



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN APLIKASI

3.1 Analisa Kebutuhan

3.1.1 Perangkat Lunak yang Digunakan

Pembuatan aplikasi ini menggunakan *software* atau perangkat lunak yaitu NetBeans IDE versi 7.3 yang berbasis *desktop*. Selain itu menggunakan Microsoft Office Access 2007 dalam membuat *database*, *plug-in Java Runtime Environment* (JRE), juga *library* pendukung aplikasi antara lain *commons-beanutils-1.8.0.jar*, *commons-collections-2.1.1.jar*, *commons-collection-3.1.jar*, *commons-digester-1.7.jar*, *commons-logging-1.1.jar*, dan *jasperreports-3.7.3.jar*. Di samping itu dalam pembuatan *database* diperlukan aplikasi yang dapat menghitung jarak tempuh, sehingga dibutuhkan *Google Maps* yang telah disediakan oleh *Google*.

3.1.2 Bahasa Pemrograman yang Digunakan

Konsep bahasa pemrograman yang digunakan dalam pembuatan aplikasi adalah konsep pemrograman berorientasi objek atau sering disebut dengan OOP (*Object Oriented Programming*), serta dengan bahasa pemrograman Java. Konsep pemrograman berorientasi objek dipilih karena dalam melakukan *maintenance*, program akan lebih mudah dibaca dan dipahami juga dapat mengontrol kerumitan program hanya dengan memperlihatkan rincian yang dibutuhkan *programmer*. Jadi, pemrograman berorientasi objek cocok digunakan dalam kasus pembuatan

software yang rumit dan kompleks karena memberikan kemudahan pada pemrogram.

3.2 Metodologi Penelitian

3.2.1 Studi Literatur

Tahap pertama dalam penelitian ini adalah dengan melakukan studi literatur menggunakan referensi buku, jurnal, dan artikel mengenai algoritma *Shortest Job First* dan *Round Robin*, serta berbagai sumber yang mendukung dalam pembuatan dan pengembangan aplikasi. Literatur tersebut digunakan sebagai pedoman dalam melakukan penelitian.

3.2.2 Pencarian dan Pendataan Kebutuhan

Pengumpulan data-data berupa laporan pengiriman barang oleh para kurir di lima wilayah yang sudah ditentukan sebelumnya. Selain itu meminta data-data tambahan seperti tanda bukti pengiriman barang. Data-data tersebut akan digunakan dalam penelitian nantinya.

3.2.3 Perancangan Desain Sistem dan Database

Setelah semua data yang dibutuhkan telah didapat, selanjutnya dilakukan perancangan dan pembuatan *database* untuk menampung semua data yang telah dikumpulkan. Kemudian dilakukan juga perancangan desain aplikasi yang nantinya akan digunakan untuk mempresentasikan data tersebut.

3.2.4 Pemrograman dan Penerapan Algoritma

Melakukan pemrograman aplikasi berbasis *desktop* menggunakan NetBeans IDE versi 7.3 dan implementasi algoritma *Shortest Job First* dan *Round Robin* yang digunakan dalam aplikasi penjadwalan ini.

3.2.5 Uji Coba Aplikasi

Aplikasi yang telah jadi dan diimplementasikan dengan Algoritma *Shortest Job First* dan *Round Robin* untuk sistem penjadwalannya, akan dicoba seberapa efisien penjadwalan yang dihasilkan oleh kedua algoritma ini. Pengujian ini dilakukan oleh Admin dari PT Royal Express dengan menggunakan aplikasi yang telah dibuat. Admin yang menjadi *user* akan meng-*input* beberapa transaksi pengiriman barang yang sudah ada sebelumnya. Kemudian transaksi akan tersimpan di tabel pengiriman dan setelah itu dapat ditampilkan laporan pengiriman berdasarkan urutan jarak dari implementasi algoritma *Shortest Job First* atau *Round Robin*. Laporan tersebut memberikan urutan pengiriman barang bagi para kurir agar jadwal pengiriman barang dapat menjadi lebih efisien dan cepat sampai ke tujuan.

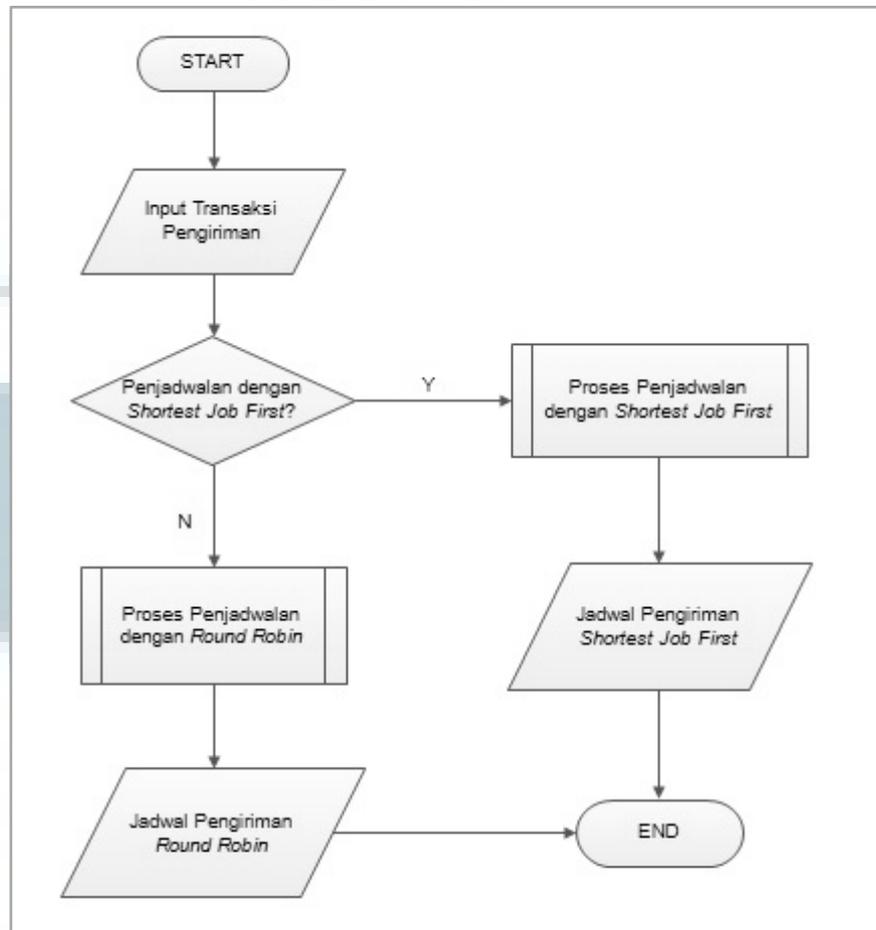
3.3 Perancangan Sistem

Perancangan sistem sebagai gambaran proses-proses utama yang ada pada aplikasi.

3.3.1 Flowchart

Flowchart atau bagan alir adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir (*flowchart*) digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Bagan alir sistem akan dipecah menjadi tiga bagian, yaitu bagan alir untuk sistem umum, bagan alir sistem penjadwalan dengan menggunakan algoritma *Shortest Job First*, dan bagan alir sistem penjadwalan dengan menggunakan algoritma *Round Robin*.

1. Bagan alir sistem



Gambar 3.1 *Flowchart* Sistem

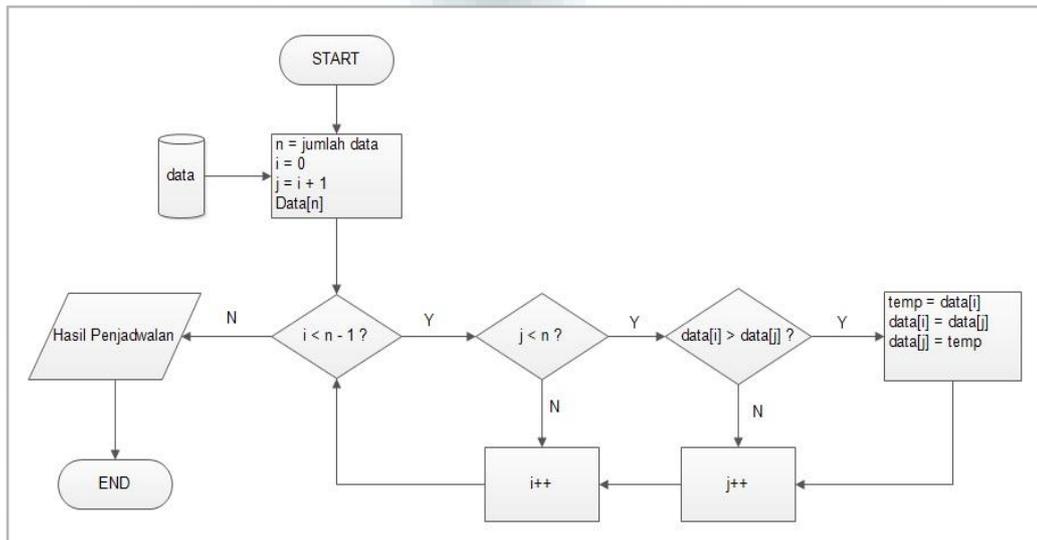
Ketika pengguna menjalankan aplikasi, sistem akan menampilkan enam buah tombol atau pilihan sebagai menu utama. Menu pertama adalah Transaksi, lalu *History*, *Report*, *Master*, *Help*, dan yang terakhir adalah *About*. Saat pengguna melakukan Transaksi maka akan muncul *form* untuk melakukan pengisian transaksi pengiriman barang yang baru dan kemudian akan disimpan dalam tabel pengiriman di *database*. Pada *form* Transaksi terdapat kode transaksi pengiriman, kode barang, kode pengirim, kode penerima, kode daerah, kode kurir,

kota asal pengiriman dan tanggal. Pengguna hanya perlu untuk mengisi kode transaksi dan kota asal pengiriman karena sudah disediakan *combo box* untuk mengisi kode-kode yang telah terhubung di *database*. Apabila terdapat kode yang belum ada, maka tersedia tombol untuk memasukkan data baru ke *database* yaitu New Barang, New Penerima, New Pengirim, New Daerah, dan New Kurir. Setelah itu klik tombol Submit, maka transaksi pun akan tersimpan.

