BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Berjalan

Analisis sistem yang sedang berjalan merupakan peninjauan atau analisis terhadap sistem yang berjalan yang didalamnya terdapat urutan kegiatan yang tepat dari tahapan-tahapan yang menerangkan proses yang dikerjakan, siapa yang mengerjakan proses tersebut, bagaimana proses itu dapat dikerjakan.

4.1.1 Analisis Prosedur yang Sedang Berjalan

Analisis sistem yang sedang berjalan pada aplikasi Informasi Transportasi Umum bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem untuk dapat dijadikan landasan usulan rancangan analisis sistem yang sedang berjalan yang dilakukan berdasarkan fungsionalitas yang ada pada aplikasi tersebut dapat dibuat diagram *Use Case*.

Adapun prosedur kerja dari sistem aplikasi yang sebelumnya adalah sebagai berikut :

1. Pengguna aplikasi membuka aplikasi transportasi umum kota Jakarta
2. Pengguna mengakses menu Info Bus Transjakarta pada aplikasi transportasi umum kota Jakarta guna mendapatkan informasi seputar Transjakarta.
3. Pengguna mengakses menu Info Taksi pada aplikasi transportasi umum kota Jakarta guna mendapatkan informasi seputar taksi.

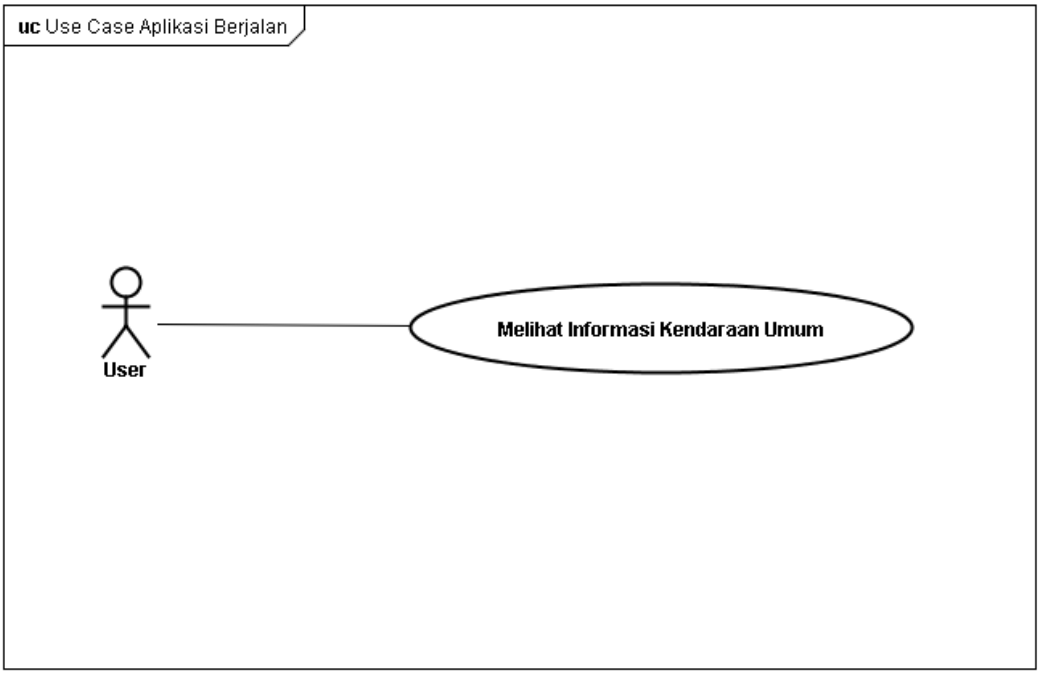
4. Pengguna mengakses menu Info angkot pada aplikasi transportasi umum kota Jakarta guna mendapatkan informasi seputar angkot.

4.1.1.1 *Use Case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk kelakukan *(behavior)* sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

4.1.1.1.1 *Use Case* Diagram pada Sistem Berjalan

Diagram *Use Case* pada aplikasi Informasi Transportasi Umum adalah sebagai berikut :



Gambar 4.1 *Use Case* Berjalan Aplikasi Informasi Transportasi Umum

1. Identifikasi *Use Case*

Identifikasi *Use Case* adalah *Use Case* yang bekerja pada sistem aplikasi dan *Use Case* yang bekerja pada sistem adalah sebagai berikut :

Tabel 4.1 Deskripsi *Use Case* Pengguna Sistem Berjalan

No	*Use Case*	Deskripsi
UC-01	Melihat Informasi Kendaraan Umum	Fungsionalitas untuk mencari informasi mengenai alat transportasi umum yang meliputi informasi taksi, bus dan angkot

2. Identifikasi Aktor

Aktor yang berperan dalam menjalankan sistem dapat dilihat pada tabel dibawah ini.

Tabel 4.2 Deskripsi Aktor Sistem yang Berjalan

No	Aktor	Deskripsi
P-01	Pengguna	Merupakan aktor yang menggunakan aplikasi Informasi transportasi umum

Tabel 4.2 menjelaskan bahwa aktor berperan sebagai pengguna aplikasi yang berinteraksi dengan *smartphone* berbasis Android.

4.1.2 Evaluasi Sistem Berjalan

Tabel 4.3 Evaluasi sistem berjalan

No	Permasalahan	Penyelesaian
1	Belum banyak informasi mengenai transportasi umum di kota Bandung	Membuat aplikasi versi *Mobile* untuk layanan informasi transportasi Kota Bandung, sehingga pengguna dapat membuka aplikasi dimanapun kapanpun.
2	Masih sedikitnya masyarakat khususnya pendatang baru yang tahu akan Informasi mengenai alat transportasi umum yang ada di Kota Bandung	Memberikan fasilitas informasi alat transportasi umum yang ada di Kota Bandung

3	Belum tersedianya fasilitas melihat peta lokasi keberadaan pengguna secara langsung.	Membuat fasilitas Melihat peta lokasi untuk mengetahui lokasi keberadaan pengguna.

4.2 Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan suatu sistem yang baru, atau memperbaharui sistem yang ada untuk meningkatkan efektifitas kerja agar dapat memenuhi hasil yang digunakan dengan tujuan memanfaatkan teknologi dan fasilitas yang tersedia. Pada bagian ini penyusun akan memberikan usulan yang merupakan sistem informasi secara komputerisasi yang diharapkan akan membantu dan mempermudah pekerjaan.

4.2.1 Tujuan Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran yang jelas dan rancang bangun yang sesuai dengan kebutuhan *user* atau pemakai sistem itu sendiri. Perancangan sistem dilakukan apabila tahap analisis sistem telah dilakukan. Maka untuk selanjutnya seorang analis sistem merancang bagaimana membentuk sistem yang baru ataupun memperbaharui sistem yang lama. Tahap inilah yang dinamakan dengan istilah dari perancangan sistem.

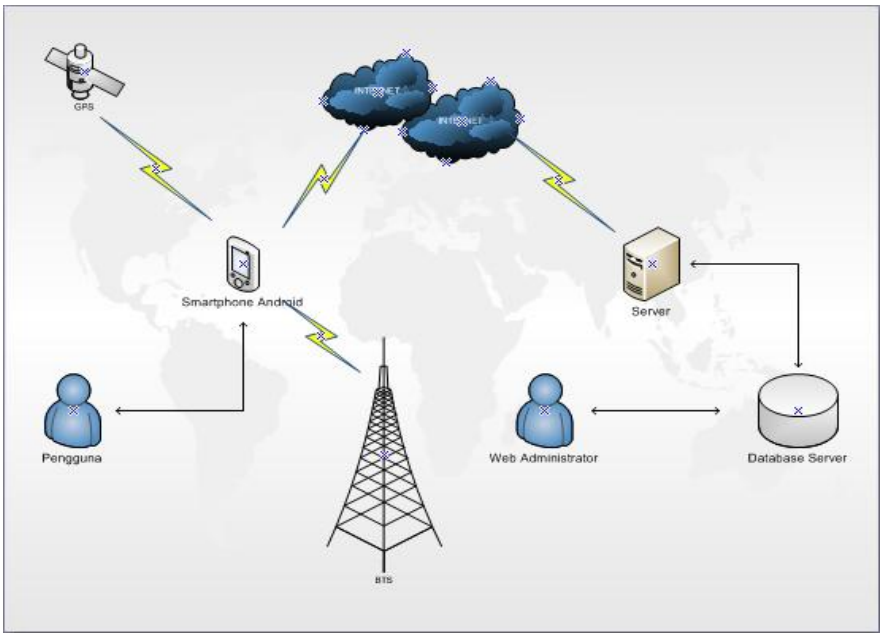
Adapun tujuan perancangan sistem yang diusulkan yaitu untuk membuat aplikasi *mobile* baru dengan menambahkan beberapa fungsi baru pada aplikasi.

4.2.2 Gambaran Umum Sistem yang Diusulkan

Sistem yang akan diusulkan untuk memperbaiki sistem yang lama secara umum yaitu dengan menambahkan beberapa fungsi aplikasi ke dalam aplikasi

mobile transportasi Kota Bandung agar pengguna dapat lebih terbantu dengan adanya aplikasi tersebut.

Arsitektur fisik sistem terdiri dari tiga bagian utama yaitu *client*, *application server*, dan *database server*. Prinsip kerja sistem secara keseluruhan dapat dilihat pada gambar berikut ini.



Gambar 4.2 Arsitektur Aplikasi Informasi Transportasi Umum

4.2.3 Perancangan Prosedur yang Diusulkan

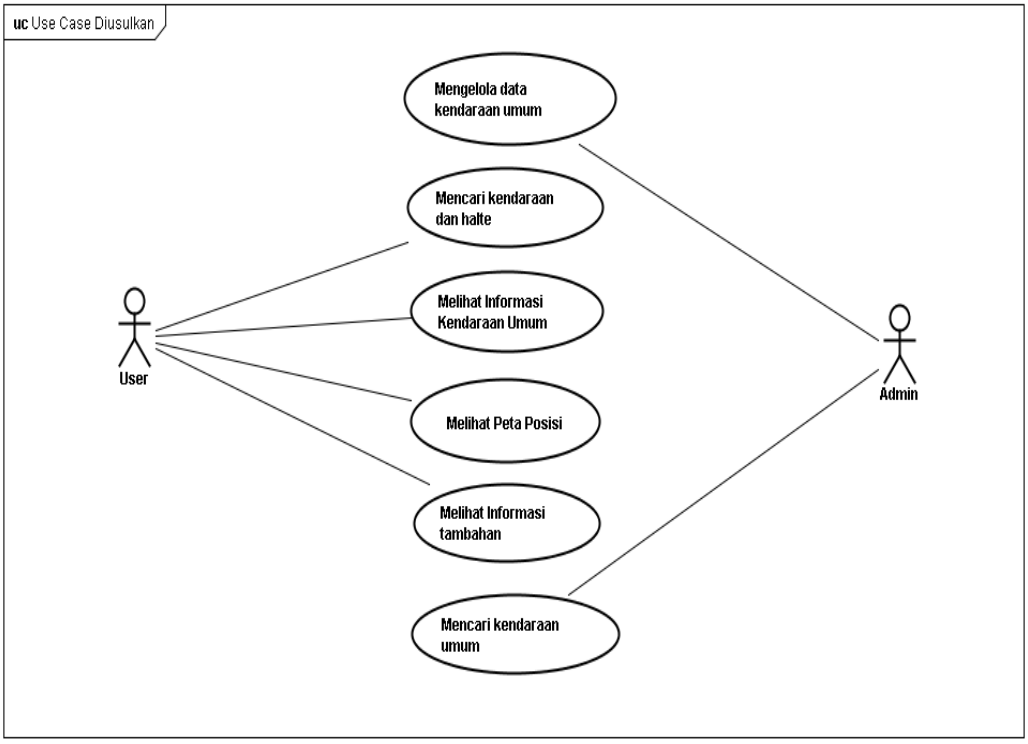
Prosedur kerja aplikasi *mobile* transportasi umum kota Bandung yang diusulkan adalah sebagai berikut :

1. Admin melakukan manipulasi data atau mengelola data transportasi umum kota Bandung yang meliputi pengelolaan data Bus Damri, angkot, taksi dan halte pada server yang dibuat menggunakan bahasa pemrograman PHP dan database MySQL.

2. Pengguna membuka aplikasi *mobile* transportasi umum kota Bandung pada *smartphone* Android.
3. Pengguna mengakses menu Pencarian yang terdiri dari pencarian kendaraan dan pencarian halte pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi seputar kendaraan yang melintas di jalan yang pengguna masukkan ataupun halte yang ada di sepanjang ruas jalan yang masukkan pada kotak pencarian.
4. Pengguna mengakses menu Info Bus Damri pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi seputar Bus Damri.
5. Pengguna mengakses menu Info Angkot pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi seputar Angkot.
6. Pengguna mengakses menu Info Taksi pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi seputar Taksi yang ada di kota Bandung.
7. Pengguna mengakses menu Lihat Peta posisi pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi mengenai lokasi keberadaan pengguna agar pengguna mengetahui posisi/nama jalan tempat ia berada.
8. Pengguna mengakses menu *help* dan *about* pada aplikasi transportasi umum kota Bandung guna mendapatkan informasi seputar petunjuk penggunaan aplikasi dan informasi singkat mengenai aplikasi.

4.2.3.1 *Use Case* Diagram

Hasil dari evaluasi sistem berjalan dan dari masalah-masalah yang telah dianalisis dicari solusi dari permasalahan tersebut dengan merancang *Use Case* Aplikasi Layanan Informasi Transportasi Umum Kota Bandung Berbasis *Mobile* Android. Diagram *Use Case* pada aplikasi yang diusulkan adalah sebagai berikut :



Gambar 4.3 *Use Case* sistem yang diusulkan

1. Identifikasi *Use Case*

Identifikasi *Use Case* adalah *Use Case* yang bekerja pada sistem aplikasi dan *Use Case* yang bekerja pada sistem adalah sebagai berikut :

Tabel 4.4 Deskripsi *Use Case* Pengguna Sistem yang Diusulkan

No	*Use Case*	Deskripsi
UC-01	Mengelola data kendaraan umum	Fungsionalitas untuk memanipulasi / mengelola data kendaraan umum pada

		server.
UC-02	Mencari Kendaraan dan halte	Fungsionalitas untuk mencari kendaraan umum yang melintas di jalan dan mencari daftar halte sepanjang ruas jalan yang dimasukkan. Pencarian berdasarkan kata kunci nama jalan.
UC-03	Melihat Informasi Kendaraan Umum	Fungsionalitas untuk mencari informasi mengenai kendaraan umum yang ada di Kota Bandung meliputi Angkot, Bus Damri dan Taksi
UC-04	Melihat Peta Posisi	Fungsionalitas untuk melihat posisi dan lokasi(pada peta) *end user* berada
UC-05	Melihat Informasi tambahan	Fungsionalitas untuk melihat informasi singkat mengenai aplikasi dan bantuan petunjuk penggunaan aplikasi Fortra.
UC-06	Mencari Kendaraan umum	Fungsionalitas untuk mencari kendaraan umum berdasarkan nama jalan pada server.

