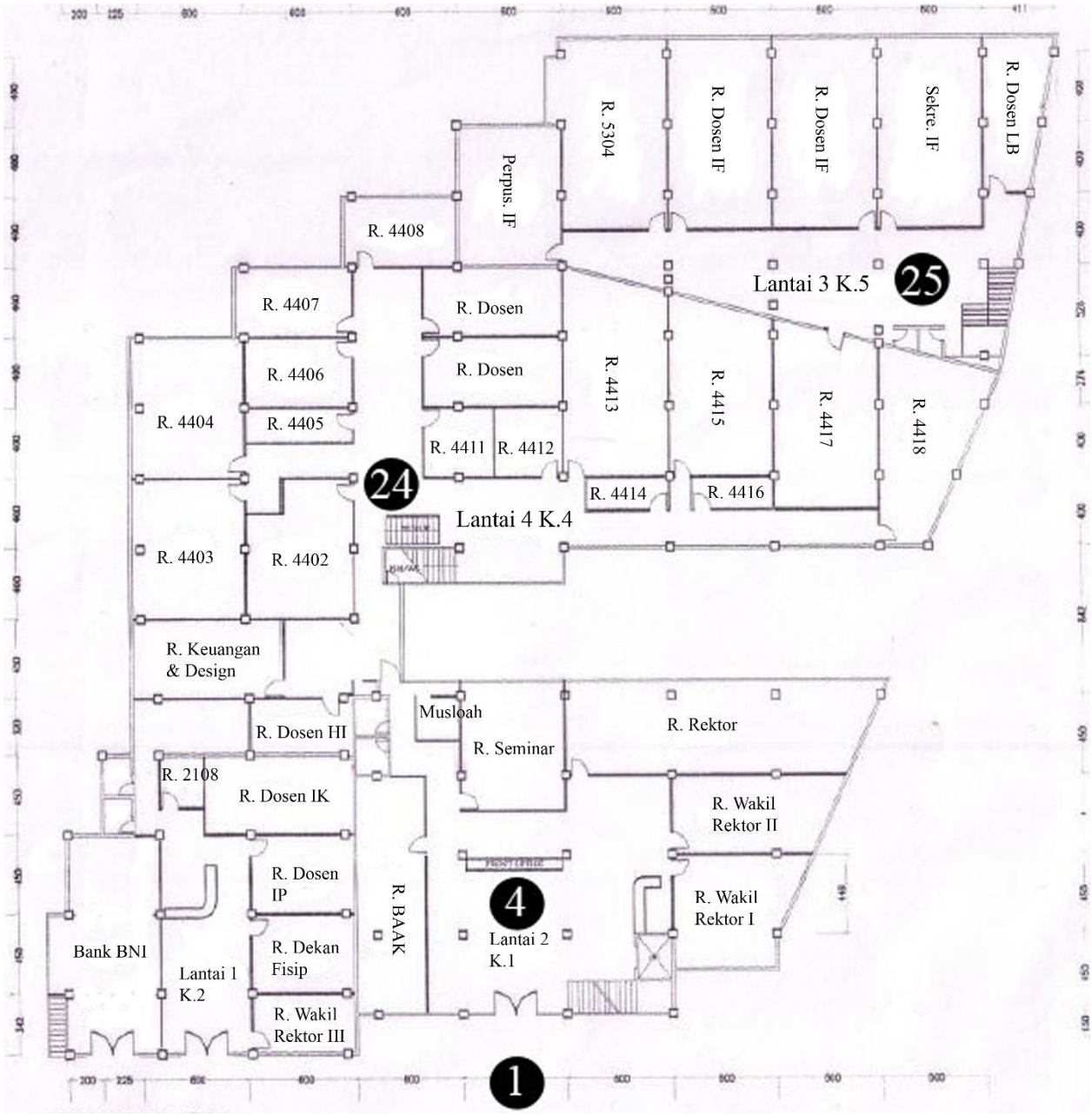
3.1.3 Analisis Algoritma Dijkstra

Algoritma Dijkstra adalah sebuah algoritma untuk memecahkan permasalahan jalur terpendek (*shortest path problem*) pada sebuah graf berarah (*directed graf*) atau graf tidak berarah (*undirected graf*) dengan bobot-bobot sisi (*edge weights*) yang bernilai tidak negatif. Bila node dari sebuah graf melambangkan kota-kota dan bobot sisi (*edge weights*) melambangkan jalur antara kota-kota tersebut, maka algoritma *Dijkstra* dapat digunakan untuk menemukan jalur terpendek antara dua tempat.

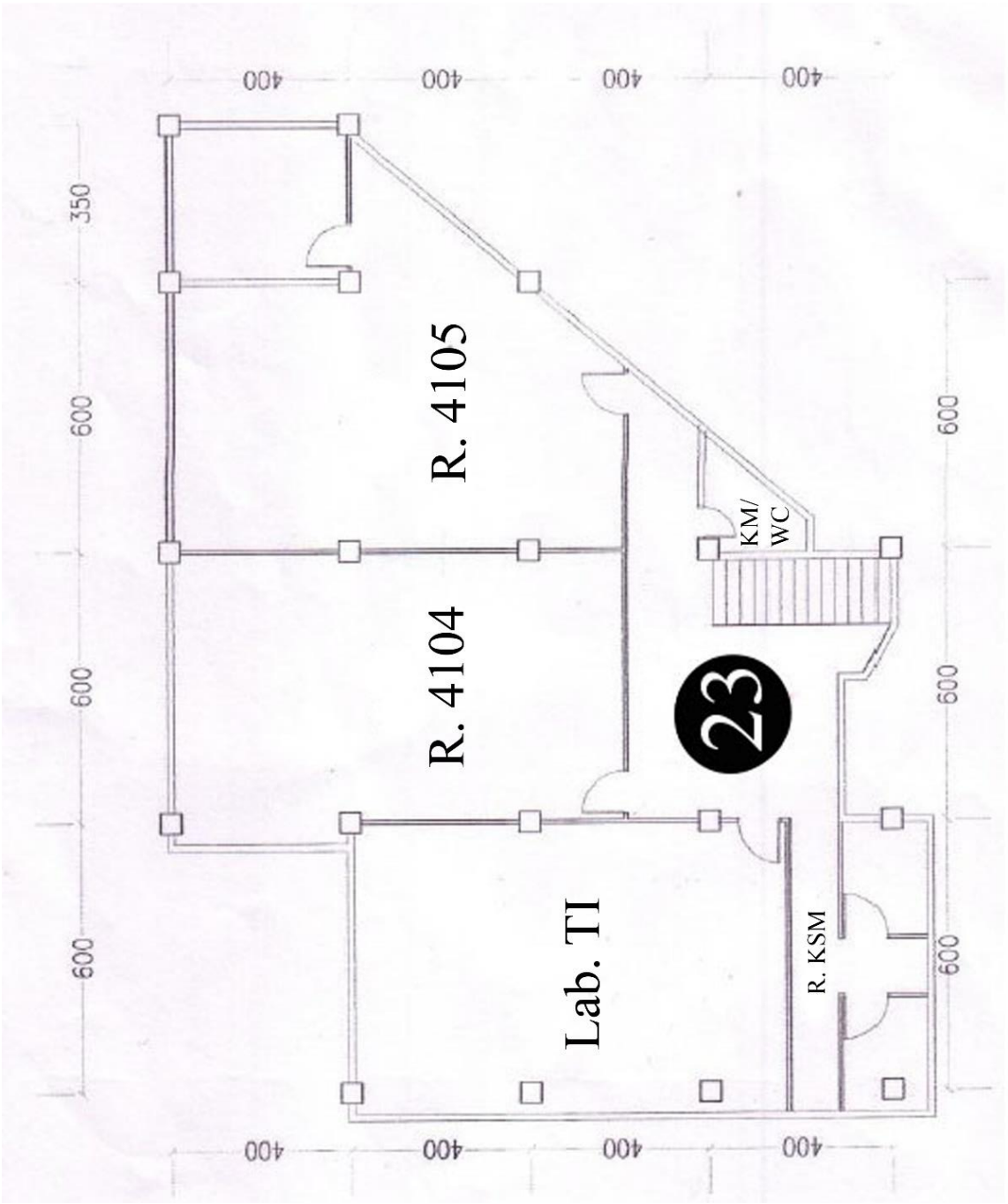
Dalam menentukan jalur terpendek dari suatu graf oleh algoritma *dijkstra* akan didapatkan jalur yang terbaik, karena pada waktu penentuan jalur yang akan

dipilih, akan dianalisis bobot dari node yang belum terpilih, lalu dipilih node dengan bobot yang terkecil. Jika ternyata ada bobot yang lebih kecil melalui node tertentu maka bobot akan dapat berubah. Algoritma *dijkstra* akan berhenti ketika semua node sudah terpilih. Sehingga akan ditemukan jalur terpendek dari seluruh node, tidak hanya untuk node dari asal dan node tujuan tertentu saja.

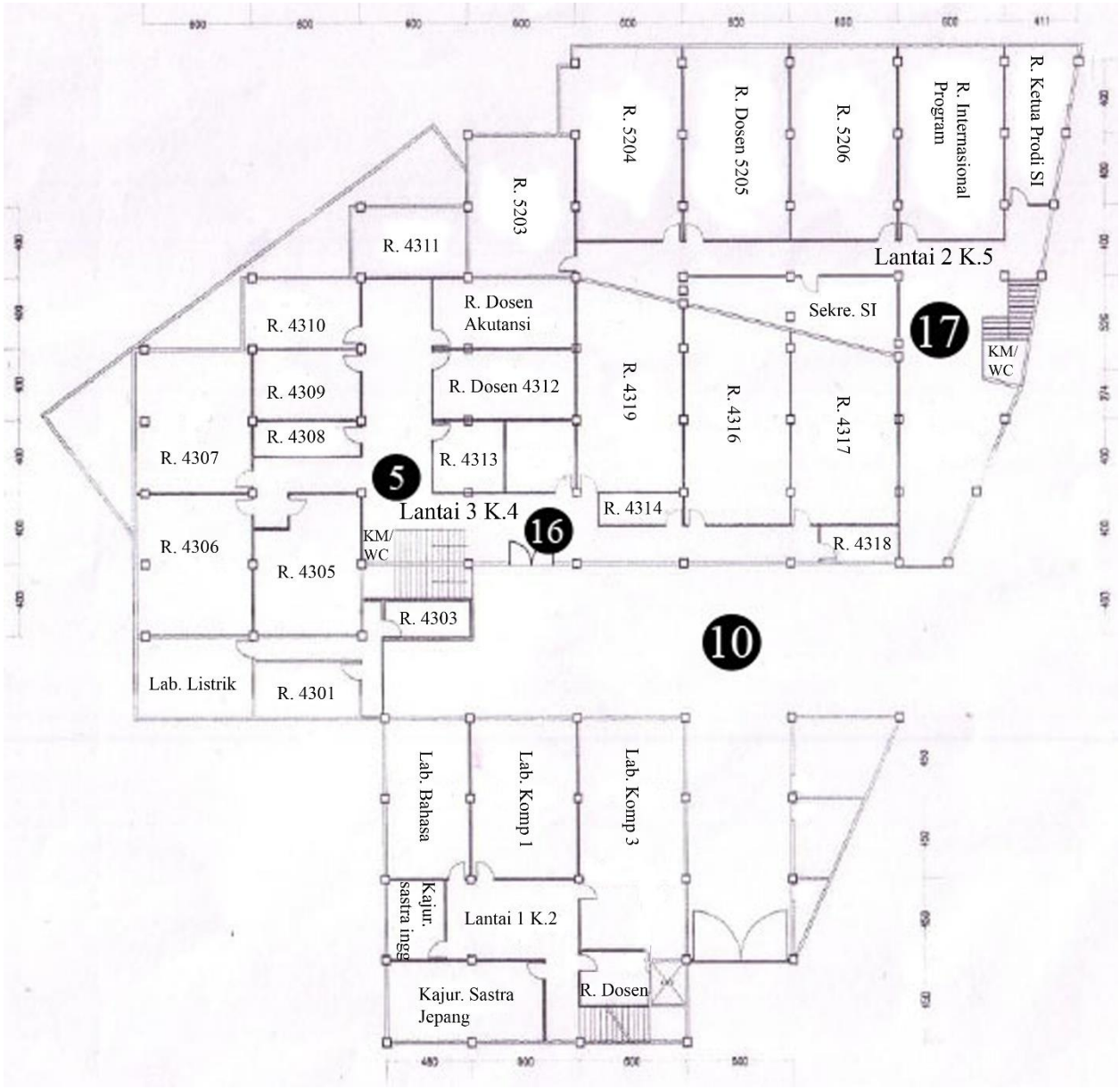
3.1.3.1 Denah Dan Node Bangunan UNIKOM



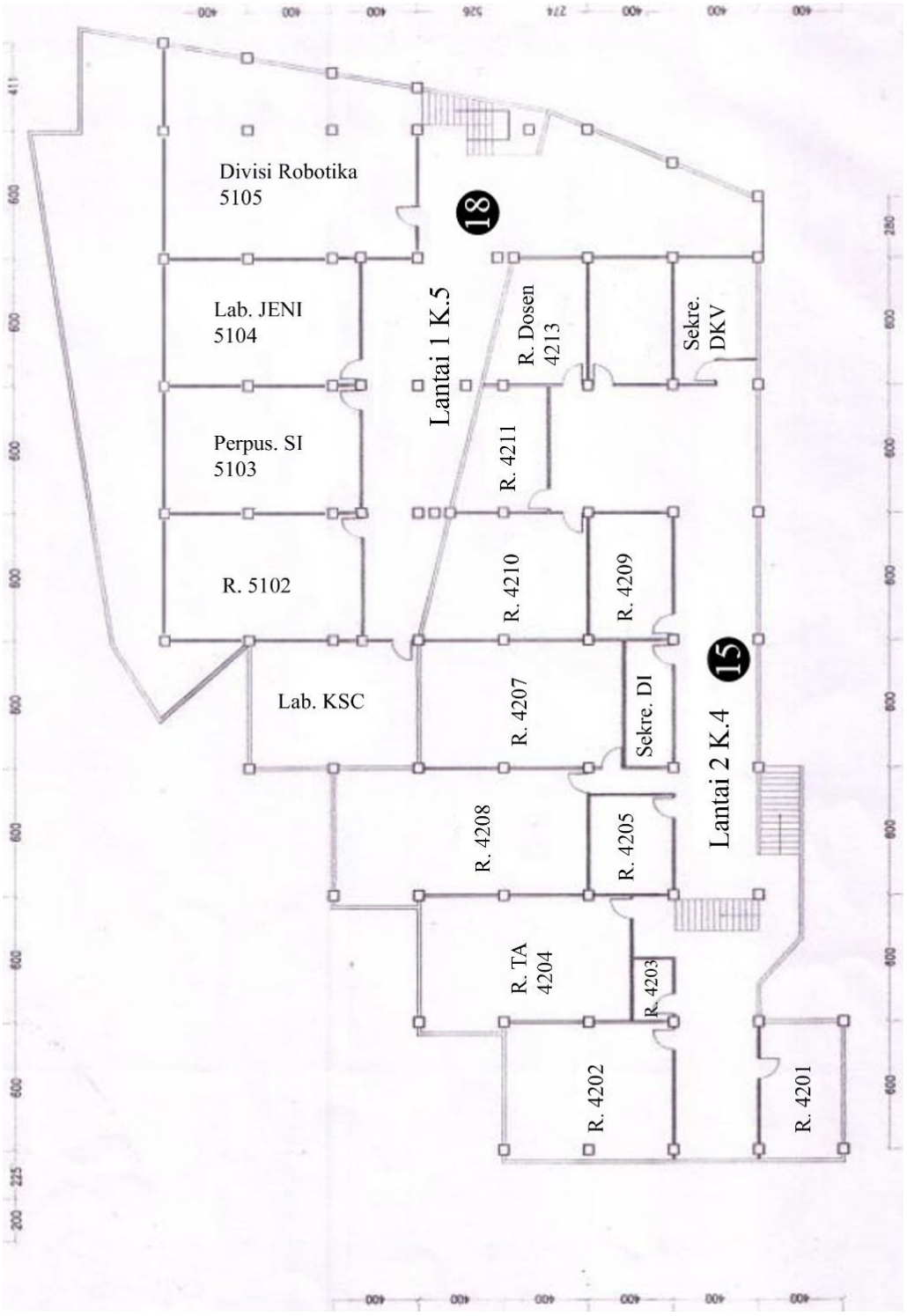
Gambar 3.3 Denah lantai 2 kampus 1, Denah lantai 1 kampus 2, Denah lantai 4 kampus 4, Denah lantai 2 kampus 5.



Gambar 3.4 Denah lantai 1 kampus 4.



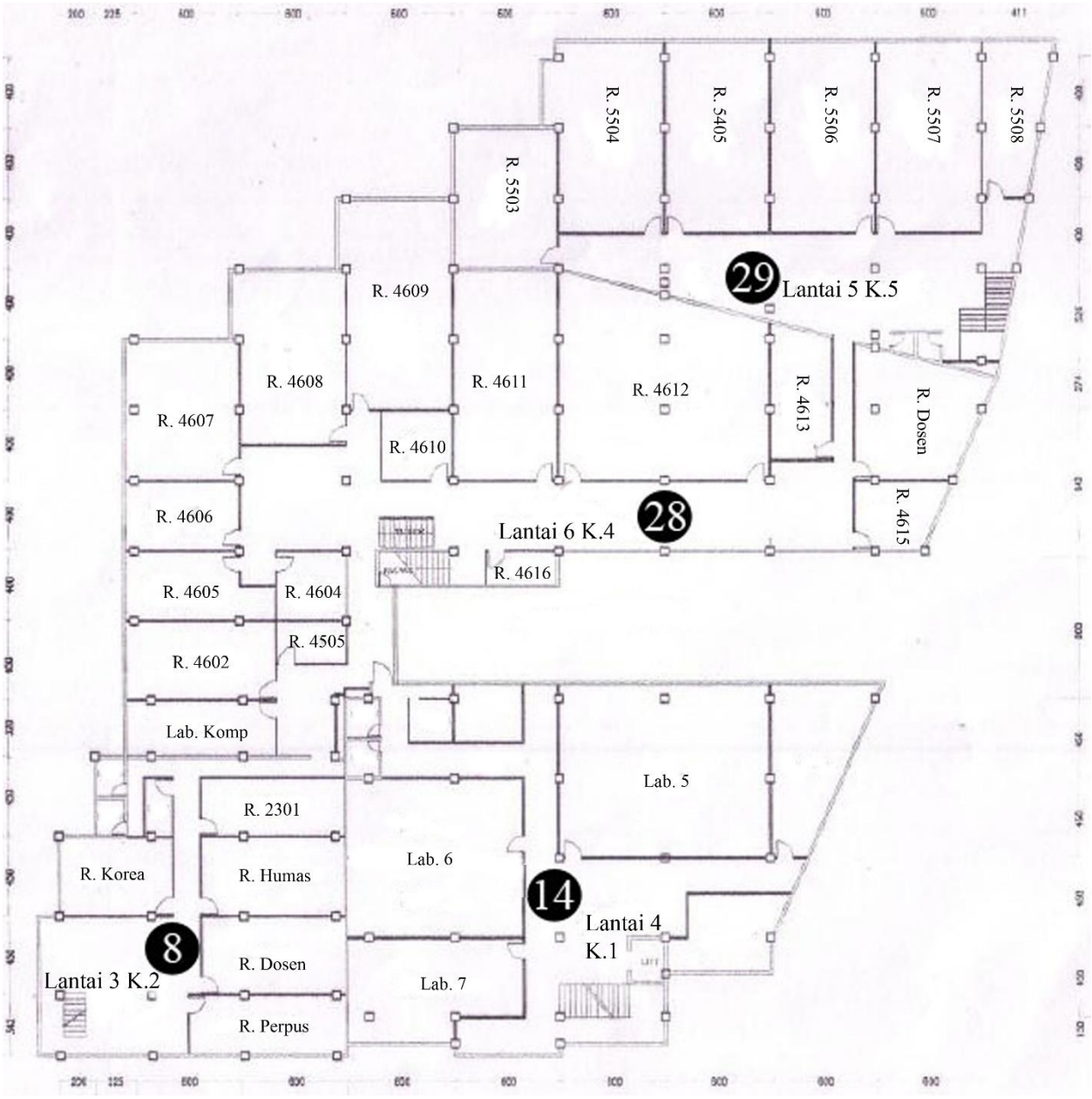
Gambar 3.5 Denah lantai 1 kampus 2, lantai 3 kampus 4, lantai 2 kampus 5.