Menu *History* berguna untuk menampilkan transaksi pengiriman yang ada pada tabel pengiriman. Pada menu *Report*, *user* dapat menampilkan laporan pengiriman berdasarkan urutan jarak dengan menggunakan algoritma *Shortest Job First* atau *Round Robin*. Pada menu *Master*, pengguna dapat melihat detail dari isi tabel pada *database* yang terdiri dari Master Barang, Master Daerah, Master Pengirim, Master Penerima dan Master Kurir. Selain itu, pengguna juga dapat melakukan *edit* atau *update*, serta hapus atau *delete* data. Untuk menu *Help* pengguna dapat melihat bantuan jika diperlukan. Menu *About* berisi versi dari *software*, *library* yang digunakan, *updates*, dan sistem operasi yang digunakan dalam pembuatan aplikasi.



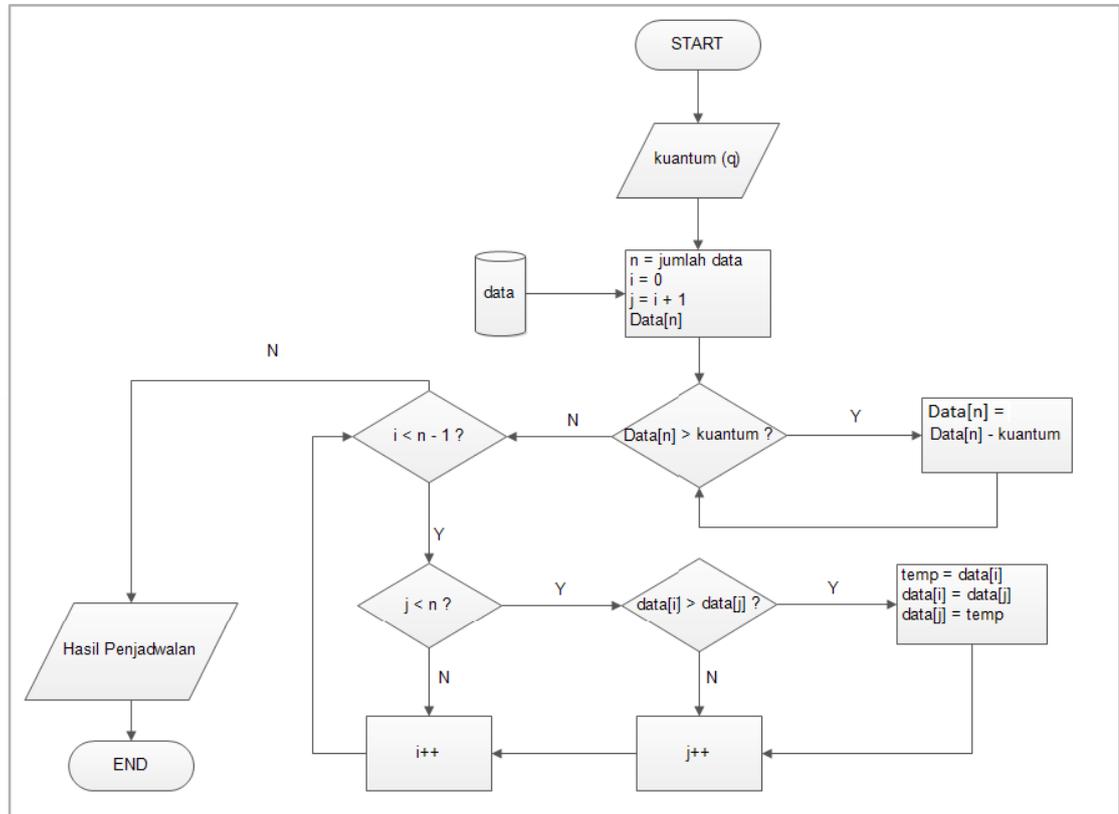
2. Bagan alir sistem penjadwalan dengan menggunakan algoritma *Shortest Job First*



Gambar 3.2 Flowchart Sistem Penjadwalan dengan Algoritma *Shortest Job First*

Saat pengguna memilih menu *Report* dan memilih *Shortest Job First* (SJF) maka pertama sistem akan mengambil data dari *database*, yaitu jarak pada tabel daerah. Kemudian dibuat *variable* *i* dan *j* untuk membuat *looping* pengurutan data. Dibuat *variable* tampung (*temp*) untuk menyimpan data yang akan dipindahkan untuk diurutkan. Setelah mendapatkan nilai yang sesuai keinginan maka dilanjutkan ke proses *output* untuk mencetak hasil penjadwalan.

3. Bagan alir sistem penjadwalan dengan menggunakan algoritma *Round Robin*



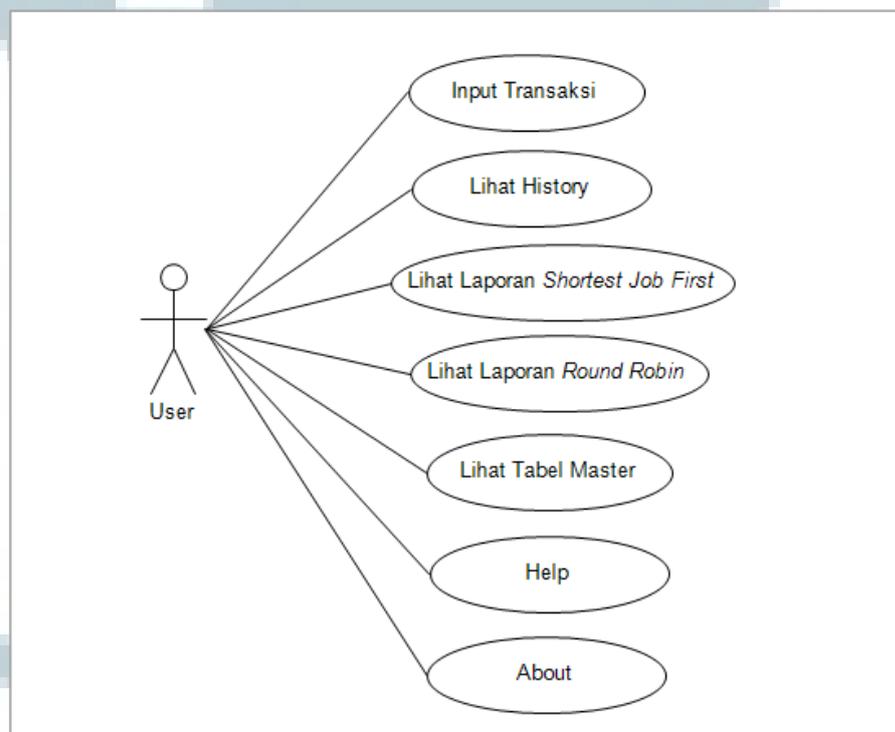
Gambar 3.3 *Flowchart* Sistem Penjadwalan dengan Algoritma *Round Robin*

Ketika pengguna memilih menu *Report* dan memilih *Round Robin*, maka pertama sistem akan meminta *user* untuk melakukan *input* kuantum atau *slice time*. Setelah itu sistem akan mengambil data jarak dari *database* yang kemudian akan dibandingkan dengan besar kuantum. Apabila jarak lebih besar daripada kuantum yang dimasukkan, maka jarak – kuantum. Hal tersebut dilakukan terus-menerus (*looping*) sampai jarak lebih kecil dari kuantum. Kemudian dibuat *variable* tampung yang berupa *ArrayList* untuk menampung data yang telah dikurangi oleh kuantum sampai tidak bisa dikurangi lagi. Kemudian jika data yang sudah dikurangi sudah lebih kecil atau sama dengan nol, maka data (yang

asli) ke n akan ditampung ke *ArrayList* baru. Setelah itu data akan dimunculkan sesuai pada isi *ArrayList* baru.