2. Identifikasi Aktor

Aktor yang berperan dalam menjalankan sistem dapat dilihat pada tabel dibawah ini.

Tabel 4.5 Deskripsi Aktor Sistem yang Diusulkan

No	Aktor	Deskripsi
P-01	Pengguna	Merupakan aktor yang menggunakan aplikasi Informasi transportasi umum
P-02	Admin	Merupakan aktor yang berperan memanipulasi / mengelola data aplikasi fortra pada server

Tabel 4.5 menjelaskan bahwa aktor berperan sebagai pengguna aplikasi yang berinteraksi dengan *smartphone* berbasis Android.

3. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam *Use Case* dapat dilihat pada tabel-tabel dibawah ini.

Tabel 4.6 Skenario *Use Case* Mengelola data kendaraan umum

Identifikasi	
Nama *Use Case*	Mengelola data kendaraan umum
Aktor	Admin
Tujuan	Memperbaharui Data aplikasi
Keadaan Awal	Sistem menampilkan halaman *Log in*
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi server	
	2. Menampilkan halaman *Log in*
3. Mengisi form *Log in* dengan memasukkan *username* dan *password* yang sesuai	
	4. Memeriksa kesesuaian antara *username* dan *password*
	5. Jika *password* dan *username* benar maka sistem akan menampikan halaman *index*/menu utama.
6. Memilih data yang akan dikelola	
	7. Jika memilih Data Bus Damri maka akan menampilkan halaman mengelola data bus Damri
8. Menambah atau mengedit atau menghapus atau mencari data bus Damri	
	9. Menambah atau mengedit atau menghapus atau mencari data bus Damri pada *database*
	10. Jika memilih Data Angkot maka akan menampilkan halaman mengelola data Angkot
11. Menambah atau mengedit atau menghapus atau mencari data angkot	
	12. Menambah atau mengedit atau menghapus atau mencari data angkot pada *database*
	13. Jika memilih Data Taksi maka akan menampilkan halaman mengelola taksi
14. Menambah atau mengedit	

atau menghapus atau mencari data taksi	
	15. Menambah atau mengedit atau menghapus atau mencari data taksi pada *database*
Skenario Alternatif	
Aksi Aktor	Reaksi Sistem
	1. Jika *password* dan *username* tidak sesuai/salah ketika mengisi form *login* maka sistem akan menampilkan Pesan "Maaf *Username* dan *Password* tidak sesuai, silakan coba kembali"
2. Menekan tombol "OK"	
	3. Sistem akan *terminate*
	4. Jika *field* belum diisi pada saat penambahan data maka akan menampilkan pesan "Data belum diisi"
5. Menekan tombol "OK"	
	6. Sistem akan *terminate*
Kondisi Akhir	Menampilkan halaman indeks/menu utama

Tabel 4.7 Skenario *Use Case* Mencari Kendaraan

Identifikasi	
Nama *Use Case*	Mencari Kendaraan dan halte
Aktor	Pengguna
Tujuan	Mencari informasi kendaraan dan halte pada jalan tertentu
Keadaan Awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi	
	2. Menampilkan menu utama
3. Memilih menu cari kendaraan	
	4. Menampilkan halaman pencarian kendaraan yang terdiri dari *button* cari angkot dan *button* cari bus damri 5. Jika pengguna memilih button cari angkot maka sistem akan menampilkan halaman pencarian angkot

6. Memasukkan kata kunci nama jalan pada kotak pencarian kemudian menekan tombol *search*	
	7. Menampilkan informasi kendaraan yang melintas sesuai dengan nama jalan yang diinputkan
	8. Jika pengguna memilih button cari bus damri maka sistem akan menampilkan halaman pencarian bus damri
9. Memasukkan kata kunci nama jalan pada kotak pencarian kemudian menekan tombol *search*	
	10. Menampilkan informasi kendaraan yang melintas sesuai dengan nama jalan yang diinputkan
11. Memilih menu cari halte	
	12. Menampilkan halaman pencarian halte
13. Memasukkan kata kunci nama jalan pada kotak pencarian kemudian menekan tombol *search*	
	14. Menampilkan *list* atau daftar halte yang ada sepanjang ruas jalan sesuai dengan nama jalan yang diinputkan
15. Memilih salah satu *list* halte	
	16. Menampilkan informasi detail mengenai halte yang telah dipilih
Skenario Alternatif – Pencarian tidak ditemukan	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pesan bahwa tidak ada kendaraan umum yang melintas pada jalan yang anda masukkan(“Periksa kembali inputan”) atau (“Hubungi taksi”)
2. Menekan tombol “back” pada *smartphone*	
	3. Kembali ke menu sebelumnya

Kondisi Akhir	Menampilkan informasi sesuai nama jalan yang diinputkan

Tabel 4.8 Skenario *Use Case* Melihat Informasi Kendaraan Umum

Identifikasi	
Nama *Use Case*	Melihat Informasi Kendaraan Umum
Aktor	Pengguna
Tujuan	Mencari informasi kendaraan umum yang ada di kota Bandung meliputi Angkot, Bus Damri dan Taksi
Keadaan Awal	Sistem menampilkan halaman utama

Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi	
	2. Menampilkan menu utama
3. Memilih menu informasi kendaraan umum yang meliputi Info Angkot, Info Damri, Info dan ino Taksi	
	4. Jika pengguna memilih menu Info Bus Damri maka sistem akan menampilkan halaman pencarian info Bus Damri yang terdiri dari tab nama trayek dan tab kode trayek
5. Memilih tab nama trayek 6. Memilih nama trayek Bus Damri pada *spinner* kemudian menekan tombol cari.	
	7. Menampilkan informasi Bus Damri
8. Memilih tab kode trayek 9. Memilih kode trayek Bus Damri pada *spinner* kemudian menekan tombol cari.	
	10. Menampilkan informasi Bus Damri
	11. Jika pengguna memilih menu Info Angkot maka sistem akan menampilkan halaman pencarian info angkot yang terdiri dari tab nama trayek dan tab kode trayek
12. Memilih tab nama trayek 13. Memilih nama trayek angkot pada *spinner* kemudian	

menekan tombol cari.	
	14. Menampilkan informasi angkot
15. Memilih tab kode trayek 16. Memilih kode trayek angkot pada *spinner* kemudian menekan tombol cari.	
	17. Menampilkan informasi angkot
	18. Jika pengguna memilih menu Info Taksi maka sistem akan menampilkan halaman informasi Taksi
19. Memilih nama taksi pada *spinner* dan menekan tombol cari informasi	
	20. Menampilkan informasi taksi
21. Menekan tombol telpon	
	22. Menghubungkan aplikasi ke nomor telpon taksi yang dituju
23. Menekan tombol *reset*	
	24. Menghapus info taksi yang telah ditampilkan
Kondisi Akhir	Menampilkan informasi transportasi umum yang dipilih

Tabel 4.9 Skenario *Use Case* Melihat Posisi

Identifikasi	
Nama *Use Case*	Melihat Peta Posisi
Aktor	Pengguna
Tujuan	Melihat peta lokasi keberadaan pengguna
Keadaan Awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi	
	2. Menampilkan menu utama
3. Menekan gambar peta yang ada pada halaman utama	
	4. Menampilkan peta lokasi sesuai tempat pengguna berada
Skenario Alternatif – Proses gagal	
Aksi Aktor	Reaksi Sistem
	1. Menampilkan pesan bahwa proses gagal dilakukan (“*internet connection unavailable*”)

2. Menekan tombol “ok”	
	3. Sistem akan berakhir
Kondisi Akhir	Menampilkan peta lokasi sesuai tempat pengguna berada

Tabel 4.11 Skenario *Use Case* Melihat Informasi Tambahan

Identifikasi	
Nama *Use Case*	Melihat Informasi Tambahan
Aktor	Pengguna
Tujuan	Mencari informasi tentang aplikasi
Keadaan Awal	Sistem menampilkan halaman informasi aplikasi
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi	
	2. Menampilkan menu utama
3. Menekan tombol *option menu*	
	4. Menampilkan menu tentang, bantuan dan keluar aplikasi.
	5. Jika pengguna memilih menu tentang maka sistem akan menampilkan informasi tentang aplikasi
	6. Jika pengguna memilih menu bantuan maka sistem akan menampilkan informasi petunjuk penggunaan aplikasi
Kondisi Akhir	Menampilkan informasi mengenai aplikasi Fortra

Tabel 4.12 Skenario *Use Case* Mencari Kendaraan Umum

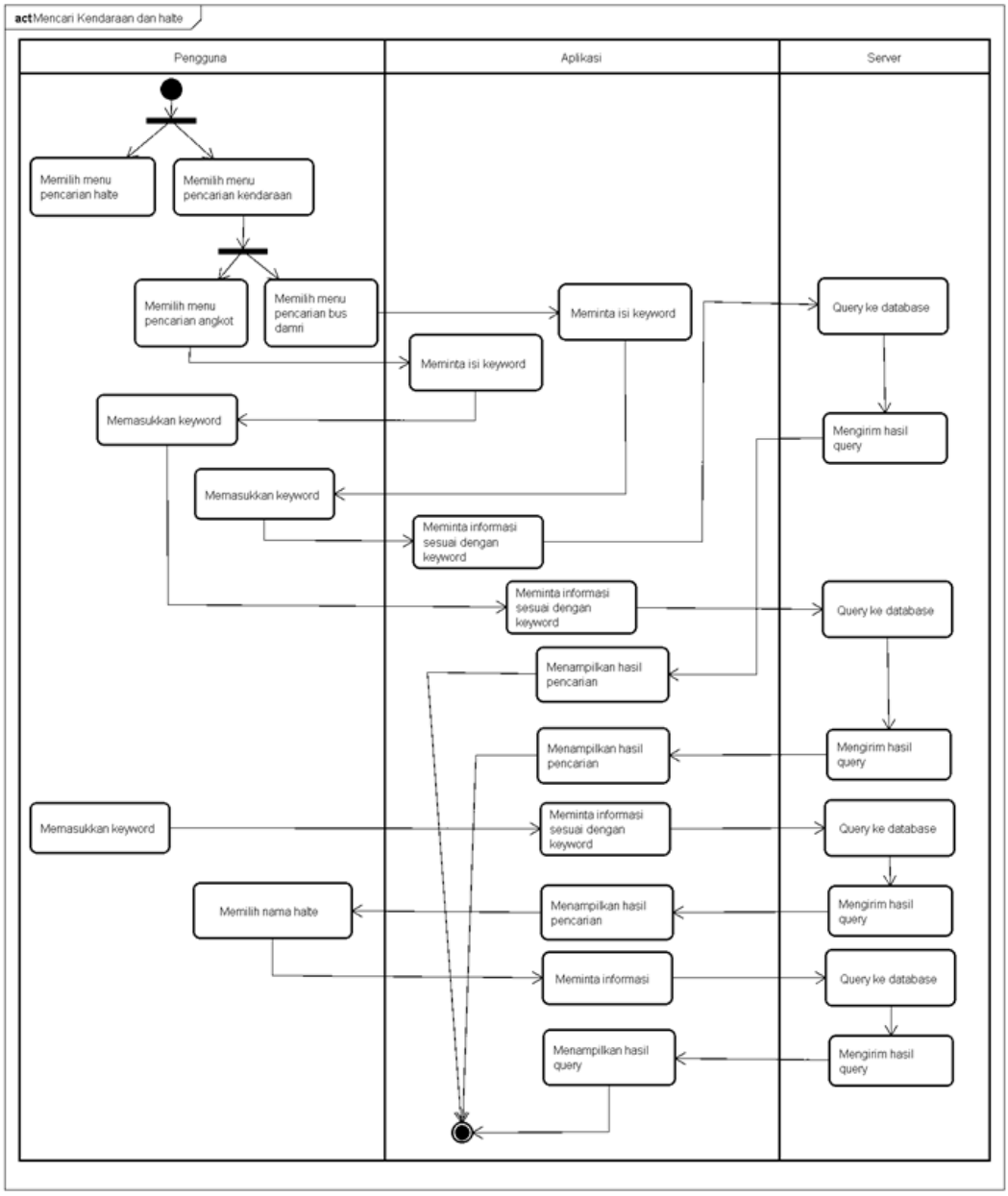
Identifikasi	
Nama *Use Case*	Mencari Kendaraan Umum
Aktor	Admin
Tujuan	Mencari informasi kendaraan pada jalan tertentu
Keadaan Awal	Sistem menampilkan halaman utama
Skenario Utama	
Aksi Aktor	Reaksi Sistem
1. Membuka aplikasi server	
	2. Menampilkan menu utama
3. Memasukkan kata kunci nama jalan pada kotak pencarian	

kemudian menekan tombol *search*	
	4. Menampilkan informasi kendaraan yang melintas sesuai dengan nama jalan yang diinputkan
Kondisi Akhir	Menampilkan informasi sesuai nama jalan yang diinputkan

4.2.3.2 *Activity* Diagram

Activity Diagram merupakan bagian dari penggambaran sistem secara fungsional menjelaskan proses-proses logika atau fungsi yang terimplementasi oleh kode program. *Activity Diagram* memodelkan *event-event* yang terjadi didalam suatu *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

