BAB III
ANALISIS KEBUTUHAN DAN PERANCANGAN

3.1. Metode Rekayasa Perangkat Lunak

Dalam membangun sebuah perangkat lunak dibutuhkan metode pengerjaan sehingga perangkat lunak yang akan dibuat dapat berjalan dengan baik dan sesuai dengan waktu pengerjaan yang telah dibuat. Adapun metode yang digunakan dalam penelitian ini ialah dengan menggunakan metode *waterfall*.

3.2. Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak meliputi analisis kebutuhan masukan, analisis kebutuhan proses, analisis kebutuhan keluaran, analisis kebutuhan antarmuka pengguna dan analisis kebutuhan alat-alat bantu untuk perancangan perangkat lunak.

3.2.1 Analisis Kebutuhan Masukan

Adapun kebutuhan masukan yang dibutuhkan dalam sistem ini dari penggunanya adalah sebagai berikut :

1. Admin
    - *Username* dan *password* untuk admin *login* ke sistem
    - Data kurir yang berupa nama kurir, *username* kurir dan *password* agar kurir dapat *login* ke sistem untuk kurir.
    - Data pengirim berupa identitas dari pengirim yang terdiri dari id barang atau nomor resi, nama pengirim, barang yang dikirim, nomor telepon pengirim, alamat tujuan pengiriman barang, koordinat lokasi yang diimplementasikan menggunakan peta pada Google Map serta kurir yang mengantarkan barang tersebut.
    - Menentukan rute terpendek dengan memasukan alamat tujuan pengiriman barang lalu admin akan menentukan sendiri rute-

rute mana saja yang bisa dilalui oleh setiap kurir nanti dengan membuat titik-titik disetiap rutenya.

- Menentukan lokasi-lokasi mana yang diantar terlebih dahulu oleh setiap kurir

2. Konsumen
   - Nomor pengiriman barang atau nomor resi yang diberikan oleh pihak perusahaan jasa pengiriman guna untuk memantau barang yang dikirim
3. Kurir
   - *Username* dan *password* untuk kurir *login* ke sistem kurir

3.2.2 Analisis Kebutuhan Proses

Adapun kebutuhan proses yang dibutuhkan dalam sistem ini adalah sebagai berikut :

1. Manajemen kurir

   Manajemen kurir adalah proses yang berkaitan dengan data kurir. Proses yang terdapat pada manajemen kurir antara lain :

   a. Tambah kurir
   b. Ubah Kurir
   c. Hapus Kurir
2. Manajemen Pengiriman Barang

   Manajemen pengiriman barang adalah proses yang berkaitan dengan data pengiriman barang. Proses yang terdapat pada manajemen pengiriman barang antara lain :

   a. Tambah barang
   b. Hapus Barang
3. Menentukan lokasi konsumen

   Setiap lokasi konsumen akan didapatkan dengan memilih lokasi konsumen tersebut dengan memanfaatkan layanan peta pada Google Map sehingga ketika admin memilih lokasi maka akan mendapatkan titik koordinat lokasi konsumen tersebut.

4. Penentuan rute terpendek

   Pada proses ini akan ditentukan alamat tujuan pengiriman barang lalu dengan menggunakan perhitungan algoritma Dijkstra maka akan dipilih sebuah rute terpendek ke alamat tujuan pengiriman barang dengan membandingkan titik-titik rute dari lokasi awal ke alamat tujuan pengiriman barang dimana terdapat jarak pada setiap titik rute yang terhubung yang telah dipilih oleh admin. Jarak tersebut didapatkan dari Google Map itu sendiri.

5. Pemantauan posisi barang

   Sebelum melakukan pemantauan barang pengirim ataupun konsumen diwajibkan untuk memasukan nomor pengiriman barang atau nomor resi yang telah diberikan oleh pihak perusahaan jasa pengiriman. Pemantaun posisi barang dilakukan agar konsumen dapat mengetahui keberadaan barangnya yang dibawa oleh kurir. Pemantauan posisi barang ini akan memanfaatkan layanan peta pada Google Map sehingga konsumen dapat mengetahui barang yang dibawa oleh kurir melalui peta secara *real time*.

6. *Login* dan *logout*

   Proses *login* digunakan oleh admin dan kurir untuk dapat masuk ke sistem dengan melakukan verifikasi *username* dan *password*. *Logout* merupakan proses untuk menghilangkan hak akses pada sistem sehingga admin dan kurir tidak dapat menggunakan sistem tersebut dan diharukan untuk *login*.

3.2.3 Analisis Kebutuhan Keluaran

Adapun keluaran dari dalam sistem ini untuk penggunanya adalah sebagai berikut :

1. Admin
   - Informasi data kurir
   - Daftar pengiriman barang
   - Rute terpendek menuju lokasi tujuan pengiriman barang

2. Konsumen
    - Pemantauan barang yang dibawa oleh kurir dengan memanfaatkan layanan peta pada Google Map secara *real time*.
    - Data barang konsumen
3. Kurir
    - Rute terpendek menuju lokasi pengiriman barang
    - Data pengiriman barang

3.2.4 Analisis Kebutuhan Antarmuka Pengguna

Kebutuhan antarmuka sistem merupakan kebutuhan halaman-halaman yang akan ditampilkan pada sistem sebagai tempat menampilkan informasi-informasi yang dibutuhkan *user*. Setiap halaman mempunyai informasi yang berbeda sesuai kebutuhan halaman masing-masing. Adapun halaman-halaman antarmuka pengguna adalah sebagai berikut :

1. Halaman *Login*
2. Halaman Tambah Kurir
3. Halaman Ubah Kurir
4. Halaman Tambah Pengiriman Barang
5. Halaman Manajemen Rute Terpendek
6. Halaman Daftar Pengiriman Barang
7. Halaman *Home* Konsumen
8. Halaman *Tracking* Barang
9. Halaman *Login* Kurir
10. Halaman *Home* Kurir

3.2.5 Analisis Kebutuhan Alat-Alat Bantu Perancangan Perangkat Lunak

Dalam mengerjakan penelitian, dibutuhkan alat-alat bantu (*tools*) untuk melakukan perancangan alur berjalannya sistem yang akan dibuat. Alat-alat bantu (*tools*) yang digunakan untuk mengerjakan dan menyelesaikan penelitian dalam hal

perancangan model analisis, basis data maupun pembangunan perangkat lunak atau perangkat keras yang dipakai ketika membangun sistem yaitu :

1. Perangkat Lunak
    a. Software Ideas Modeler
    b. Sublime Text Editor
    c. PHP MyAdmin
    d. Firebase
    e. Xampp
    f. Pencil
    g. DbDesigner
    h. Windows 10
    i. GitHub

2. Perangkat Keras
    a. Komputer
        - Prosesor Core i3 atau lebih tinggi
        - RAM 2 GB
    b. Smartphone
        - GPS

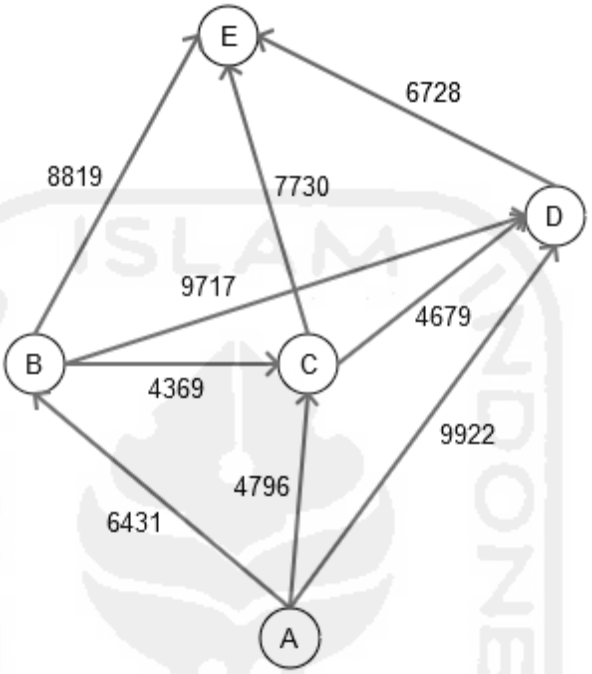
3.3. Analisis Kasus

Pada analisis kasus akan diberikan sebuah contoh kasus dalam permasalahan pengiriman barang guna untuk mencoba sistem dapat menyelesaikan permasalahan dalam kasus tersebut. Berikut merupakan contoh kasus dalam permasalahan pengiriman barang.