3.3.2 Use Case Diagram

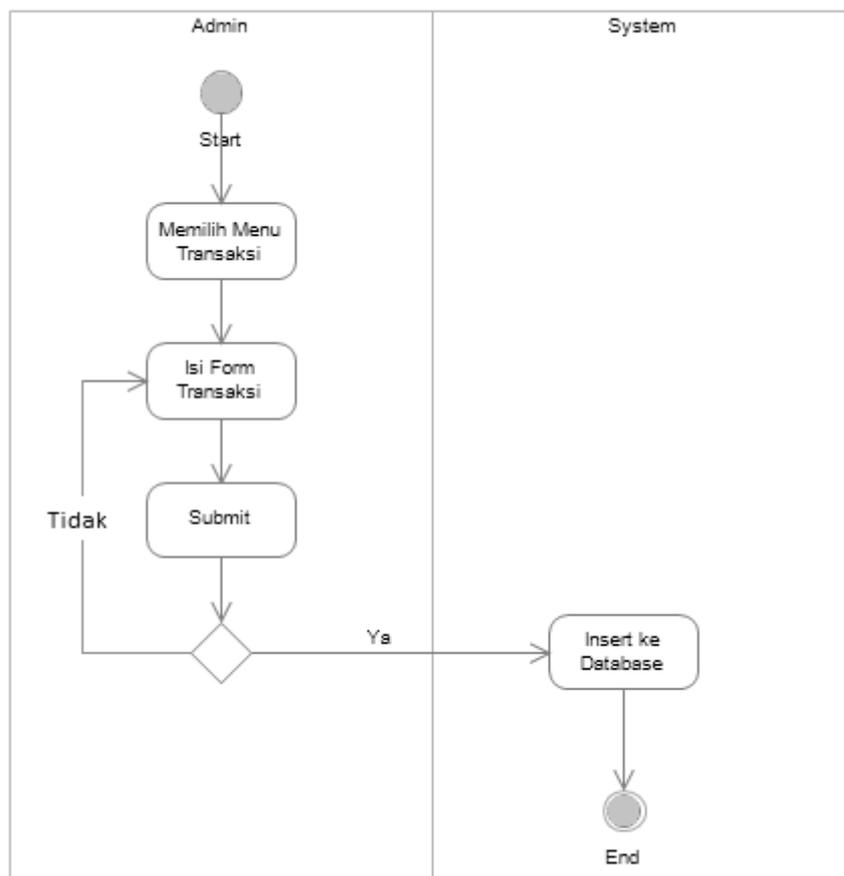
Aplikasi sistem penjadwalan dapat diakses oleh *user* melalui *Personal Computer* (PC) atau *Notebook*. Aktivitas yang dapat dilakukan oleh *user*, yaitu memilih menu *input* transaksi, lihat *history* pengiriman, lihat laporan yang dibagi menjadi dua yaitu secara *Shortest Job First* atau *Round Robin*, dan terdapat pula tombol cetak atau *print*. Lalu menu master untuk melihat dan meng-*edit* isi dari tabel yang ada di *database*, menu *Help* untuk melihat bantuan dan menu *About* untuk melihat versi *software* dan sistem operasi yang digunakan dalam membuat aplikasi.



Gambar 3.4 *Use Case* Sistem

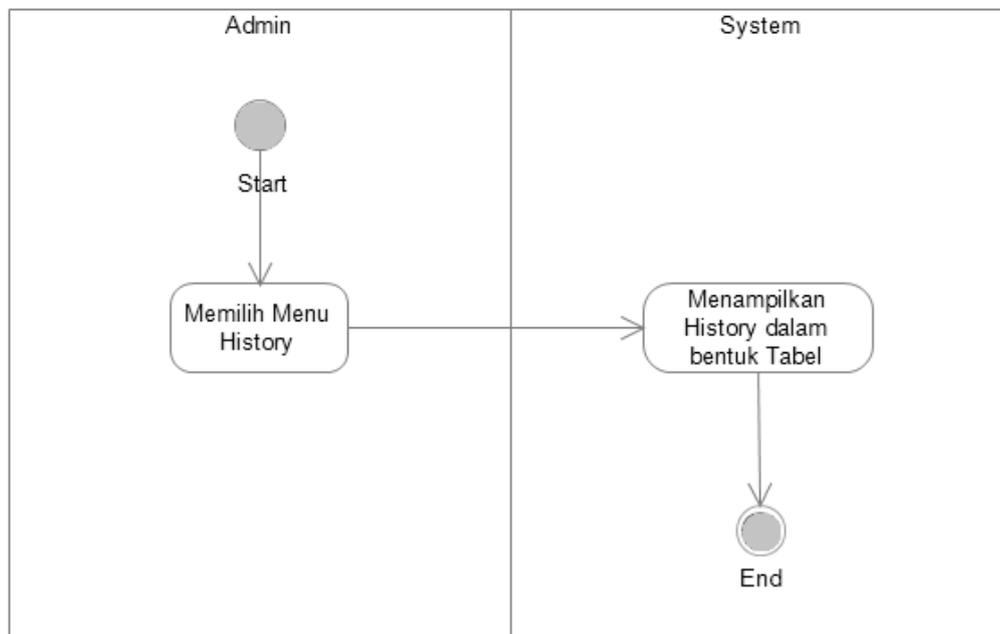
3.3.3 Activity Diagram

Aplikasi ini membutuhkan data yang berasal dari *input-an user*. Berikut adalah *activity diagram* untuk alur data yang terjadi dalam aplikasi saat menu transaksi dipilih.



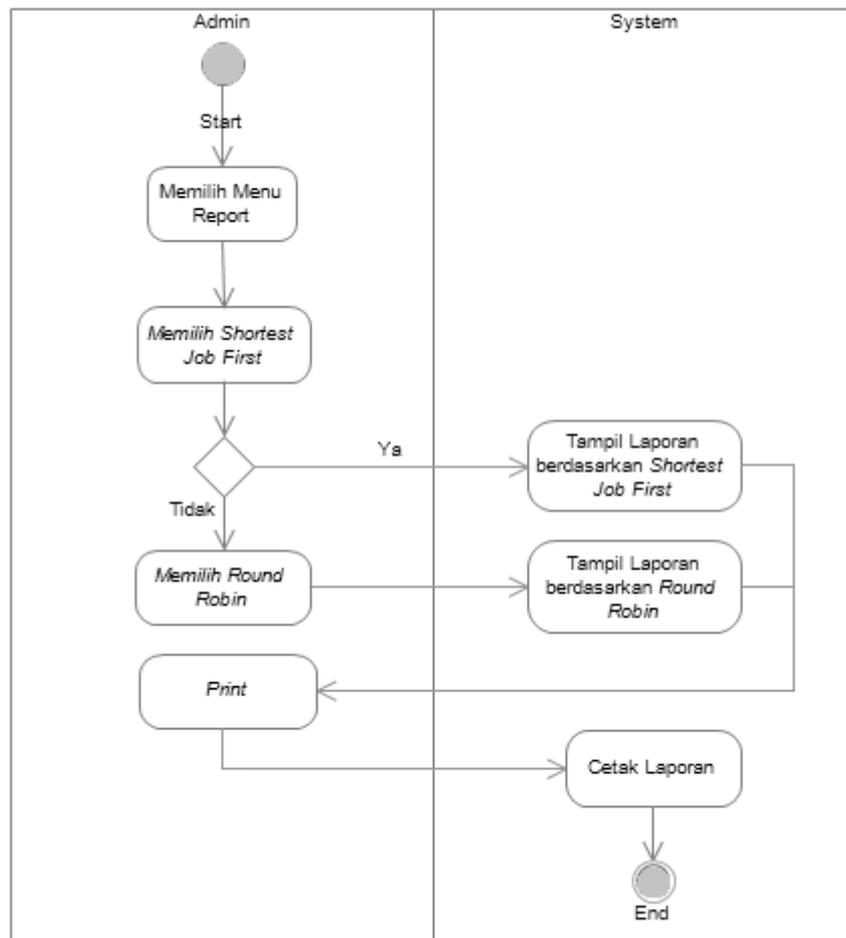
Gambar 3.5 *Activity Diagram* Menu Transaksi

Aplikasi ini juga terdapat menu *History* yang berguna untuk menampilkan histori dari pengiriman barang terbaru. Dengan demikian *activity diagram* untuk menu *History* dipaparkan sebagai berikut.



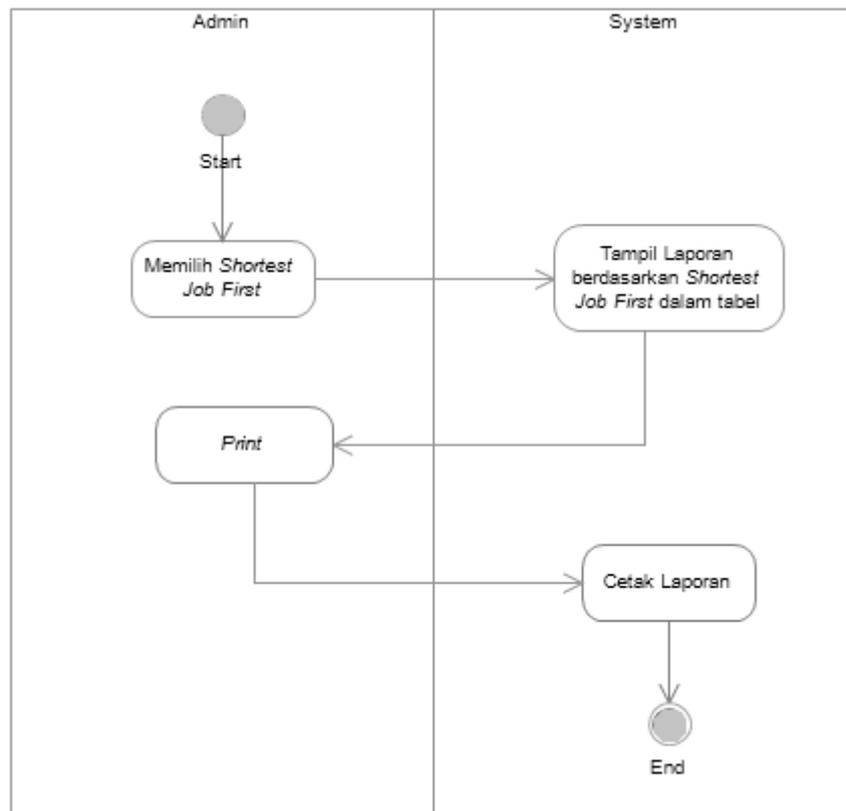
Gambar 3.6 Activity Diagram Menu History

Menu berikutnya dalam aplikasi adalah *Report*. Menu ini meminta *input user* untuk memilih *Shortest Job First* atau *Round Robin*. Setelah *user* memilih dari dua pilihan algoritma, maka akan ditampilkan laporan pengiriman berdasarkan algoritma yang telah dipilih. Lalu apabila *user* ingin mencetak laporan, maka ada tombol *Print* yang telah disediakan pada aplikasi. Gambar 3.7 merupakan *activity diagram* dari menu *Report*.