1. *Activity Diagram* Mencari Kendaraan dan halte

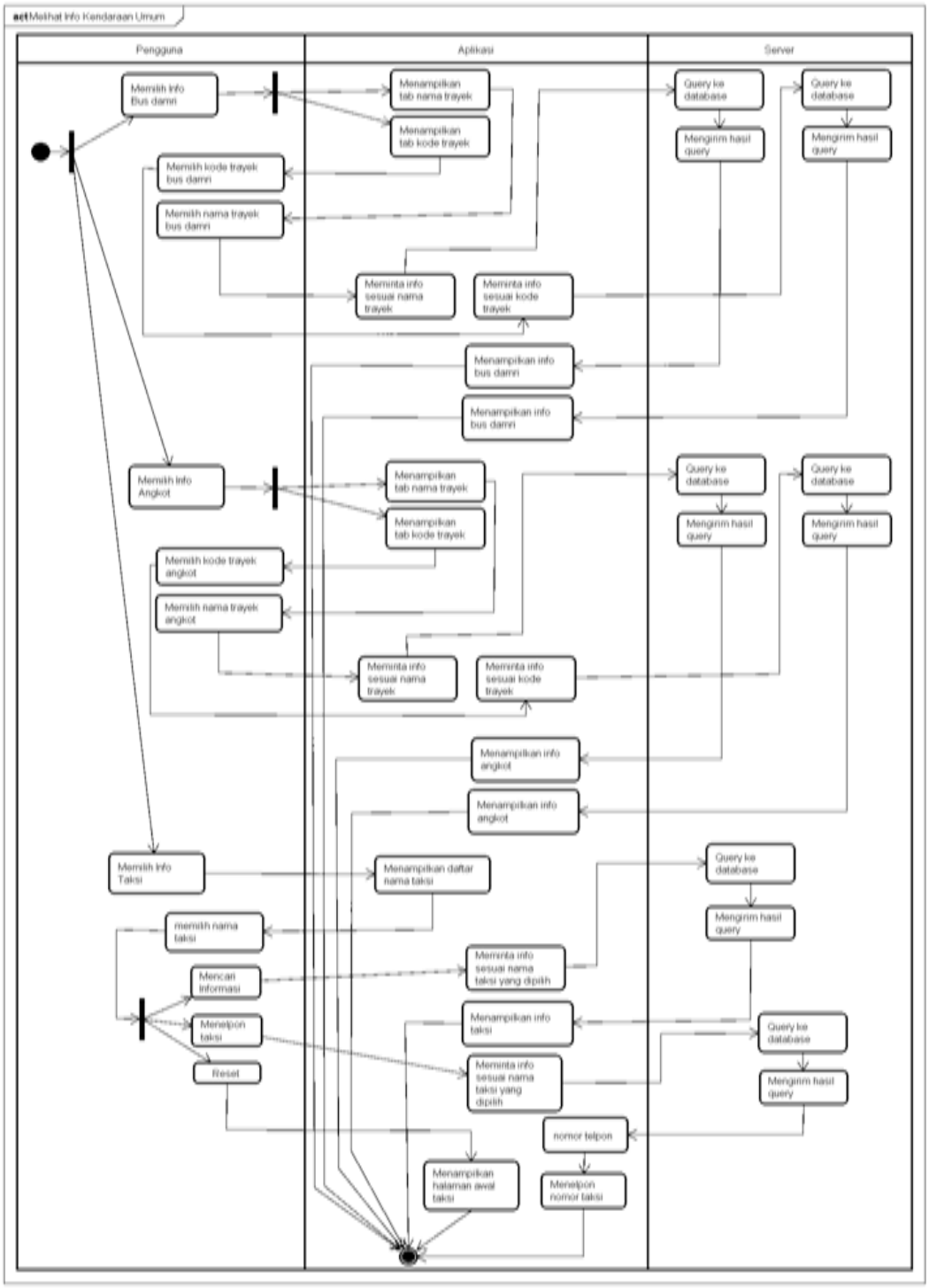


Gambar 4.4 *Activity Diagram* Mencari Kendaraan

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi Fortra yang dilakukan oleh pengguna, dimulai ketika pengguna memilih menu pencarian yang terdiri dari pencarian kendaraan umum yang melintas dan pencarian daftar halte pada menu utama, kemudian pengguna

memasukkan kata kunci, aplikasi akan meminta informasi sesuai dengan kata kunci yang dimasukkan pada server, server akan mengeksekusi *query* ke *database* dan mengirimkan hasil *query* ke aplikasi.

2. *Activity Diagram* Melihat Info Kendaraan Umum



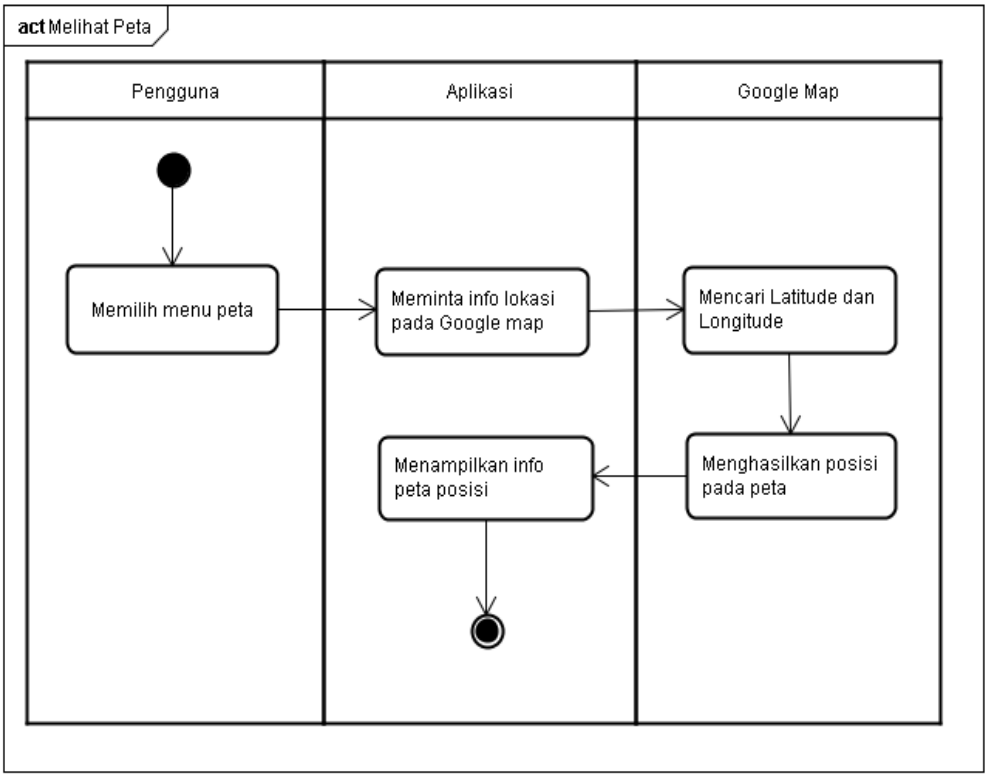
Gambar 4.5 *Activity Diagram* Melihat Info Kendaraan Umum

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi Fortra yang dilakukan oleh pengguna, dimulai ketika pengguna memilih menu info kendaraan umum yang terdiri dari Info Bus Damri, Info Angkot dan Info Taksi.

Jika pengguna memilih menu info bus damri maka aplikasi akan menampilkan tab nama trayek dan tab kode trayek, kemudian jika pengguna memilih tab nama trayek dan memilih salah satu nama trayeknya maka aplikasi akan meminta informasi sesuai dengan nama trayek yang dipilih ke server, server akan mengeksekusi *query* ke database dan mengirimkan hasil *query* ke aplikas. Jika pengguna memilih tab kode trayek dan memilih salah satu kode trayeknya maka aplikasi akan meminta informasi sesuai dengan kode trayek yang dipilih ke server, server akan mengeksekusi *query* ke database dan mengirimkan hasil *query* ke aplikasi. Sedangkan untuk info angkot prosedurnya sama dengan menu info bus damri.

Jika pengguna memilih menu info taksi maka aplikasi akan menampilkan daftar nama taksi, kemudian pengguna memilih nama taksi kemudian dapat memilih aksi yang disediakan yaitu mencari informasi, menelpon taksi dan *reset*. Jika pengguna memilih mencari informasi maka aplikasi akan meminta info sesuai dengan nama taksi yang dipilih ke server, selanjutnya server akan mengeksekusi *query* ke *database* dan mengirimkan hasil *query* pada aplikasi. Jika pengguna memilih menelpon taksi, maka aplikasi akan menghubungi nomor telpon taksi yang dipilih dan jika pengguna memilih *reset* maka aplikasi akan menampilkan halaman info taksi pada keadaan kosong/awal.

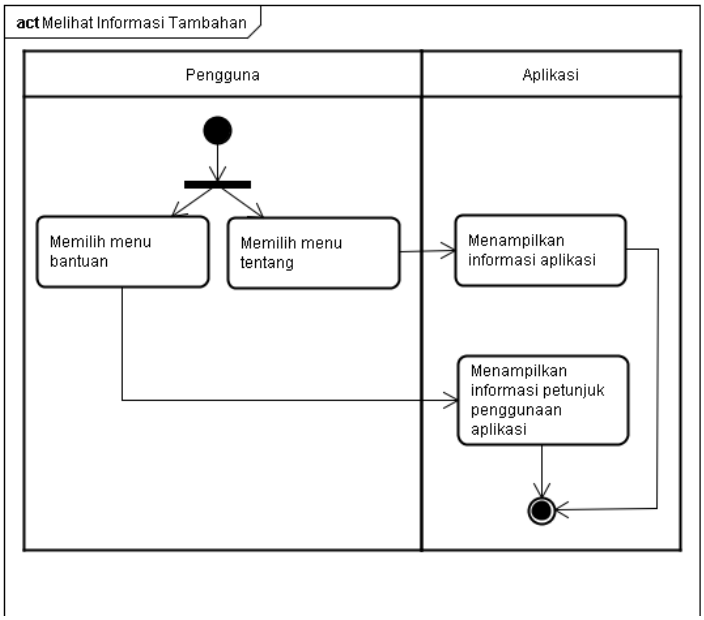
3. *Activity Diagram* Melihat Peta



Gambar 4.6 *Activity Diagram* Melihat Peta

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi Fortra yang dilakukan oleh pengguna, dimulai ketika pengguna memilih menu peta pada menu utama, kemudian aplikasi akan meminta info lokasi pada Google *map*, kemudian Google map akan mencari nilai latitude dan longitude dan mengirimkan informasi posisi pengguna pada peta ke aplikasi.

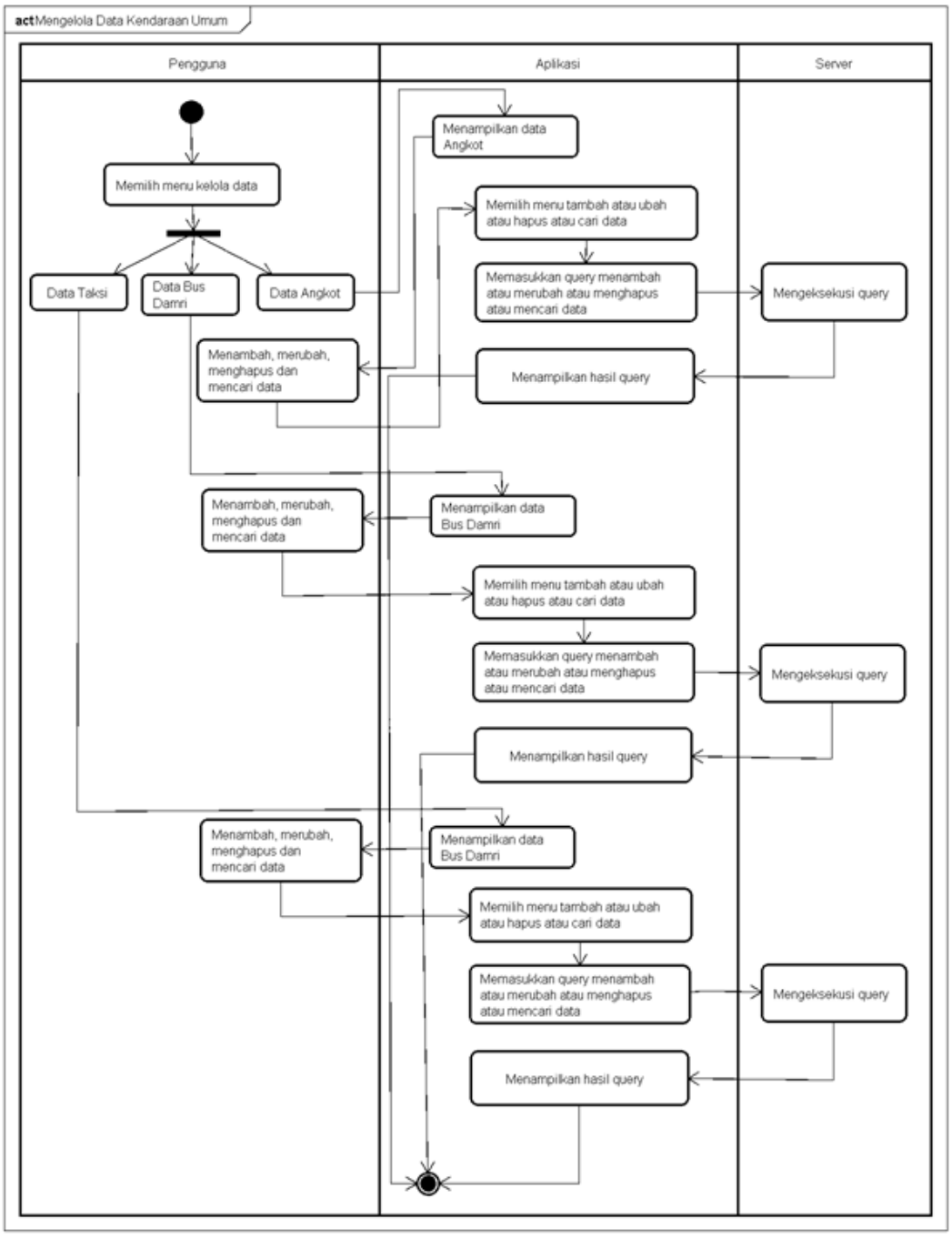
4. *Activity Diagram* Melihat Informasi Tambahan



Gambar 4.7 *Activity Diagram* Melihat Informasi Tambahan

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi Fortra yang dilakukan oleh pengguna, jika pengguna memilih menu tentang aplikasi pada *option menu* maka sistem akan menampilkan informasi mengenai aplikasi. Sedangakan jika pengguna memilih menu bantuan maka sistem akan menampilkan informasi petunjuk penggunaan aplikasi.