Contoh Kasus :

Seorang kurir diberikan tugas untuk mengantar barang dari lokasi pusat perushaan pengiriman jasa ke lokasi konsumen di Perumahan Dayu Permai No A8 Jalan Kalurang Km 8,5. Rute yang ditempuh oleh kurir dalam melakukan aktivitas pengiriman barang sehari-hari umumnya hanyalah rute yang sering (biasa) dilalui ataupun rute yang dianggap terpendek berdasarkan persepsi pribadi/orang lain yang

pada kenyataannya, hal tersebut belum tentu benar. Untuk itu diberikan alternatif-alternatif rute yang digambarkan dalam bentuk graf berarah dengan bobot yang tak bernilai negatif. Gambar 3.1 merupakan graf dari contoh analisis kasus.



Gambar 3.1 Graf Contoh Analisis Kasus

Penyelesaian Dari Analisis Kasus :

Diketahui tabel 3.1 merupakan tabel jarak antar titik.

Tabel 3.1 Tabel Jarak Contoh Kasus

Titik Rute	A	B	C	D	E
A	0	6431	4796	9922	0
B	0	0	4369	9717	8819
C	0	0	0	4679	7730
D	0	0	0	0	6728

Setelah jarak antar titik rute diketahui, maka berikutnya lakukan iterasi pertama dimana iterasi pertama dimulai dari titik A. Iterasi pertama dapat dilihat pada tabel 3.2.

Tabel 3.2 Tabel Perhitungan Dijkstra Iterasi Pertama Contoh Kasus

Iterasi 1	**A**	**B**	**C**	**D**	**E**	**L**
A	0_A	6431_A	$\underline{4796}_A$	9992_A	∞_A	{ A }

Iterasi pertama :

- Masukkan jarak antar titik A keseluruh titik yang dapat dilalui oleh titik A di mana jarak tersebut merupakan jarak dari titik A.
- Jika ada rute yang tidak dilalui oleh titik A maka jarak bernilai ∞.
- Bandingkan keseluruhan jarak yang dilalui oleh titik A, kemudian pilih jarak yang terpendek.
- Dipilih jarak 4796 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik A ke titik C, lalu beri tanda pada jarak tersebut.
- Karena jarak yang dipilih merupakan jarak dari titik A ke titik C maka titik C dipilih sebagai iterasi berikutnya.

Selanjutnya lakukan iterasi kedua. Iterasi kedua dapat dilihat pada tabel 3.3.

Tabel 3.3 Tabel Perhitungan Dijkstra Iterasi Kedua Contoh Kasus

Iterasi 2	**A**	**B**	**C**	**D**	**E**	**L**
A	0_A	6431_A	4796_A	9992_A	∞_A	{ A, C }
C	0_A	$\underline{6431}_A$	4796_A	9475_C	12526_C	{ A, C, B }

Iterasi kedua :

- Titik C sekarang bernilai 4796 dari titik A.

- Masukkan jarak antar titik C keseluruh titik yang dapat dilalui oleh titik C di mana jarak tersebut merupakan akumulasi jarak titik C.
- Jika ada rute yang tidak dilalui oleh titik C maka jarak bernilai jarak sebelumnya.
- Jika ada rute yang dilalui oleh titik C dan tidak memiliki jarak maka masukan jarak dari titik C ke titik tersebut.
- Jika ada rute yang dilalui oleh titik C dan telah memiliki jarak maka bandingkan jarak sebelumnya dengan jarak baru.
- Karena titik C ke titik D mempunyai jarak (4796 + 4679) = 9575 dan telah ada jarak sebelumnya di titik D dari titik A maka bandingkan kedua jarak tersebut. Kemudian pilih yang terpendek.
- Dipilih jarak 9575 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik C
- Bandingkan keseluruhan jarak yang dilalui oleh titik C, kemudian pilih jarak yang terpendek.
- Dipilih jarak 6431 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik A ke titik B, lalu beri tanda pada jarak tersebut.
- Karena jarak yang dipilih merupakan jarak dari titik A ke titik B maka titik B dipilih sebagai iterasi berikutnya.

Selanjutnya lakukan iterasi ketiga. Iterasi ketiga dapat dilihat pada tabel 3.4.

Tabel 3.4 Tabel Perhitungan Dijkstra Iterasi Ketiga Contoh Kasus

Iterasi 3	**A**	**B**	**C**	**D**	**E**	**L**
A	0_A	6431_A	4796_A	9992_A	∞_A	{ A, C }
C	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B }
B	0_A	6431_A	4796_A	$\underline{9475}_C$	12526_C	{ A, C, B, D }

Iterasi ketiga :

- Titik B sekarang bernilai 6431 dari titik A.
- Masukkan jarak antar titik B keseluruh titik yang dapat dilalui oleh titik B di mana jarak tersebut merupakan akumulasi jarak titik B.
- Jika ada rute yang tidak dilalui oleh titik B maka jarak bernilai jarak sebelumnya.
- Jika ada rute yang dilalui oleh titik B dan tidak memiliki jarak maka masukan jarak dari titik B ke titik tersebut.
- Jika ada rute yang dilalui oleh titik B dan telah memiliki jarak maka bandingkan jarak sebelumnya dengan jarak baru.
- Karena titik B ke titik D mempunyai jarak (6431 + 9717) = 16148 dan telah ada jarak sebelumnya di titik D dari titik C maka bandingkan kedua jarak tersebut. Kemudian pilih yang terpendek.
- Dipilih jarak 9575 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik C
- Karena titik B ke titik E mempunyai jarak (6431 + 8819) = 15250 dan telah ada jarak sebelumnya di titik E dari titik C maka bandingkan kedua jarak tersebut. Kemudian pilih yang terpendek.
- Dipilih jarak 12526 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik C
- Bandingkan keseluruhan jarak yang dilalui oleh titik B, kemudian pilih jarak yang terpendek.
- Dipilih jarak 9575 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik C ke titik D, lalu beri tanda pada jarak tersebut.
- Karena jarak yang dipilih merupakan jarak dari titik C ke titik D maka titik D dipilih sebagai iterasi berikutnya.

Selanjutnya lakukan iterasi keempat. Iterasi keempat dapat dilihat pada tabel 3.5.

Tabel 3.5 Tabel Perhitungan Dijkstra Iterasi Keempat Contoh Kasus

Iterasi 4	A	B	C	D	E	L
A	0_A	6431_A	4796_A	9992_A	∞_A	{ A, C }
C	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B }
B	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B, D }
D	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B, D, E }

Iterasi keempat :

- Titik D sekarang bernilai 9475 dari titik C.
- Masukkan jarak antar titik D keseluruh titik yang dapat dilalui oleh titik D di mana jarak tersebut merupakan akumulasi jarak titik D.
- Jika ada rute yang tidak dilalui oleh titik D maka jarak bernilai jarak sebelumnya.
- Jika ada rute yang dilalui oleh titik D dan tidak memiliki jarak maka masukan jarak dari titik D ke titik tersebut.
- Jika ada rute yang dilalui oleh titik D dan telah memiliki jarak maka bandingkan jarak sebelumnya dengan jarak baru.
- Karena titik D ke titik E mempunyai jarak (9475 + 6728) = 16203 dan telah ada jarak sebelumnya di titik D dari titik C maka bandingkan kedua jarak tersebut. Kemudian pilih yang terpendek.
- Dipilih jarak 12526 sebagai jarak terpendek di mana jarak tersebut merupakan jarak dari titik C.
- Karena tidak ada jarak yang bisa dibandingkan dan titik E merupakan titik tujuan maka iterasi pun berhenti.

Setelah iterasi berhenti maka selanjutnya cari rutenya. Rute yang dipilih dapat dilihat pada tabel 3.6.