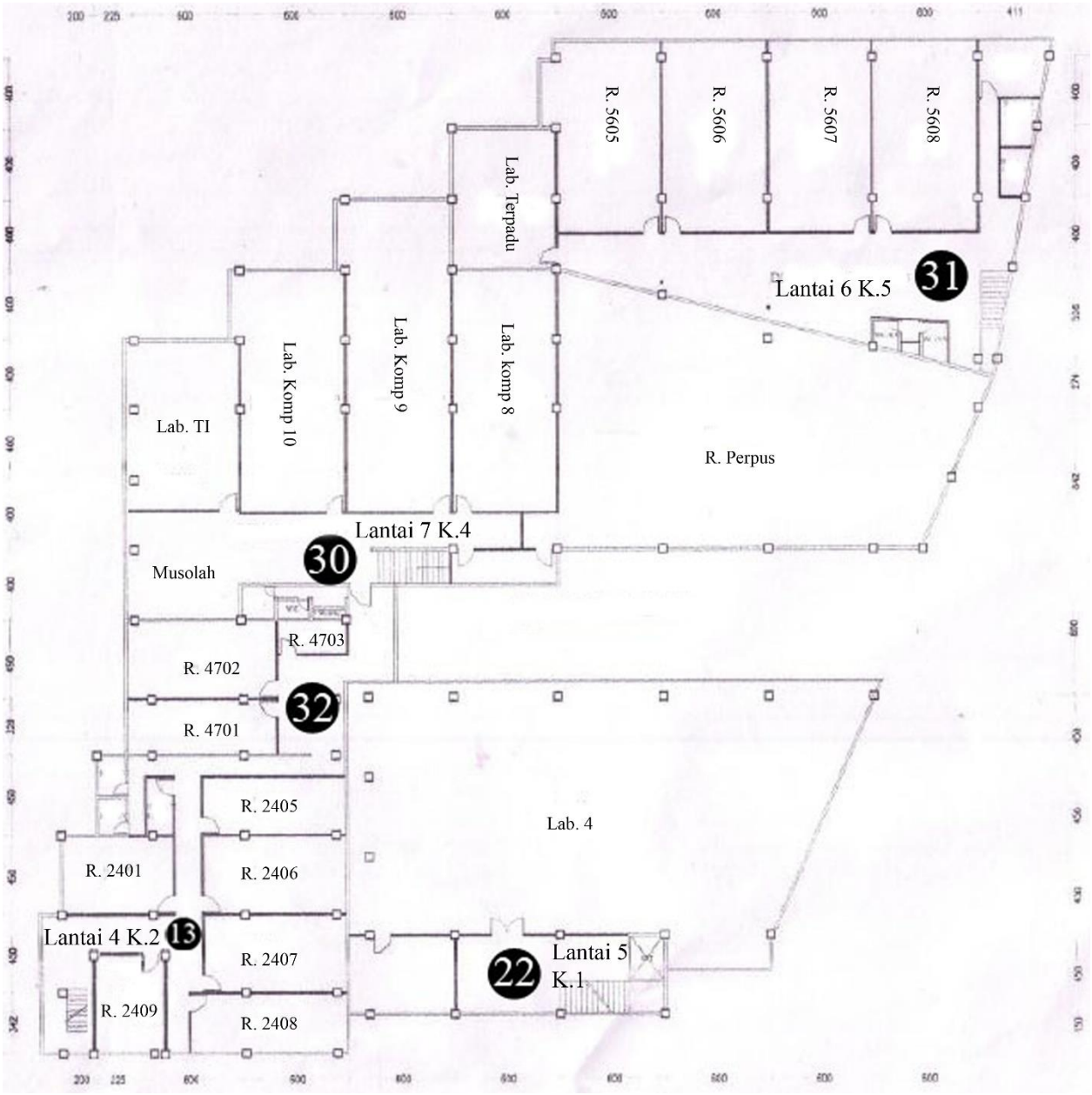
Gambar 3.6 Denah lantai 1 kampus 5, lantai 2 kampus 4.



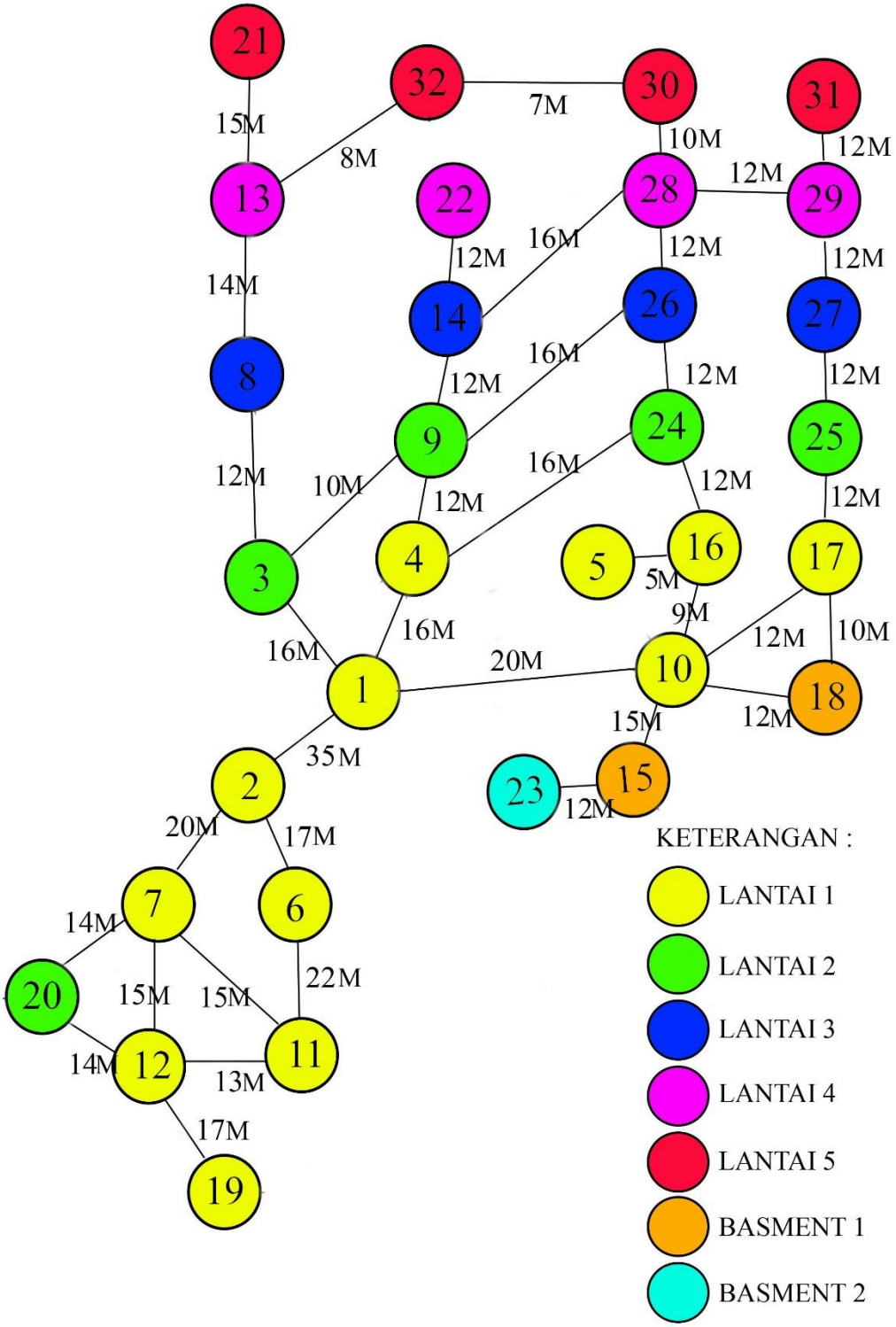
Gambar 3.7 Denah lantai 4 kampus 5, lantai 5 kampus 4, lantai 3 kampus 1, lantai 2 kampus 2



Gambar 3.8 Denah lantai 5 kampus 5, lantai 6 kampus 4, lantai 3 kampus 2, lantai 4 kampus 1.



Gambar 3.9 Denah lantai 6 kampus 5, lantai 7 kampus 4, lantai 4 kampus 2, lantai 5 kampus 1.



Gambar 3.10 Node Bangunan UNIKOM

Tabel 3.1 Keterangan Node

Node	Keterangan
1	Halaman UNIKOM
2	Depan Kampus Hukum
3	Kampus 2 lantai 2
4	Front Oficce(FO)
5	Kampus 4 lantai 3
6	Dalam Kampus Hukum
7	Parkiran
8	Kampus 2 lantai 3
9	Kampus 1 lantai 3
10	Lorong Kampus
11	Parkiran Hukum
12	Kampus Dago
13	Kampus 2 lantai 4
14	Kampus 1 lantai 4
15	Kampus 4 lantai 2
16	Kampus 4 lantai 3

node	Keterangan
17	Kampus 5 lantai 2
18	Kampus 5 lantai 1
19	Keluar Kampus Dago
20	Kampus Dago lantai 2
21	Kampus 2 lantai 5
22	Kampus 1 lantai 5
23	Kampus 4 lantai 1
24	Kampus 4 lantai 4
25	Kampus 5 lantai 3
26	Kampus 4 lantai 5
27	Kampus 5 lantai 4
29	Kampus 5 lantai 5
30	Kampus 4 lantai 7
31	Kampus 5 lantai 6
32	Kampus 2 lantai 4

3.1.3.2 Cara Kerja Algoritma *Dijkstra*

Algoritma ini mencari panjang lintasan terpendek dari node asal ke node tujuan dalam sebuah graf.

Langkah-langkah dalam menentukan lintasan terpendek pada algoritma *Dijkstra* yaitu :

1. Pada awalnya inisialisasikan node asal (V1) dan node tujuan (V2).
2. Buat 2 buah list, *open list* dan *closed list.* Keduanya tidak ada data atau kosong, dan formatnya {node, bobot, node induk}.
3. Masukkan V1 ke *open list.*
4. Pilih 1 node dengan bobot terkecil pada *open list*, tambahkan kedalam *closed list.*
5. Cari node yang bertetangga langsung dari node sebelumnya, yang masuk terakhir dalam *closed list*. Tambahkan bobot dengan node yang terkait, apabila sudah ada dalam *closed list* abaikan.
6. Apabila dalam *open list* terdapat node yang sudah ada bandingkan, lalu cari yang terkecil, dan perbaharui. Bila ternyata jumlah bobotnya sama dalam node yang sama, maka abaikan.
7. Apakah data dalam *open list* kosong ? jika belum ulangi langkah 4.
8. Dalam *closed list* cari V2, telusuri jalur berdasarkan node induk sampai mengacu ke node asal (V1), dan balikkan urutan node.
9. Lintasan terpendek ditemukan bersama bobotnya.

Algoritma *dijkstra* digunakan untuk pencarian jalur terpendek dari suatu graf, sehingga akan didapatkan jalur yang akan ditempuh.

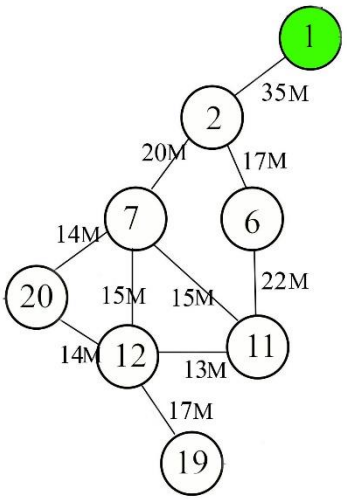
3.1.3.3 Penerapan Cara Kerja Algoritma Dijkstra

Gambar 3.9 menunjukkan jalur bangunan UNIKOM yang berupa graf yang memiliki 8 buah node dan 10 bobot yang saling terhubung,contoh kasus pencarian dimulai dari halaman utama bangunan UNIKOM atau node satu (1) ke bangunan kampus dago atau node dua belas (12) sebagai node yang dituju.

Langkah awalnya inisialisasikan node 1 sebagai node awal (V1) dan node 12 sebagai node tujuan (V2), buat 2 buah *list*, *open list* dan *closed list*.

Tabel 3.2 Hasil Iterasi Ke-1

Open List	Closed List
{1,0,null}	-

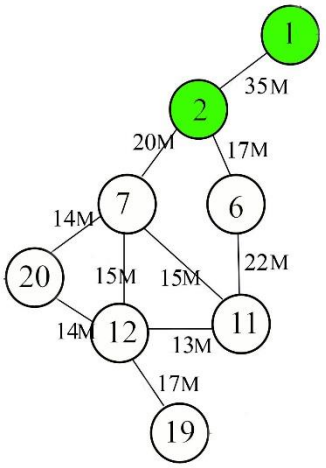


Gambar 3.11 Hasil Iterasi Ke-1

Tabel 3.2 masukkan node 1 ke dalam *open list*, node 1 memiliki bobot 0, dan node induknya *null* karena sebagai awal proses pencarian.

Tabel 3.3 Hasil Iterasi Ke-2

Open List	Closed List
{2,35,1}	{1,0,null}

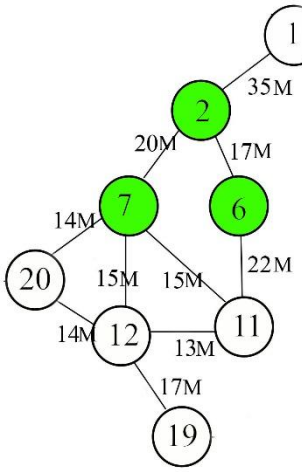


Gambar 3.12 Hasil Iterasi Ke-2

Dari tabel 3.2 Proses dari bobot terkecil lebih dahulu, dikarenakan ada satu yaitu, node 1 maka masukkan kedalam *closed list*. Node yang bertetangga dengan node 1 adalah, node 2, lalu masukkan ke dalam *open list*, node 2 memiliki bobot 35, dan node induknya node 1.

Tabel 3.4 Hasil Iterasi Ke-3

Open List	Closed List
{6,52,2}	{1,0,null}
{7,55,2}	{2,35,1}

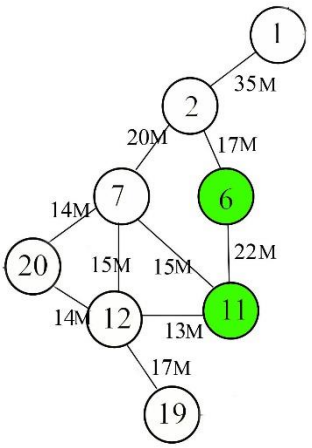


Gambar 3.13 Hasil Iterasi Ke-3

Dari tabel 3.3 Proses dari bobot terkecil lebih dahulu yaitu, node 2 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 2 adalah node 6 dan node 7, lalu masukkan ke dalam open list, karena node 1 sudah masuk ke dalam *closed list* diabaikan dan node 6 memiliki bobot 52, dan node induknya node 2, node 7 memiliki bobot 55, dan node induknya node 2.

Tabel 3.5 Hasil Iterasi Ke-4

Open List	Closed List
{7,55,2}	{1,0,null}
{11,74,6}	{2,35,1}
	{6,52,2}

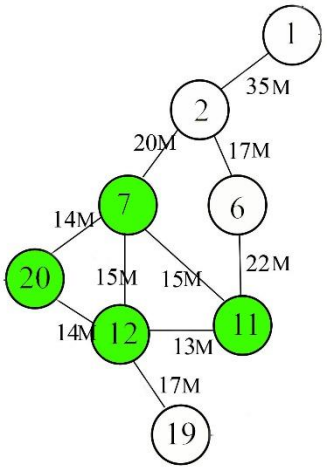


Gambar 3.14 Hasil Iterasi Ke-4

Dari tabel 3.4 Proses dari bobot terkecil lebih dahulu yaitu, node 6 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 6 adalah node 1 dan node 11, lalu masukkan ke dalam *open list*, karena node 1 sudah masuk ke dalam *closed list* diabaikan. Node 11 memiliki bobot 74, dan node induknya node 6.

Tabel 3.6 Hasil Iterasi Ke-5

Open List	Closed List
{11,74,6}	{1,0,null}
{11,70,7}	{2,35,1}
{12,70,7}	{6,52,2}
{20,69,7}	{7,55,2}

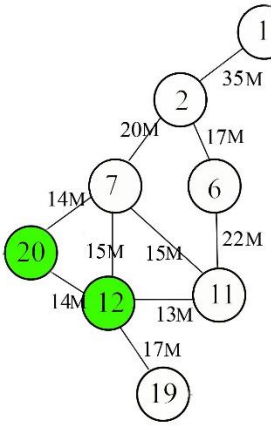


Gambar 3.15 Hasil Iterasi Ke-5

Dari tabel 3.5 Proses dari bobot terkecil lebih dahulu yaitu, node 7 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 7 adalah node 2, node 11, node 12 dan node 20, lalu masukkan ke dalam *open list*, karena node 2 sudah masuk ke dalam *closed list* diabaikan, karena node 11 sudah ada dalam *open list* bandingkan bobotnya, ambil yang terkecil. Sehingga node 11 memiliki bobot 70, dan node induknya node 7. Node 12 memiliki bobot 70, dan node induknya node 7. Node 20 memiliki bobot 69, dan node induknya node 7.

Tabel 3.7 Hasil Iterasi Ke-6

Open List	Closed List
{11,74,6}	{1,0,null}
{11,70,7}	{2,35,1}
{12,70,7}	{6,52,2}
{12,83,20}	{7,55,2}
	{20,69,7}

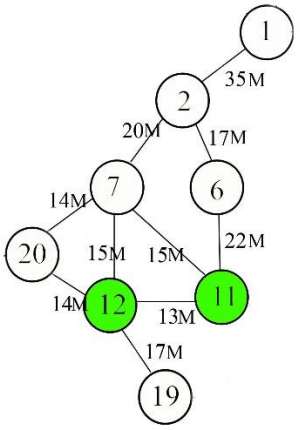


Gambar 3.16 Hasil Iterasi Ke-6

Dari tabel 3.6 Proses dari bobot terkecil lebih dahulu yaitu, node 20 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 20 adalah node 7 dan node 12, lalu masukkan ke dalam *open list*, karena node 7 sudah masuk ke dalam *closed list* diabaikan. Node 20 memiliki bobot 83, dan node induknya node 12.

Tabel 3.8 Hasil Iterasi Ke-7

Open List	Closed List
{11,74,6}	{1,0,null}
{12,70,7}	{2,35,1}
{12,83,20}	{6,52,2}
{12,83,11}	{7,55,2}
	{20,69,7}
	{11,70,7}



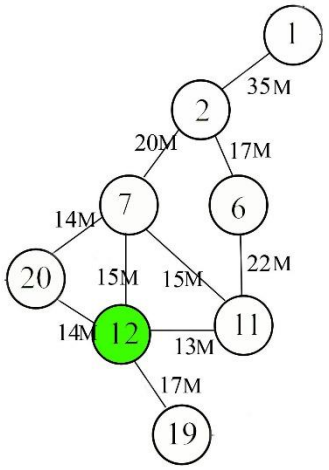
Gambar 3.17 Hasil Iterasi Ke-7

Dari tabel 3.7 Proses dari bobot terkecil lebih dahulu yaitu, node 11 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 11 adalah node 6, node 7 dan node 12, lalu masukkan ke dalam *open list*, karena node 6 dan node 7 sudah masuk ke dalam *closed list* diabaikan. Node 11 memiliki bobot 83, dan node induknya node 12.

Tabel 3.9 Hasil Iterasi Ke-8

Open List	Closed List
{11,74,6}	{1,0,null}
{12,83,20}	{2,35,1}
{12,83,11}	{6,52,2}
	{7,55,2}
	{20,69,7}

	{11,70,7}
	{12,70,7}

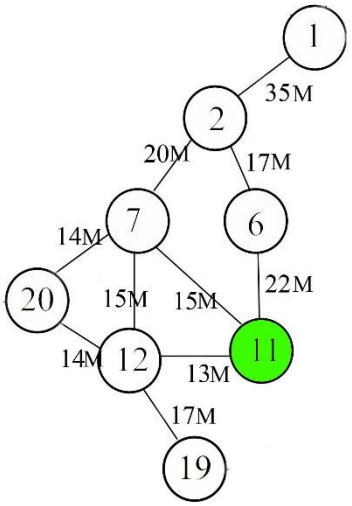


Gambar 3.18 Hasil Iterasi Ke-8

Dari tabel 3.8 Proses dari bobot terkecil lebih dahulu yaitu, node 12 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 12 adalah node 7, node 11 dan node 20, lalu masukkan ke dalam *open list*, karena node 7, node 11 dan node 20 sudah masuk ke dalam *closed list* diabaikan.

Tabel 3.10 Hasil Iterasi Ke-9

Open List	Closed List
{12,83,20}	{1,0,null}
{12,83,11}	{2,35,1}
{12,87,11}	{6,52,2}
	{7,55,2}
	{20,69,7}
	{11,70,7}
	{12,70,7}
	{11,74,6}

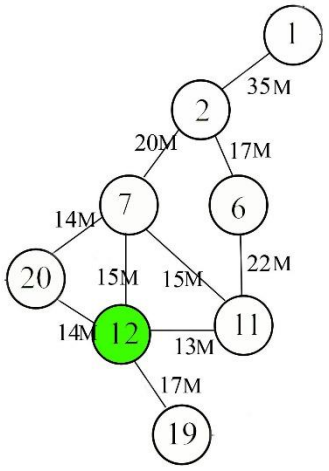


Gambar 3.19 Hasil Iterasi Ke-9

Dari tabel 3.9 Proses dari bobot terkecil lebih dahulu yaitu, node 11 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 11 adalah node 6, node 7 dan node 12, lalu masukkan ke dalam *open list*, karena node 6, node 7 dan node 12 sudah masuk ke dalam *closed list* diabaikan. Node 11 memiliki bobot 87, dan node induknya node 12.

Tabel 3.11 Hasil Iterasi Ke-10

Open List	Closed List
{12,83,11}	{1,0,null}
{12,87,11}	{2,35,1}
	{6,52,2}
	{7,55,2}
	{20,69,7}
	{11,70,7}
	{12,70,7}
	{11,74,6}
	{12,83,20}

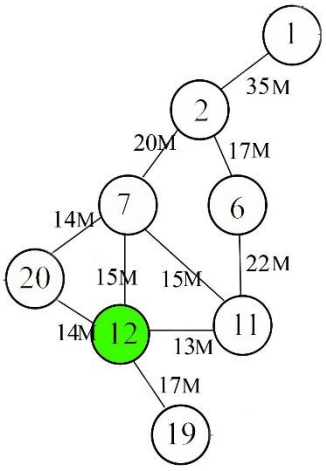


Gambar 3.20 Hasil Iterasi Ke-10

Dari tabel 3.10 Proses dari bobot terkecil lebih dahulu yaitu, node 12 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 12 adalah node 7, node 11 dan node 20, lalu masukkan ke dalam *open list*, karena node 7, node 11 dan node 20 sudah masuk ke dalam *closed list* diabaikan.

Tabel 3.12 Hasil Iterasi Ke-11

Open List	Closed List
{12,87,11}	{1,0,null}
	{2,35,1}
	{6,52,2}
	{7,55,2}
	{20,69,7}
	{11,70,7}
	{12,70,7}
	{11,74,6}
	{12,83,20}
	{12,83,11}

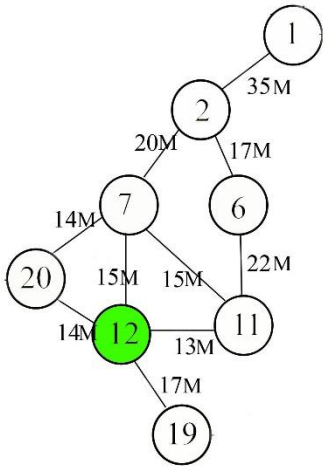


Gambar 3.21 Hasil Iterasi Ke-11

Dari tabel 3.11 Proses dari bobot terkecil lebih dahulu yaitu, node 12 dan masukkan kedalam *closed list*, node yang bertetangga dengan node 12 adalah node 7, node 11 dan node 20, lalu masukkan ke dalam *open list*, karena node 7, node 11 dan node 20 sudah masuk ke dalam *closed list* diabaikan.