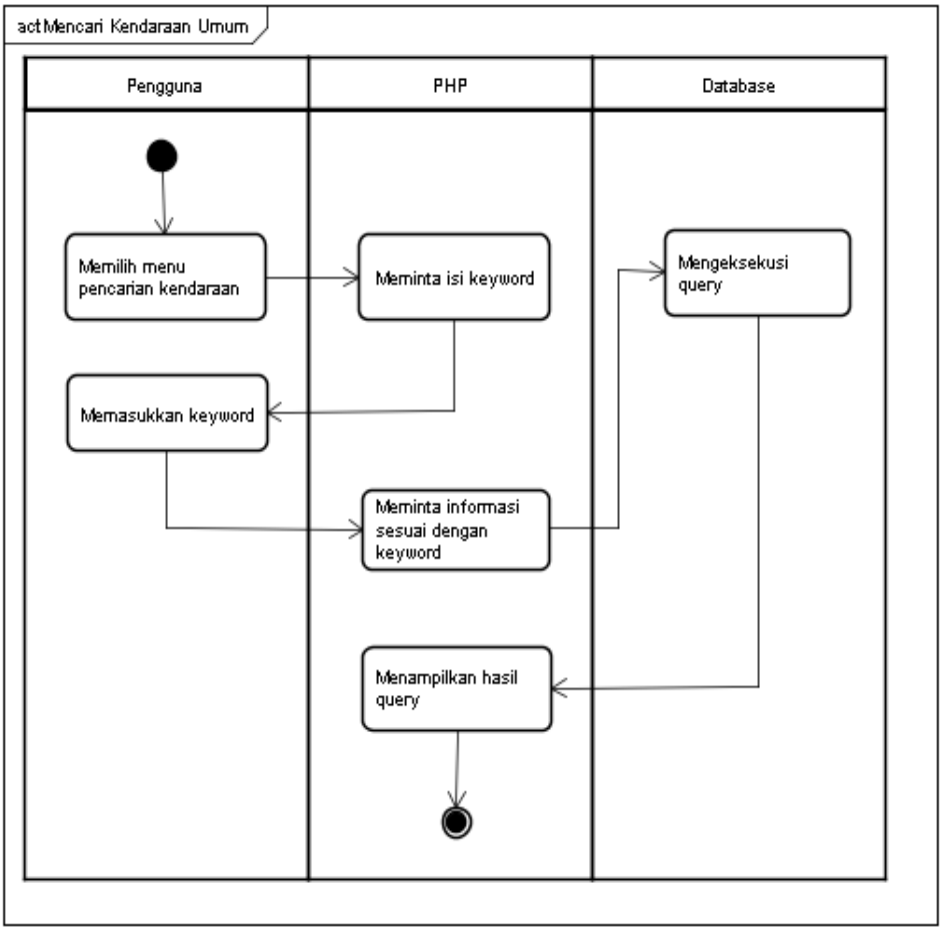
5. *Activity Diagram* Mengelola Data Kendaraan Umum



Gambar 4.8 *Activity Diagram* Mengelola Data Kendaraan Umum

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi *Backend* Fortra yang dilakukan oleh Admin *Webservice*, pada awalnya saat menu utama muncul Admin dapat memilih menu pengelolaan data yang terdiri dari data angkot, data bus damri dan data taksi. Jika Admin memilih menu data Angkot dimana hanya ada *query* menambah, merubah, menghapus dan mencari data Bus Damri yang kemudian *query* tersebut akan dieksekusi di *database* dan ditampilkan pada website Fortra. Kemudian jika Admin memilih menu data Bus Damri dimana hanya ada *query* menambah, merubah, menghapus dan mencari data Bus Damri yang kemudian *query* tersebut akan dieksekusi di *database* dan ditampilkan pada website Fortra. Sedangkan jika Admin memilih menu data Taksi dimana hanya ada *query* menambah, merubah, menghapus dan mencari data Bus Damri yang kemudian *query* tersebut akan dieksekusi di *database* dan ditampilkan pada website Fortra.

6. *Activity Diagram* Mencari Kendaraan Umum

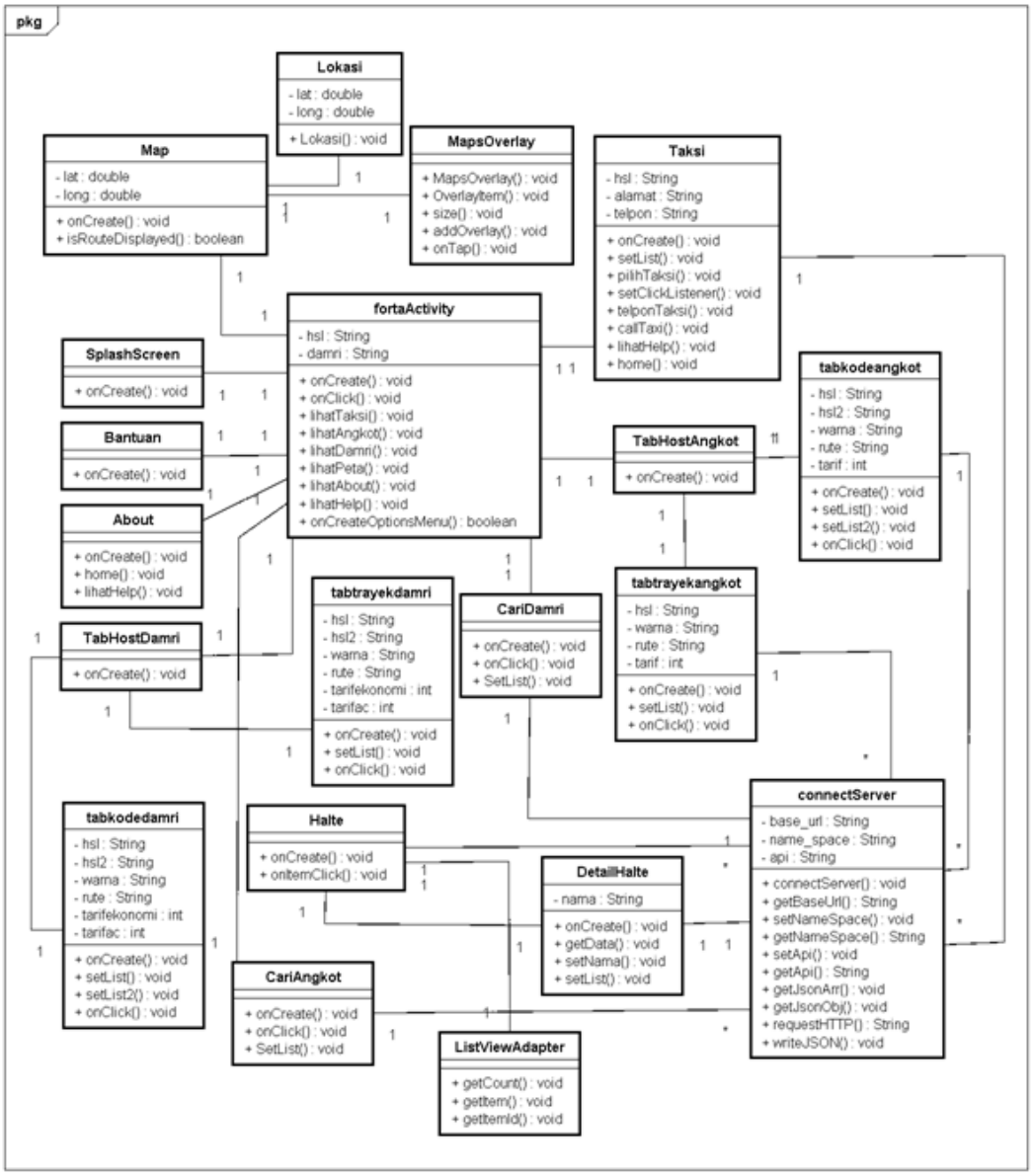


Gambar 4.9 *Activity Diagram* Mencari Kendaraan Umum

Gambar *activity diagram* diatas menjelaskan tentang aktivitas yang terjadi pada menu aplikasi *Backend* Fortra yang dilakukan oleh Admin *Webservice*, pada awalnya saat menu utama muncul Admin memilih menu pencarian kendaraan dan memasukkan kata kunci pencarian berdasarkan nama jalan dimana hanya ada *query* mencari data Angkot dan Bus Damri yang kemudian *query* tersebut akan di *query* di database dan ditampilkan pada website Fortra.

4.2.3.3 *Class* Diagram

Class Diagram adalah sebuah spesifikasi dari fungsionalitas yang menghasilkan objek dan merupakan inti dari pengembangan aplikasi ini. Diagram ini juga merupakan gambaran keadaan atribut atau properti dari sistem yang melakukan manipulasi fungsi atau metode. Berikut ini gambaran *class* diagram dari sistem Aplikasi Fasilitas Umum.

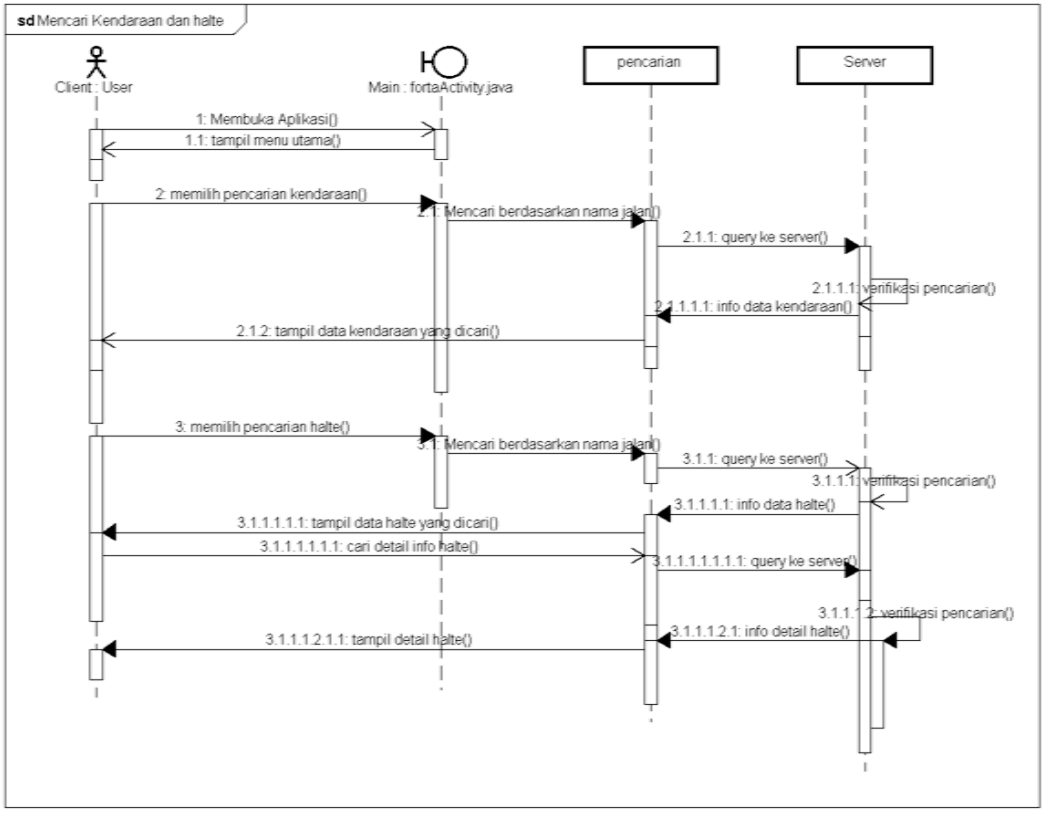


Gambar 4.10 *Class Diagram* aplikasi transportasi umum

4.2.3.4 *Sequence* Diagram

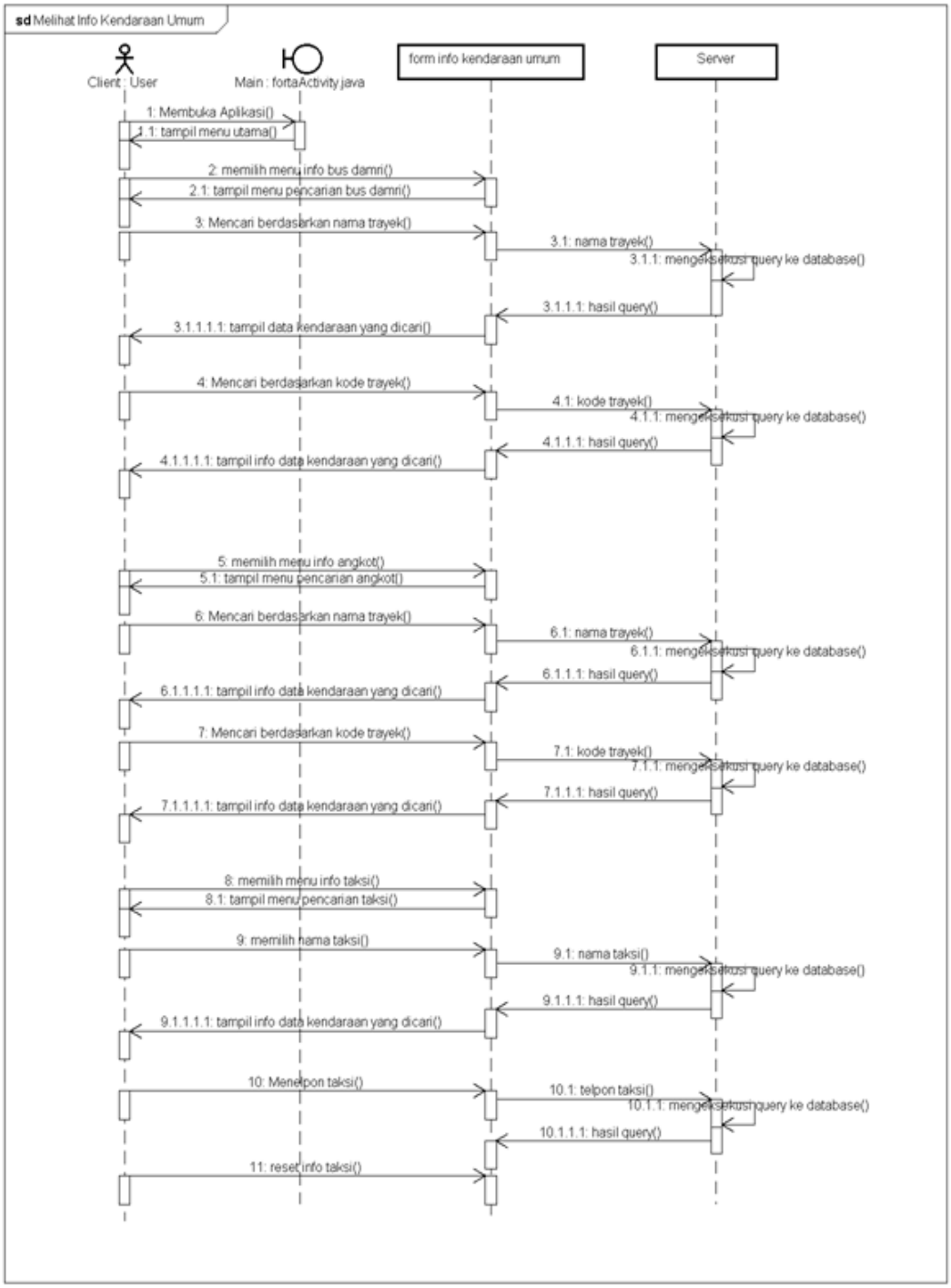
Pada tahap ini *Sequence Diagram* menjelaskan secara detil urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case*. Interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