Tabel 3.6 Tabel Pemilihan Rute Terpendek Contoh Kasus

Rute	A	B	C	D	E	L
A	0_A	6431_A	4796_A	9992_A	∞_A	{ A, C }
C	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B }
B	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B, D }
D	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, B, D, E }
E	0_A	6431_A	4796_A	9475_C	12526_C	{ A, C, E }

Pemilihan Rute :

- Karena titik E merupakan titik tujuan, lihat jarak pada titik E.
- Jarak pada titik E merupakan jarak dari titik C, maka lihat titik C.
- Jarak pada titik C merupakan jarak dari titik A.
- Titik A merupakan titik awal.
- Maka rute terpendek yang dipilih ialah A - C - E.

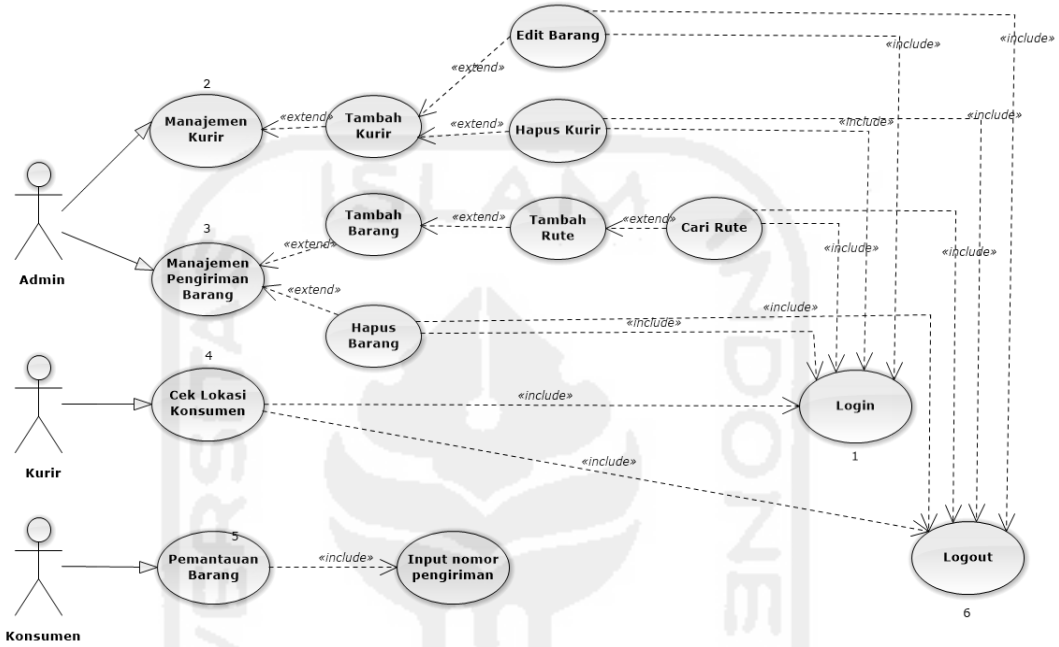
3.4. Pemodelan Analisis

Pemodelan pada sistem ini menggunakan metodologi pemodelan dengan UML (*Unified Modelling Language*). Diagram yang digunakan dalam sistem ini antara lain *Use Case* Diagram, *Activity* Diagram dan *Class* Diagram. Metode yang digunakan didalam perancangan sistem ini disesuaikan dengan menggunakan konsep OOP (*Object Oriented Programming*) atau PBO (Pemrograman Berorientasi Objek).

3.4.1 Perancangan *Use Case* Diagram

Use Case diagram digunakan untuk menggambarkan skenario interaksi antara *user* dengan aktivitas-aktivitas yang terdapat didalam sistem. Didalam diagram ini terdapat beberapa *user* dimana setiap *user* mempunyai aktivitas-

aktivitas yang dapat dilakukan terhadap sistem. Diagram ini ditujukan agar dalam pembangunan sistem tidak terjadi kesalahan skenario interaksi antar *user* yang menggunakan sistem dengan aktivitas yang terjadi dalam sistem. *Use Case* diagram pada sistem ini dapat dilihat pada gambar 3.2**.**



Gambar 3.2 *Use Case* Diagram

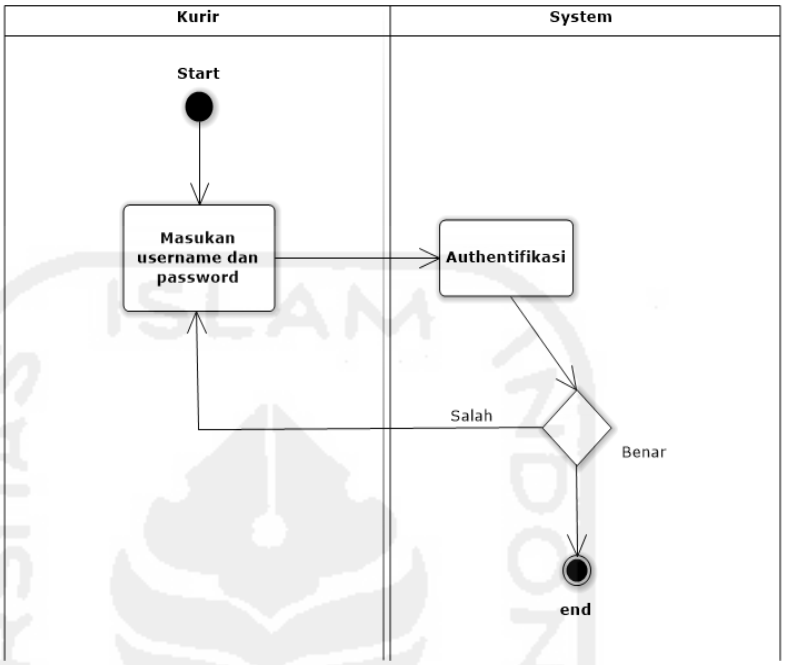
3.4.2 Perancangan *Activity* Diagram

Activity Diagram merupakan diagram yang menggambarkan aktivitas-aktivitas apa saja yang terjadi selama sistem dijalankan. Diagram ini menjelaskan proses-proses apa saja yang terjadi dalam sistem mulai dari aktivitas tersebut dimulai sampai dengan berakhirnya aktivitas tersebut. Dari *use* case diagram yang telah dibuat, maka dapat dihasilkan empat aktivitas yang dapat dibentuk menjadi empat *activity* diagram sebagai berikut :

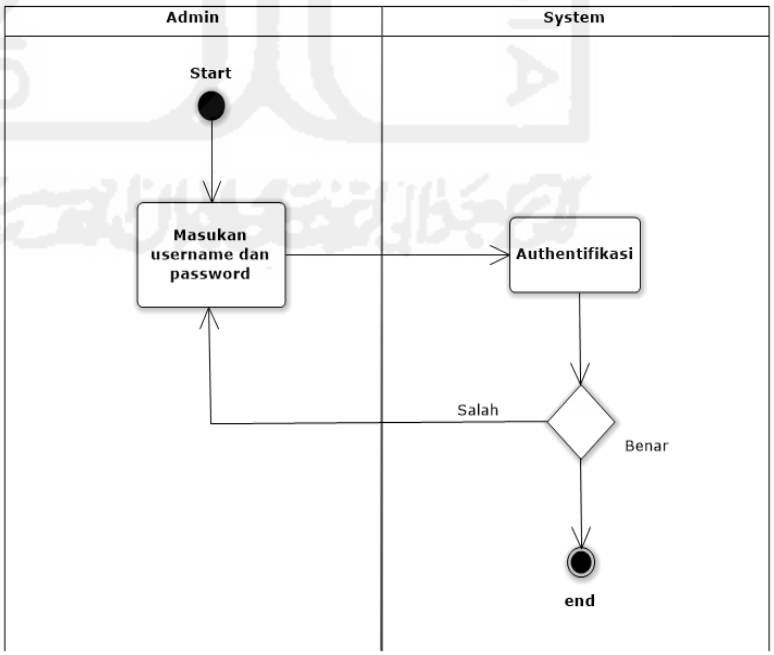
1. *Login*

   Sebelum admin atau kurir masuk ke sistem, admin atau kurir diminta untuk *login* terlebih dahulu dengan memasukan *username* dan *password* agar dapat menggunakan hak mereka pada sistem. Gambar 3.3 dan gambar 3.4 merupakan gambaran umum yang menjelaskan

proses *login* admin dan *login* kurir, sistem akan verifikasi *username* dan *password* yang telah dimasukan oleh admin.