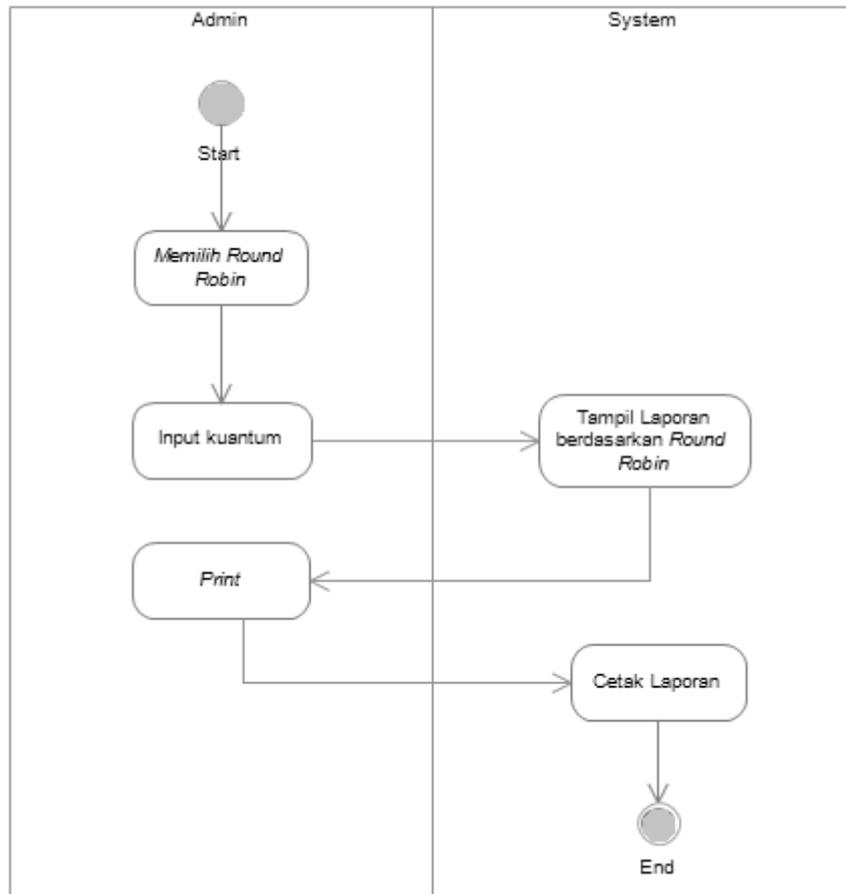
Gambar 3.7 *Activity Diagram Menu Report*

Gambar 3.8 menunjukkan *activity diagram* dari menu *Shortest Job First* serta gambar 3.9 merupakan *activity diagram* dari menu *Round Robin*.



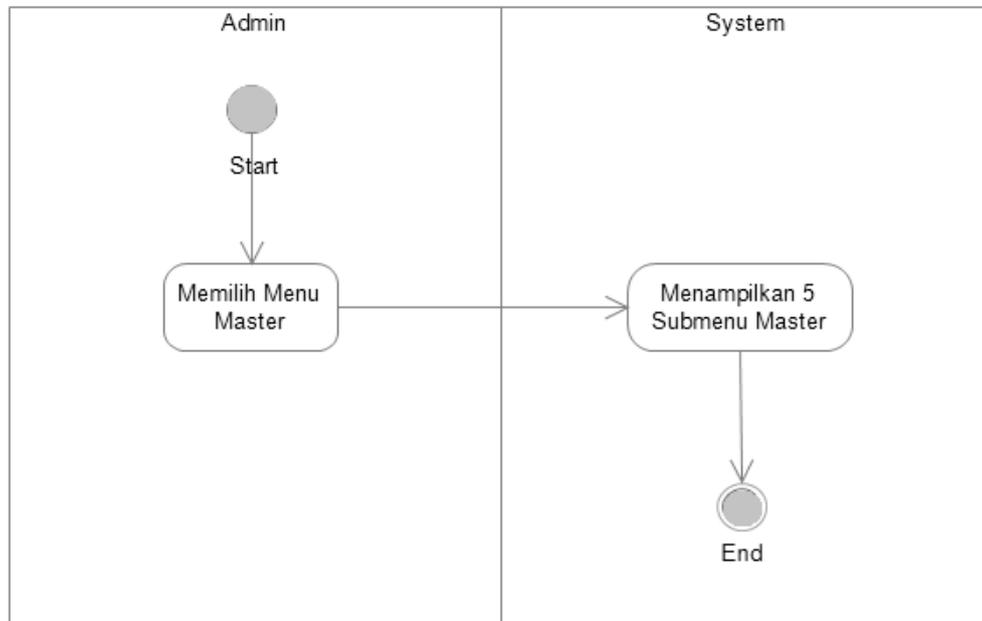
Gambar 3.8 Activity Diagram Menu Shortest Job First

UMMN



Gambar 3.9 Activity Diagram Menu Round Robin

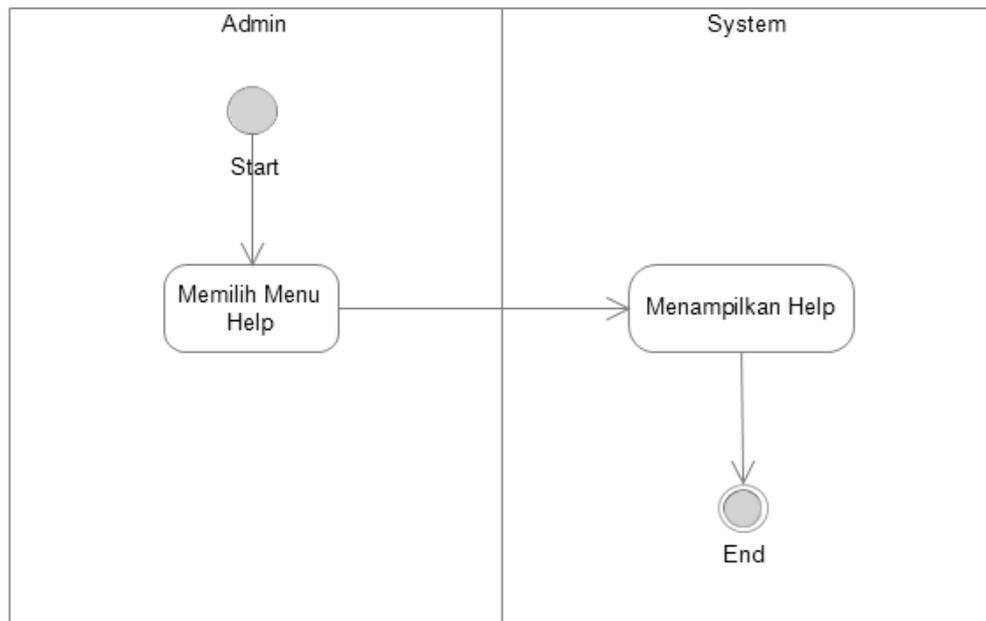
Selanjutnya menu Master yang akan menampilkan lima buah submenu yaitu Master Barang, Master Pengirim, Master Penerima, Master Daerah, dan Master Kurir. Menu ini memiliki fungsi bagi *user* untuk mengetahui detail dari suatu kode. Selain itu juga terdapat tombol 'EDIT' untuk melakukan *update* atau *delete*. Gambar 3.10 merupakan *activity diagram* dari menu Master.



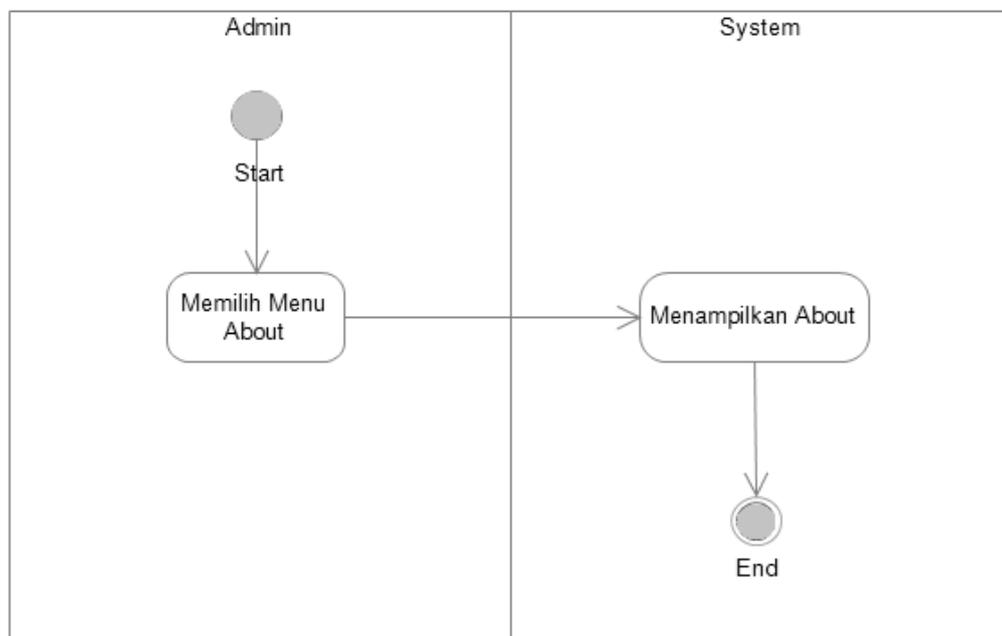
Gambar 3.10 *Activity Diagram Menu Master*

Dua menu terakhir adalah menu *Help* dan *About*. Pada menu *Help*, user dapat melihat sedikit informasi dan bantuan dalam menggunakan aplikasi. Sedangkan untuk menu *About* berisi versi *software* yang digunakan, serta sistem operasi yang digunakan dalam pembuatan aplikasi. Gambar 3.11 dan 3.12 merupakan *activity diagram* dari menu *Help* dan *About*.