Tabel 3.13 Hasil Iterasi Ke-12

Open List	Closed List
-	{1,0,null}
	{2,35,1}
	{6,52,2}
	{7,55,2}
	{20,69,7}
	{11,70,7}
	{12,70,7}
	{11,74,6}
	{12,83,20}
	{12,83,11}
	{12,87,11}



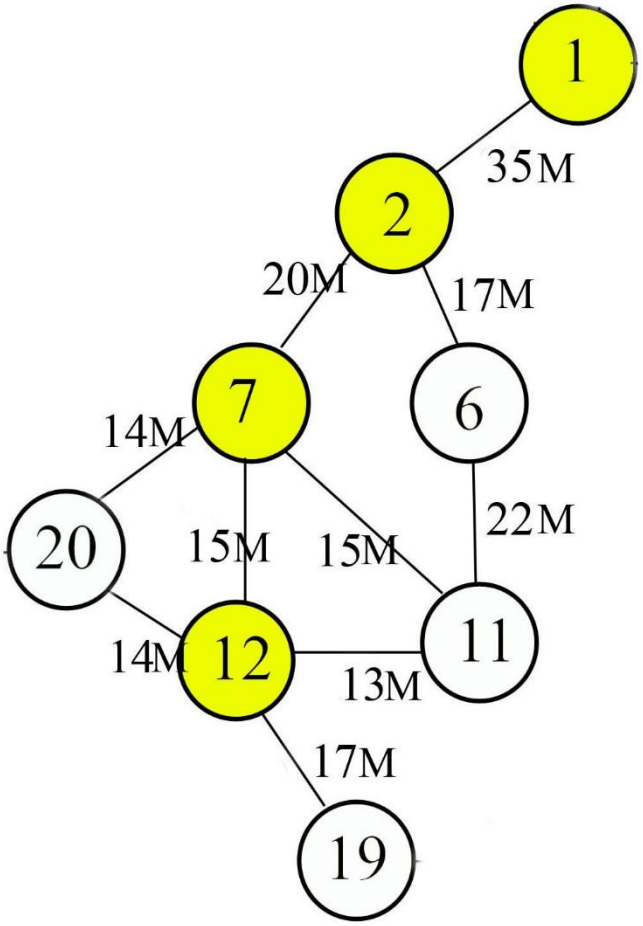
Gambar 3.22 Hasil Iterasi Ke-12

Dari tabel 3.12 Proses dari bobot terkecil lebih dahulu, karena tinggal 1 maka node 12 masukkan kedalam *closed list*, node yang bertetangga dengan node 12 adalah node 7, node 11 dan node 20, lalu masukkan ke dalam *open list*.

Pencarian akan berhenti dikarenakan *open list* sudah kosong, didalam *closed list* telusuri V2 atau node tujuan dengan mengacu pada node induk, dan balikkan urutan nodenya, sehingga pencarian jalur terdapat 4 yaitu:

1. 1 – 2 – 7 – 12: 70.
2. 1 – 2 – 7 – 20 – 12: 83.
3. 1 – 2 – 7 – 11 – 12: 83.
4. 1 – 2 – 6 – 11 – 12: 87.

Maka jalur terpendek dari graf yang dimulai dari node satu (1) ke node duabelas (12) diperoleh jalur 1. **1-2-7-12** dengan jumlah bobot **70**. Dapat dilihat pada gambar 3.21.



Gambar 3.23 Sesudah pencarian jalur terpendek

3.1.4 Analisis Kebutuhan non-Fungsional

Analisis *non fungsional* dilakukan untuk menghasilkan spesifikasi kebutuhan *non fungsional*. Spesifikasi kebutuhan *non fungsional* adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika 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 pemakai dan kategori pemakai, serta kontrol terhadap sistem.

3.1.4.1 Analisis Pemakai

Pemakai atau user dari aplikasi yang akan dibangun terdiri dari dua pemakai, yaitu pengguna dan pengelola.

Tabel 3.14 Pemakai

Nama Pemakai	Kualifikasi
Pengguna	Pengguna yaitu mahasiswa atau mahasiswa baru maupun orangtua mahasiswa yang memiliki kemampuan dasar di bidang komputer, memiliki pengalaman browsing di internet. pengguna hanya dapat melihat konten yang tersedia.
Pengelola	Kualifikasi yang harus dimiliki pengelola yaitu memiliki kemampuan dasar di bidang komputer, mengerti tentang jaringan dan web server (MySql dan phpmyadmnin). Pengelola dapat mengubah atau menghapus data ruangan dan gambar.

3.1.4.2 Analisi Perangkat Keras

Spesifikasi perangkat keras yang digunakan adalah sebagai berikut:

1. Prosesor: 1,0 GHz
2. Memori: 512 Mb
3. Harddisk: 80 MB *free space*
4. VGA Card: 64 MB
5. Monitor: color 14’
6. Keyboard: Standar
7. Mouse: Standar
8. LAN card atau Modem untuk koneksi internet

3.1.4.3 Analisis Perangkat Lunak

Kebutuhan perangkat lunak merupakan faktor-faktor yang harus dipenuhi untuk merancang sebuah perangkat lunak sehingga perangkat lunak tersebut sesuai

dengan maksud dan tujuan perangkat lunak tersebut di buat. Beberapa perangkat lunak pendukung pembangun sebagai berikut :

1. Windows sebagai sistem operasi.
2. Google Chrome 30.x sebagai web browser.
3. MySQL sebagai DBMS.
4. Tool Antarmuka design Microsoft Visio 2013.
5. Adobe dreamweaver sebagai pembangun web.

Sedangkan untuk user dapat menggunakan :

1. Windows sebagai sistem operasi.
2. Mozilla Firefox 2.x sebagai web browser.

3.1.4.4 Analisis Webhosting

Webhosting digunakan untuk menyimpan data yang ada di internet. Untuk itu webhosting yang digunakan dapat mengeksekusi PHP dan mempunyai MySql dalam media penyimpanan databasenya. Selain itu webhosting dapat yang digunakan harus dapat aktif 24 jam dan memiliki spesifikasi minimum seperti:

a. Bandwidth 100GB/bulan.
b. Minimum upload speed 128 Kbit/s, download 128Kbit/s
c. Apache version 2.2.13.
d. PHP versi 5.2.
e. MySql versi 5.0.81-community.
f. Mendukung versi mail, XML, GD2, curl.

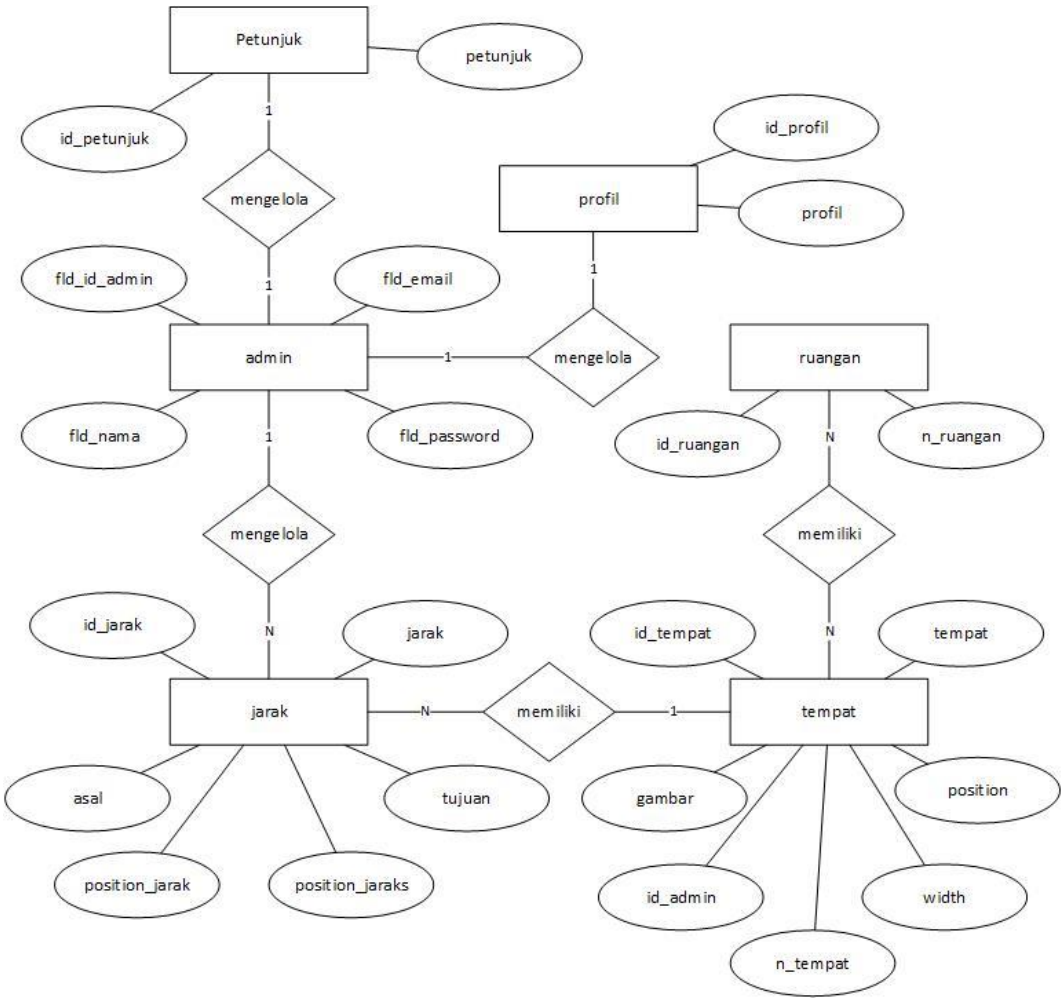
3.2 Perancangan Sistem

3.2.1 ERD (Entity Relationship Diagram)

Usulan untuk diagram E-R yaitu terdapat kunci yang unik (*primary key*) pada setiap entitas (tabel induk) yang dapat membedakan dengan atribut lainnya sehingga tabel tersebut dapat dijadikan referensi untuk tabel yang lainnya. Usulan

untuk tabel yang akan dibuat untuk membantu dalam pengolahan data adalah user dan virtual tour.

Dari usulan diatas maka dapat dibuat diagram E-R untuk sistem baru yang terdri dari beberapa tabel yang dibutuhkan untuk menyimpan data. Untuk melihat keterhubungan antar tabel yang ada maka akan digambarkan sebagai berkut:



Gambar 3.24 ERD Pemandu Lokasi UNIKOM

Tabel 3.15 Kamus Data Entity Relationship Diagram (ERD)

No	Nama Entitas atau Relasi	Atribut
1.	Admin	fld_Id_admin, fld_email, fld_nama, fld_password.
2.	Ruangan	id_ruangan, n_ruangan.
3.	Jarak	Id_jarak, asal, tujuan, _jarak, position_jarak, position_jaraks.

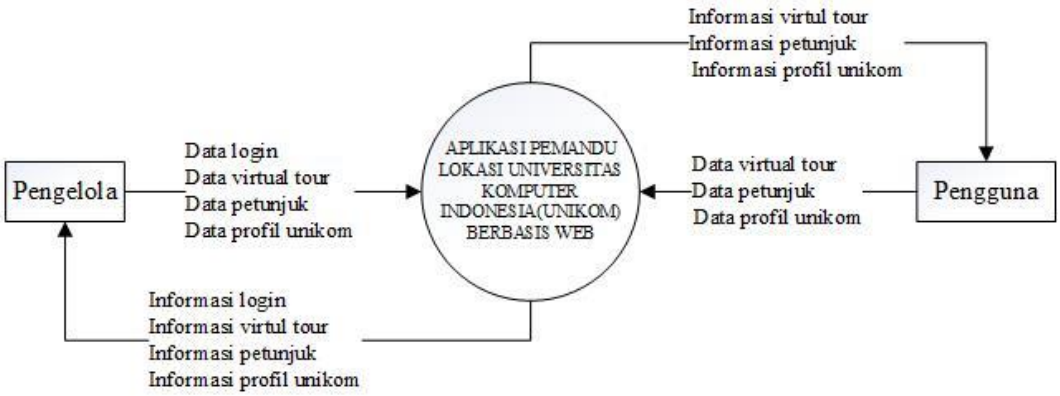
No	Nama Entitas atau Relasi	Atribut
4.	Tempat	Id_tempat, tempat, gambar, position, id_admin, n_tempat, width.
5	Petunjuk	Id_petunjuk, petunjuk.
6	Profil	Id_profil, profil.

3.2.2 Perancangan Proses

Analisis kebutuhan fungsional akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan, analisis kebutuhan fungsional dapat didefinsikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Tahapan ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan dari rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. Alat bantu yang digunakan untuk menggambarkan system secara umum yang akan dibangun yaitu diagram konteks dan *data flow diagram*. Untuk menjelaskan bagaimana suatu masukan diproses pada sistem maka digunakan spesifikasi proses dan kamus data untuk mengetahui aliran data yang mengalir pada sistem.

3.2.2.1 Diagram Konteks

Diagram konteks adalah arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar. Adapun diagram konteks untuk sistem yang akan dibangun dapat dilihat pada gambar 3.24:



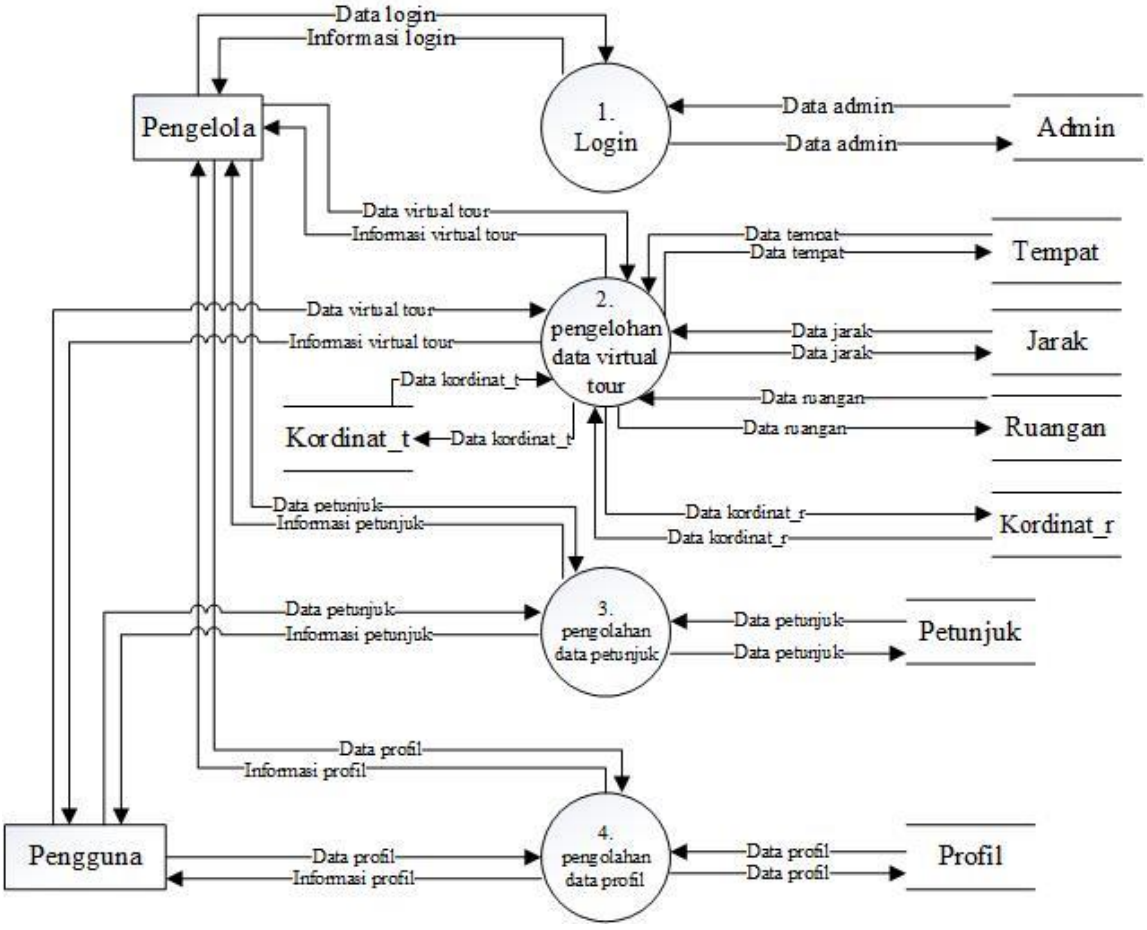
Gambar 3.25 Diagram Konteks

3.2.2.2 Data Flow Diagram (DFD)

Data Flow Diagram merupakan suatu media yang digunakan untuk menggambarkan aliran data yang mengalir pada suatu sistem informasi. DFD pada pembangunan pemandu lokasi universitas komputer Indonesia (UNIKOM) yang terdiri dari beberapa bagian, berikut data flow diagram selengkapnya:

3.2.2.2.1 DFD Level 1

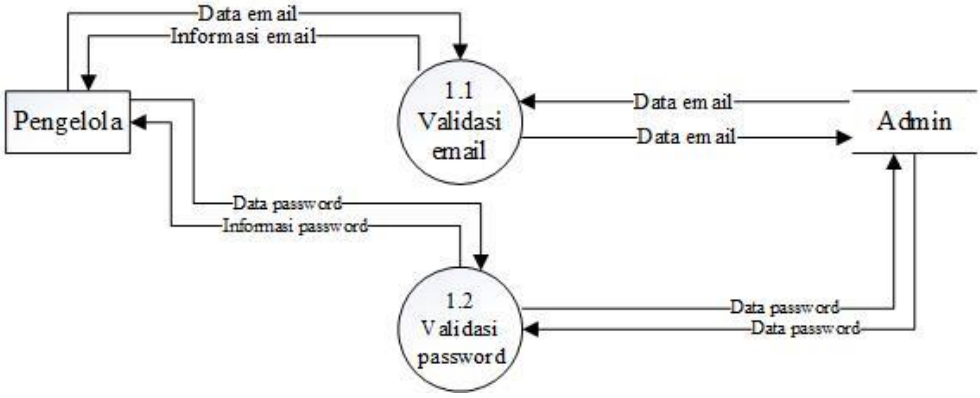
Diagram alir data level satu dari sistem pembangunan pemandu lokasi Universitas Komputer Indonesia (UNIKOM) dapat dilihat pada gambar 3.25 sebagai berikut.



Gambar 3.26 Data Flow Diagram Level 1

3.2.2.2.2 DFD Level 2 Login

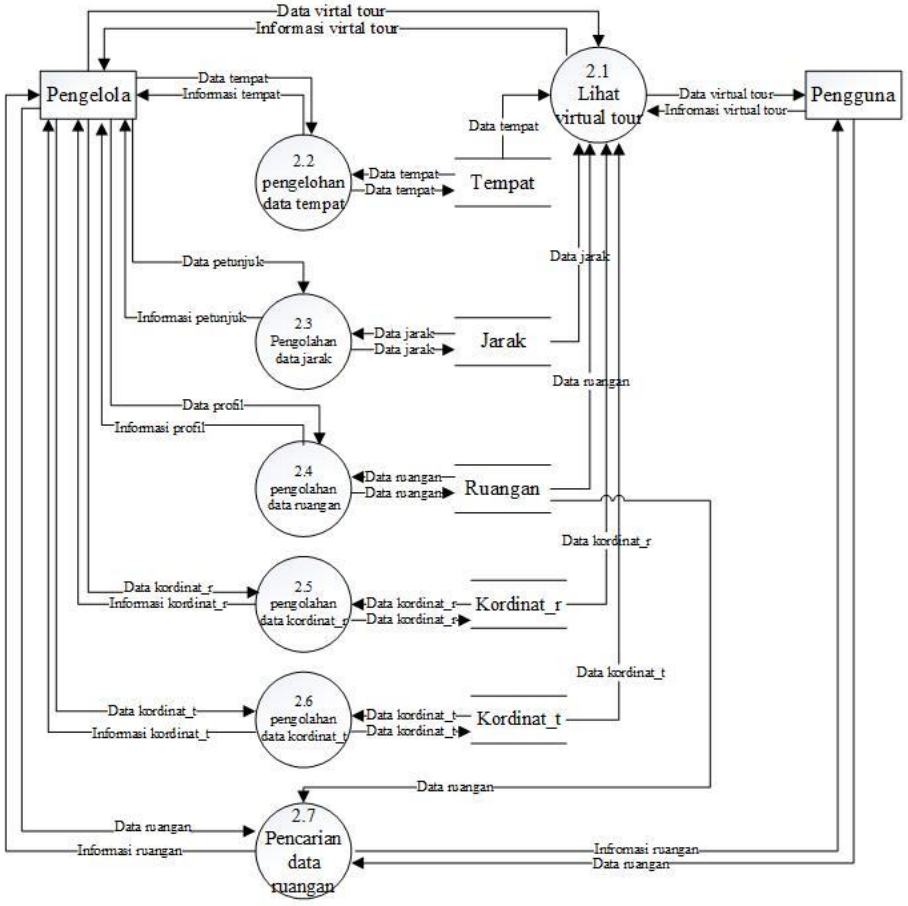
Diagram alir data pada level 2 proses 1 Data Login dapat dilihat pada Gambar 3.26:



Gambar 3.27 Data Flow Diagram Level 2 Proses 1 Login

3.2.2.2.3 DFD Level 2 Proses Pengolahan Data Virutaltour

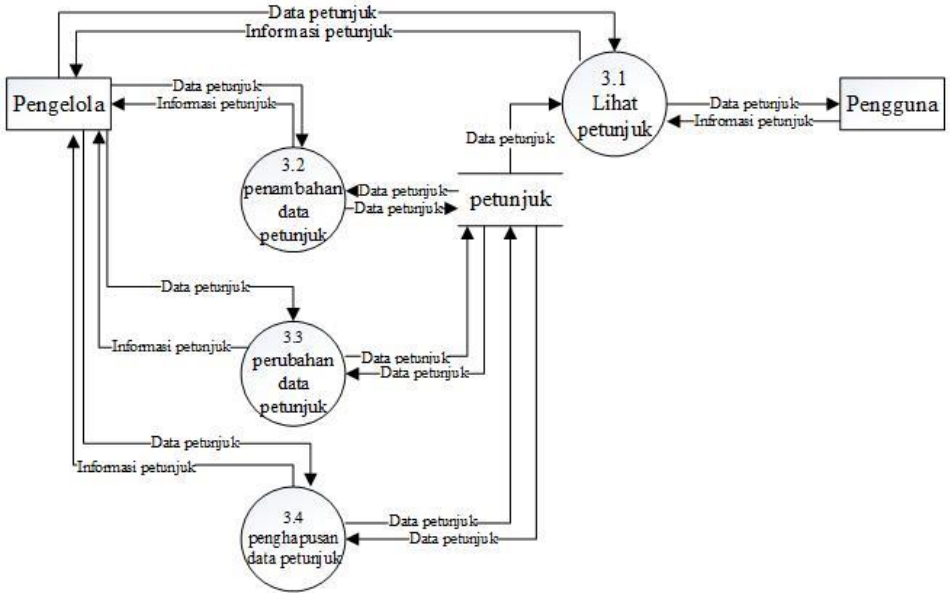
Diagram alir data pada level 2 proses 2 Data Proses Pengolahan Data Virutaltour dapat dilihat pada Gambar 3.27:



Gambar 3.28 Data Flow Diagram Level 2 Proses 2 Pengolahan Data Virutaltour

3.2.2.2.4 DFD Level 2 Proses Pengolahan Data Petunjuk

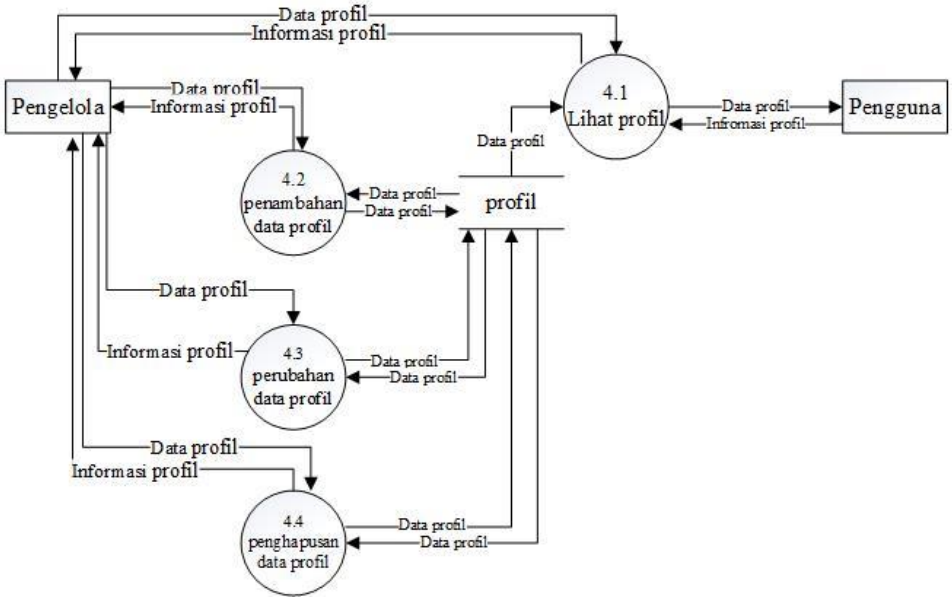
Diagram alir data pada level 2 proses 3 Data Proses Pengolahan Data Petunjuk dapat dilihat pada Gambar 3.28:



Gambar 3.29 Data Flow Diagram Level 2 Proses 3 Pengolahan Data petunjuk

3.2.2.2.5 DFD Level 2 Proses Pengolahan Data Profil UNIKOM

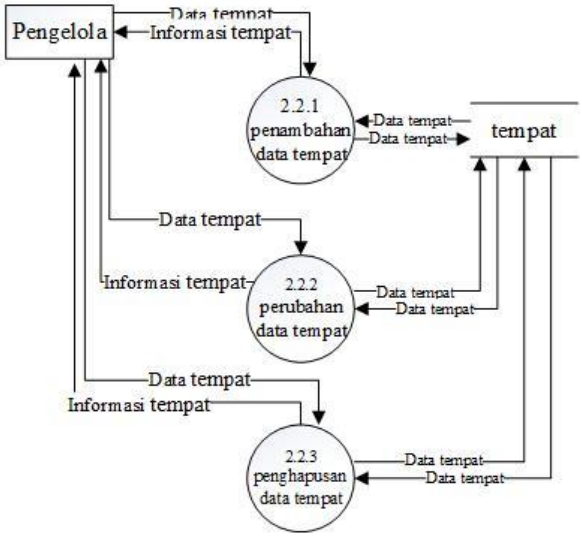
Diagram alir data pada level 2 proses 4 Data Proses Pengolahan Data Profil dapat dilihat pada Gambar 3.29:



Gambar 3.30 Data Flow Diagram Level 2 Proses 4 Pengolahan Data Profil UNIKOM

3.2.2.2.6 DFD Level 3 Proses Pengolahan Data Tempat

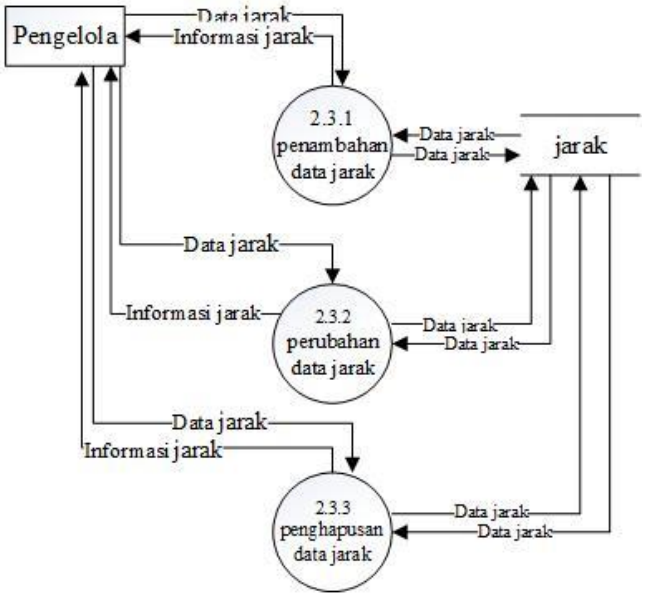
Diagram alir data pada level 3 proses 1 Data Proses Pengolahan Data Tempat dapat dilihat pada Gambar 3.30:



Gambar 3.31 Data Flow Diagram Level 3 Proses 1 Pengolahan Data Tempat

3.2.2.2.7 DFD Level 3 Proses Pengolahan Data Jarak

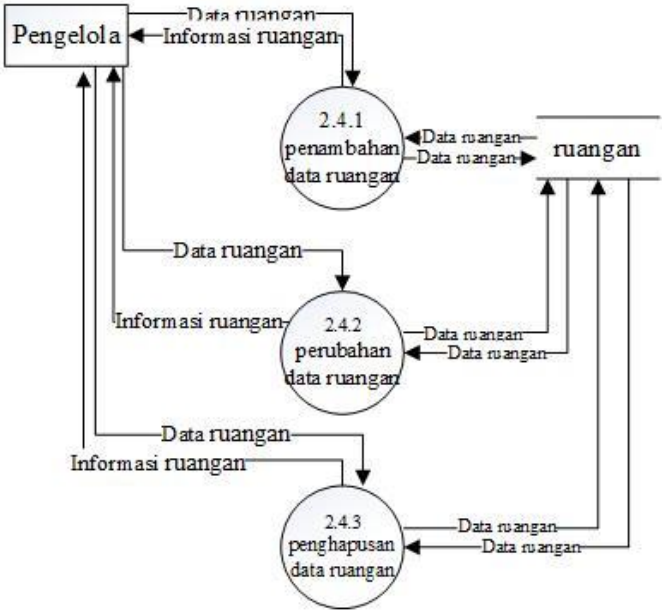
Diagram alir data pada level 3 proses 2 Data Proses Pengolahan Data Jarak dapat dilihat pada Gambar 3.31:



Gambar 3.32 Data Flow Diagram Level 3 Proses 2 Pengolahan Data Jarak

3.2.2.2.8 DFD Level 3 Proses Pengolahan Data Ruangan

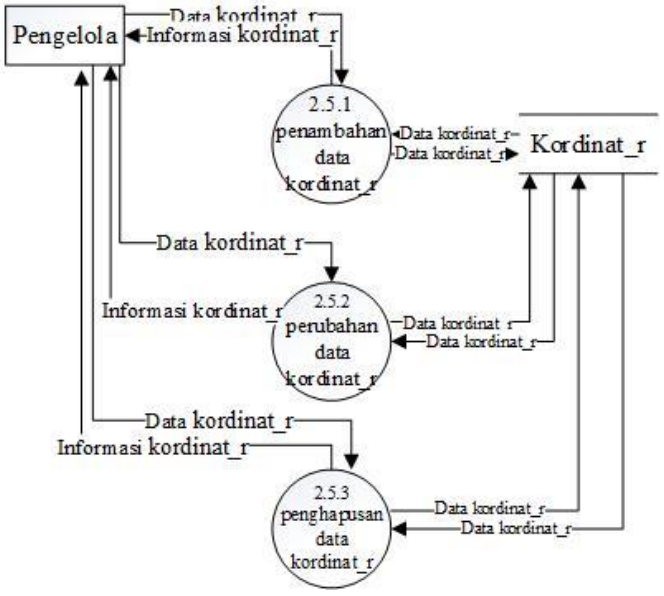
Diagram alir data pada level 3 proses 3 Data Proses Pengolahan Data Ruangan dapat dilihat pada Gambar 3.32:



Gambar 3.33 Data Flow Diagram Level 3 Proses 3 Pengolahan Data Ruangan

3.2.2.2.9 DFD Level 3 Proses Pengolahan Data Kordinat_r

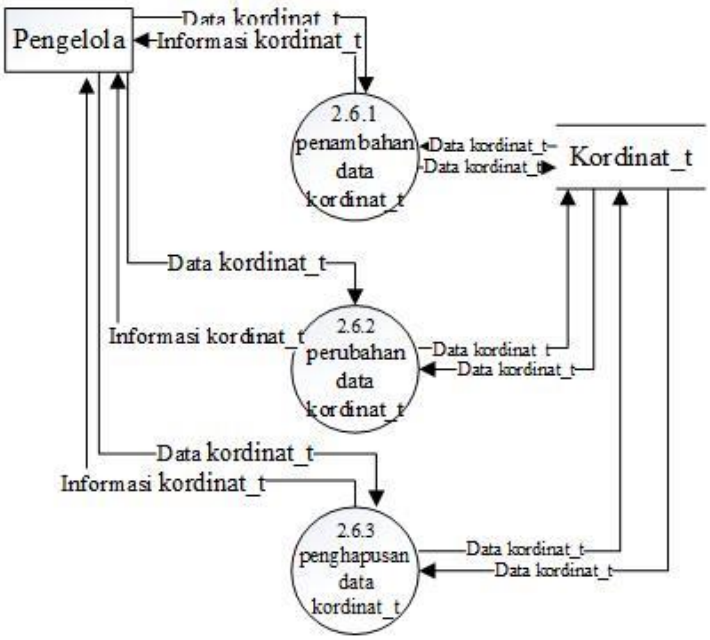
Diagram alir data pada level 3 proses 4 Data Proses Pengolahan Data Kordinat_r dapat dilihat pada Gambar 3.33:



Gambar 3.34 Data Flow Diagram Level 3 Proses 4 Pengolahan Data Kordinat_r

3.2.2.2.10 DFD Level 3 Proses Pengolahan Data Kordinat_t

Diagram alir data pada level 3 proses 5 Data Proses Pengolahan Data Kordinat_t dapat dilihat pada Gambar 3.34:



Gambar 3.35 Data Flow Diagram Level 3 Proses 5 Pengolahan Data Kordinat_t

3.2.3 Spesifikasi Proses

Spesifikasi Proses digunakan untuk menggambarkan proses model aliran yang terdapat pada DFD. Spesifikasi proses dari DFD sistem pembangunan pemandu lokasi Universitas Komputer Indonesia (UNIKOM) dijelaskan pada table 3.16.

Tabel 3.16 Spefikasi Proses DFD

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
1.	No. Proses	1
	Nama	Login
	Source	Admin
	Input	Data Login
	Output	Informasi Login
	Destination	Admin
	Logika Proses	1. Admin memasukan data login ke dalam sistem. 2. Sistem mengecek data login ke tabel *Admin* a. Jika data pada form login belum lengkap, maka sistem menampilkan pesan “Kedua kolom harus diisi” b. Jika data login salah, maka sistem menampilkan pesan “Kombinasi Username & Password Salah... !” 3. Sistem menampilkan Layar Home Admin apabila login sukses.

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
2.	No. Proses	2
	Nama	Pengolahan Data Virtual Tour
	Source	Admin, pengguna.
	Input	Data Virutal Tour
	Output	Informasi Virutal Tour
	Destination	Admin, pengguna.
	Logika Proses	Jika user memilih menu virtual tour sistem menampilkan gambar dalam bentuk 360° dan info jarak.
3.	No. Proses	3
	Nama	Pengolahan Data Petunjuk
	Source	Admin, pengguna.
	Input	Data Petunjuk
	Output	Informasi Petunjuk
	Destination	Admin, pengguna.
	Logika Proses	Jika user melihat petunjuk sistem menampillkan infromasi petunjuk tentang cara menggunakan aplikasi.
4.	No. Proses	4
	Nama	Pengolahan Data Profil
	Source	Admin, pengguna.
	Input	Data Profil
	Output	Informasi Profil
	Destination	Admin, pengguna.
	Logika Proses	Jika user melihat profil sistem menampillkan infromasi profil UNIKOM.
5.	No. Proses	1.1
	Nama	Validasi Email
	Source	Admin.
	Input	Data Email
	Output	Informasi Email
	Destination	Email
	Logika Proses	Admin memasukan email benar atau tidaknya email.
6.	No. Proses	1.2
	Nama	Validasi Password
	Source	Admin
	Input	Data password
	Output	Informasi password
	Destination	Admin
	Logika Proses	Admin memasukan password benar atau tidaknya password.
7.	No. Proses	2.1
	Nama	Lihat Virtual Tour
	Source	Admin, pengguna.
	Input	Data Virutal tour
	Output	Informasi Virtual tour
	Destination	Admin, pengguna
	Logika Proses	Jika user ingin lihat Virtual tour maka sistem menampilkan form virtual tour yang terdapat gambar dan jarak.
8.	No. Proses	2.2
	Nama	Pengolahan Data Tempat
	Source	Admin
	Input	Data Tempat
	Output	Informasi Tempat
	Destination	Admin

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
	Logika Proses	Jika admin ingin mengolah data tempat maka sistem menampilkan form pengolahan data tempat yang di dalamnya terdapat proses menambah, mengubah dan menghapus data tempat.
9.	No. Proses	2.3
	Nama	Pengolahan Data Jarak
	Source	Admin
	Input	Data Tempat
	Output	Informasi Tempat
	Destination	Admin
	Logika Proses	Jika admin ingin mengolah data jarak maka sistem menampilkan form pengolahan data jarak yang di dalamnya terdapat proses menambah, mengubah dan menghapus data jarak.
10.	No. Proses	2.4
	Nama	Pengolahan Data Ruangan
	Source	Admin
	Input	Data Ruangan
	Output	Informasi Ruangan
	Destination	Admin
	Logika Proses	Jika admin ingin mengolah data ruangan maka sistem menampilkan form pengolahan data ruangan yang di dalamnya terdapat proses menambah, mengubah dan menghapus data ruangan.
11.	No. Proses	2.5
	Nama	Pengolahan Data Kordinat_r
	Source	Admin
	Input	Data Kordinat_r
	Output	Informasi Kordinat_r
	Destination	Admin
	Logika Proses	Jika admin ingin mengolah data kordinat_r maka sistem menampilkan form pengolahan data kordinat_r yang di dalamnya terdapat proses menambah, mengubah dan menghapus data kordinat_r.
10.	No. Proses	2.6
	Nama	Pengolahan Data Kordinat_t
	Source	Admin
	Input	Data Kordinat_t
	Output	Informasi Kordinat_t
	Destination	Admin
	Logika Proses	Jika admin ingin mengolah data kordinat_t maka sistem menampilkan form pengolahan data kordinat_t yang di dalamnya terdapat proses menambah, mengubah dan menghapus data kordinat_t.
12.	No. Proses	2.7
	Nama	Pencarian Data Ruangan
	Source	Admin
	Input	Data Ruangan
	Output	Informasi Ruangan
	Destination	Admin
	Logika Proses	Jika user ingin mencari data ruangan maka sistem menampilkan form pencarian data ruangan yang di dalamnya terdapat ruangan yang ingin dicari.
13	No. Proses	3.1
	Nama	Lihat Petunjuk .
	Source	Admin, pengguna.
	Input	Data petunjuk.

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
	Output	Informasi petunjuk.
	Destination	Admin, pengguna.
	Logika Proses	Jika user ingin lihat petunjuk maka sistem menampilkan form petunjuk yang berisi tentang cara menggunakan aplikasi.
14.	No. Proses	3.2
	Nama	Penambahan Data Petunjuk .
	Source	Admin
	Input	Data petunjuk.
	Output	Informasi petunjuk.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data petunjuk maka sistem menampilkan form pengisian petunjuk.
15.	No. Proses	3.3
	Nama	Perubahan Data Petunjuk .
	Source	Admin
	Input	Data petunjuk.
	Output	Informasi petunjuk.
	Destination	Admin
	Logika Proses	Jika user ingin merubah data petunjuk maka sistem menampilkan form data petunjuk yang diinginkan.
16.	No. Proses	3.4
	Nama	Penghapusan Data Petunjuk .
	Source	Admin
	Input	Data petunjuk.
	Output	Informasi petunjuk.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data petunjuk maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
17.	No. Proses	4.1
	Nama	Lihat Profil.
	Source	Admin, pengguna.
	Input	Data Profil.
	Output	Informasi Profil.
	Destination	Admin, pengguna.
	Logika Proses	Jika user ingin lihat Profil maka sistem menampilkan form Profil yang berisi tentang profil unikom.
18.	No. Proses	4.2
	Nama	Penambahan Data Profil.
	Source	Admin
	Input	Data Profil.
	Output	Informasi Profil.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data Profil maka sistem menampilkan form pengisian Profil.
19.	No. Proses	4.3
	Nama	Perubahan Data Profil.
	Source	Admin
	Input	Data Profil.
	Output	Informasi Profil.
	Destination	Admin

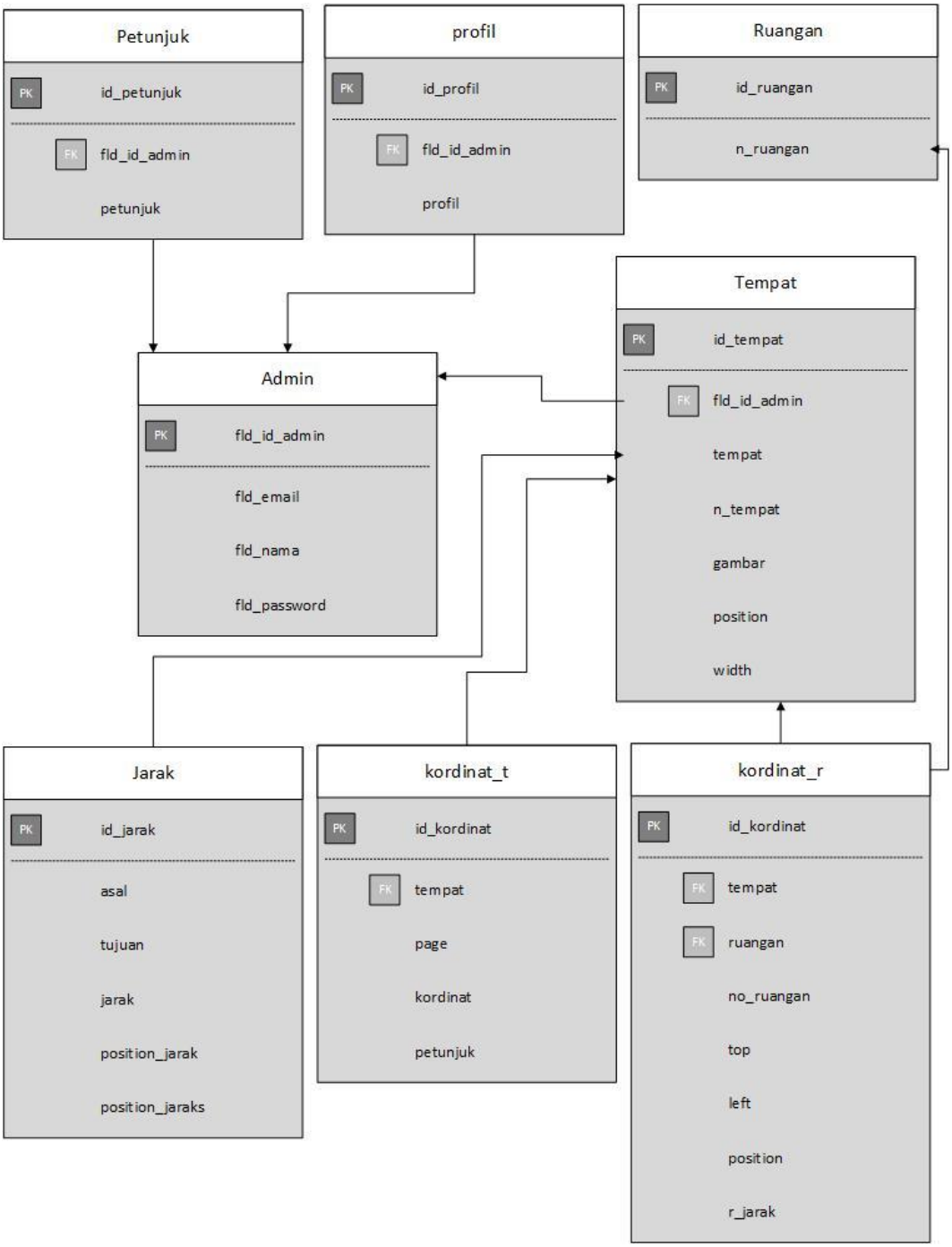
Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
	Logika Proses	Jika user ingin merubah data Profil maka sistem menampilkan form data Profil yang diinginkan.
20	No. Proses	4.4
	Nama	Penghapusan Data Profil.
	Source	Admin
	Input	Data Profil.
	Output	Informasi Profil.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data Profil maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
21.	No. Proses	2.2.1
	Nama	Penambahan Data Tempat.
	Source	Admin
	Input	Data Tempat.
	Output	Informasi Tempat.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data Tempat maka sistem menampilkan form pengisian Tempat.
22	No. Proses	2.2.2
	Nama	Perubahan Data Tempat.
	Source	Admin
	Input	Data Tempat.
	Output	Informasi Tempat.
	Destination	Admin
	Logika Proses	Jika user ingin merubah data Tempat maka sistem menampilkan form data Tempat yang diinginkan.
23.	No. Proses	2.2.3
	Nama	Penghapusan Data Tempat.
	Source	Admin
	Input	Data Tempat.
	Output	Informasi Tempat.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data Tempat maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
24.	No. Proses	2.3.1
	Nama	Penambahan Data Jarak.
	Source	Admin
	Input	Data Jarak.
	Output	Informasi Jarak.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data Jarak maka sistem menampilkan form pengisian Jarak.
25.	No. Proses	2.3.2
	Nama	Perubahan Data Jarak.
	Source	Admin
	Input	Data Jarak.
	Output	Informasi Jarak.
	Destination	Admin