1. *Sequence* Mencari Kendaraan dan halte



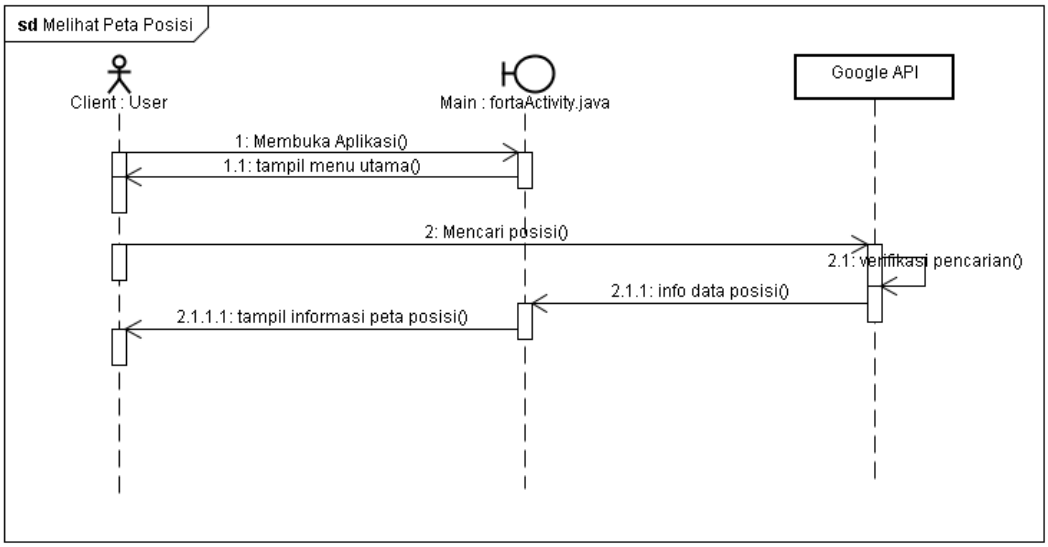
Gambar 4.11 *Sequence Diagram* Mencari Kendaraan

2. *Sequence* Melihat Info Kendaraan Umum



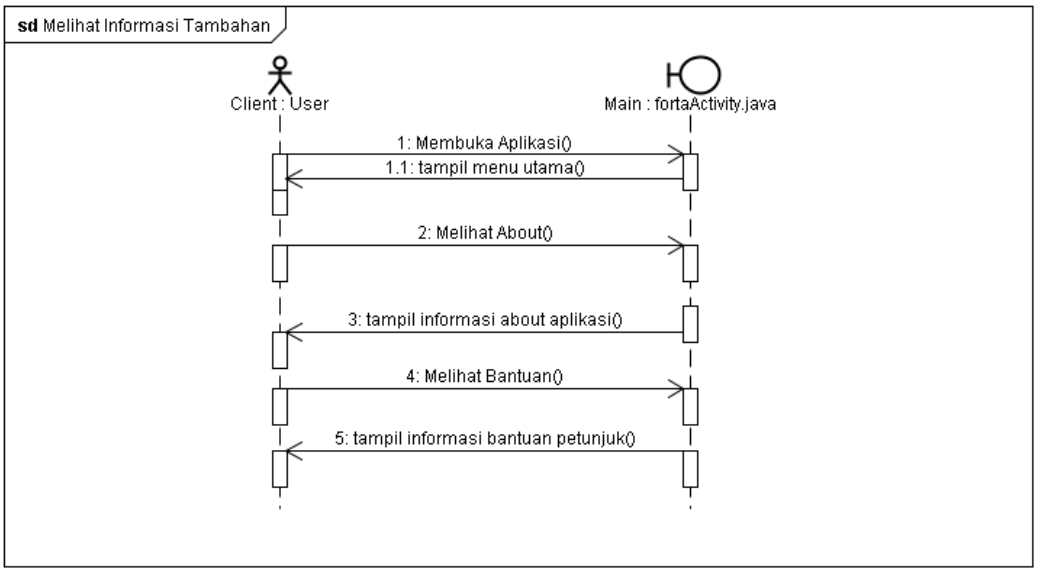
Gambar 4.12 *Sequence Diagram* Melihat Info Kendaraan Umum

3. *Sequence* Melihat Peta Posisi



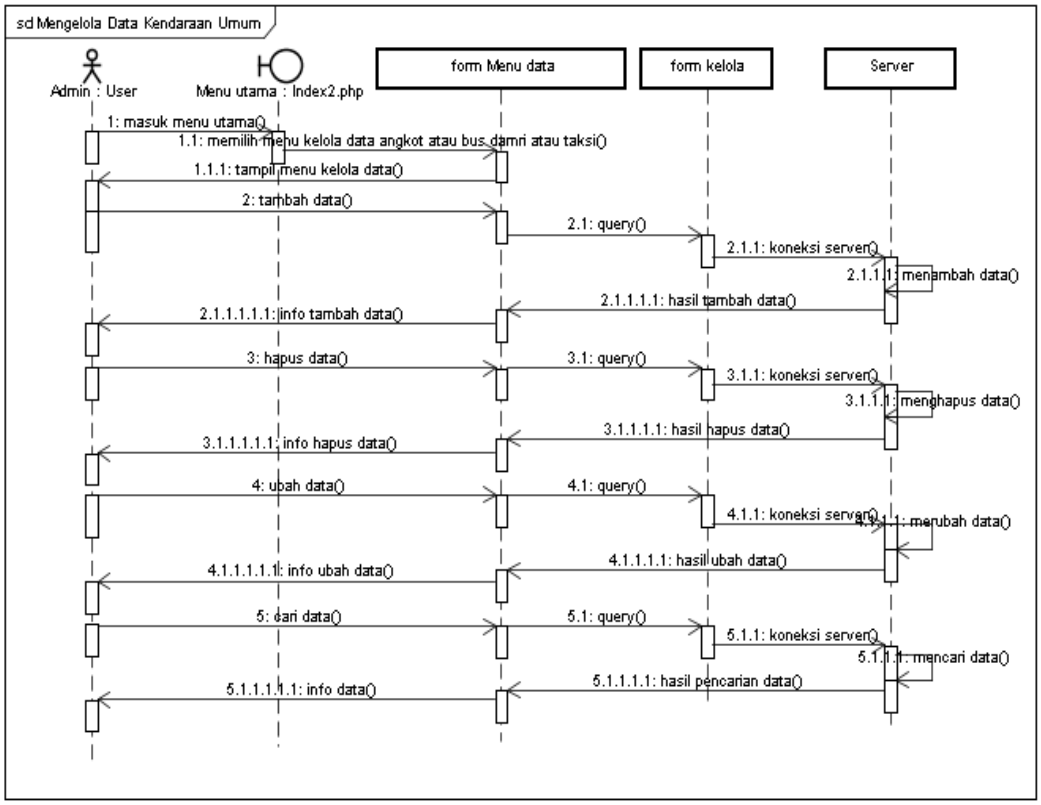
Gambar 4.13 *Sequence Diagram* Melihat Peta Posisi

4. *Sequence* Melihat Informasi Tambahan



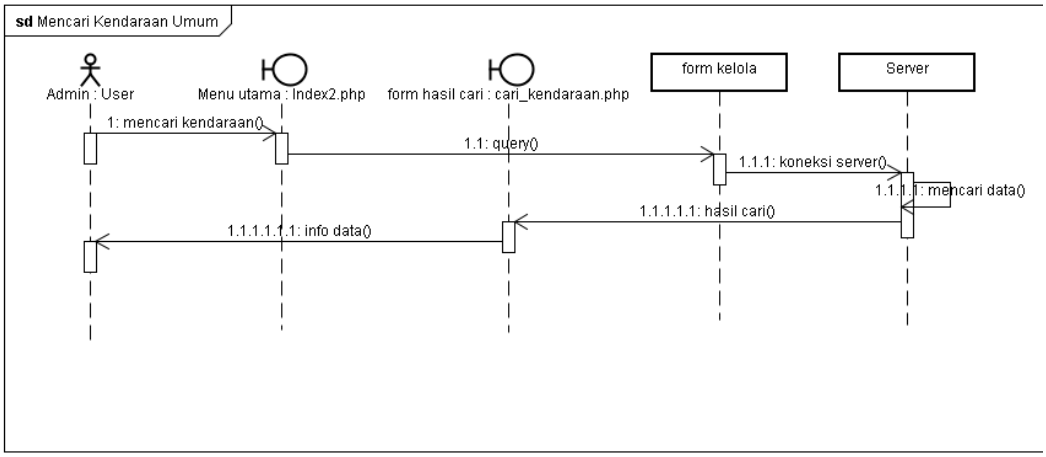
Gambar 4.14 *Sequence Diagram* Melihat Informasi Tambahan

5. *Sequence* Mengelola Data Kendaraan Umum



Gambar 4.15 *Sequence Diagram* Melihat About

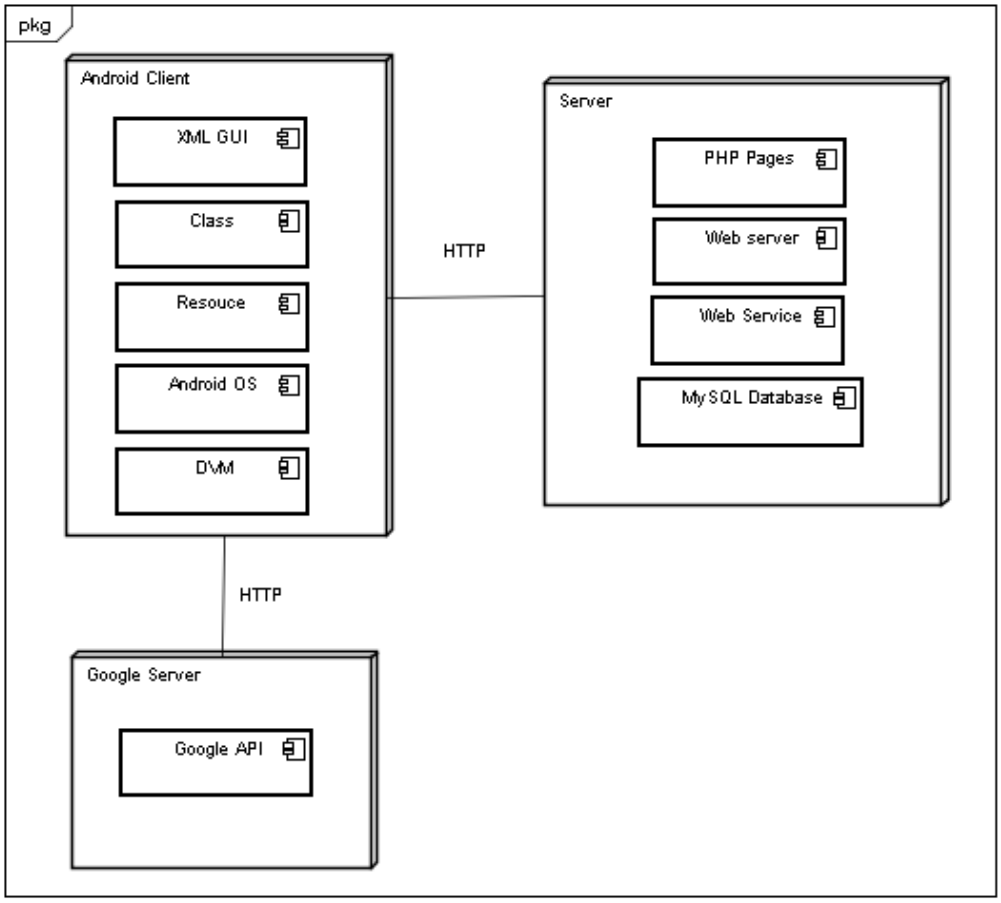
6. *Sequence* Mencari Kendaraan Umum



Gambar 4.16 *Sequence Diagram* Mencari Kendaraan Umum

4.2.3.5 *Deployment Diagram*

Deployment diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*. *Deployment diagram* menggambarkan detail bagaimana komponen dibentuk dan didistribusikan (*deploy*) dalam infrastruktur sistem. Dimana komponen akan terletak pada mesin, *server* atau peranti keras. Bagaimana jaringan pada lokasi tersebut, misalnya *server*, *client* dan hal-hal lain yang bersifat fisik.