U
M
M
N



Gambar 3.11 *Activity Diagram Menu Help*

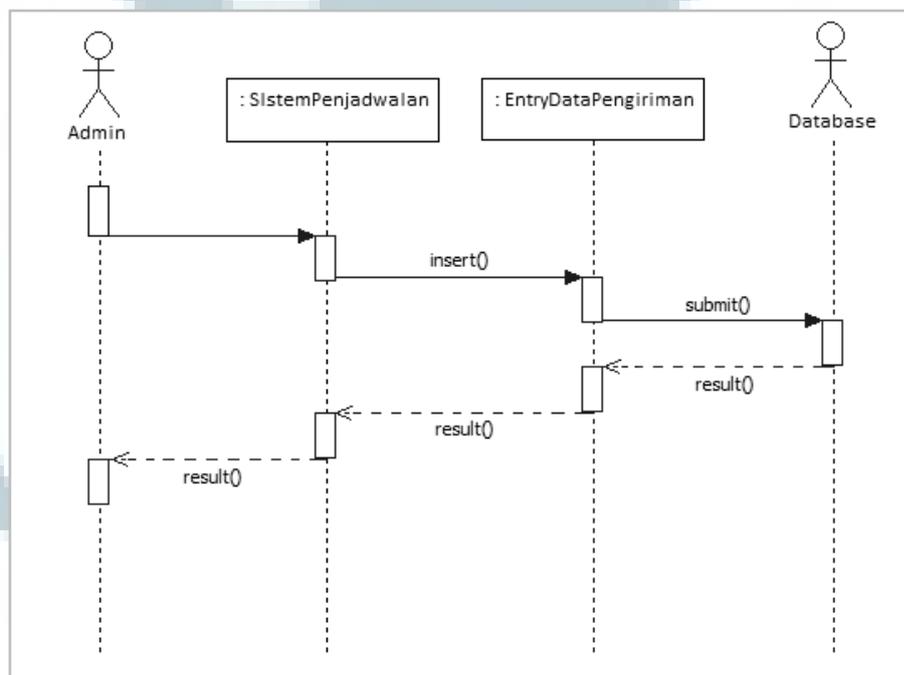


Gambar 3.12 *Activity Diagram Menu About*

3.3.4 Sequence Diagram

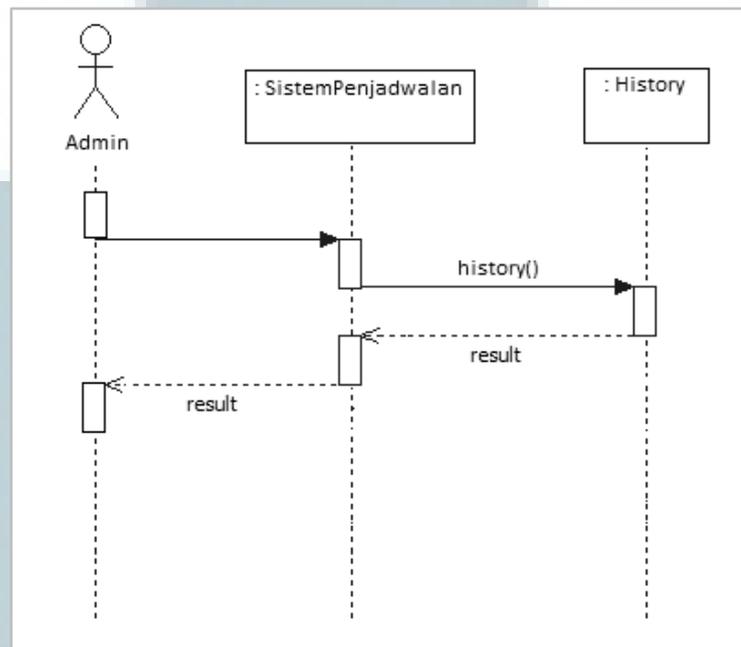
Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

Pada saat *user* menjalankan aplikasi maka *class* SistemPenjadwalan akan dijalankan. Selanjutnya pada saat menu Transaksi dipilih maka *class* SistemPenjadwalan akan *men-trigger method insert()*. Kemudian *class* EntryDataPengiriman akan *men-trigger method submit* yang berfungsi dalam melakukan *insert* data ke dalam *database*. *Sequence diagram* saat menu Transaksi dipilih dapat dilihat pada gambar 3.13.



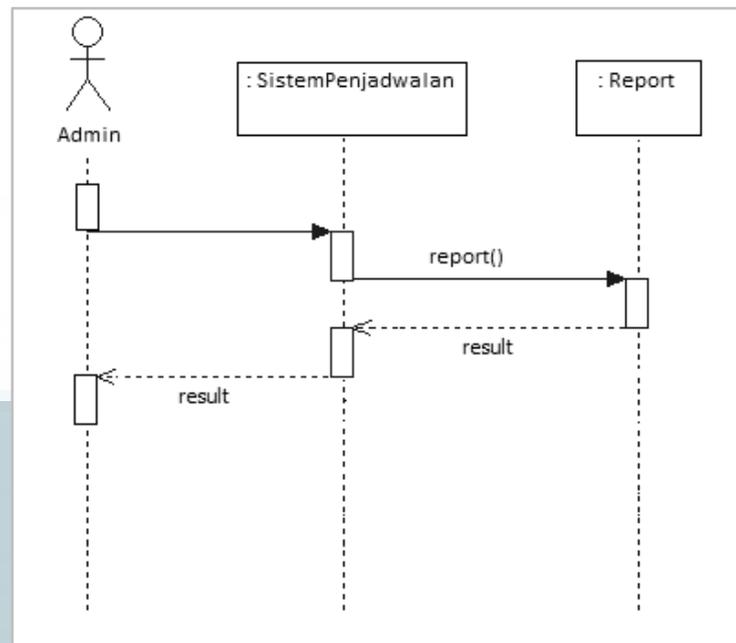
Gambar 3.13 *Sequence Diagram* Transaksi

Setelah itu pada saat *user* memilih menu *History*, maka *class* *SistemPenjadwalan* akan *men-trigger method history()*. Gambar 3.14 menunjukkan *sequence diagram* ketika menu *History* dipilih.

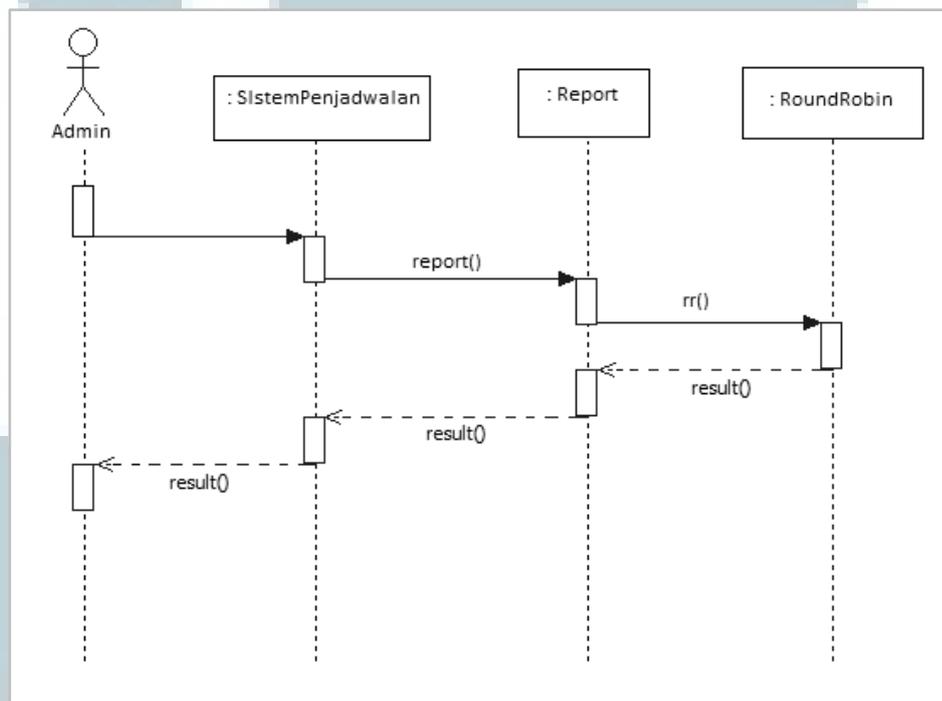


Gambar 3.14 *Sequence Diagram History*

Selanjutnya pada saat menu *Report* dipilih, maka *class* *SistemPenjadwalan* akan *men-trigger method report()*. Gambar 3.15 merupakan *sequence diagram* dari menu *Report*. Kemudian jika *user* memilih untuk menampilkan *report* dengan menggunakan algoritma *Round Robin*, maka *class* *Report* akan *men-trigger method rr()*. Gambar 3.16 merupakan *sequence diagram* menu *Report* berdasarkan algoritma *Round Robin*.