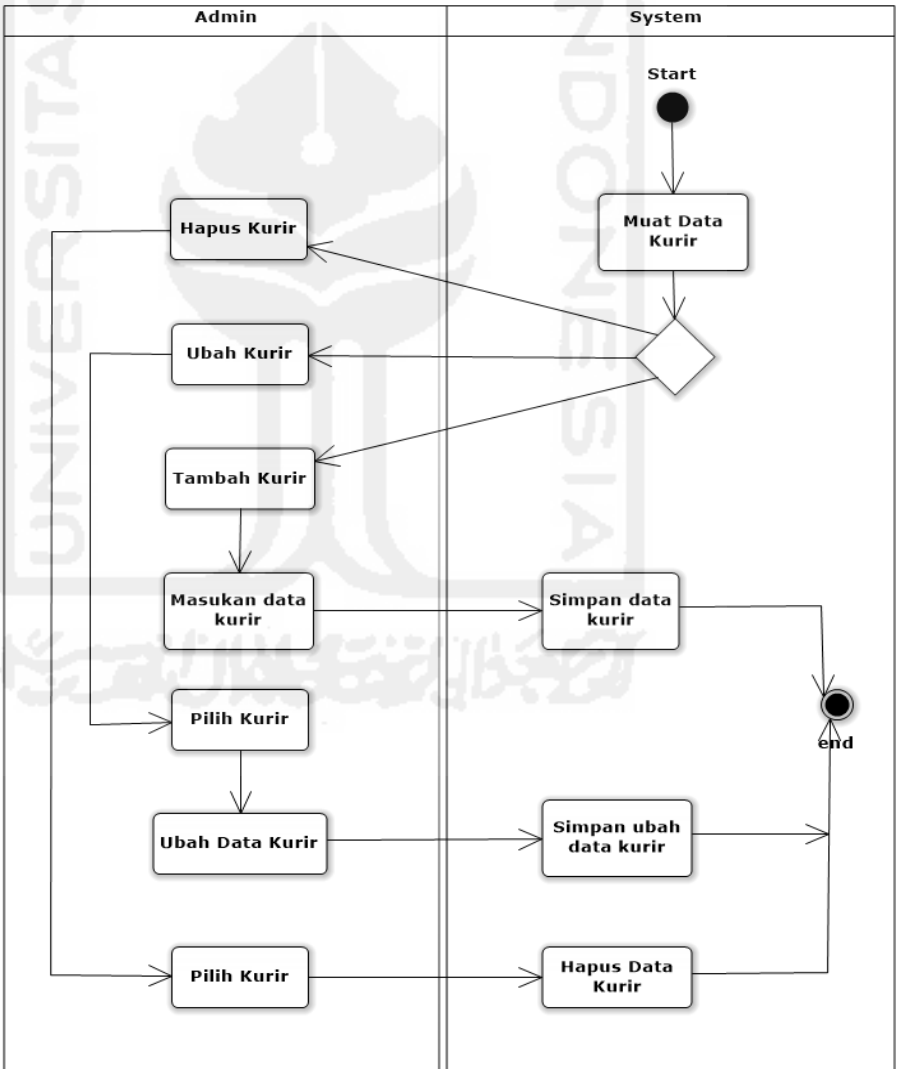
Gambar 3.3 *Activity* Diagram *Login* Kurir



Gambar 3.4 *Activity* Diagram *Login* Admin

2. Manajemen Kurir

Manajemen kurir yang dimaksud ialah memberikan kurir akses untuk masuk ke sistem kurir. Admin harus mengisi data ke dalam *form* yang telah disediakan oleh sistem yaitu *username* dan *password* dimana data tersebut akan digunakan oleh kurir untuk masuk ke sistem kurir. Gambar 3.5 merupakan gambaran umum yang menjelaskan proses tambah kurir, ubah kurir dan hapus kurir. Setelah data tersimpan, kurir dapat masuk ke sistem kurir sesuai dengan *username* dan *password* yang telah dimasukan oleh admin.

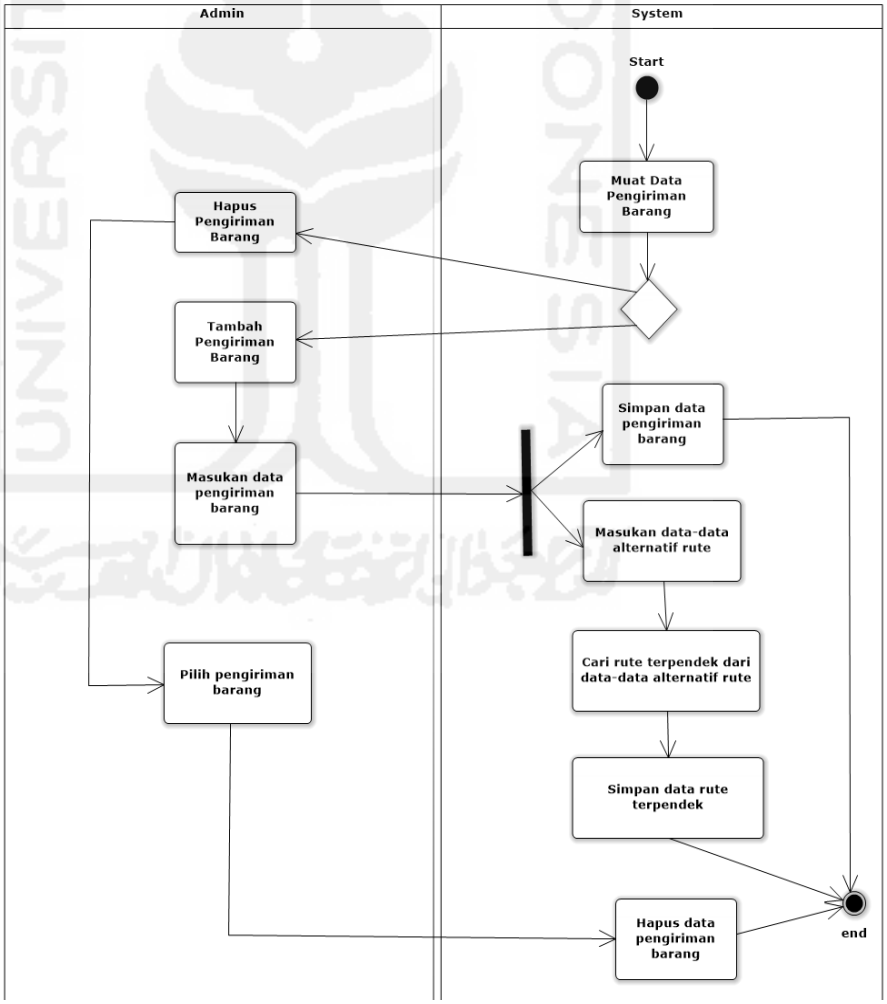


Gambar 3.5 *Activity* Diagram Manajemen Kurir

3. Manajemen Pengiriman Barang

Admin harus memasukan data pengiriman barang dimana data pengiriman barang merupakan data barang yang nantinya akan dikirim oleh kurir.

Gambar 3.6 merupakan gambaran umum dari proses tambah dan hapus pengiriman serta memanajemeni rute pengiriman barang ke lokasi tujuan pengiriman barang tersebut. Setelah admin memasukan data pengiriman barang, admin dapat membuat atau mengatur sendiri rute-rute mana saja yang nantinya dapat dilalui oleh kurir, lalu sistem akan mencari rute terpendek dari beberapa rute yang telah dimasukan oleh admin.