Gambar 4.17 *Deployment Diagram*Aplikasi Transportasi Umum

4.2.4 Perancangan Grafis dan Antarmuka

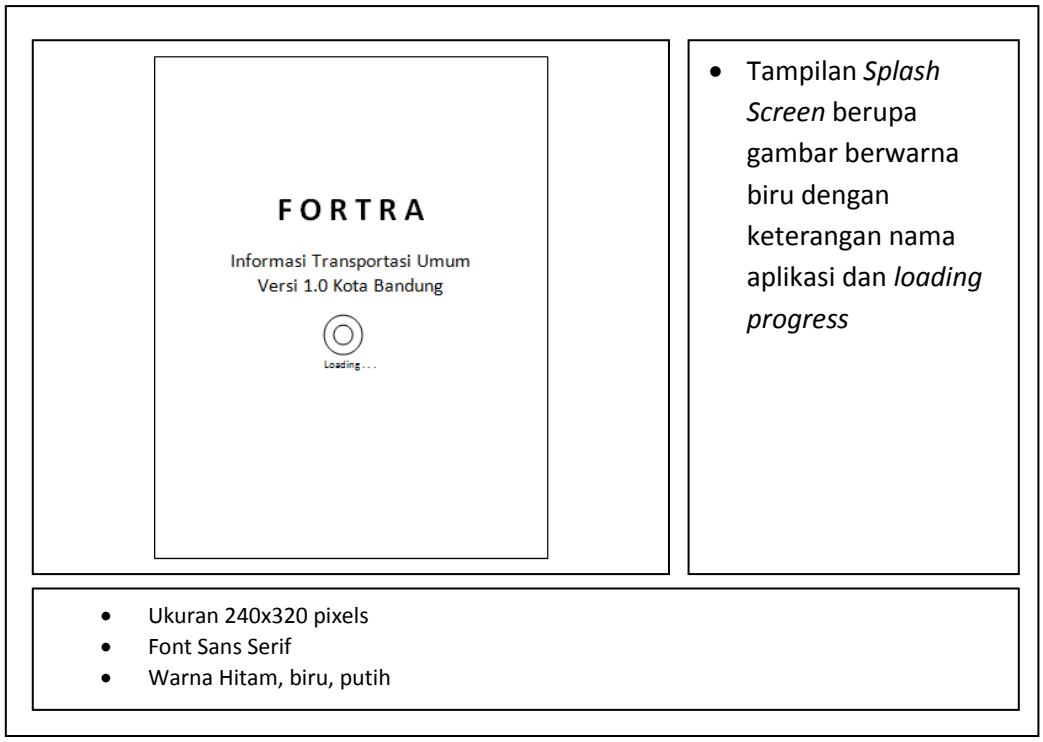
Berdasarkan perancangan sistem yang telah dibuat maka akan dibuatkan suatu aplikasi yang sesuai dengan kebutuhan pengguna. Tujuan dari perancangan ini adalah merancang antarmuka yang efektif untuk aplikasi. Efektif artinya siap digunakan, dan hasilnya sesuai dengan kebutuhan. Kebutuhan disini adalah kebutuhan penggunanya. Pengguna sering menilai sistem dari sisi antarmuka, bukan dari fungsinya melainkan dari *user interface*-nya. Jika desain *user interface* buruk, maka itu sering menjadi alasan untuk tidak menggunakan aplikasi. Selain itu *interface* yang buruk menyebabkan pengguna membuat kesalahan fatal.

Perancangan grafis dan antarmuka pada aplikasi ini terdiri dari dua bagian utama yaitu desain grafis dan antarmuka *frontend application* dan *backend application*.

4.2.4.1 Perancangan antarmuka *Frontend Application*

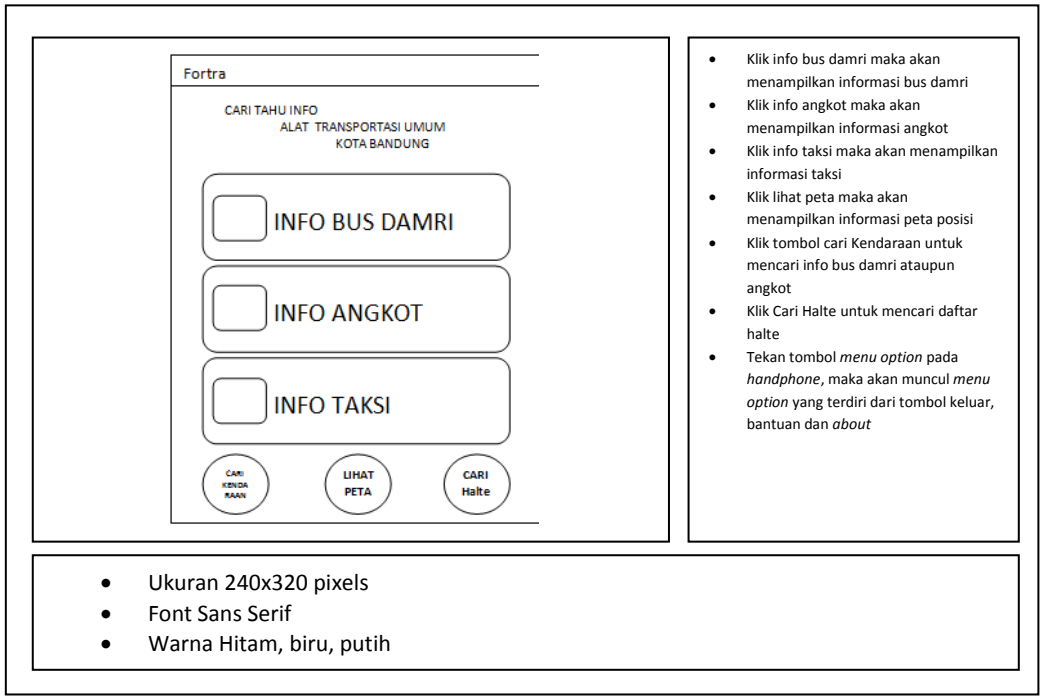
Untuk memudahkan proses pembuatan aplikasi *mobile* layanan informasi transportasi umum kota Bandung ini maka terlebih dahulu membuat rancangan tampilannya. Perancangan ini meliputi rancangan antarmuka *splash screen*, menu utama, pencarian Bus damri, pencarian angkot, pencarian taksi, *Map*, dan informasi tambahan (tentang dan bantuan).

1. Perancangan antarmuka *splash screen*



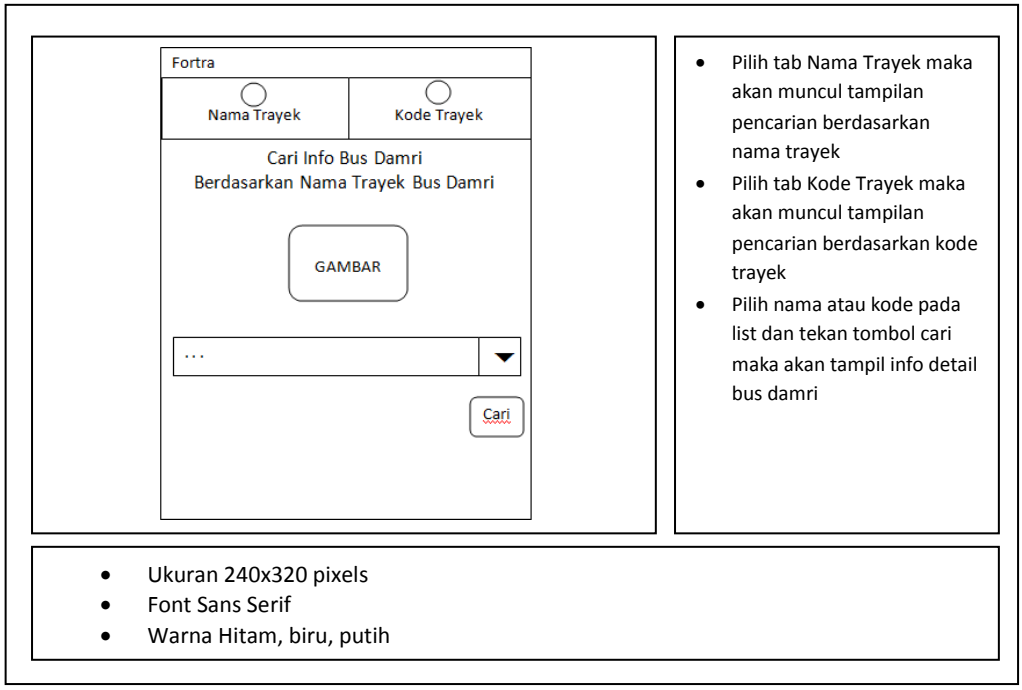
Gambar 4.18 Tampilan *Splash screen* aplikasi *frontend*

2. Perancangan antarmuka Menu Utama aplikasi *frontend*



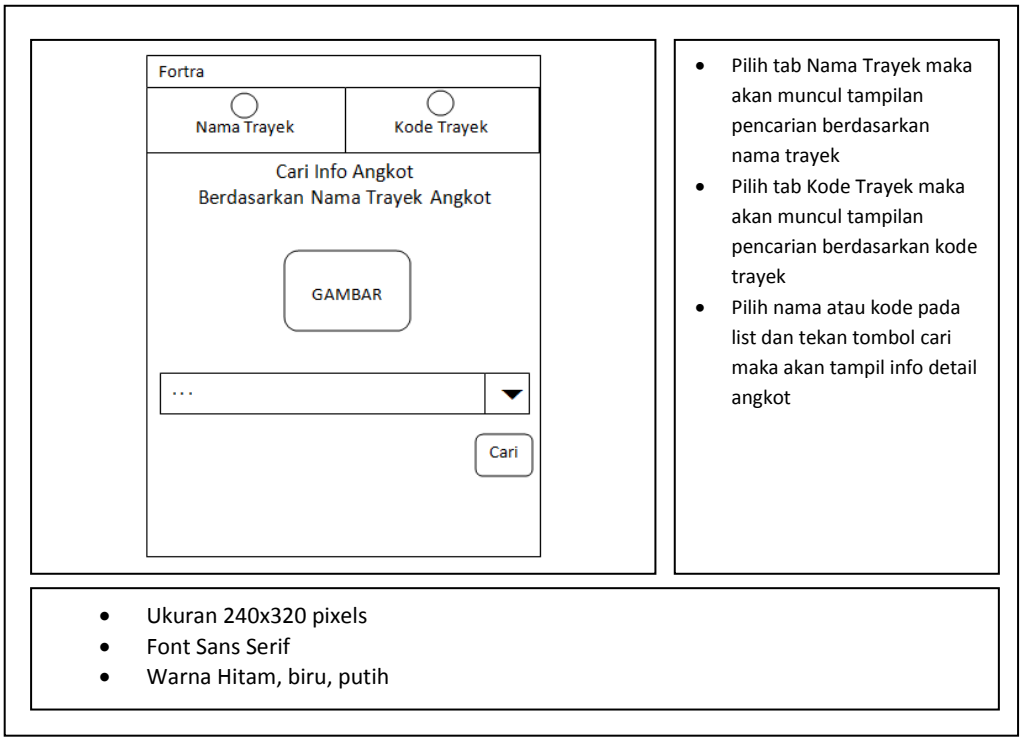
Gambar 4.19 Tampilan Menu Utama aplikasi *frontend*

3. Perancangan Antarmuka Info Bus Damri



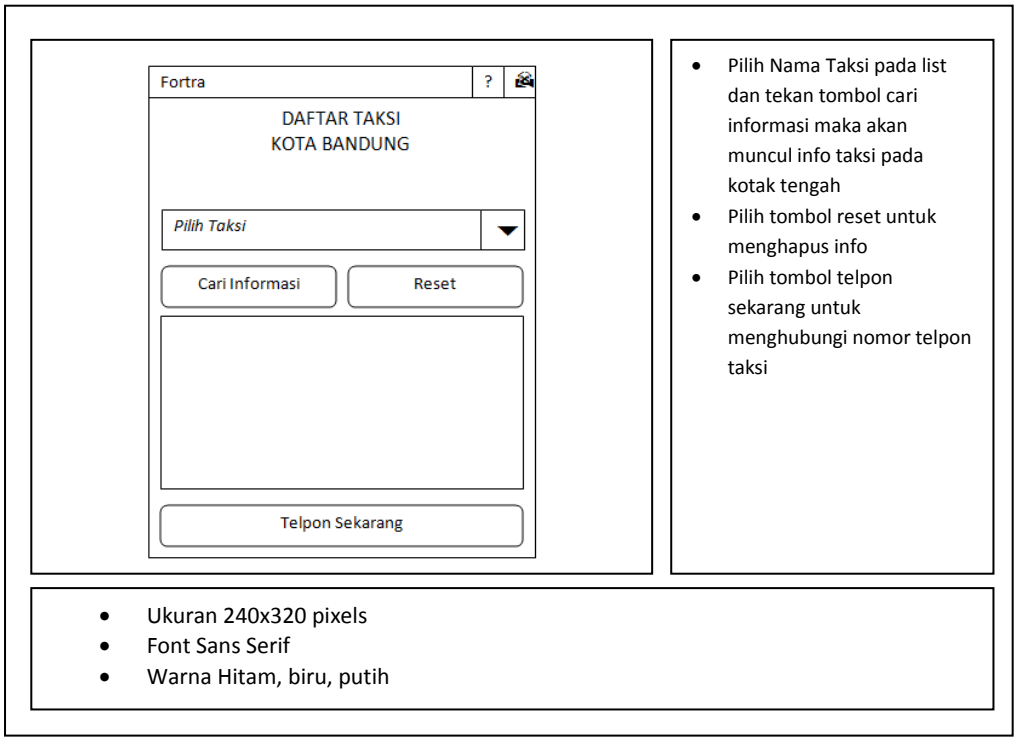
Gambar 4.20 Tampilan Antarmuka Info Bus Damri

4. Perancangan Antarmuka Info Angkot



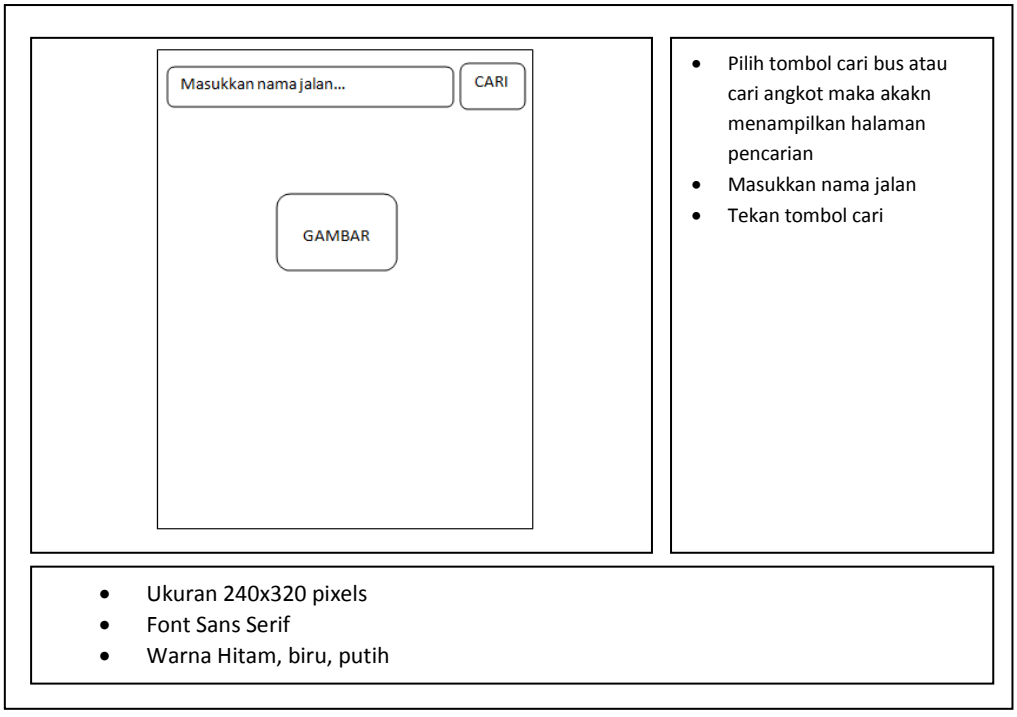
Gambar 4.21 Tampilan Antarmuka Info Angkot