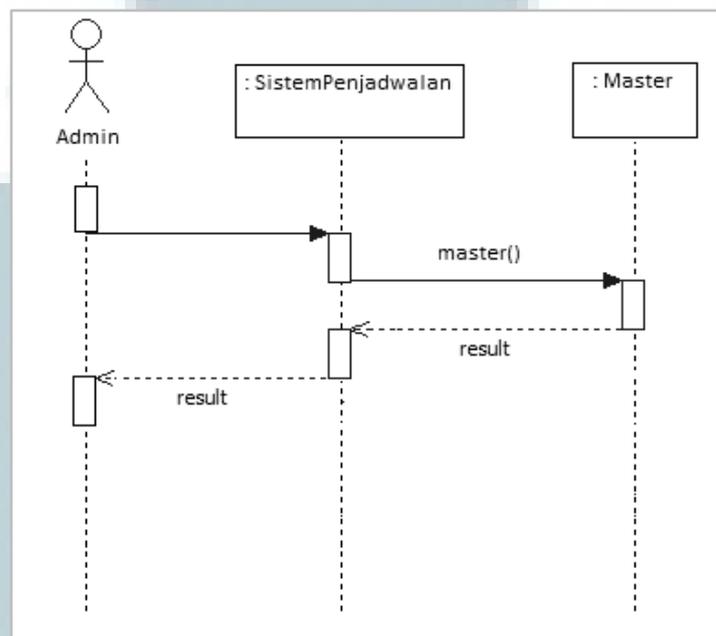


Gambar 3.15 *Sequence Diagram Report*



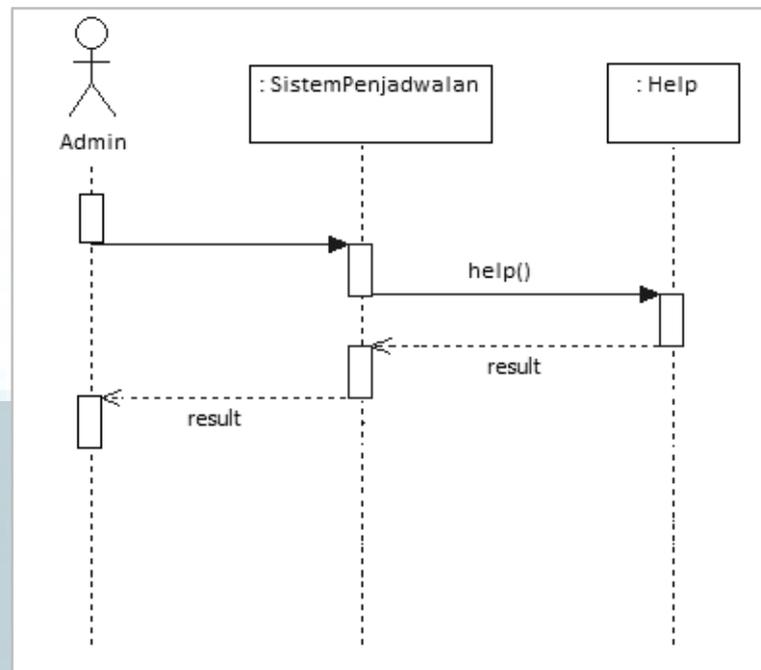
Gambar 3.16 *Sequence Diagram Report Round Robin*

Pada menu selanjutnya terdapat menu *Master*. Pada saat menu *Master* dipilih, maka *class* *SistemPenjadwalan* akan *men-trigger method master()*. *Sequence diagram* dari menu *Master* dapat dilihat pada gambar 3.17.

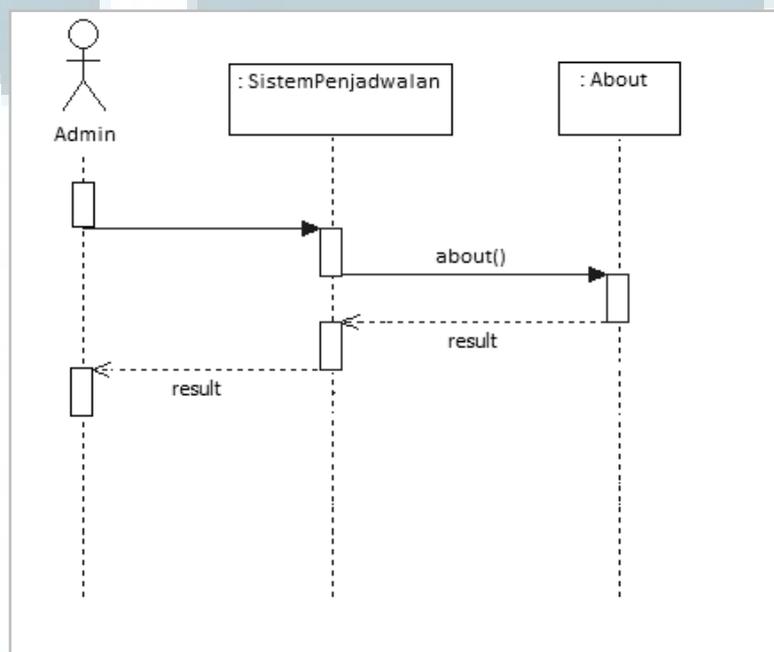


Gambar 3.17 *Sequence Diagram Master*

Kemudian menu selanjutnya adalah menu *Help* dan menu *About*. *Class* *SistemPenjadwalan* akan *men-trigger method help()* jika menu *Help* dipilih, dan *method about()* jika menu *About* dipilih. Gambar 3.18 dan 3.19 merupakan *sequence diagram* dari menu *Help* dan *About*.



Gambar 3.18 *Sequence Diagram Help*

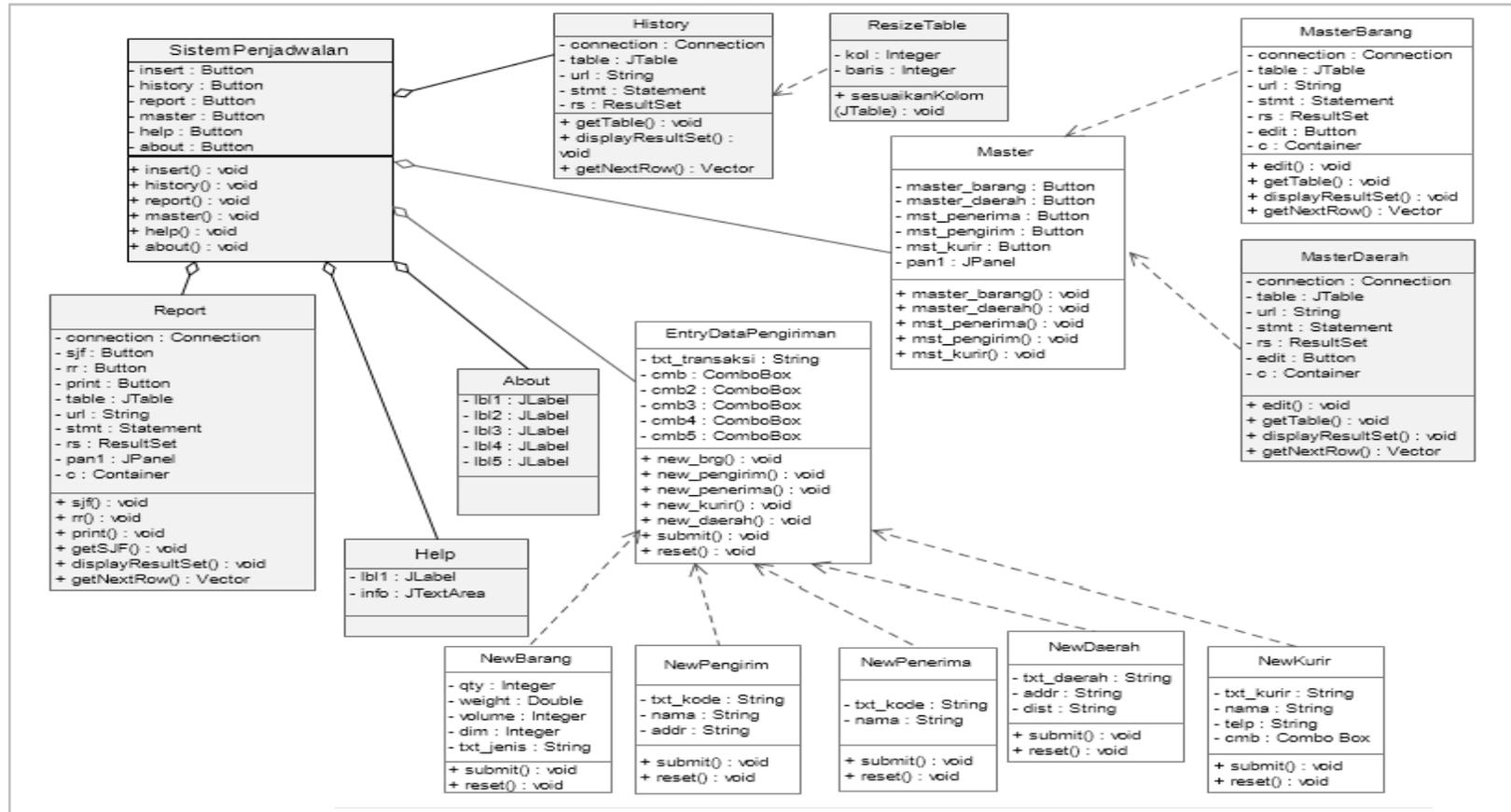


Gambar 3.19 *Sequence Diagram About*

3.3.5 Class Diagram

Prosedur kerja aplikasi secara umum dapat dijelaskan dengan menggunakan diagram. Proses bermula dari pemilihan menu utama oleh *user*. Ketika *user* memilih menu Transaksi, maka akan muncul *window* baru dengan bentuk *form* baru yang harus di-*input* oleh *user*. Setelah *form* diisi, maka data akan tersimpan dalam *database*. Saat memilih menu *Report*, *user* dapat memilih *Shortest Job First* atau *Round Robin* untuk menampilkan laporan berdasarkan algoritma tersebut. Setelah itu *user* juga dapat melakukan cetak laporan. Berikut ini adalah representasi dari *class diagram* aplikasi sistem penjadwalan pengiriman barang.