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
	Logika Proses	Jika user ingin merubah data Jarak maka sistem menampilkan form data Jarak yang diinginkan.
26.	No. Proses	2.2.3
	Nama	Penghapusan Data Jarak.
	Source	Admin
	Input	Data Jarak.
	Output	Informasi Jarak.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data Jarak maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
27.	No. Proses	2.4.1
	Nama	Penambahan Data Ruangan.
	Source	Admin
	Input	Data Ruangan.
	Output	Informasi Ruangan.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data ruangan maka sistem menampilkan form pengisian ruangan.
28.	No. Proses	2.4.2
	Nama	Perubahan Data Ruangan.
	Source	Admin
	Input	Data Ruangan.
	Output	Informasi Ruangan.
	Destination	Admin
	Logika Proses	Jika user ingin merubah data ruangan maka sistem menampilkan form data ruangan yang diinginkan.
29.	No. Proses	2.4.3
	Nama	Penghapusan Data Ruangan.
	Source	Admin
	Input	Data Ruangan.
	Output	Informasi Ruangan.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data ruangan maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
30.	No. Proses	2.5.1
	Nama	Penambahan Data Kordinat_r.
	Source	Admin
	Input	Data Kordinat_r.
	Output	Informasi Kordinat_r.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data Kordinat_r maka sistem menampilkan form pengisian Kordinat_r.
31.	No. Proses	2.4.2
	Nama	Perubahan Data Kordinat_r.
	Source	Admin
	Input	Data Kordinat_r.
	Output	Informasi Kordinat_r.
	Destination	Admin

Tabel Spesifikasi Proses DFD		
No	**Proses**	**Keterangan**
	Logika Proses	Jika user ingin merubah data Kordinat_r maka sistem menampilkan form data Kordinat_r yang diinginkan.
32.	No. Proses	2.4.3
	Nama	Penghapusan Data Kordinat_r.
	Source	Admin
	Input	Data Kordinat_r.
	Output	Informasi Kordinat_r.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data Kordinat_r maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.
33.	No. Proses	2.6.1
	Nama	Penambahan Data Kordinat_t.
	Source	Admin
	Input	Data Kordinat_t.
	Output	Informasi Kordinat_t.
	Destination	Admin
	Logika Proses	Jika user ingin menambah data Kordinat_t maka sistem menampilkan form pengisian Kordinat_t.
34.	No. Proses	2.6.2
	Nama	Perubahan Data Kordinat_t.
	Source	Admin
	Input	Data Kordinat_t.
	Output	Informasi Kordinat_t.
	Destination	Admin
	Logika Proses	Jika user ingin merubah data Kordinat_t maka sistem menampilkan form data Kordinat_t yang diinginkan.
35.	No. Proses	2.6.3
	Nama	Penghapusan Data Kordinat_t.
	Source	Admin
	Input	Data Kordinat_t.
	Output	Informasi Kordinat_t.
	Destination	Admin
	Logika Proses	Jika user ingin menghapus data Kordinat_t maka sistem menampilkan pertanyaan. Apakah yakin menghapus data tersebut. Jika memilih tidak akan kembali ke form penghapusan data dan memilih ia akan di hapus.

3.2.4 Skema Relasi

Proses relasi antar *file* merupakan gabungan antar *file* yang memiliki *primary key* (kunci utama) yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* (atribut) kunci tersebut. Pada proses ini elemen-elemen data dikelompokkan menjadi satu *file database* beserta *entitas* dan hubungannya. Skema relasi pembangunan aplikasi pemandu lokasi Universitas Komputer Indonesia (UNIKOM) adalah sebagai berikut:



Gambar 3.36 Skema Relasi

3.2.5 Struktur tabel

Berikut adalah tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem informasi yang akan dibangun:

Tabel 3.17 Admin

Nama Field	Type	Length	Keterangan
Fld_id_admin	Int	1	Primary Key
Fld_email	Varchar	100	
Fld_nama	Varchar	50	
Fld_password	Varchar	50	

Tabel 3.18 Ruangan

Nama Field	Type	Length	Keterangan
Id_ruangan	Int	5	Primary Key
N_ruangan	Varchar	25	

Tabel 3.19 Jarak

Nama Field	Type	Length	Keterangan
Id_jarak	Int	11	Primary Key
Asal	Int	4	
tujuan	Int	4	
jarak	Int	4	
Position_jarak	Int	5	
Position_jaraks	Int	5	

Tabel 3.20 Tempat

Nama Field	Type	Length	Keterangan
Id_tempat	Int	11	Primary Key
id_admin	Int	1	
tempat	Varchar	50	
n_tempat	Varchar	50	

gambar	Varchar	150	
position	Int	10	
width	Int	6	

Tabel 3.21 Kordinat_r

Nama Field	Type	Length	Keterangan
Id_kordinat	Int	5	Primary Key
tempat	Int	5	
ruangan	Varchar	50	
No_ruangan	Varchar	50	
top	Int	5	
left	Int	5	
position	Int	10	
r_jarak	Int	3	

Tabel 3.22 Kordinat_t

Nama Field	Type	Length	Keterangan
Id_kordinat	Int	5	Primary Key
tempat	Int	5	
page	Varchar	50	
Kordinat	Varchar	20	
petunjuk	Varchar	50	

Tabel 3.23 Profil

Nama Field	Type	Length	Keterangan
id_profil	Int	1	Primary Key
profil	Text		

Tabel 3.24 Petunjuk

Nama Field	Type	Length	Keterangan
id_petunjuk	Int	1	Primary Key
petunjuk	Text		

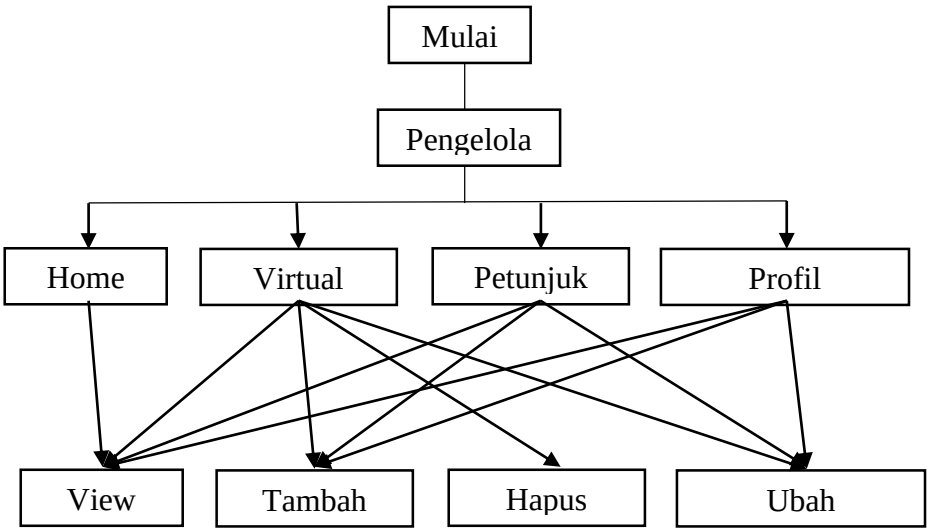
3.2.6 Perancangan Arsitektur

Perancangan arsitektur merupakan perancangan yang dibuat sebelum program aplikasi dibuat. Dengan perancangan arsitektur akan mempermudah proses pembangunan web.

3.2.6.1 Perancangan struktur menu

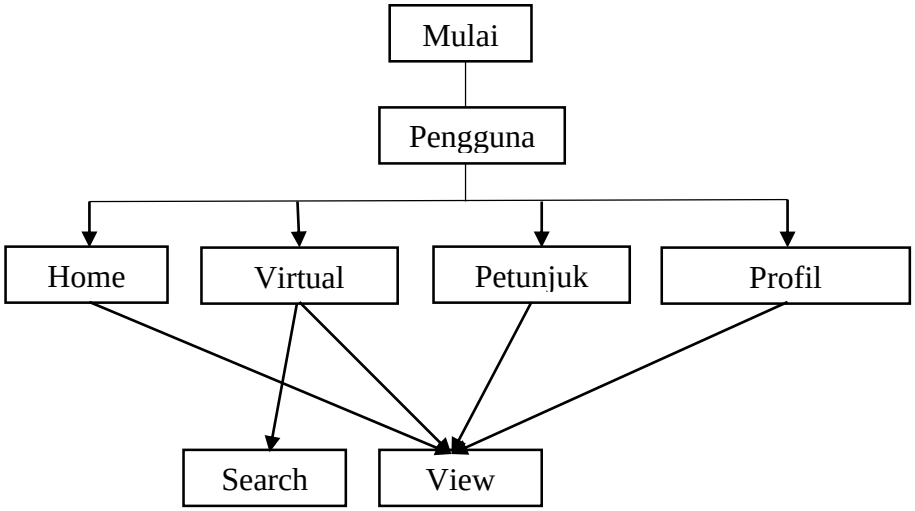
Berikut adalah perancangan struktur menu :

a. Struktur menu pengelola



Gambar 3.37 Struktur menu administrator

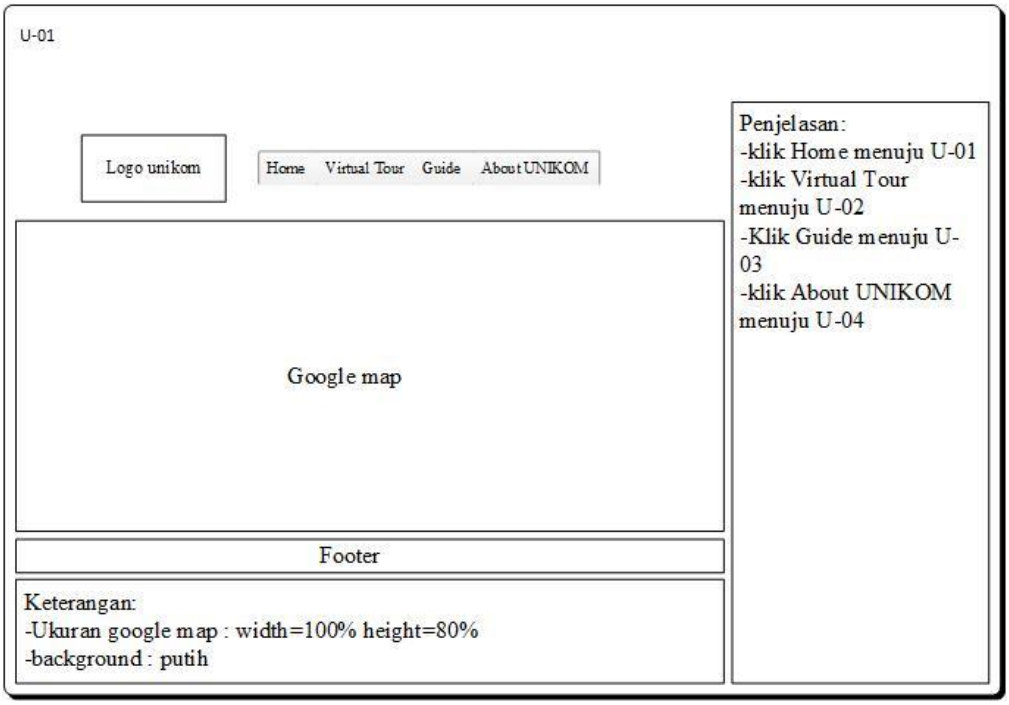
b. Struktur menu pengguna



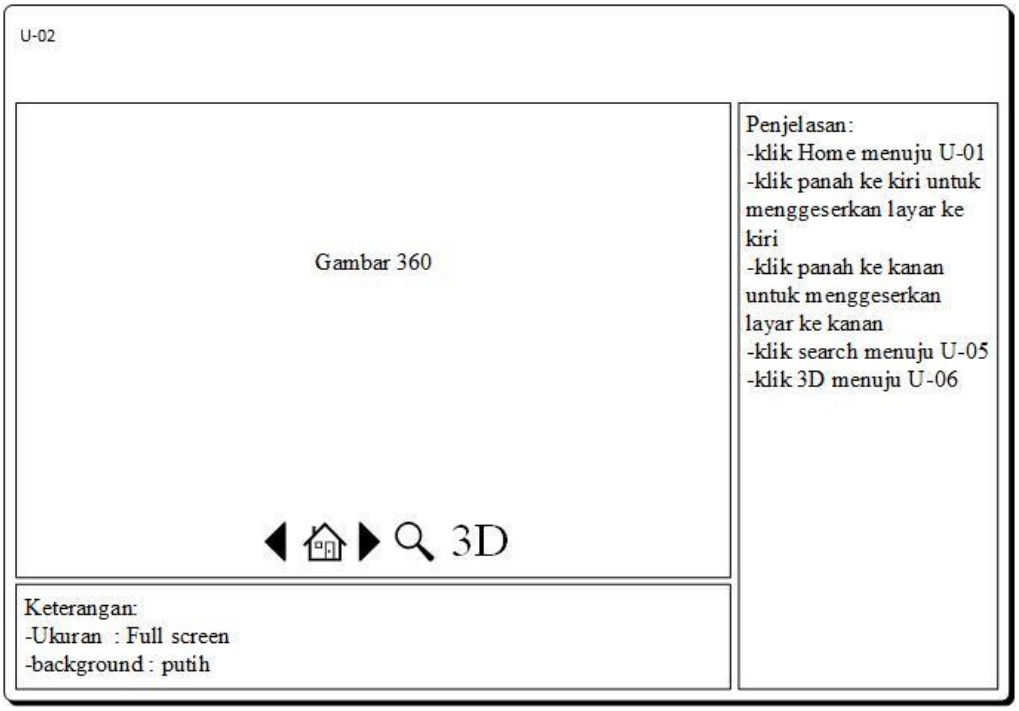
Gambar 3.38 Struktur menu pengguna

3.2.6.2 Perancangan antar muka

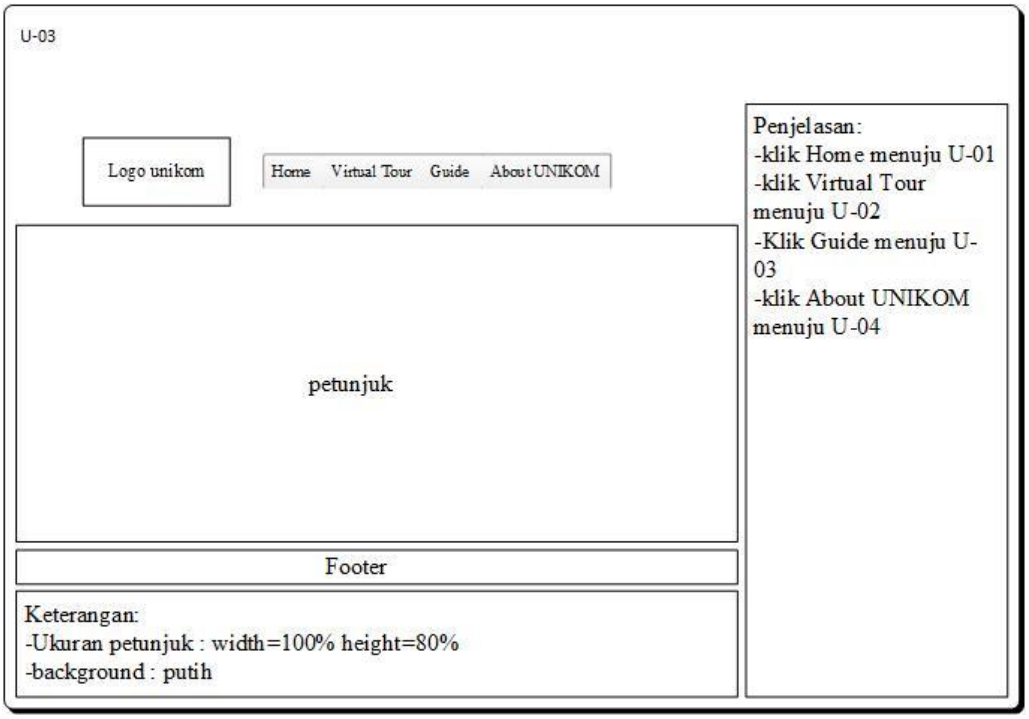
3.2.6.2.1 Perancangan antar muka halaman utama