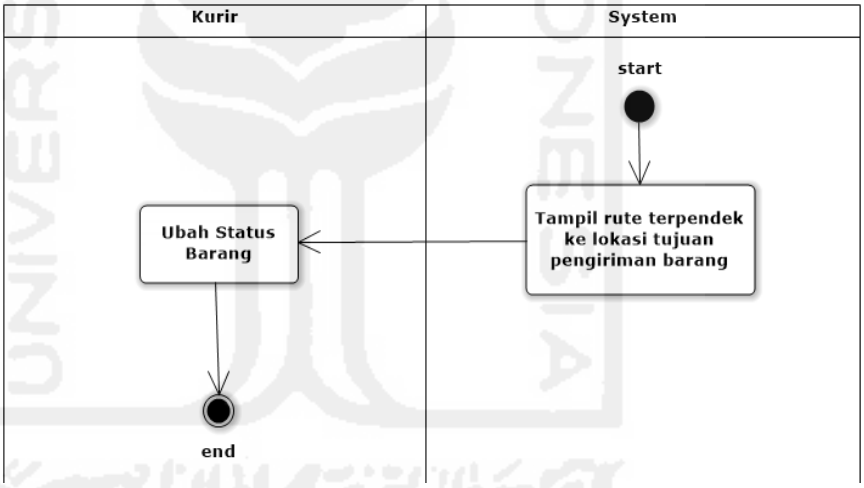


Gambar 3.6 *Activity* Diagram Manajemen Pengiriman Barang

4. Cek Lokasi Konsumen

   Kurir yang telah dapat masuk ke dalam sistem kurir nantinya akan mengetahui barang apa yang harus dikirimkan serta akan mendapatkan rute terpendek ke lokasi tujuan pengiriman. Kurir hanya tinggal mengikuti rute terpendek yang telah didapatkannya untuk mengirim barang tersebut.

   Gambar 3.7 merupakan gambaran umum dari proses cek lokasi konsumen. Setelah kurir mendapatkan rute pengiriman barang terpendek dan telah tiba ke lokasi konsumen maka kurir dapat mengubah status barang yang dikirimkan tersebut sebagai konfirmasi bahwa barang yang dikirimkan oleh kurir tersebut telah diterima oleh konsumen.



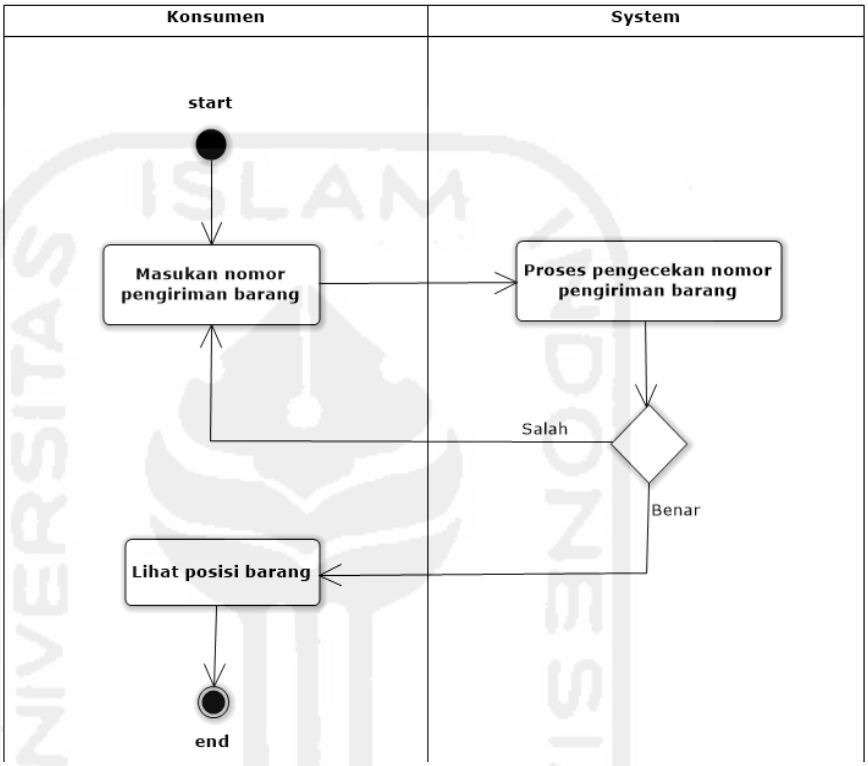
Gambar 3.7 *Activity* Diagram Cek Lokasi Konsumen

5. Pemantauan Barang

   Pemantauan barang merupakan proses utama yang berjalan di dalam sistem, dimana tujuan utama penelitian ini adalah pemantauan pengiriman barang.

   Gambar 3.8 merupakan gambaran umum dari pemantauan barang. Sebelum konsumen memantau barang, konsumen diharuskan untuk memasukan nomor pengiriman barang atau nomor resi yang diberikan

oleh pihak perusahaan jasa pengiriman barang sesuai dengan barang yang dikirimkan. Setelah nomor pengiriman barang sesuai maka konsumen dapat memantau barang yang dibawa oleh kurir melalui Google Map secara *realtime*.

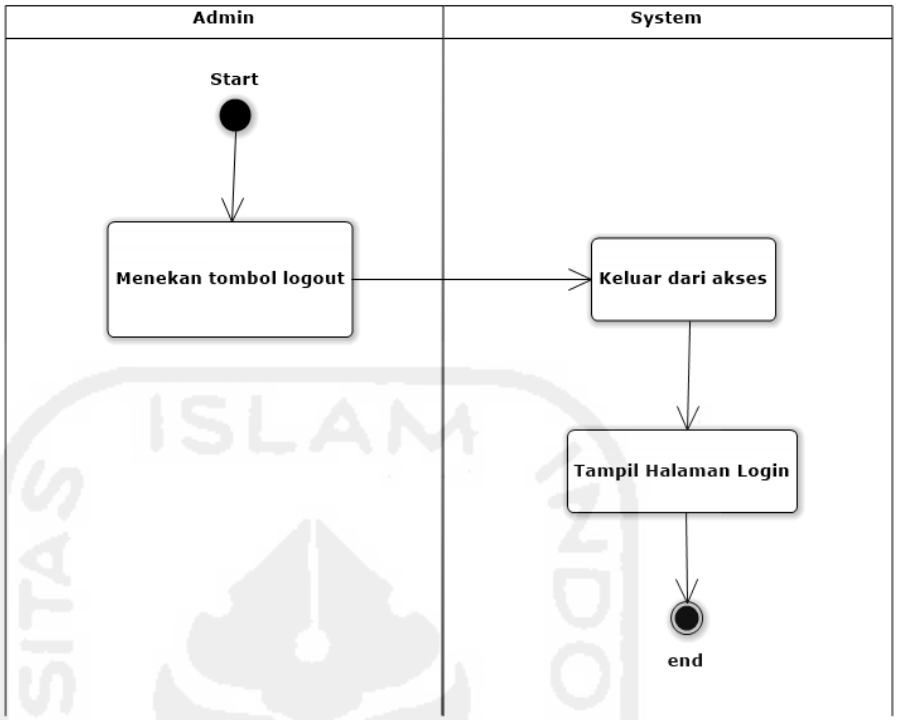


Gambar 3.8 *Activity* Diagram Pemantauan Barang

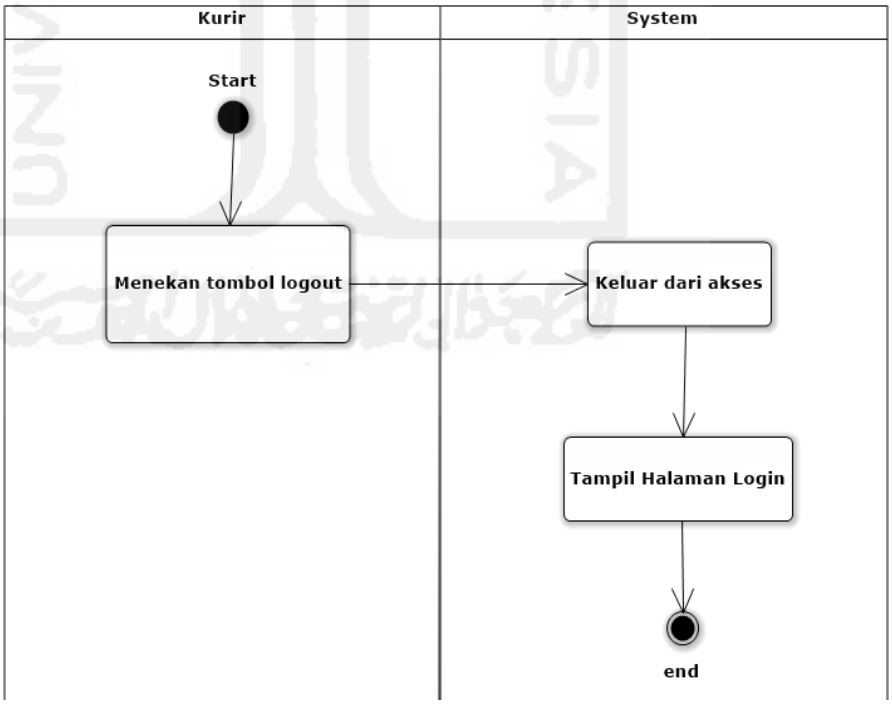
6. *Logout*

Logout merupakan proses menghilangkan hak akses admin atau kurir atas sistem.