U
M
M
N



Gambar 3.20 Class Diagram Sistem

3.3.6 Perancangan Database

Database yang digunakan adalah Microsoft Office Access 2007. Berikut struktur tabel-tabel yang terdapat di dalamnya.

1. Nama tabel : barang
Fungsi : menyimpan elemen-elemen dari barang yang akan dikirim.

Tabel 3.1 Barang

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_barang	Autonumber	Integer	<i>Primary key, unique</i>
2	jumlah	Number	Integer	
3	berat	Number	Double	
4	volume	Number	Integer	
5	dimensi	Number	Integer	
6	jenis	Varchar	20	Jenis barang (Paket/Dokumen)

2. Nama tabel : daerah
Fungsi : menyimpan wilayah, alamat tujuan pengiriman serta jarak dari perusahaan.

Tabel 3.2 Daerah

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_daerah	Varchar	5	<i>Primary key, unique</i>
2	wilayah	Varchar	15	Wilayah tujuan
3	alamat	Varchar	40	Alamat tujuan
4	jarak	Number	Double	Jarak perusahaan sampai tujuan

3. Nama tabel : pengirim

Fungsi : menyimpan data dari pengirim barang.

Tabel 3.3 Pengirim

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_pengirim	Varchar	12	<i>Primary key, unique</i>
2	nm_pengirim	Varchar	20	Nama pengirim
3	alamat_pengirim	Varchar	35	
4	kota	Varchar	15	Kota pengirim

4. Nama tabel : penerima

Fungsi : menyimpan data dari penerima barang.

Tabel 3.4 Penerima

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_penerima	Varchar	6	<i>Primary key, unique</i>
2	nm_penerima	Varchar	20	Nama penerima

5. Nama tabel : kurir

Fungsi : menyimpan data kurir.

Tabel 3.5 Kurir

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_kurir	Varchar	3	<i>Primary key, unique</i>
2	nm_kurir	Varchar	10	Nama kurir
3	no_telp	Varchar	12	Nomor telepon kurir
4	kendaraan	Varchar	5	Kendaraan kurir (Motor/Mobil)

6. Nama tabel : pengiriman

Fungsi : menyimpan data transaksi pengiriman yang menghubungkan data dari tabel-tabel sebelumnya.

Tabel 3.6 Pengiriman

No	Nama Field	Tipe Data	Panjang	Keterangan
1	kd_pengiriman	Varchar	25	<i>Primary key, unique</i>
2	kd_barang	Number	Integer	<i>Foreign key</i>
3	kd_pengirim	Varchar	12	<i>Foreign key</i>
4	kd_penerima	Varchar	6	<i>Foreign key</i>
5	kd_daerah	Varchar	5	<i>Foreign key</i>
6	kd_kurir	Varchar	3	<i>Foreign key</i>
7	tanggal	Date/Time	Short Date	
8	kota_asal	Varchar	12	

7. Nama tabel : rr

Fungsi : menampung data yang akan ditampilkan pada laporan dengan urutan algoritma *Round Robin*.

Tabel 3.7 Rr

No	Nama Field	Tipe Data	Panjang	Keterangan
1	ID	Number	Integer	Variabel untuk mengurutkan data yang akan ditampilkan di laporan
2	kd_pengiriman	Varchar	25	
3	berat	Number	Double	

4	jumlah	Number	Integer	
5	kota_asal	Varchar	12	
6	alamat	Varchar	40	
7	tanggal	Date/Time	Short Date	
8	nm_penerima	Varchar	20	
9	jarak	Number	Double	

3.3.7 Perancangan Antarmuka

a. Menu utama

Tampilan menu utama memuat enam buah tombol navigasi untuk ke tampilan lainnya atau menu selanjutnya yang terdiri dari “Transaksi”, “History”, “Report”, “Master”, “Help”, dan “About”. Gambar 3.21 merupakan rancangan tampilan dari menu utama.

Transaksi	History	Report	Master	Help	About
-----------	---------	--------	--------	------	-------

Gambar 3.21 Rancangan Antarmuka Menu Utama

b. Menu Transaksi

Menu transaksi merupakan *window* baru yang berbentuk *form* pengisian untuk transaksi pengiriman baru. Pada *form* ini terdapat dua buah *text field* yang digunakan untuk menerima *input* dari *user*. Selain itu terdapat lima buah *combo box* yang telah terhubung dari *database* dan memunculkan kode, sehingga *user* hanya tinggal memilih kode mana yang akan dimasukkan. Kemudian terdapat *text field* untuk

tanggal hari ini yang sudah otomatis tercantum. Setelah *user* selesai meng-*input*, maka terdapat dua tombol untuk melakukan ‘Submit’ atau ‘Reset’. Tombol ‘Submit’ untuk meng-*insert* data transaksi ke dalam *database*, sedangkan tombol ‘Reset’ untuk menghapus *input* dari *user* sebelumnya jika terdapat kesalahan. Selain itu juga terdapat lima buah tombol untuk melakukan *insert* data baru ke dalam lima tabel Master jika belum terdaftar, antara lain ‘New Barang’, ‘New Penerima’, ‘New Pengirim’, ‘New Daerah’, dan ‘New Kurir’. Gambar 3.22 merupakan rancangan tampilan dari menu transaksi.

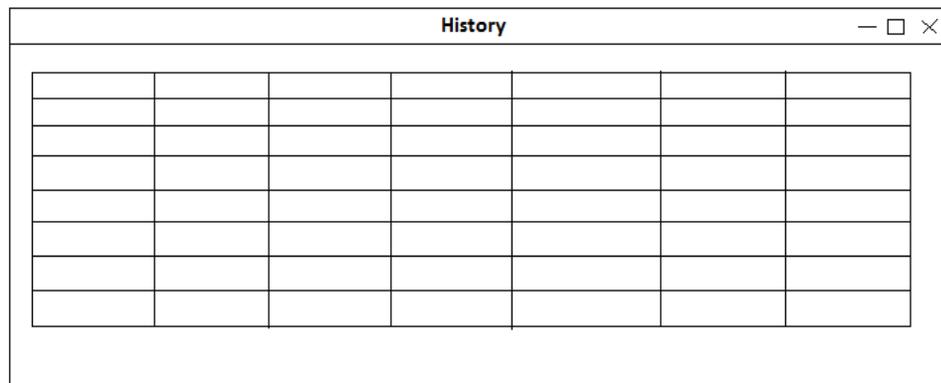
The image shows a window titled "New Transaction" with a standard window control bar (minimize, maximize, close). The window contains the following elements:

- Kode Pengiriman**: A text input field.
- Kode Barang**: A dropdown menu.
- Kode Pengirim**: A dropdown menu.
- Kode Penerima**: A dropdown menu.
- Kode Daerah**: A dropdown menu.
- Kode Kurir**: A dropdown menu.
- Tanggal**: A text input field with a placeholder "mm/dd/yyyy".
- Kota Asal**: A text input field.
- Five buttons: "New Barang", "New Penerima", "New Pengirim", "New Daerah", and "New Kurir", arranged horizontally.
- Two buttons: "RESET" and "SUBMIT", arranged horizontally below the previous buttons.

Gambar 3.22 Rancangan Antarmuka Menu Transaksi

c. Menu History

Menu *history* akan menampilkan *window* baru yang berisi *history* dari pengiriman barang terbaru dengan bentuk tabel. Gambar 3.23 merupakan rancangan tampilan dari menu *history*.