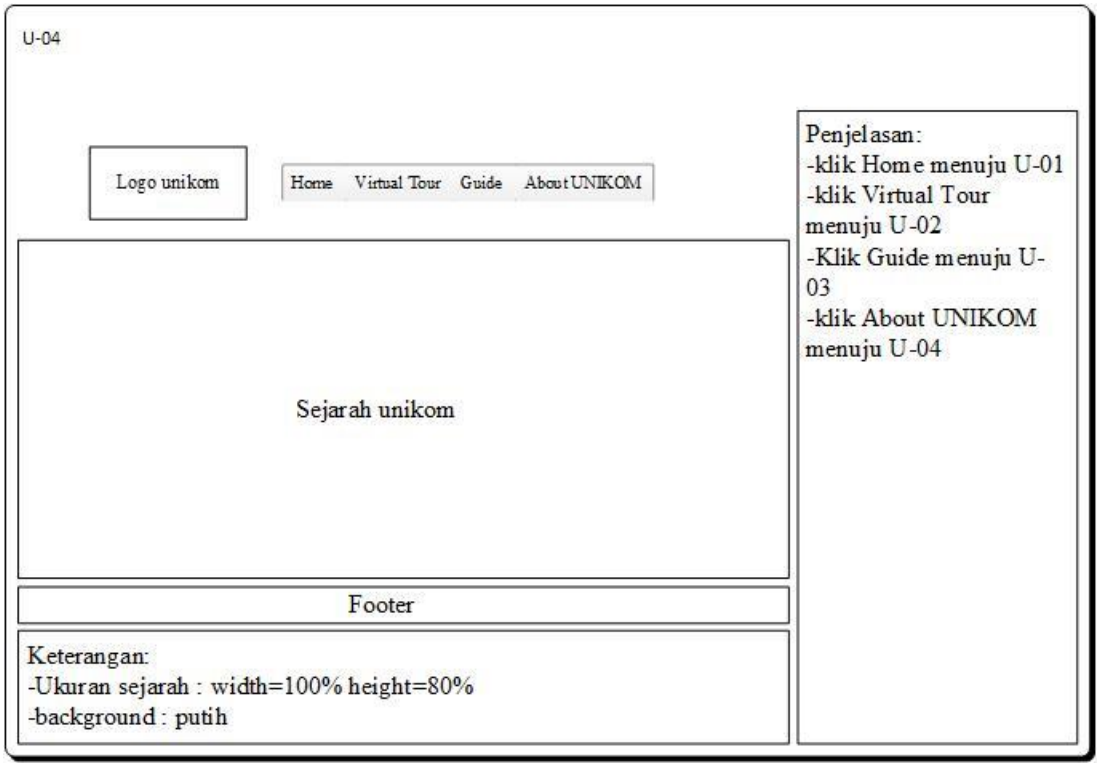
Gambar 3.39 Halaman utama



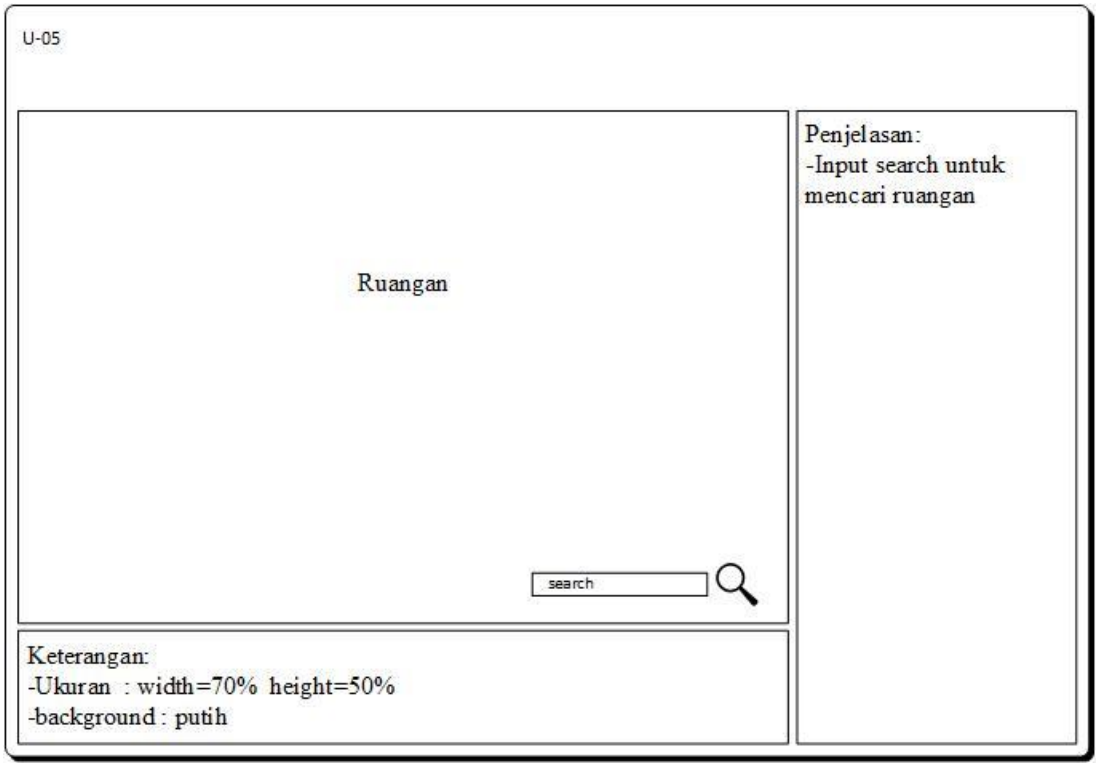
Gambar 3.40 Halaman Virtual Tour



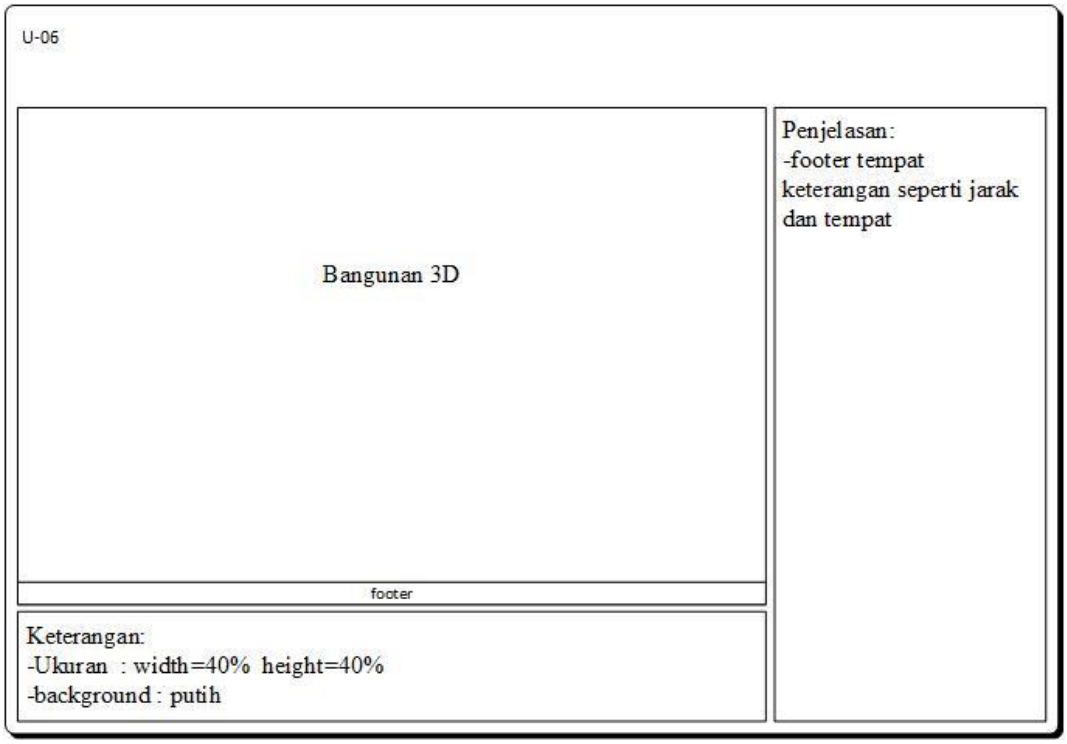
Gambar 3.41 Halaman Guide



Gambar 3.42 Halaman About UNIKOM

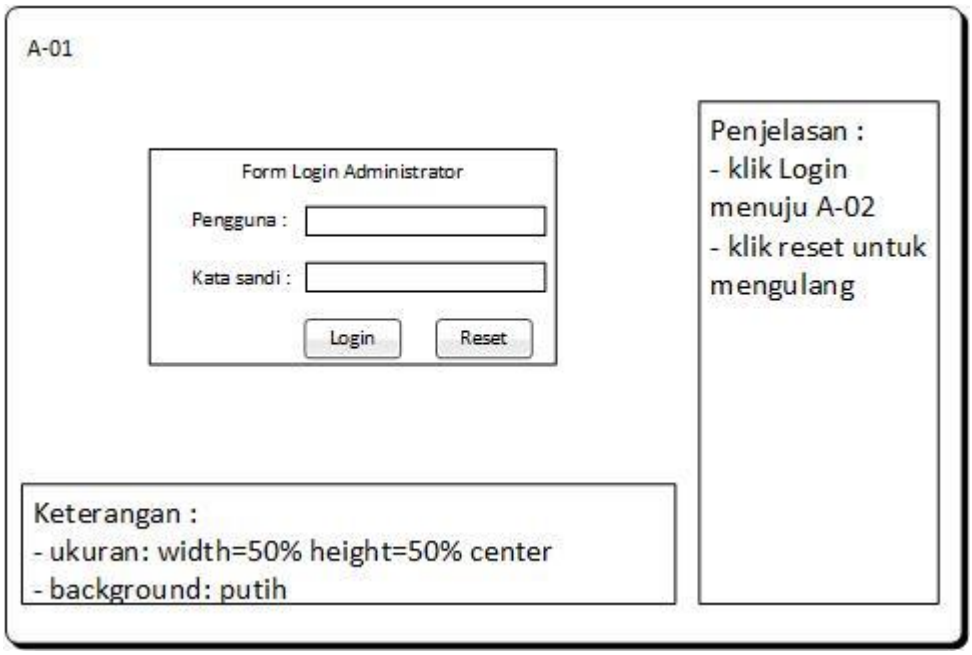


Gambar 3.43 Halaman mencari ruangan



Gambar 3.44 Halaman Bangunan 3D UNIKOM

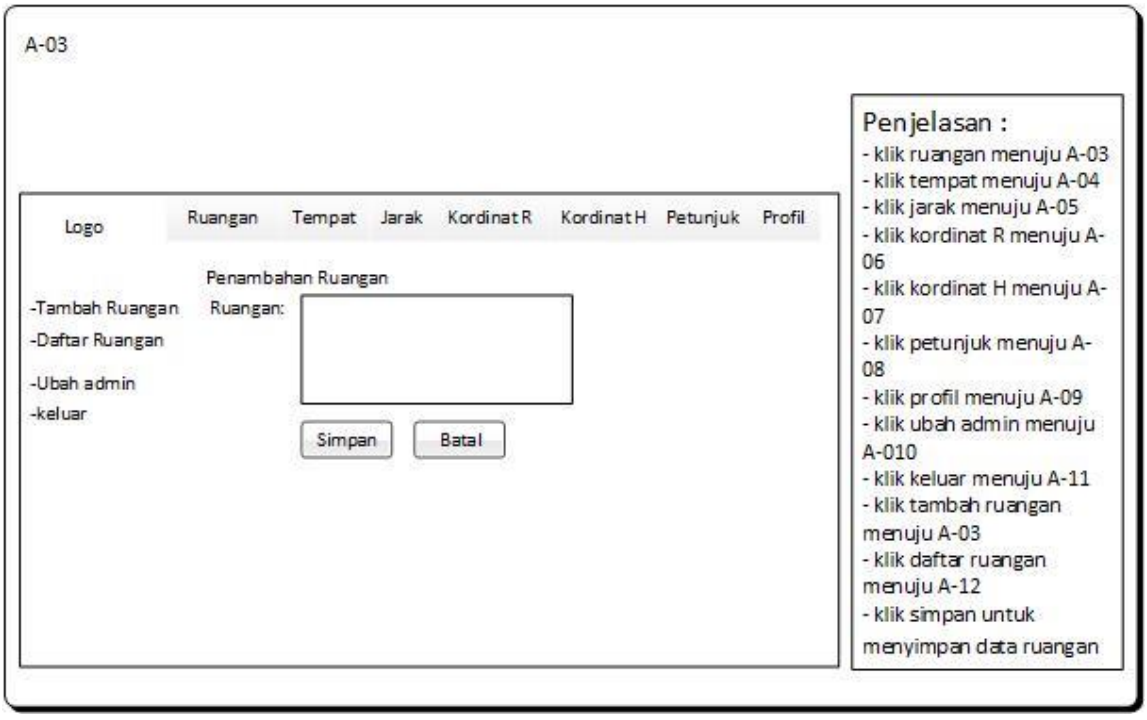
3.2.6.2.2 Perancangan antar muka halaman admin