Proses ini dibuat agar ketika admin atau kurir berhasil *login* dan ada *user* lain yang akan menggunakan komputer atau *smartphone* miliknya admin atau kurir itu, maka *user* tersebut dapat mendapatkan hak akses atas sistem tersebut. Hal ini dikarenakan admin atau kurir tersebut belum menghilangkan hak akses atas sistem tersebut. Gambar 3.9 dan gambar 3.10 merupakan gambaran umum yang menjelaskan proses *logout* admin dan *logout* kurir.



Gambar 3.9 *Activity* Diagram *Logout* Kurir

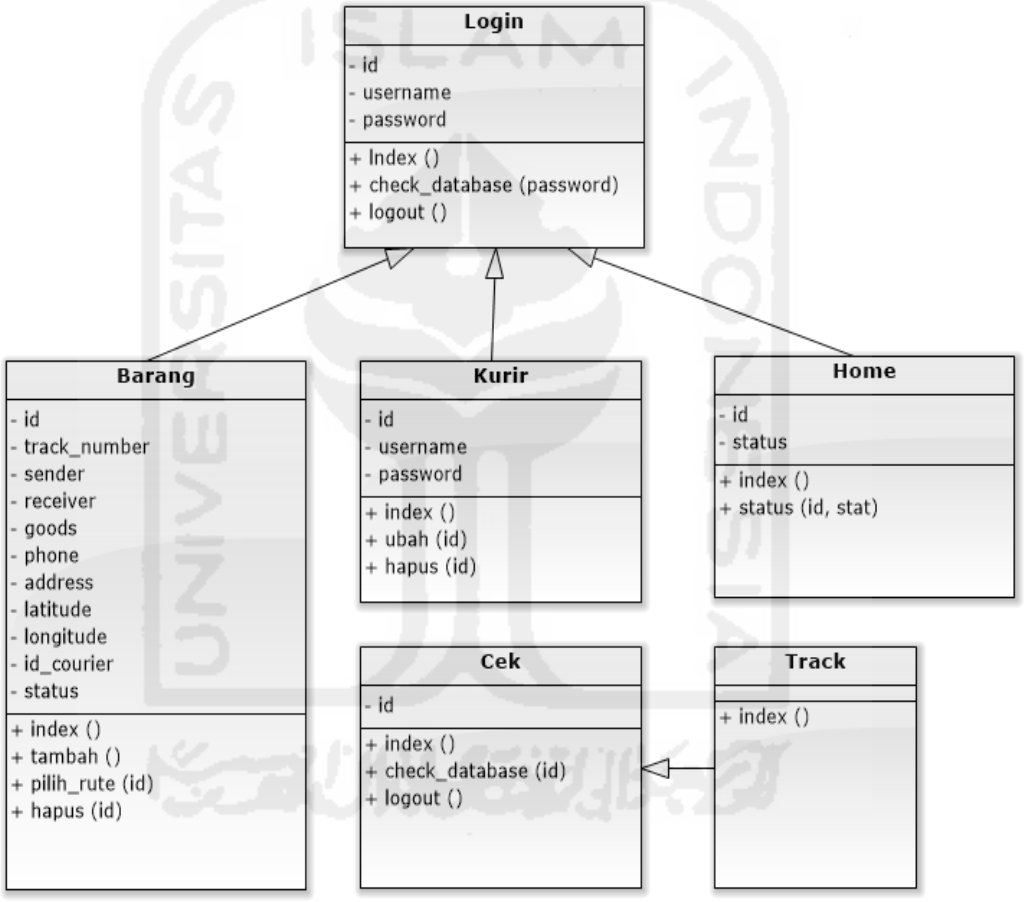


Gambar 3.10 *Activity* Diagram *Logout* Admin

3.4.3 Perancangan *Class* Diagram

Class diagram menggambarkan struktur statis *class* didalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem.

Gambar 3.11 merupakan gambaran umum yang menjelaskan dari *class* diagram pada sistem ini. Terdapat beberapa *class* serta relasi antar *class* pada sistem ini yaitu *class* barang, *class* kurir, *class* home yang berelasi dengan *class login* serta *class* track yang berelasi dengan *class* track.



Gambar 3.11 *Class* Diagram

3.5. Perancangan Basis Data

Basis data dibutuhkan sebagai tempat untuk meyimpan serta mengumpulkan informasi kedalam komputer secara sistematik. Informasi tersebut

nantinya akan digunakan untuk ditampilkan di sistem. Adapun struktur tabel pada sistem ini antara lain :

1. Tabel Admin

   Tabel 3.7 merupakan skema tabel admin yang berisi data admin untuk masuk ke sistem admin. Data administrator yang disimpan meliputi id, *username, fullname* dan *password*.

   Tabel 3.7 Tabel Admin

No	Atribut	Tipe Data	Indekx
1	id	integer(10)	Primary Key
2	username	varchar(50)	
3	fullname	varchar(50)	
4	password	varchar(100)	

2. Tabel *Courier*

   Tabel 3.8 merupakan skema tabel *courier* yang berisi data kurir untuk masuk ke sistem kurir serta. Data *courier* yang disimpan meliputi id_*courie*r, *username, password, latitude, longitude*.

   Tabel 3.8 Tabel *Courier*

No	Atribut	Tipe Data	Indekx
1	id_courier	integer(10)	Primary Key
2	username	varchar(50)	
3	password	varchar(100)	
4	latitude	varchar(255)	
5	longitude	varchar(255)	

3. Tabel *Goods*

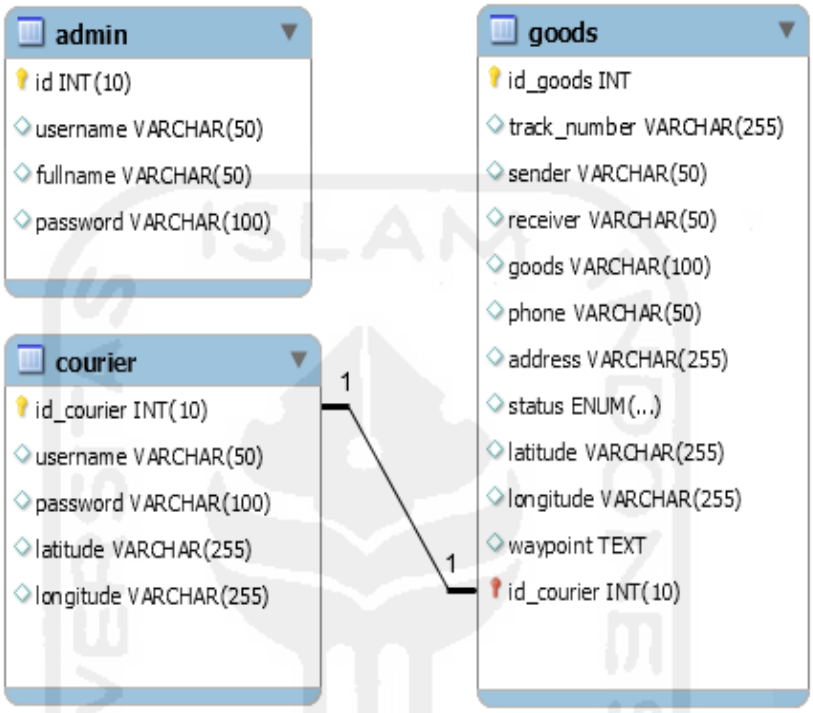
Tabel 3.9 merupakan skema tabel *goods* yang berisi data barang yang akan dikirimkan. Data *goods* yang disimpan meliputi id_*goods*, *track_number*, *sender*, *receiver*, *goods*, *phone*, status, *latitude*, *longitude*, *waypoint* dan id_*courier*.