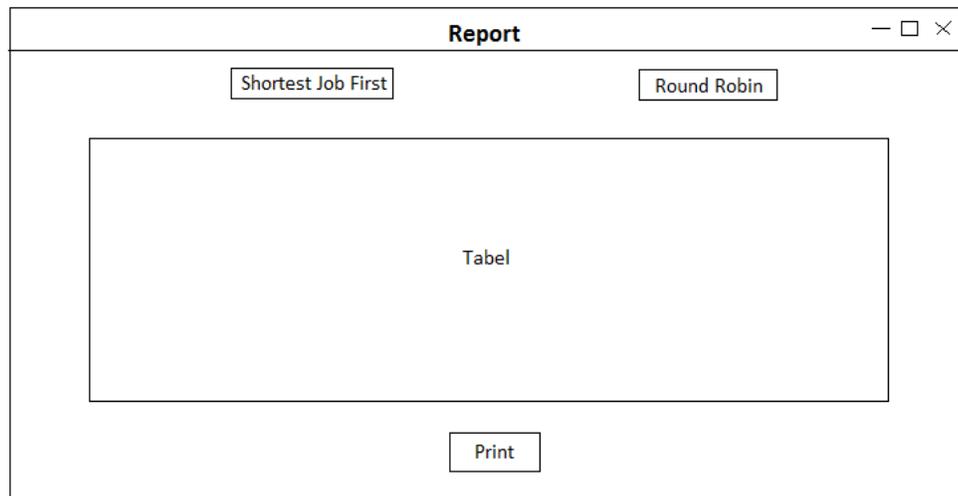
History						

Gambar 3.23 Rancangan Antarmuka Menu *History*

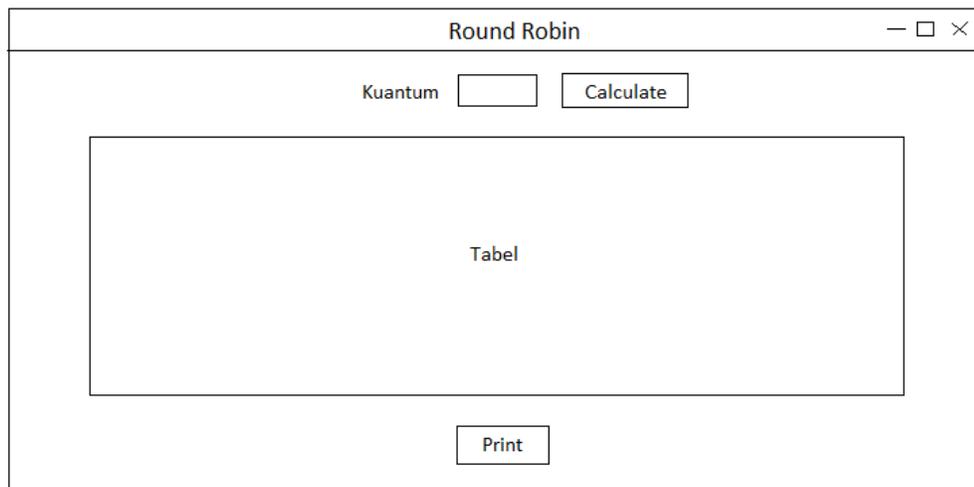
d. Menu Report

Menu *report* akan menampilkan *window* baru yang berisi laporan pengiriman barang berdasarkan algoritma *Shortest Job First* atau *Round Robin*. Kemudian ada tombol *Print* untuk mencetak laporan yang diinginkan. Gambar 3.24 merupakan rancangan tampilan dari menu *report* berdasarkan algoritma *Shortest Job First* dan gambar 3.25 ketika *user* menekan tombol *Round Robin*.

UMMN



Gambar 3.24 Rancangan Antarmuka Menu *Report*

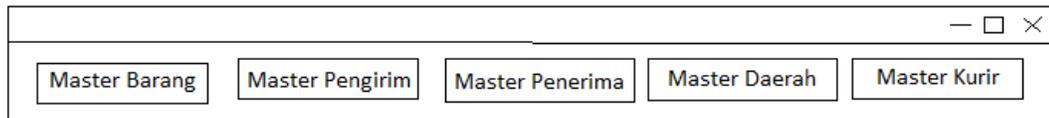


Gambar 3.25 Rancangan Antarmuka Menu *Round Robin*

e. Menu *Master*

Menu *Master* menyediakan submenu untuk lima tabel yang berada dalam *database*. Ketika *user* memilih salah satu dari tabel, maka akan muncul *detail* dari

tabel. Selain itu *user* juga dapat untuk melakukan *update* serta *delete* data yang ada pada tabel. Gambar 3.26 merupakan rancangan submenu dari menu *Master*.



Gambar 3.26 Rancangan Antarmuka Submenu Menu *Master*

Ketika tombol *Master Barang* dipilih, maka akan muncul *window* baru seperti berikut.



Gambar 3.27 Rancangan Antarmuka Menu *Master Barang*

Tampilan di atas sama untuk master tabel lainnya, dengan isi tabel sesuai dengan master yang dipilih. Pada saat tombol 'EDIT' dipilih, maka akan keluar tampilan seperti di bawah.

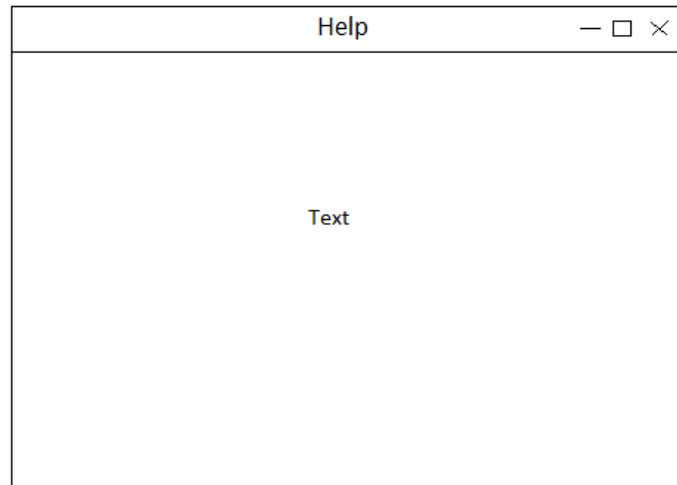
The image shows a window titled "Edit Barang" with standard window controls (minimize, maximize, close). Inside the window, there are six input fields arranged vertically, each with a label to its left: "Kode Barang" (with a dropdown arrow), "Jumlah", "Berat", "Volume", "Dimensi", and "Jenis". At the bottom of the window, there are two buttons: "Hapus" and "APPLY".

Gambar 3.28 Rancangan Antarmuka Edit Barang

Tampilan di atas akan memunculkan atribut apa saja yang terdapat dalam tabel barang. Hal tersebut sama dengan tampilan *edit* tabel lainnya yang akan memunculkan atribut sesuai tabel yang dipilih nanti. Pada kode terdapat *combo box*, sehingga *user* hanya perlu memilih kode dan atribut akan muncul pada *text field* yang telah disediakan. Tombol 'Hapus' berguna untuk menghapus *record* data dari *database*, sedangkan tombol 'APPLY' berguna untuk menyimpan data yang sudah di-*edit* atau *update*.

f. Menu *Help*

Menu ini berfungsi untuk memberikan bantuan kepada pengguna dalam menjalankan aplikasi jika mengalami kesulitan. Gambar 3.29 merupakan rancangan tampilan dari menu *Help*.

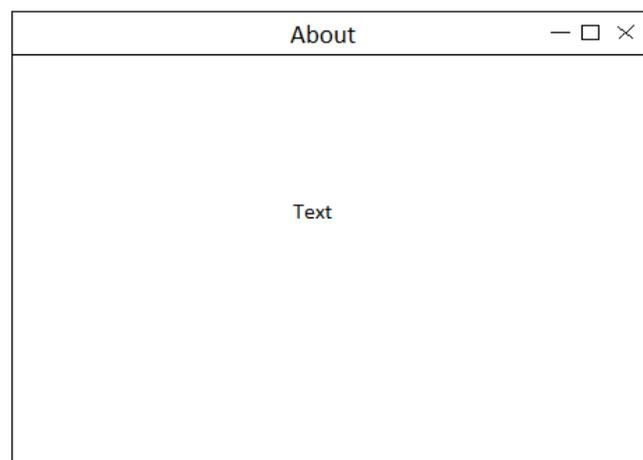


Gambar 3.29 Rancangan Antarmuka Menu *Help*

g. Menu *About*

Menu ini akan menampilkan versi dari *software* yang digunakan dalam pembuatan aplikasi, *update* dari *software*, dan sistem operasi apa yang dipakai.

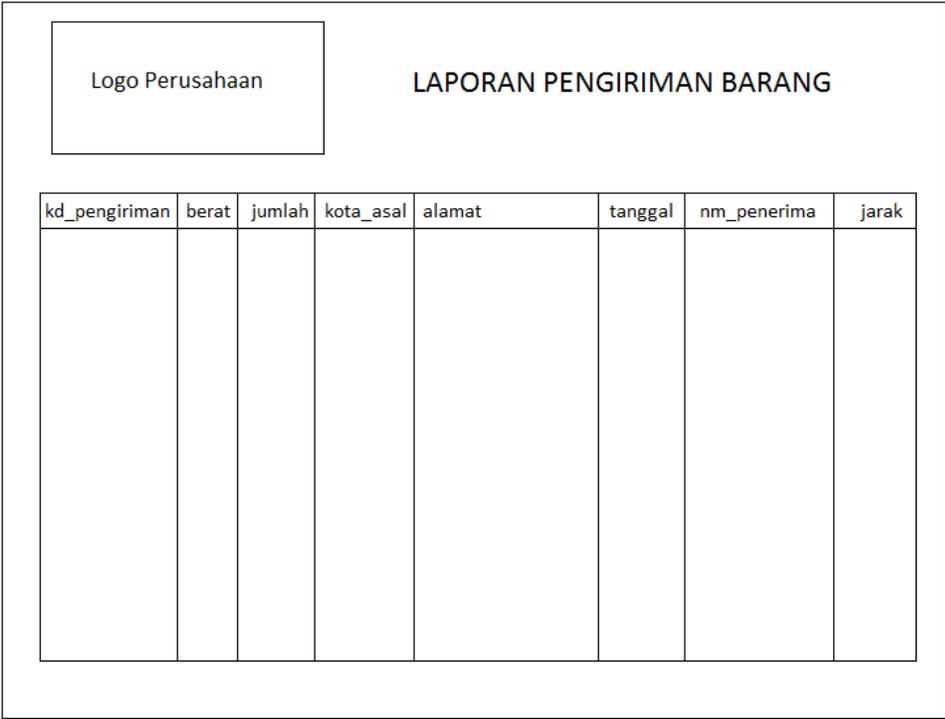
Gambar 3.30 merupakan rancangan tampilan dari menu *About*.



Gambar 3.30 Rancangan Antarmuka Menu *About*

h. Rancangan Laporan

Saat *user* memilih tombol *Print* pada menu *Report*, maka akan muncul tampilan di *window* baru berupa laporan pengiriman barang berdasarkan algoritma yang sudah dipilih. Gambar 3.31 merupakan rancangan dari laporan pengiriman barang.



The image shows a report layout for 'LAPORAN PENGIRIMAN BARANG'. It features a header section with a box for the company logo and the title. Below the header is a table with eight columns: 'kd_pengiriman', 'berat', 'jumlah', 'kota_asal', 'alamat', 'tanggal', 'nm_penerima', and 'jarak'. The table body is currently empty.

kd_pengiriman	berat	jumlah	kota_asal	alamat	tanggal	nm_penerima	jarak

Gambar 3.31 Rancangan Laporan Pengiriman Barang