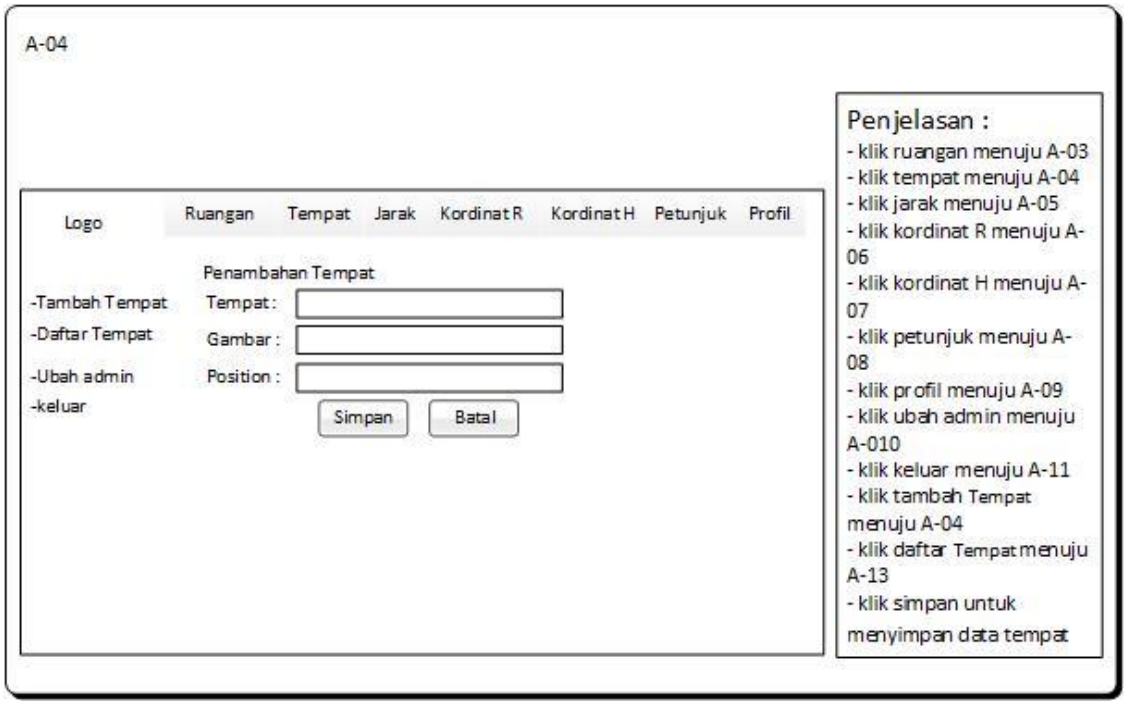
Gambar 3.45 Halaman admin login



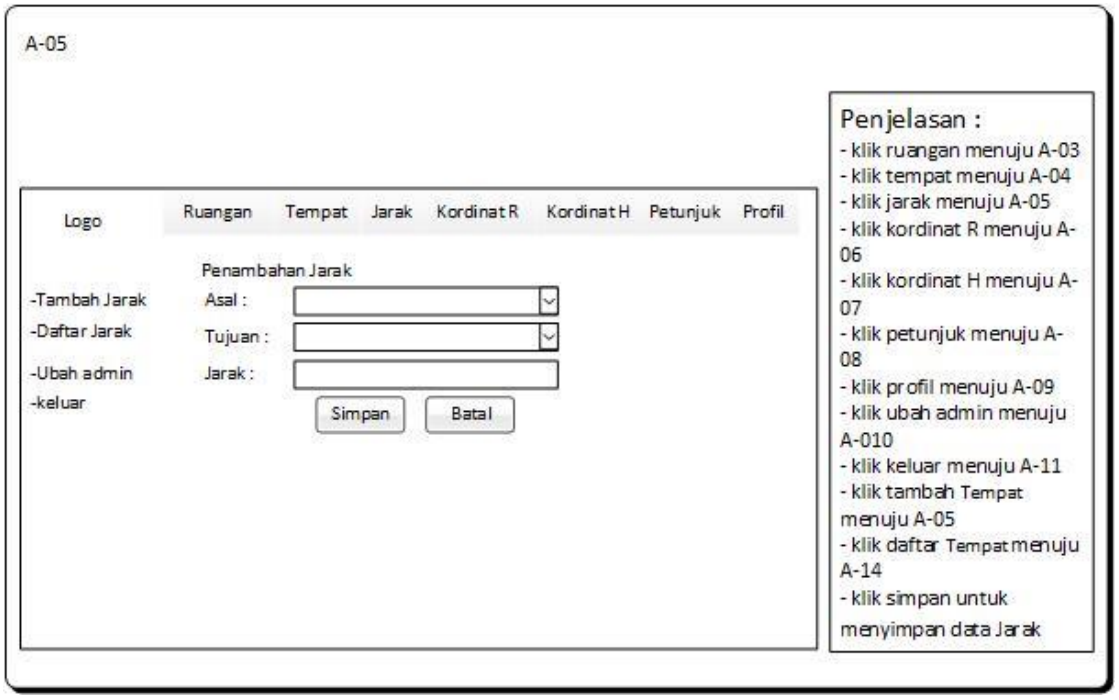
Gambar 3.46 Halaman admin Home



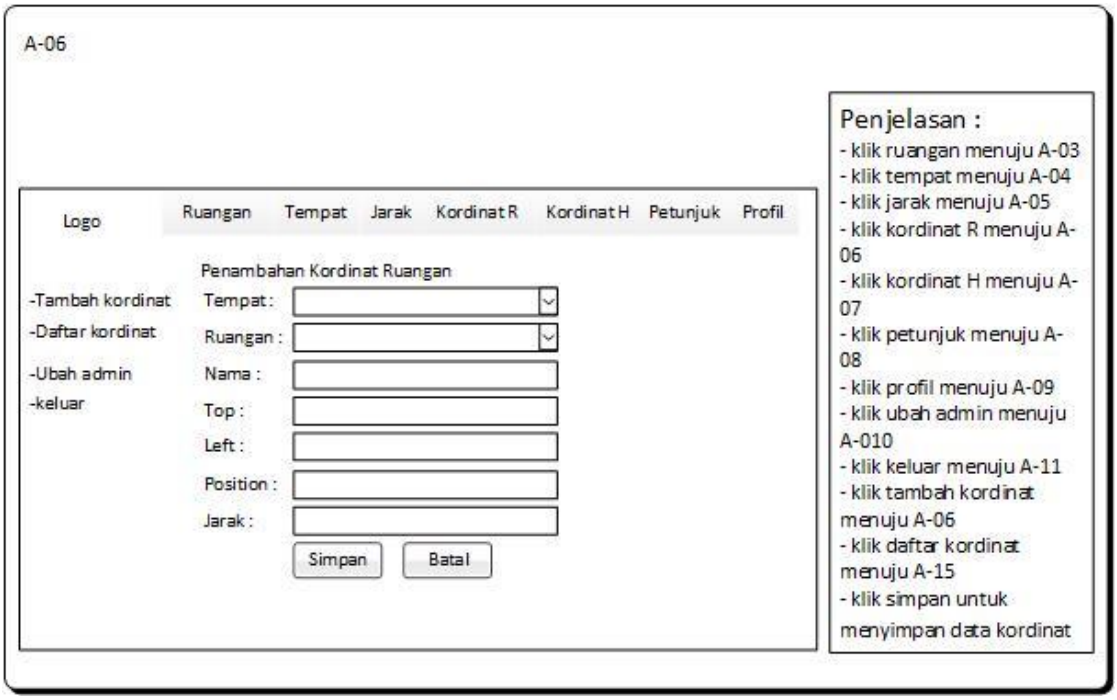
Gambar 3.47 Halaman Ruangan



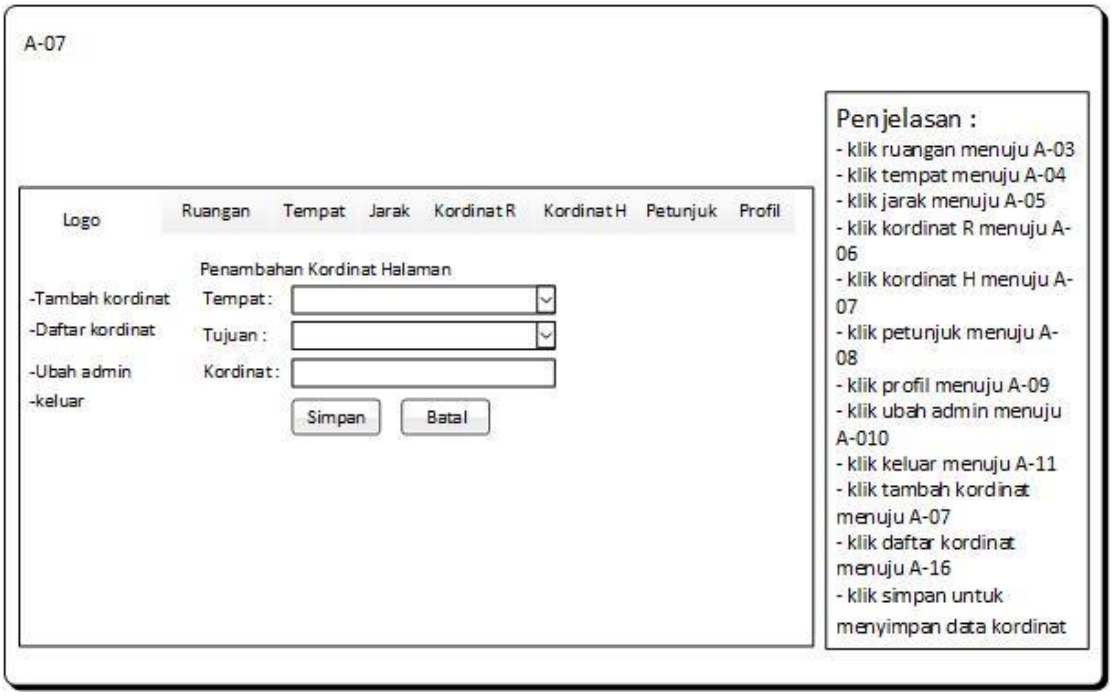
Gambar 3.48 Halaman Tempat



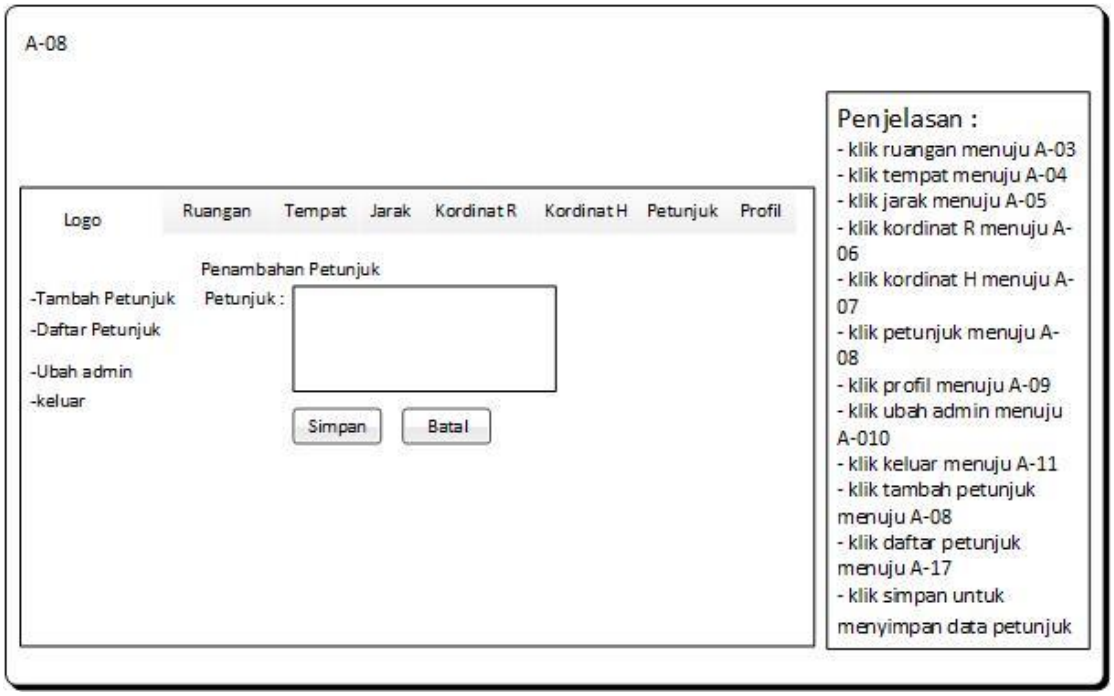
Gambar 3.49 Halaman Jarak



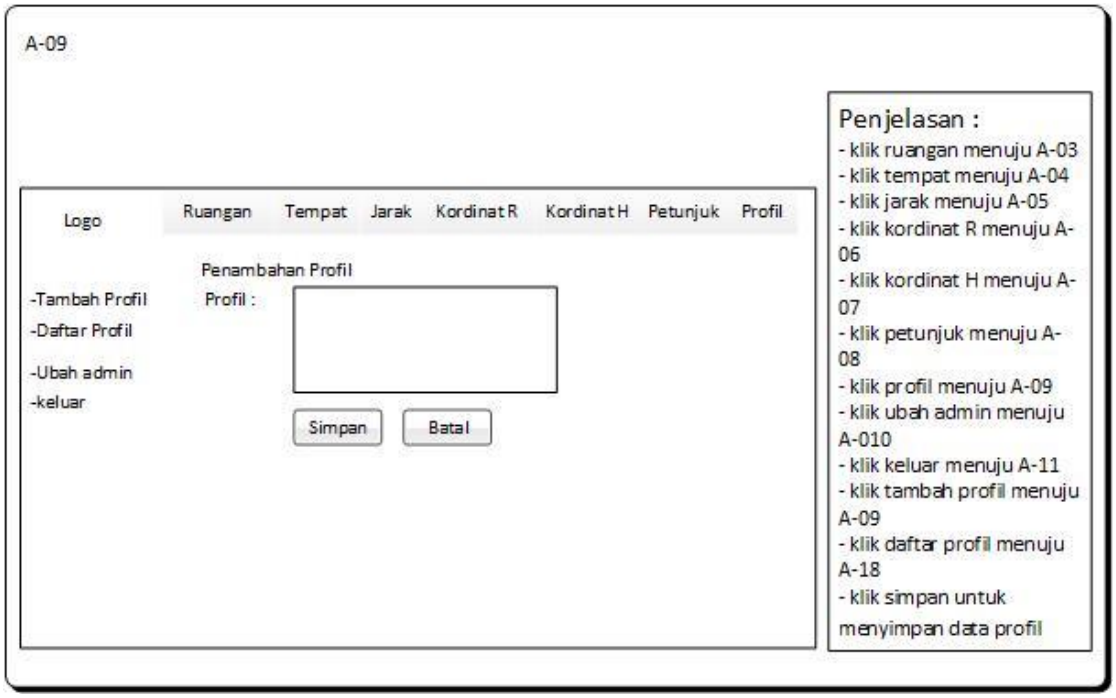
Gambar 3.50 Halaman Kordinat R



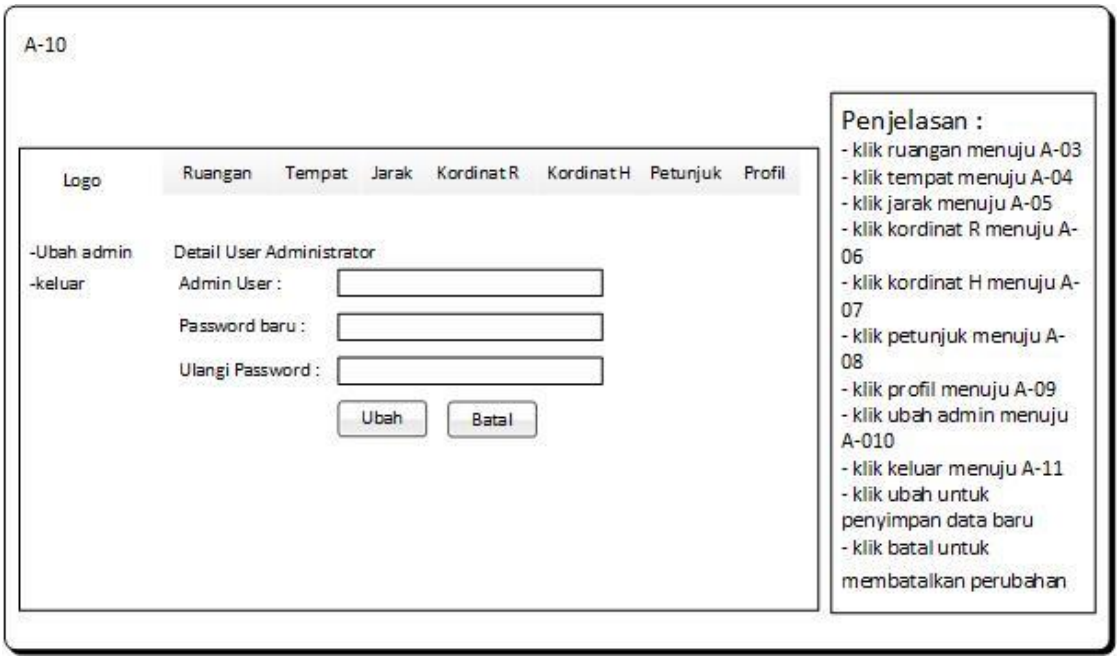
Gambar 3.51 Halaman Kordinat H



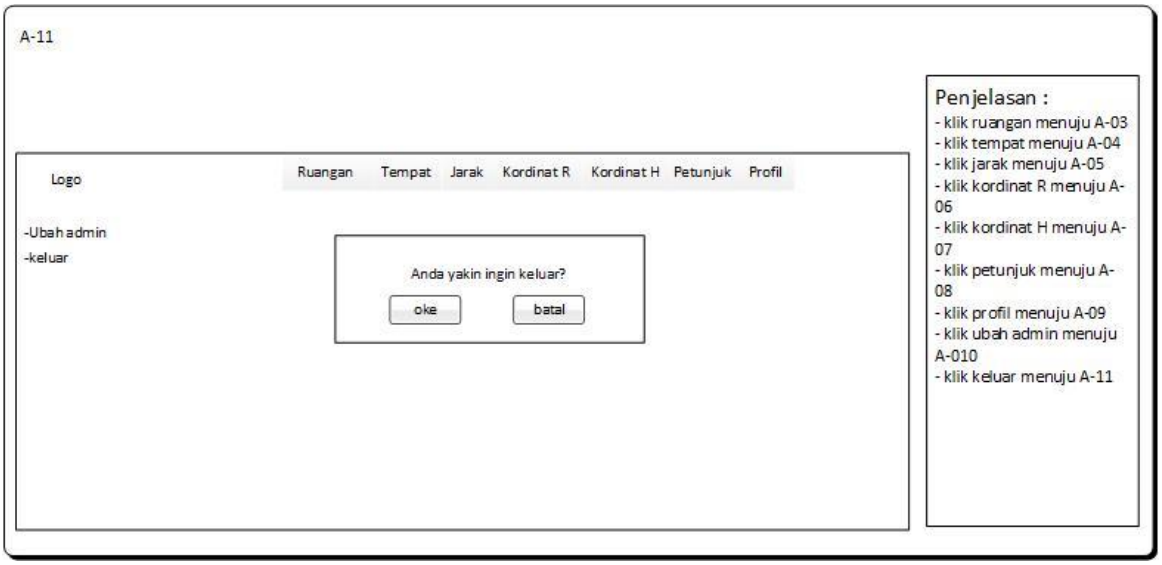
Gambar 3.52 Halaman Petunjuk



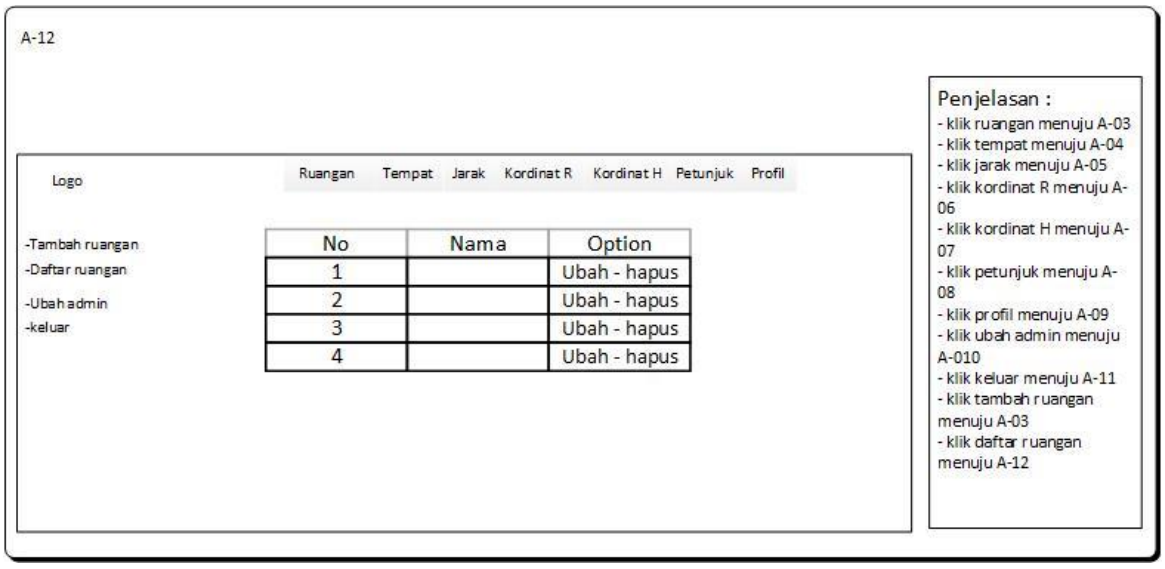
Gambar 3.53 Halaman Profil



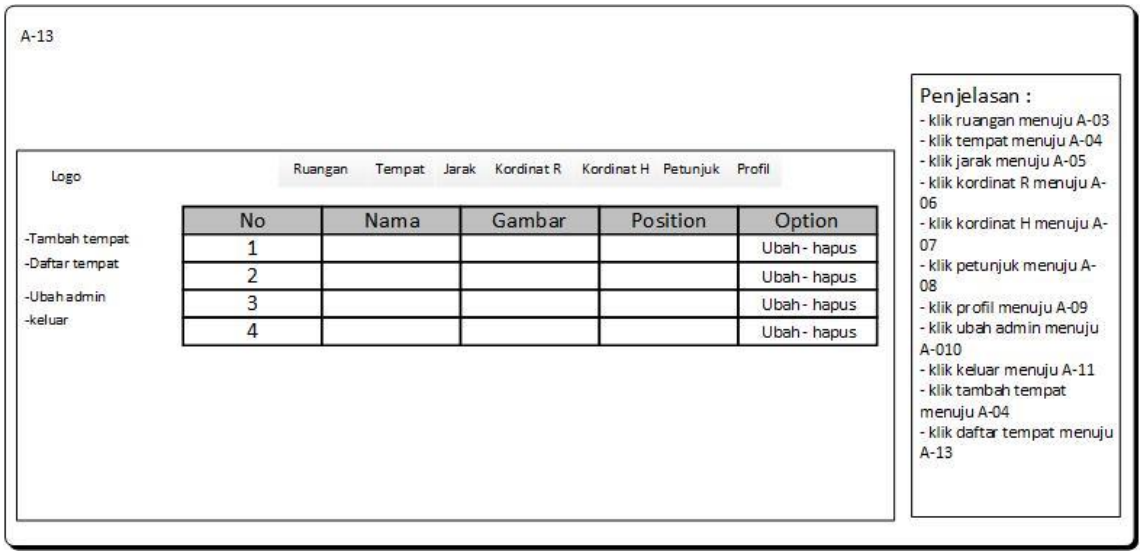
Gambar 3.54 Halaman Ubah Admin



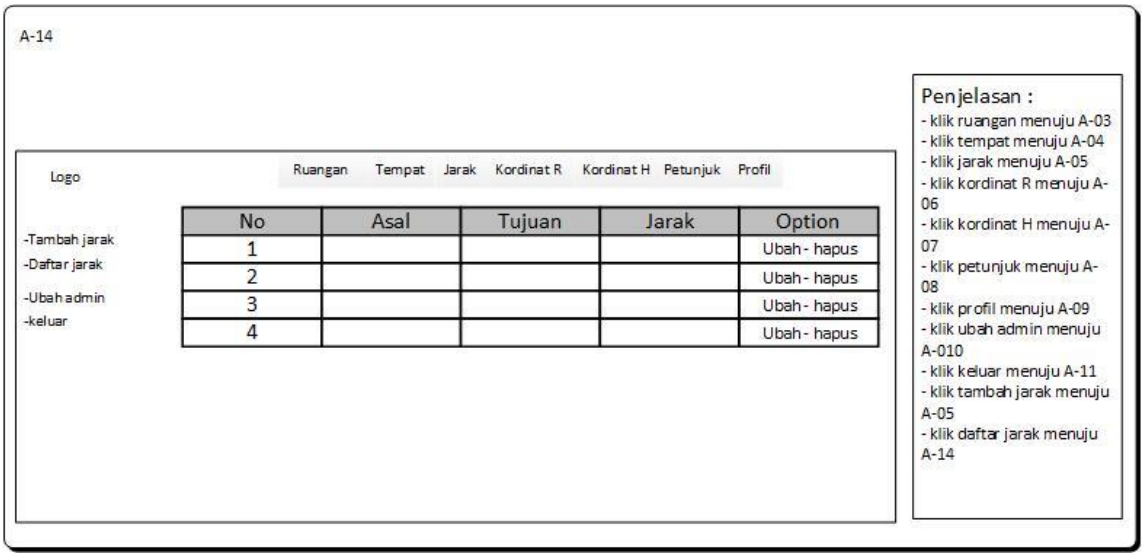
Gambar 3.55 Halaman Keluar



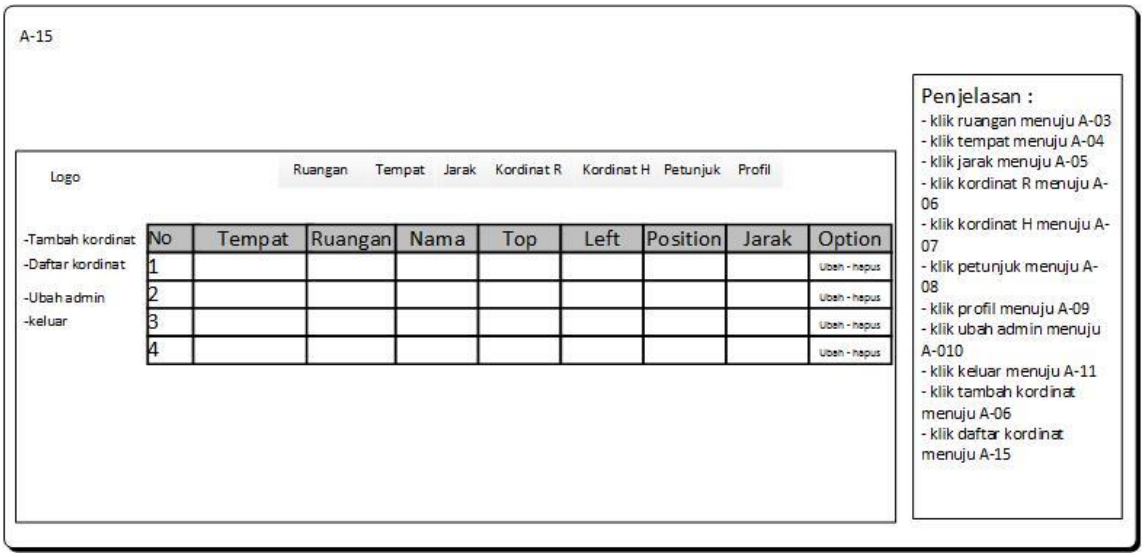
Gambar 3.56 Halaman Daftar Ruangan



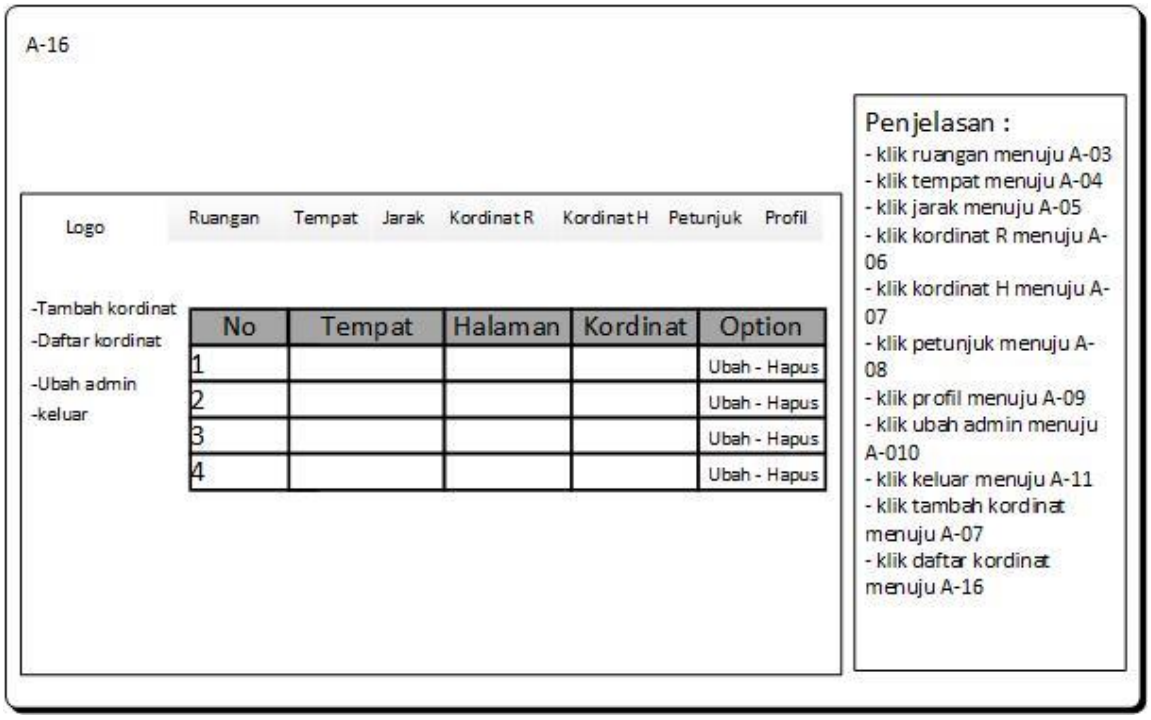
Gambar 3.57 Halaman Daftar Tempat



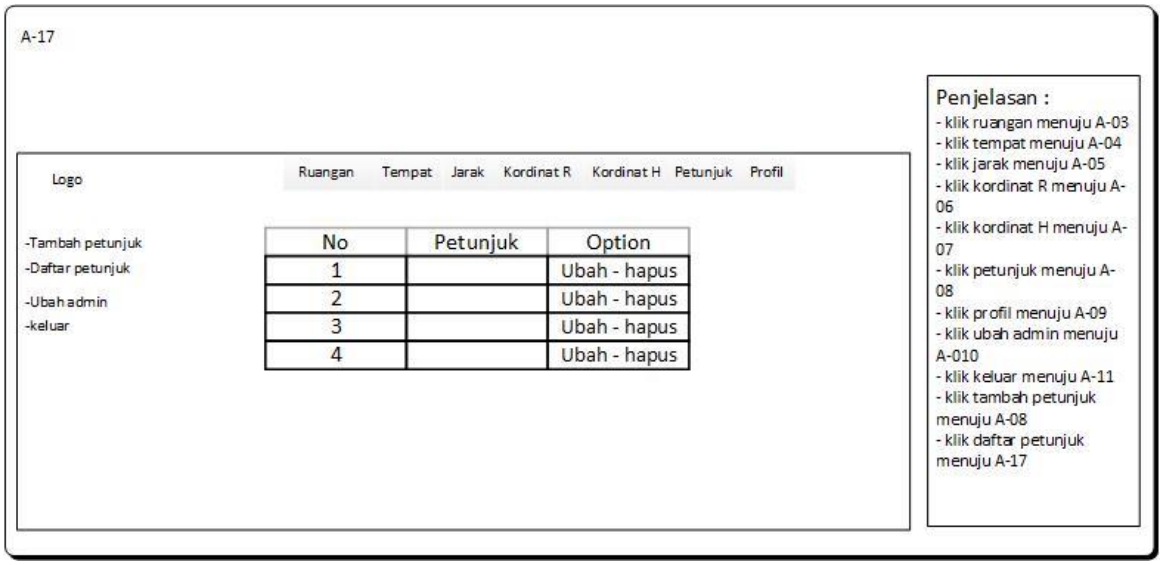
Gambar 3.58 Halaman Daftar Jarak



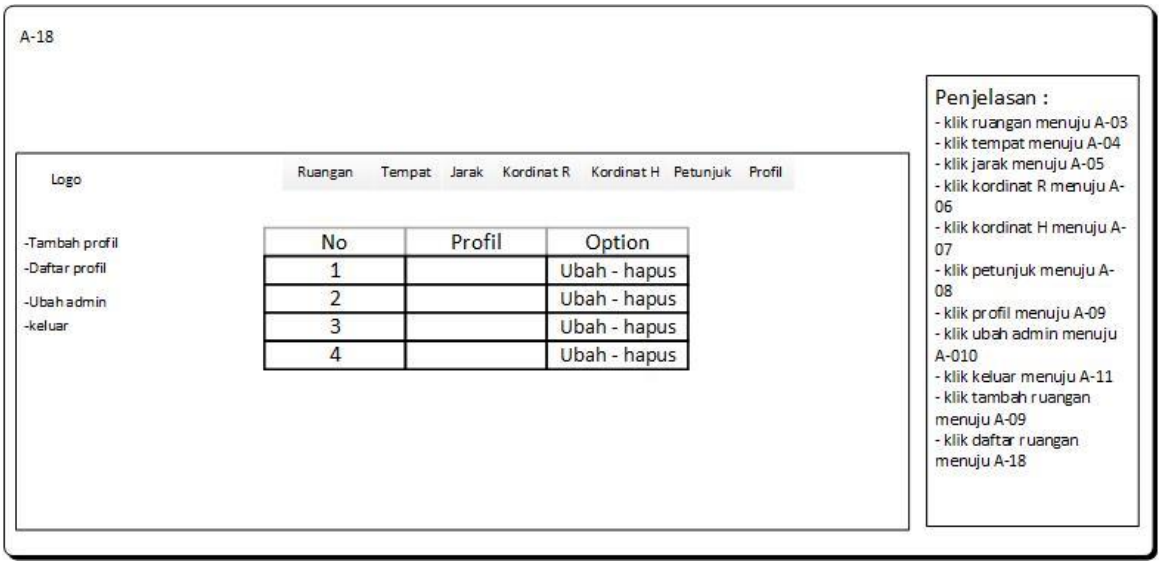
Gambar 3.59 Halaman Daftar Kordinat R



Gambar 3.60 Halaman Daftar Kordinat H



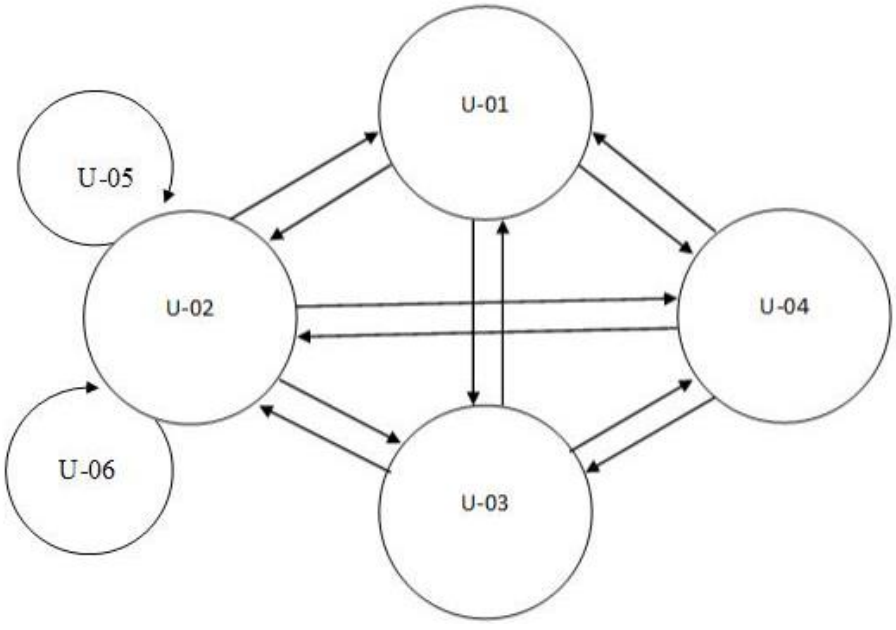
Gambar 3.61 Halaman Daftar Petunjuk



Gambar 3.62 Halaman Daftar Profil