5. Perancangan Antarmuka Info Taksi



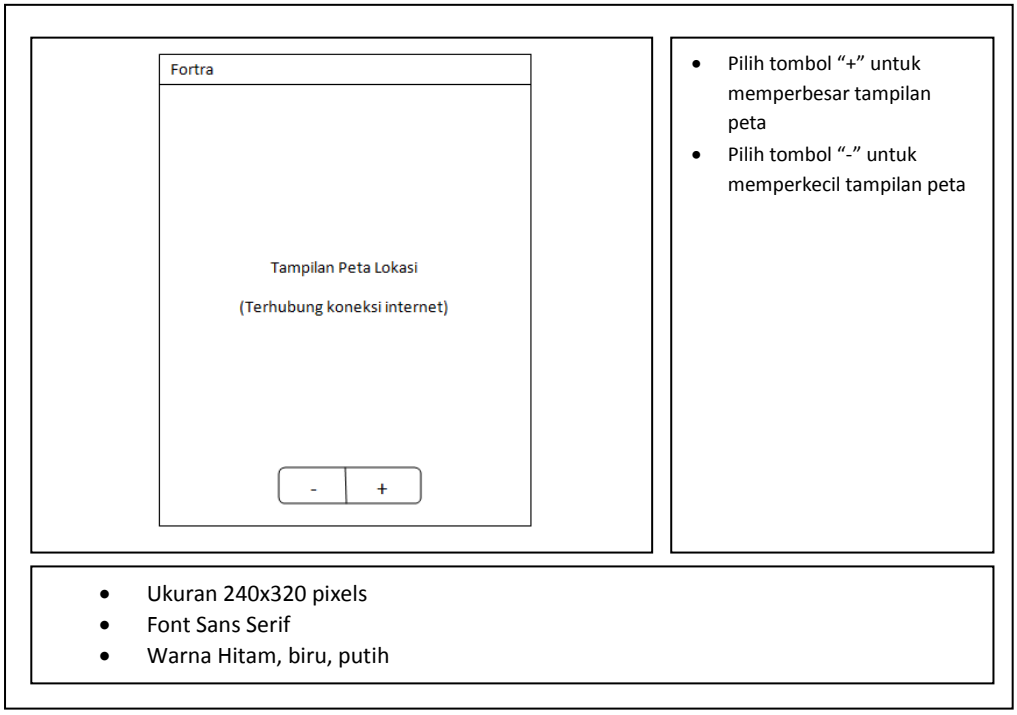
Gambar 4.22 Tampilan Antarmuka Info Taksi

6. Perancangan Antarmuka Pencarian Kendaraan



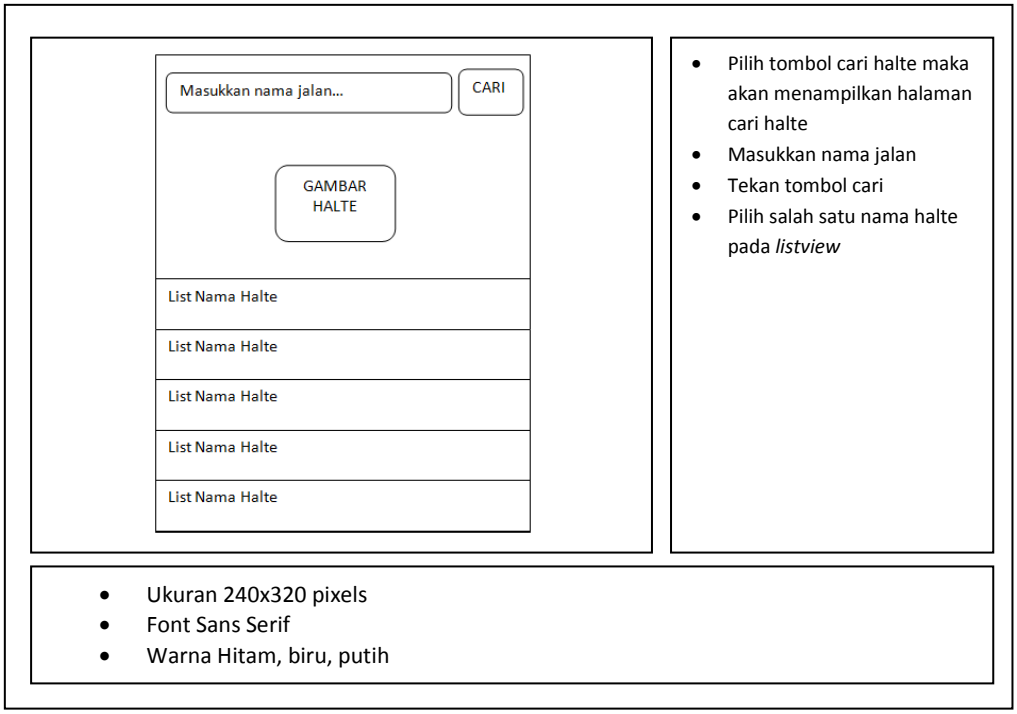
Gambar 4.23 Tampilan Antarmuka Pencarian Kendaraan

7. Perancangan Antarmuka Peta Posisi



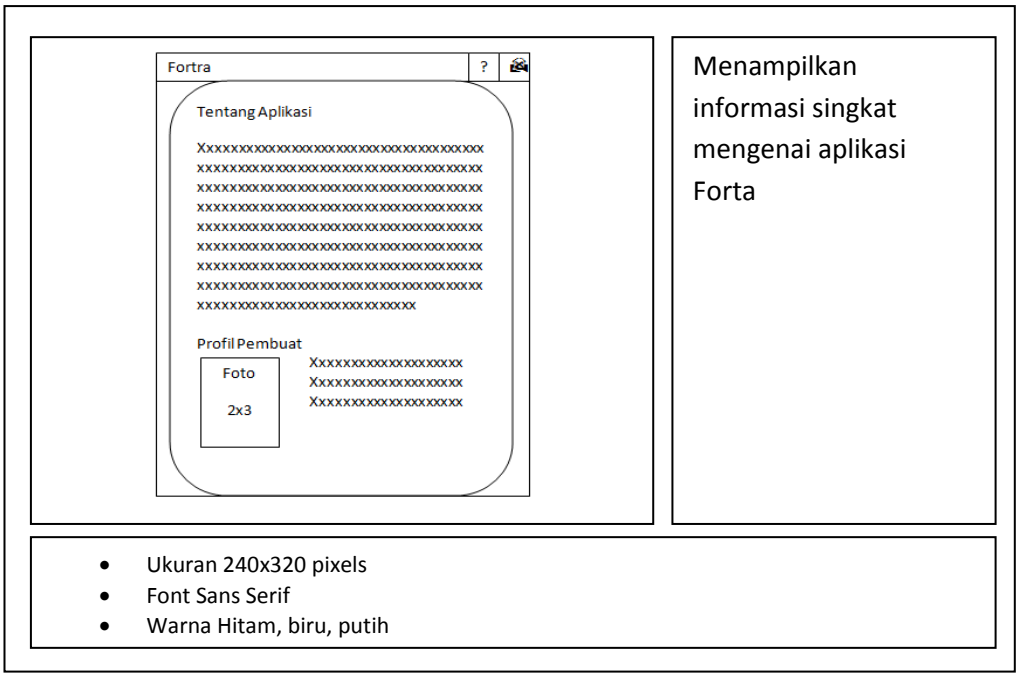
Gambar 4.24 Tampilan Antarmuka Peta Posisi

8. Perancangan Antarmuka Pencarian Halte



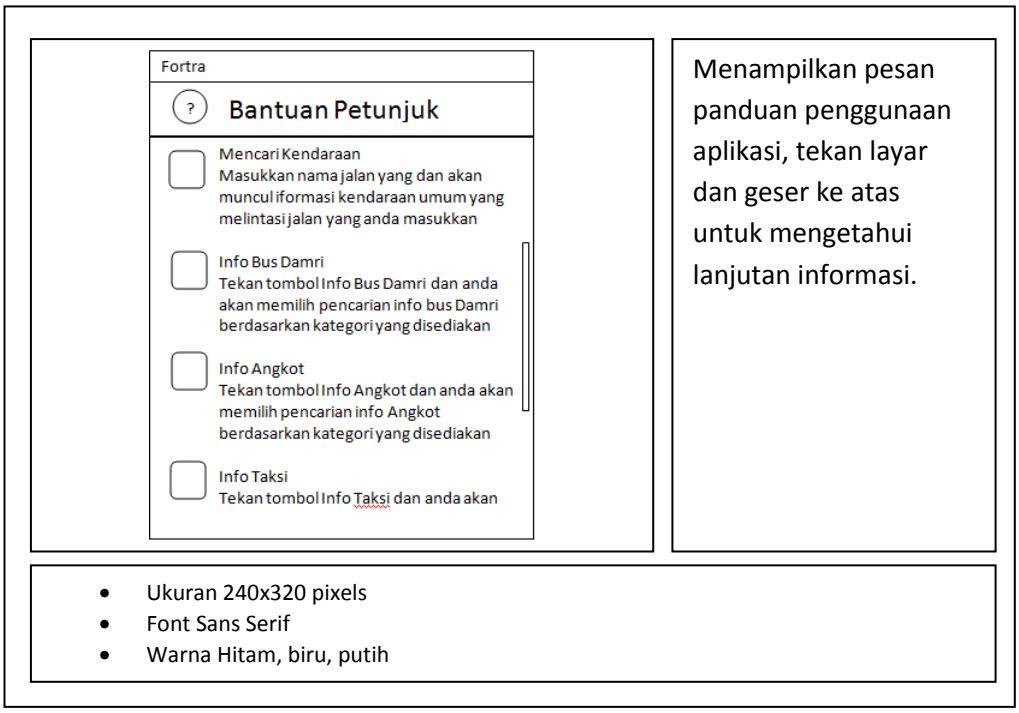
Gambar 4.25 Tampilan Antarmuka Pencarian Halte

9. Perancangan Antarmuka *About*



Gambar 4.26 Tampilan Antarmuka *About*

10. Perancangan Antarmuka Bantuan

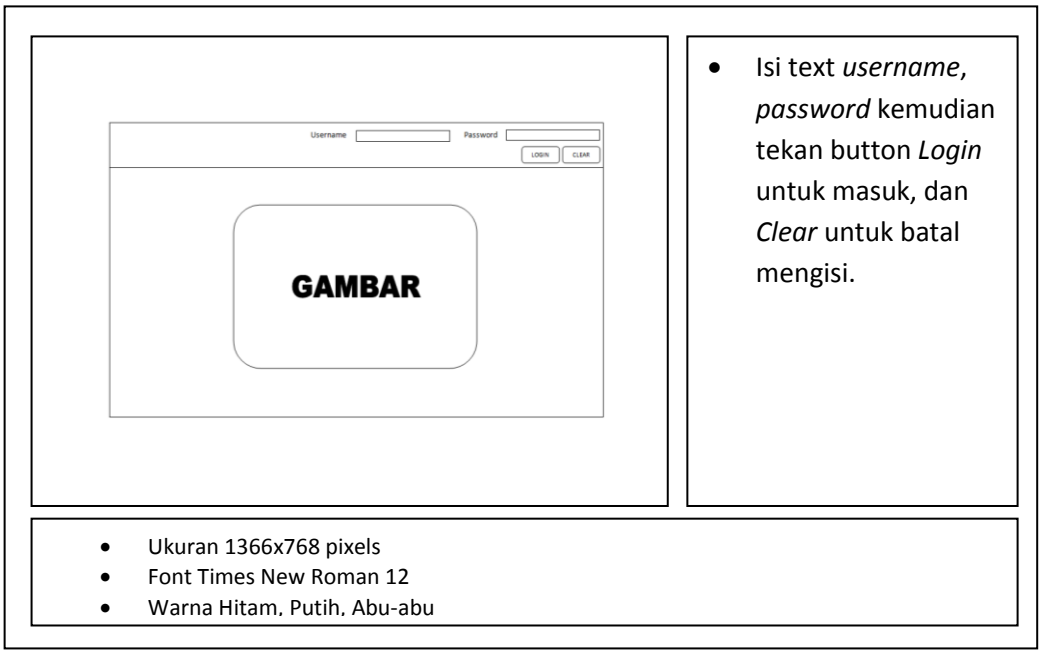


Gambar 4.27 Tampilan Antarmuka Bantuan

4.2.4.2 Perancangan antarmuka *Backend Application*

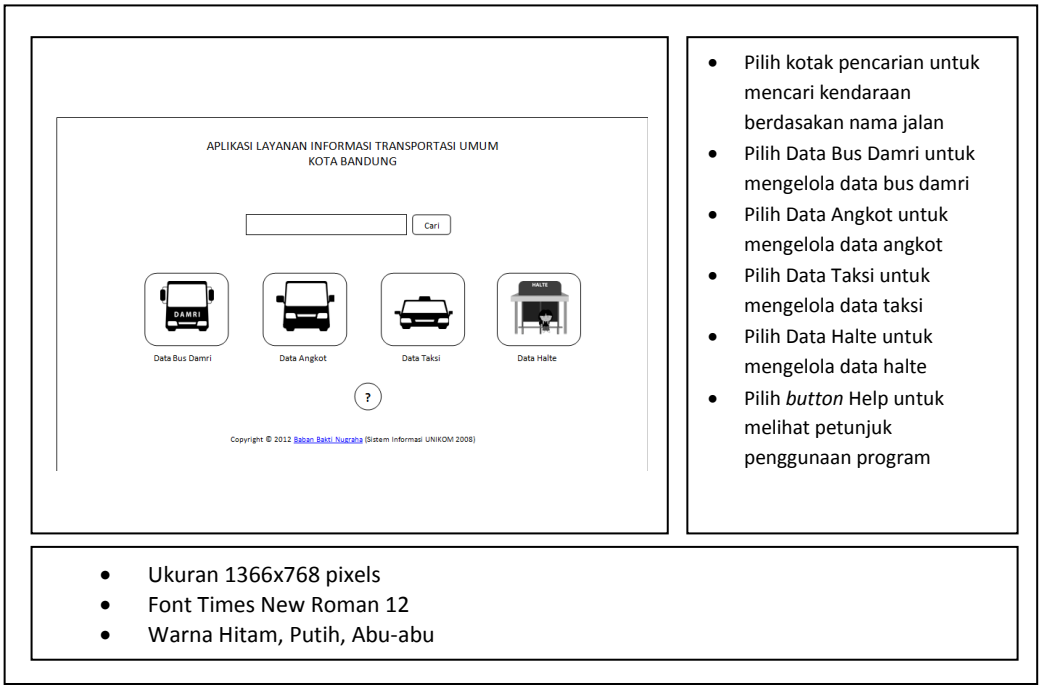
Tidak jauh berbeda dengan aplikasi *frontend*, aplikasi *backend* pun juga harus dibuat terlebih dahulu perrancangan tampilannya agar mudah dalam pengoperasiannya. Perancangan ini meliputi rancangan antarmuka *Login*, menu utama, halaman tambah data, halaman ubah data, halaman cari data, halaman *help*.