Tabel 3.9 Tabel *Goods*

No	Atribut	Tipe Data	Indekx
1	id_goods	integer(10)	Primary Key
2	track_number	varchar(255)	
3	sender	varchar(50)	
4	receiver	varchar(50)	
5	goods	varchar(100)	
6	phone	varchar(50)	
7	address	varchar(255)	
8	status	enum('on-the-way','sent')	
9	latitude	varchar(255)	
10	longitude	varchar(255)	
11	waypoint	text	
12	id_courier	int(10)	Foreign Key

Berdasarkan dengan tabel-tabel yang telah dibentuk terdapat tabel yang memiliki relasi yang sesuai dengan bentuk relasinya. Gambar 3.12 merupakan

gambar relasi antar tabel dalam sistem. Gambar tersebut menggambarkan relasi antar tabel yang satu dengan yang lainnya untuk mendukung berjalannya sistem sesuai kebutuhannya.



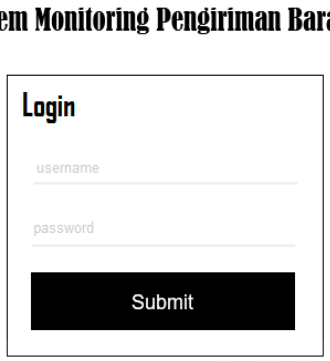
Gambar 3.12 Relasi Tabel

3.6. Perancangan Antarmuka Pengguna

Pada tahap perancangan antarmuka ini akan diberikan gambaran umum tentang perancangan antamuka sistem yang akan ditampilkan pada *user*. Adapun perancangan antarmuka yaitu :

1. Halaman *Login* Admin

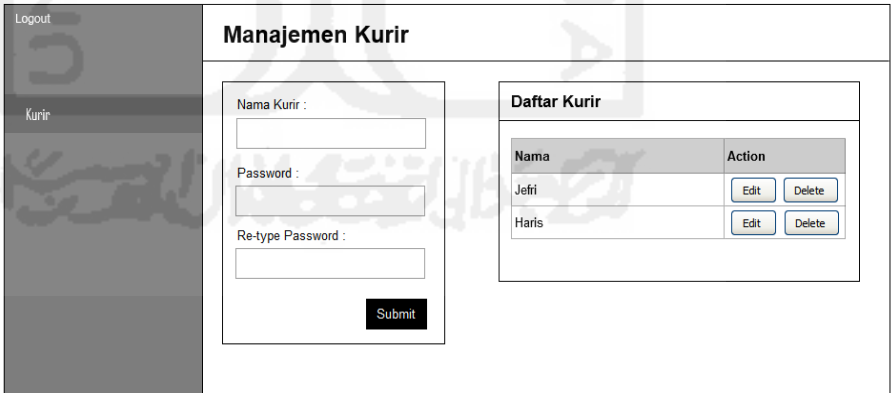
   Pada gambar 3.13 merupakan perancangan antarmuka halaman *login* admin. Sebelum admin mengakses halaman admin, admin diminta untuk memasukan *username* dan *password* untuk dapat masuk ke halaman admin.



Gambar 3.13 Rancangan Halaman *Login* Admin

2. Halaman Tambah Kurir

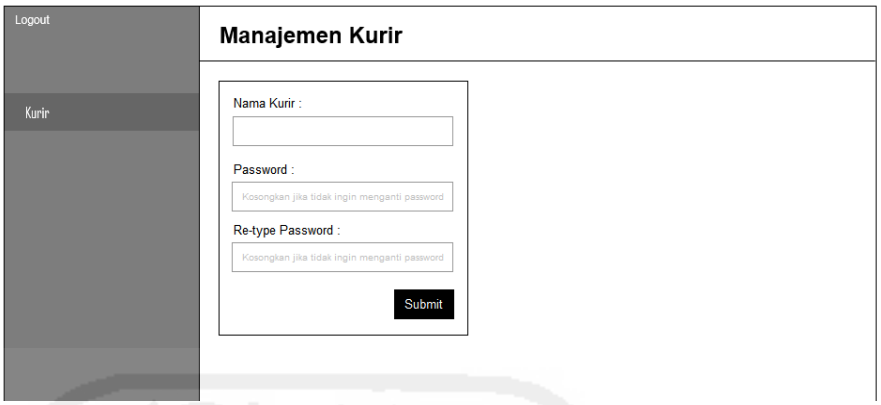
   Pada gambar 3.14 merupakan perancangan antarmuka halaman manajemen kurir. Setelah Admin berhasil masuk ke sistem, admin akan diminta untuk memasukan data kurir yang nantinya data tersebut akan digunakan oleh kurir untuk *login* ke sistem kurir. Setelah data kurir dimasukkan maka data kurir tersebut akan tampil ke dalam daftar kurir. Pada tabel daftar kurir admin dapat merubah ataupun menghapus data kurir yang diinginkan.



Gambar 3.14 Rancangan Halaman Tambah Kurir

3. Halaman Ubah Kurir

   Pada gambar 3.15 merupakan perancangan antarmuka halaman ubah kurir. Halaman ini berisi data kurir yang ingin diubah seperti *username* kurir ataupun *password* kurir.



Gambar 3.15 Rancangan Halaman Ubah Kurir

4. Halaman Tambah Pengiriman Barang

   Pada gambar 3.16 merupakan perancangan antarmuka halaman manajemen pengiriman barang. Halaman ini merupakan data masukan tentang pengiriman barang.

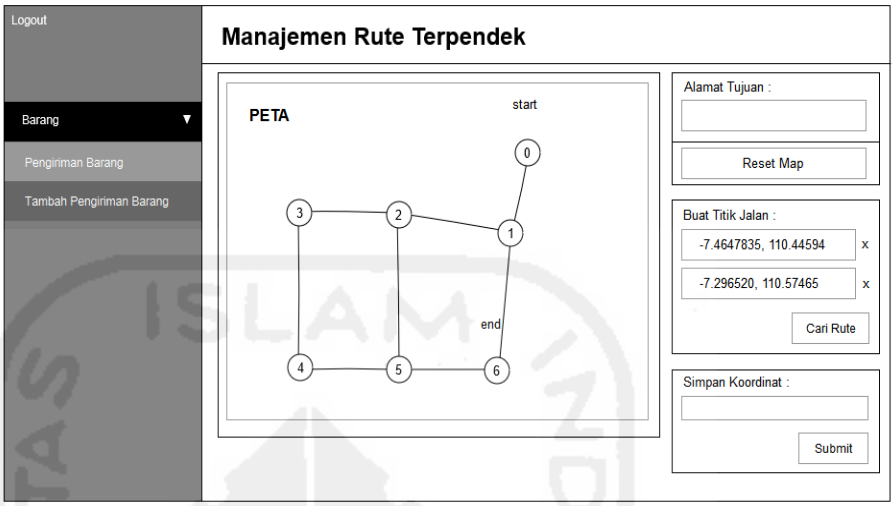


Gambar 3.16 Rancangan Halaman Tambah Pengiriman

5. Halaman Manajemen Rute Terpendek

   Pada gambar 3.17 merupakan perancangan antarmuka halaman manajemen rute terpendek. Setelah memasukan data pengiriman barang selanjutnya admin diminta untuk mencari beberapa alternatif rute ke lokasi tujuan pengiriman barang, lalu dengan menggunakan

algoritma Dijkstra maka akan dipilih rute terpendek dari beberapa alternatif rute yang telah dimasukan.



Gambar 3.17 Rancangan Halaman Manajemen Rute Terpendek

6. Halaman Daftar Pengiriman Barang

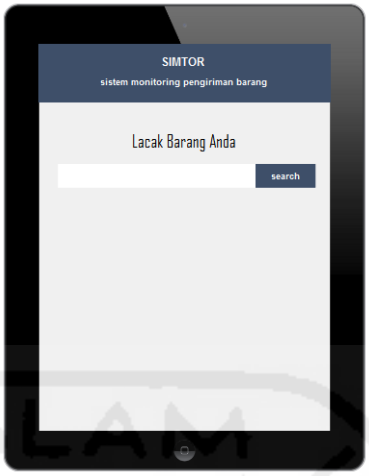
Pada gambar 3.18 merupakan perancangan antarmuka halaman daftar pengiriman barang. Halaman ini berisi tentang daftar pengiriman barang yang sebelumnya sudah dimasukkan oleh admin.