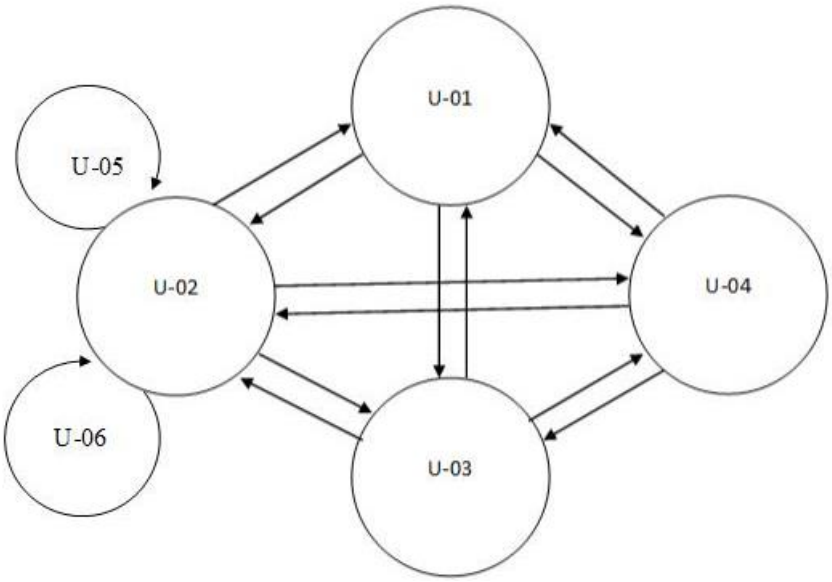
3.2.7 Jaringan Simantik

3.2.7.1 Jaringan Simantik Halaman Utama Untuk Pengguna



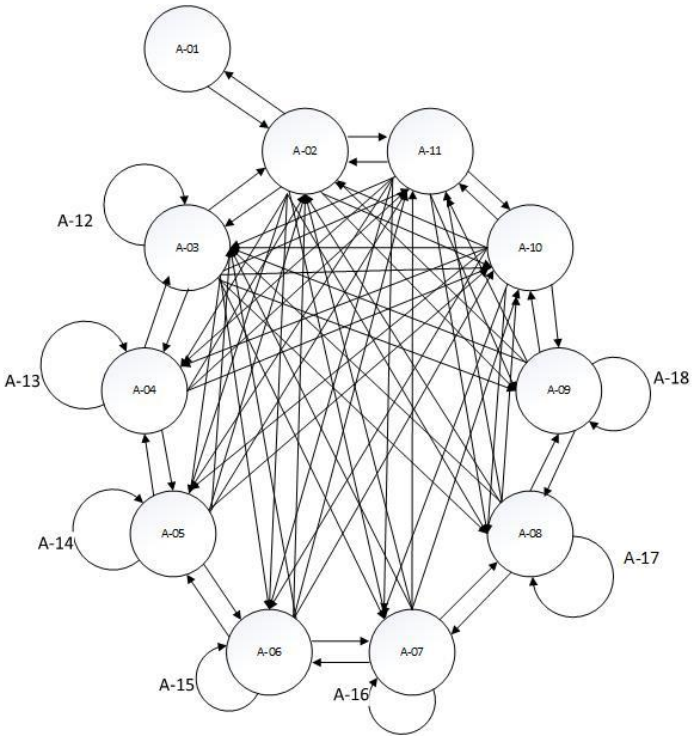
Gambar 3.63 Jaringan Simantik Halaman Utama Untuk Pengguna

3.2.7.2 Jaringan Simantik Halaman Utama Untuk Admin



Gambar 3.64 Jaringan Simantik Halaman Utama Untuk Admin

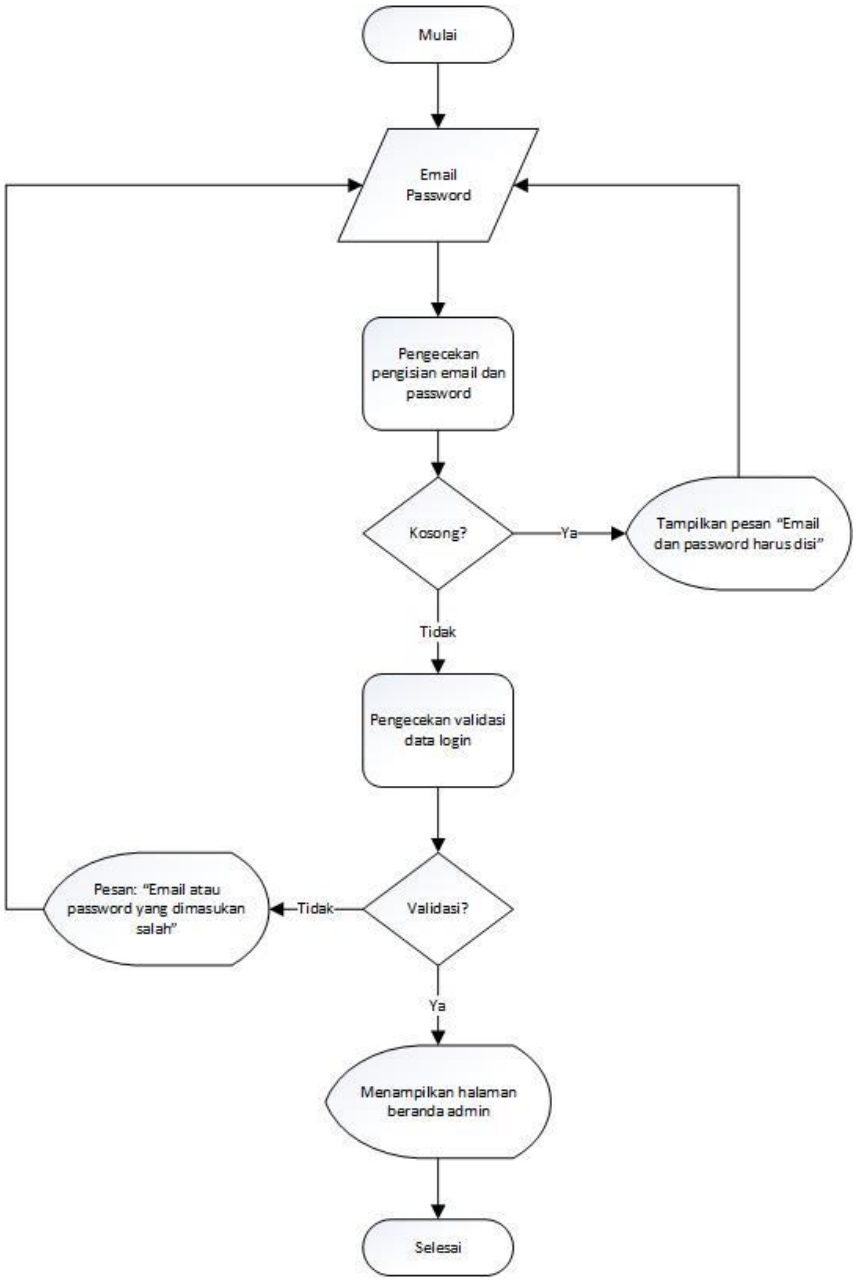
3.2.7.3 Jaringan Simantik Halaman Admin



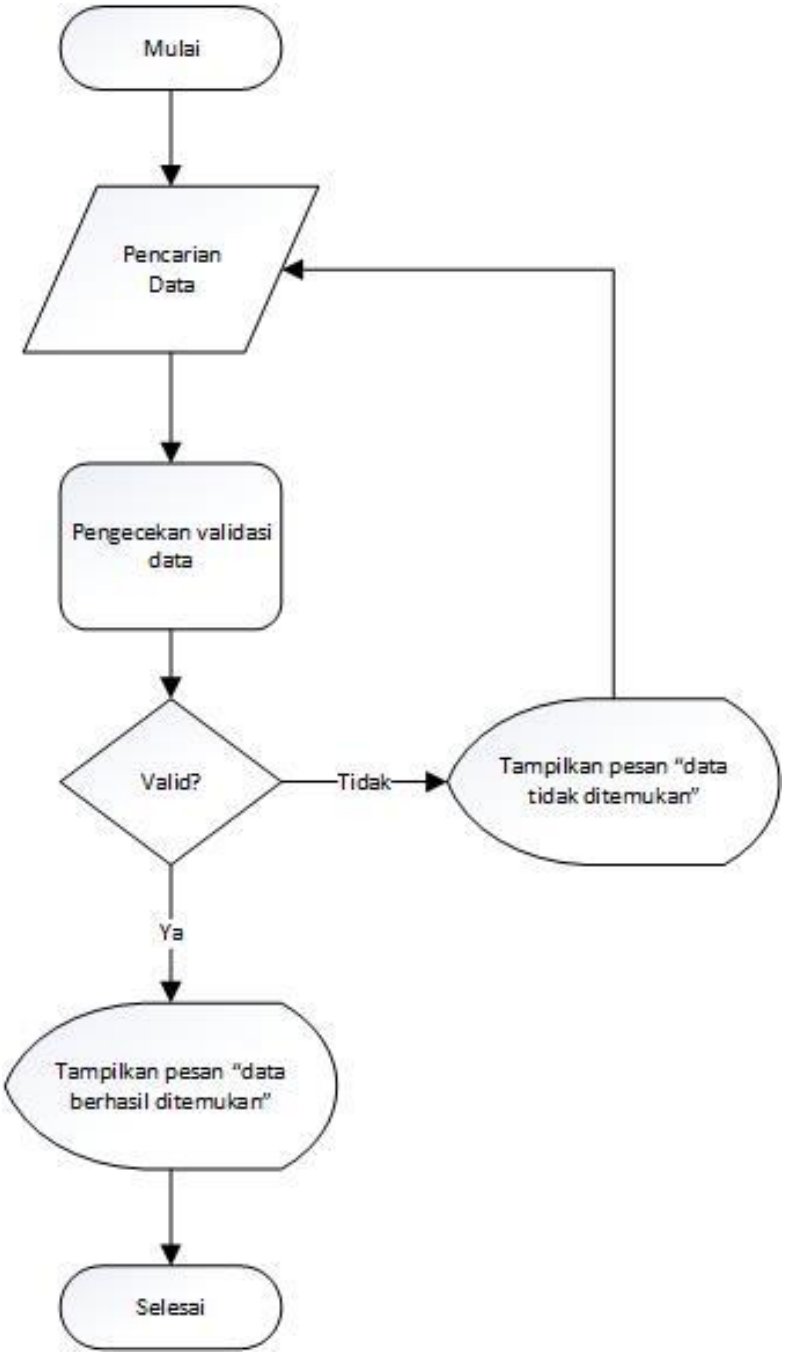
Gambar 3.65 Jaringan Simantik Halaman Admin

3.2.8 Perancangan Prosedural

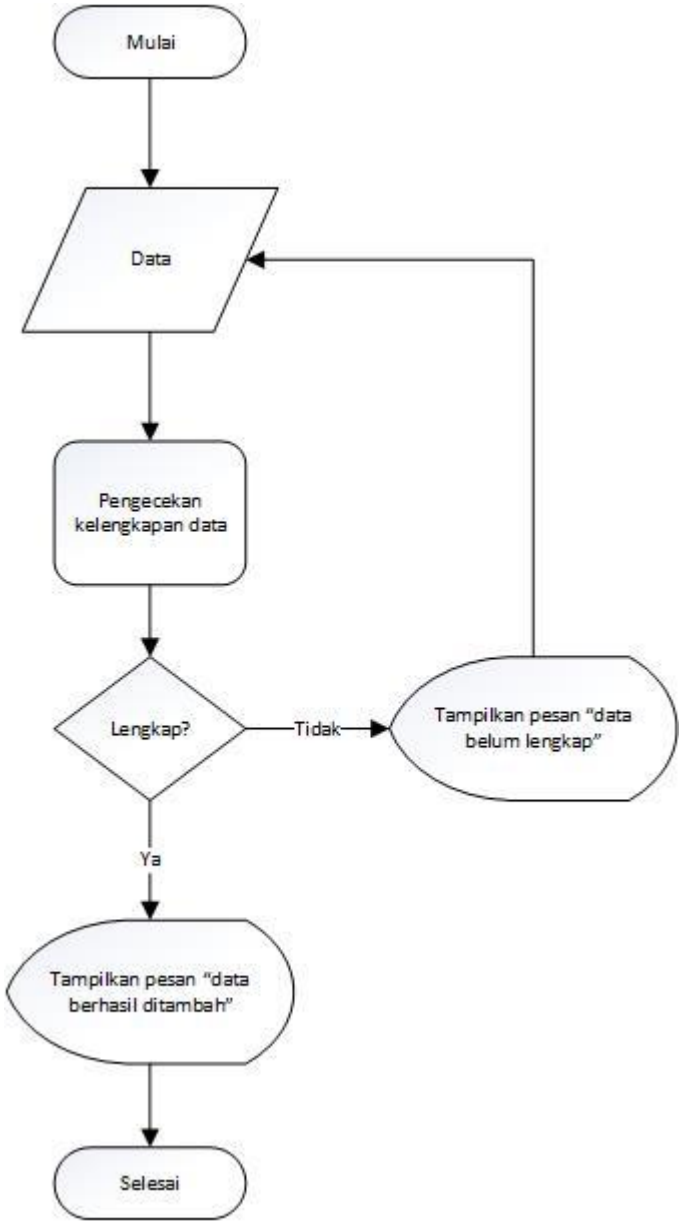
Perancangan prosedural adalah perancangan yang merepresentasikan algoritma dari perangkat lunak dengan tipe operasi yang berbeda-beda dengan menggunakan *flowchart* yang merupakan diagram dengan simbol-simbol. Untuk setiap prosedur yang melakukan interaksi terhadap sistem dengan menginputkan pada *textbox* diasumsikan telah melewati validasi *textbox* tersebut harus diisi.



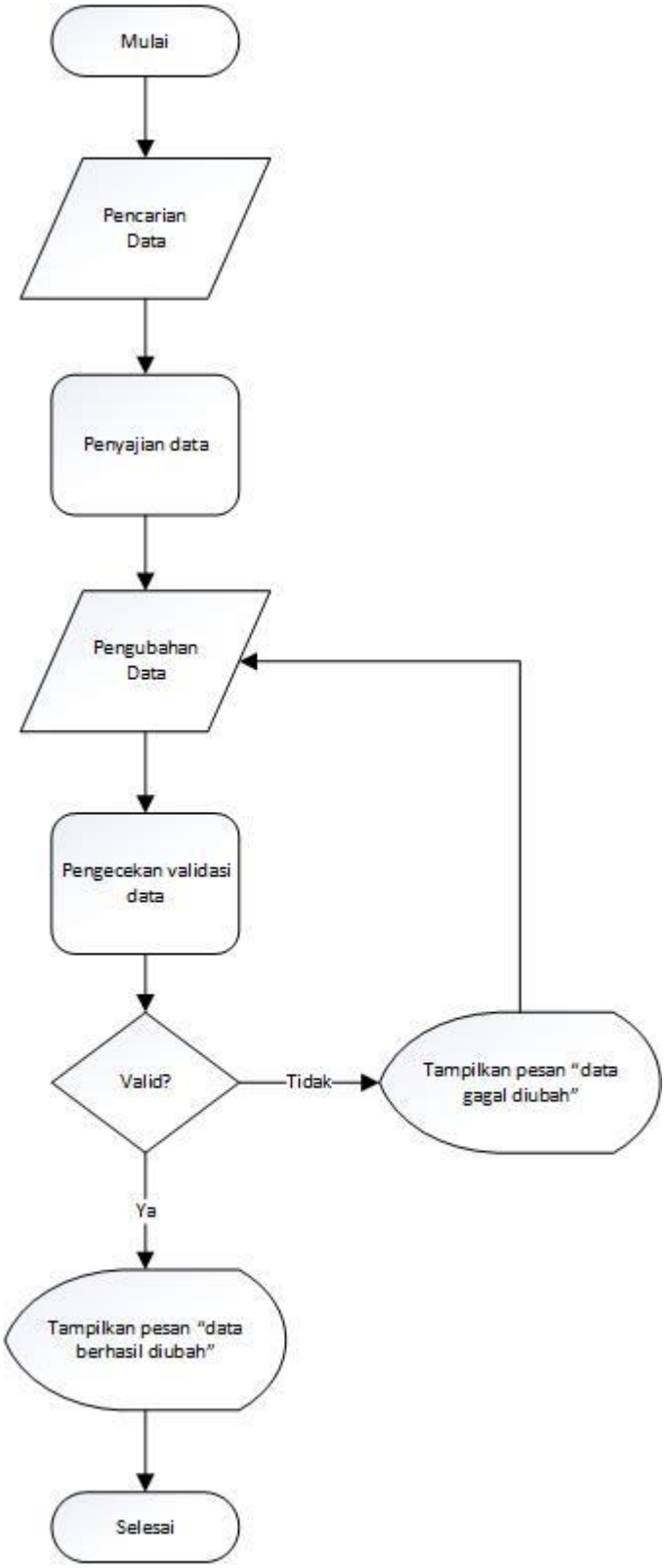
Gambar 3.66 Perancangan Prosedural Login



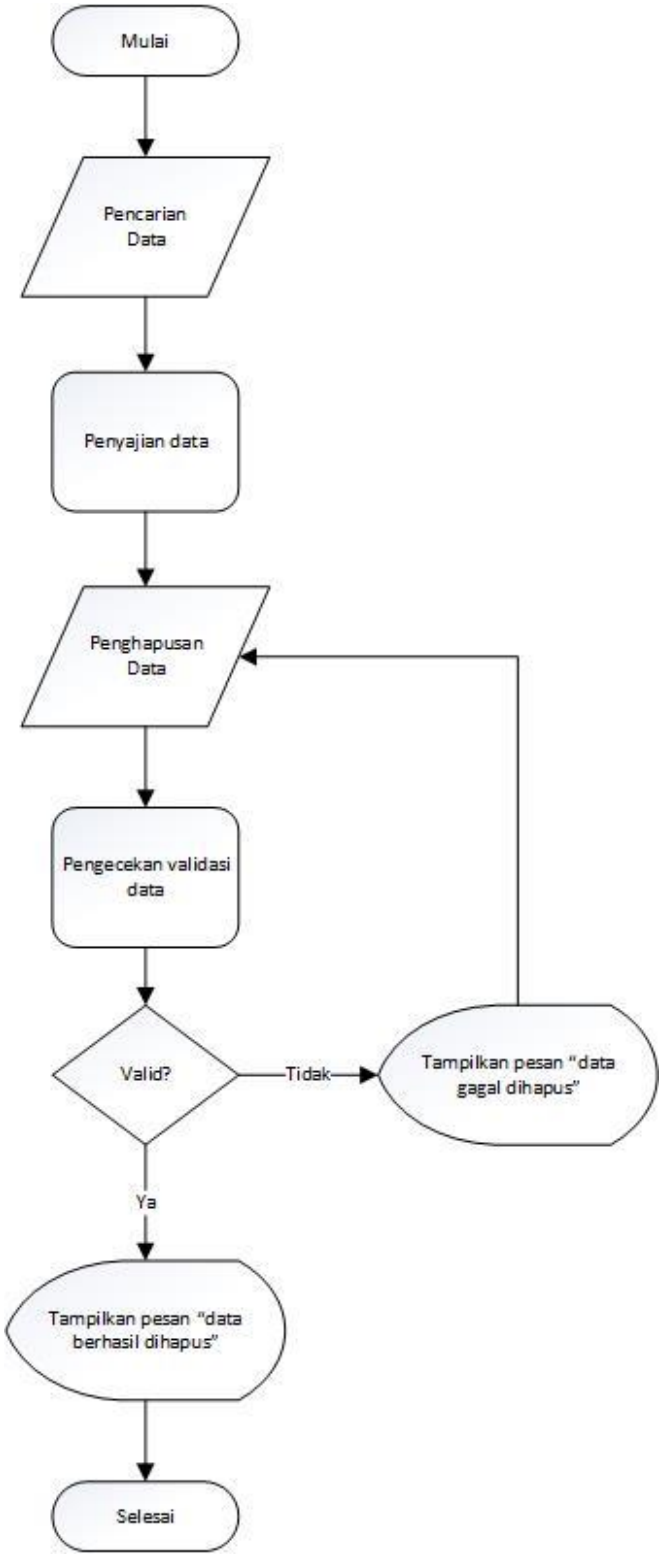
Gambar 3.67 Perancangan Prosedural Pencarian



Gambar 3.68 Perancangan Prosedural Penambahan Data



Gambar 3.69 Perancangan Prosedural Perubahan Data



Gambar 3.70 Perancangan Prosedural Penghapusan Data