1. Perancangan Antarmuka *Login* Menu Admin



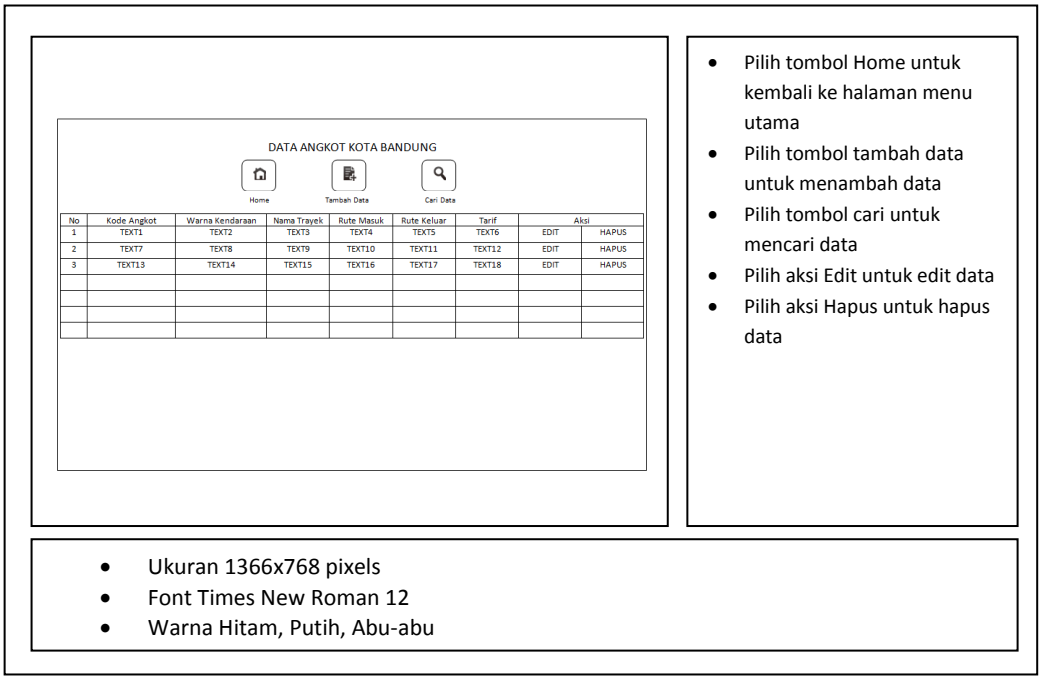
Gambar 4.28 Tampilan Antarmuka *Login* Menu Admin

2. Perancangan Antarmuka Menu Utama / *home*



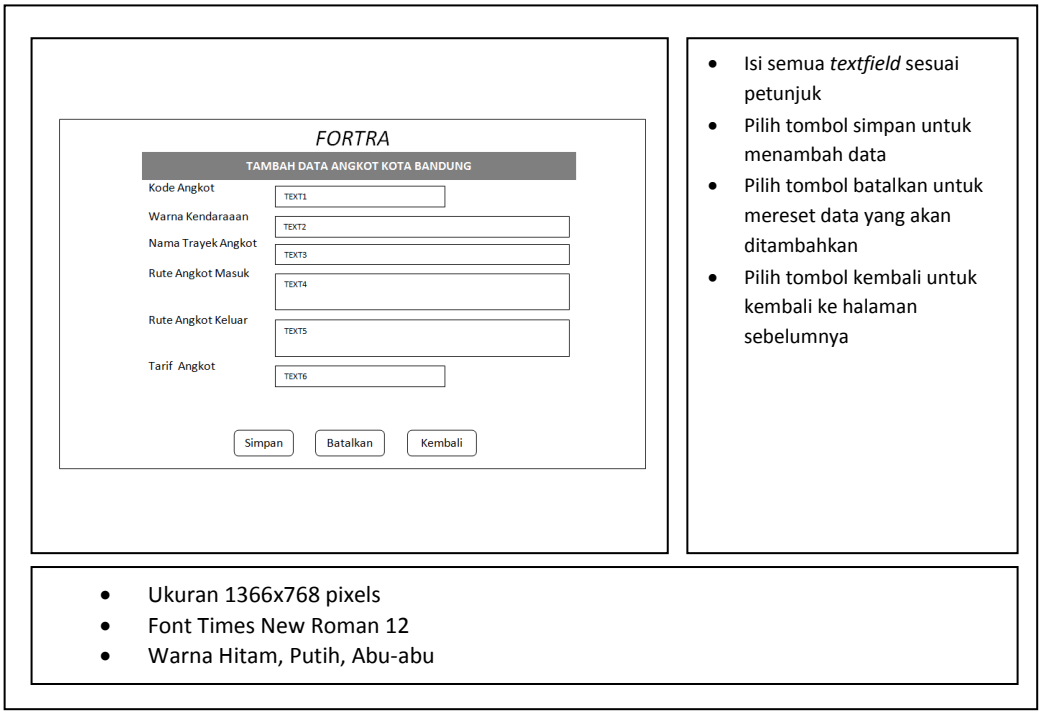
Gambar 4.29 Tampilan Antarmuka Menu Utama / *home*

3. Perancangan Antarmuka Pengelolaan Data



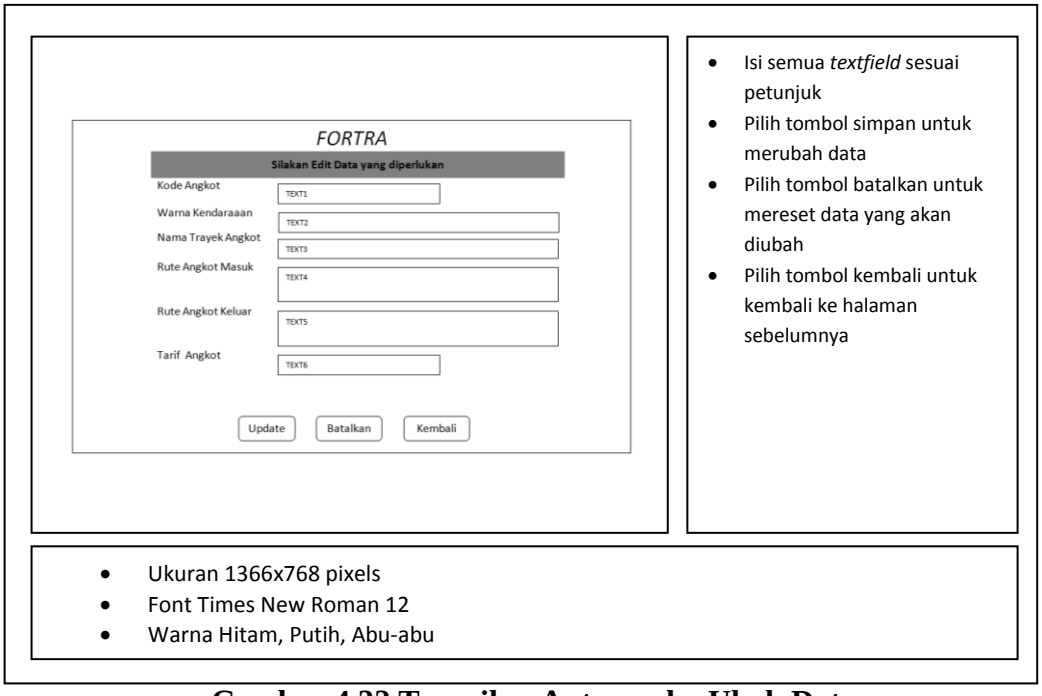
Gambar 4.30 Tampilan Antarmuka Pengelolaan Data

4. Perancangan Antarmuka Tambah Data



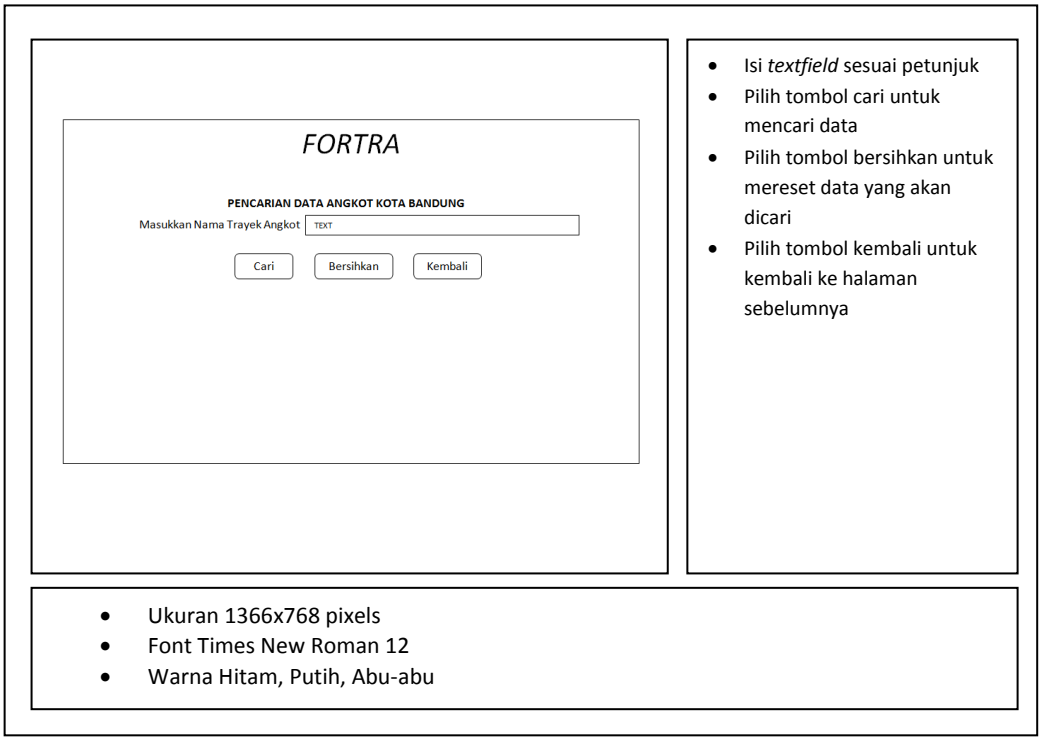
Gambar 4.31 Tampilan Antarmuka Tambah Data

5. Perancangan Antarmuka Ubah Data



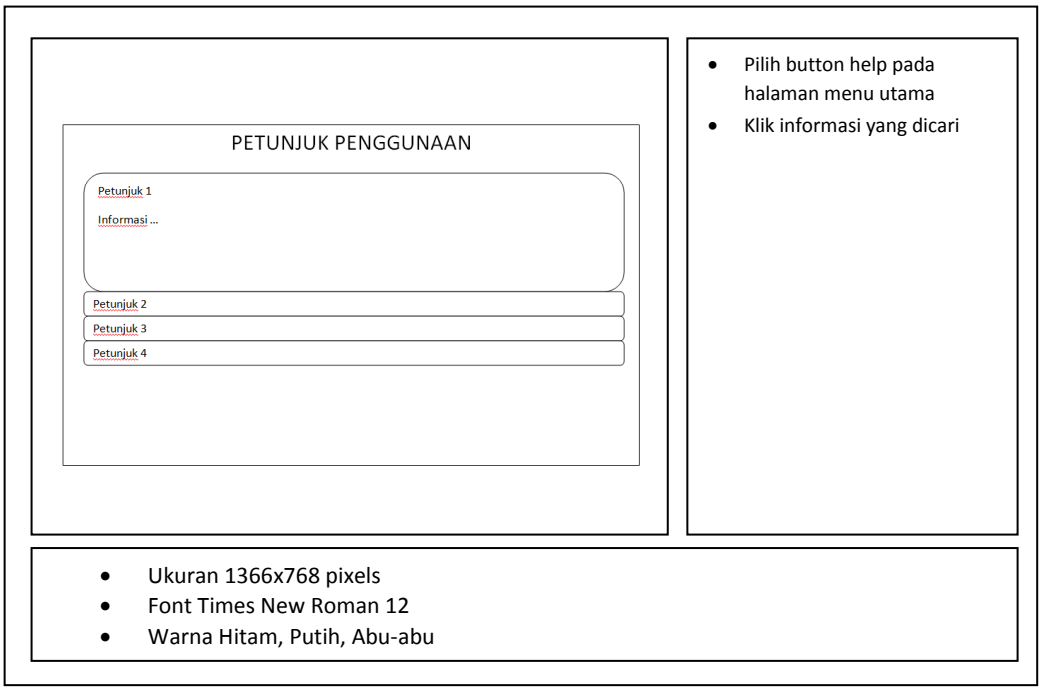
Gambar 4.32 Tampilan Antarmuka Ubah Data

6. Perancangan Antarmuka Pencarian Data



Gambar 4.33 Tampilan Antarmuka Pencarian Data

7. Perancangan Antarmuka *Help*



Gambar 4.34 Tampilan Antarmuka *Help*