Gambar 3.18 Rancangan Halaman Daftar Pengantaran Barang

7. Halaman Home Konsumen

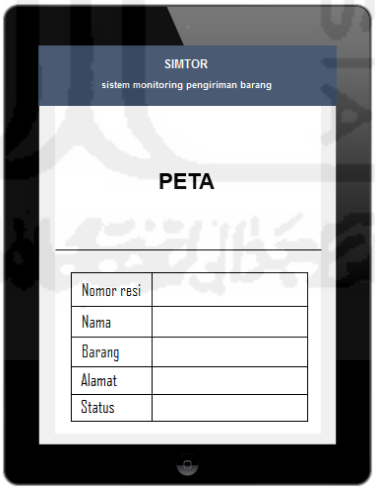
Pada gambar 3.19 merupakan perancangan antarmuka halaman sistem untuk konsumen agar dapat memantau barang yang dikirim, konsumen diminta untuk memasukan nomor pengiriman.



Gambar 3.19 Rancangan Halaman *Home* Konsumen

8. Halaman *Tracking* Barang

   Pada gambar 3.20 merupakan perancangan antarmuka halaman *tracking* dimana setelah nomor *tracking* atau kode barang yang dimasukan oleh konsumen sesuai maka konsumen telah dapat memantau barangnya.

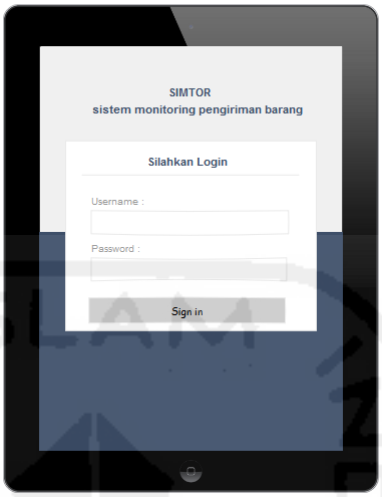


Gambar 3.20 Rancangan Halaman *Tracking* Barang

9. Halaman *Login* Kurir

   Pada gambar 3.21 merupakan perancangan antarmuka halaman *login* kurir. Sebelum kurir mengakses halaman kurir, kurir diminta untuk

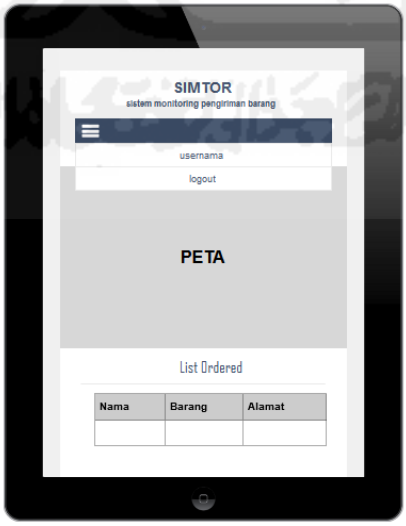
memasukan *username* dan *password* untuk dapat masuk ke sistem kurir.



Gambar 3.21 Rancangan Halaman *Login* Kurir

10. Halaman *Home* Kurir

Pada gambar 3.22 merupakan perancangan antarmuka halaman *home* kurir. Halaman home kurir berisi data pengiriman barang yang akan diantarkan oleh kurir tersebut. Kurir tersebut juga akan mendapatkan rute terpendek ke lokasi tujuan pengiriman barang.



Gambar 3.22 Rancangan Halaman *Home* Kurir

3.7. Perancangan Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan tahap penting yang akan menentukan kualitas dari sistem setelah melewati tahap pengembangan. Pengujian sistem juga memberikan pandangan mengenai sistem secara objektif dan independen, yang bermanfaat untuk mengetahui tingkat resiko pada saat implementasinya. Ada dua pendekatan pengujian untuk sistem ini, yaitu :

1. Pengujian Fungsional atau *Black Box Testing*

   Pengujian fungsional atau *black box testing* digunakan untuk mengetahui apakah sistem telah berfungsi dengan benar. Pengujian *black box* merupakan metode perancangan data uji yang didasarkan pada spesifikasi sistem. Data uji dibangkitkan, dieksekusi pada sistem, dan kemudian keluaran dari sistem akan dicek apakah telah sesuai dengan yang diharapkan. Tabel 3.10 merupakan contoh tabel untuk pengujian *black box* pada sistem ini.

   Tabel 3.10 Contoh Tabel Pengujian *Black Box*

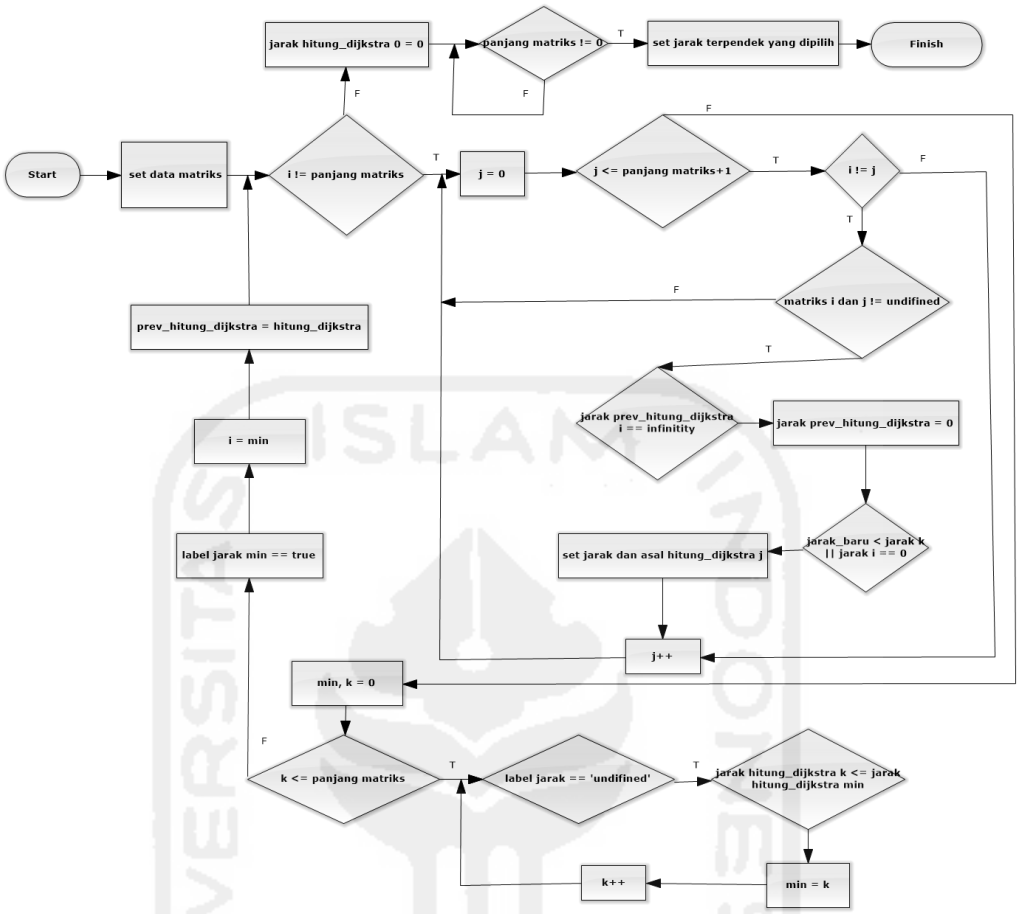
No	Skenario pengujian	Test case	Hasil yang diharapkan	Hasil pengujian	Kesimpulan

2. Pengujian Komputasi

   Pengujian komputasi digunakan untuk mengetahui seberapa besar sistem dapat menemukan cara untuk menyelesaikan permasalahan-permasalahan dari data masukan dengan menggunakan suatu algoritma.

3.8. *Flowchart*

Pada gambar 3.23 merupakan penjelasan alur algoritma Dijkstra dalam menentukan rute terpendek dari beberapa alternatif rute dari titik awal menuju lokasi tujuan.



Gambar 3.23 *Flowchart* Algoritma Dijkstra