BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Dalam analisis sistem dilakukan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponen sistem, dengan maksud untuk mengidentifikasikan dan mengevaluasi masalah atau hambatan apa saja yang mungkin muncul sehingga dapat segera ditemukan dan dilakukan tindakan-tindakan yang diperlukan untuk memenuhi kebutuhan-kebutuhan tersebut agar sistem yang dibangun dapat berjalan dengan baik dan sesuai dengan harapan.

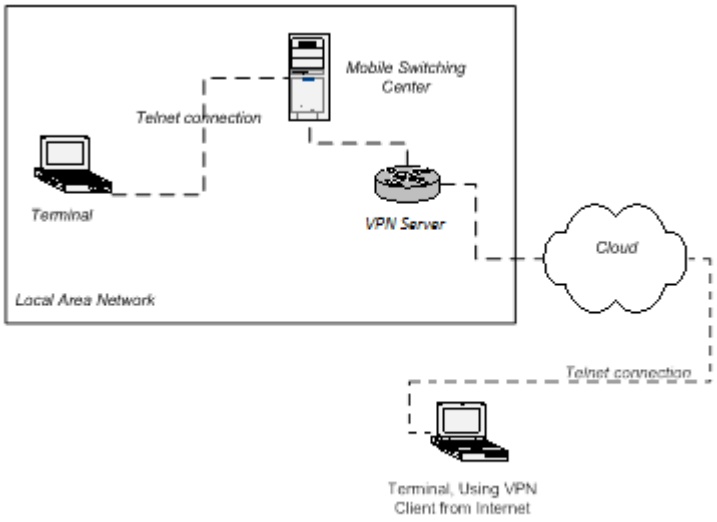
Tahap analisis sistem dilakukan setelah tahap perencanaan sistem dan sebelum tahap perancangan sistem. Tahap analisis merupakan tahap yang paling kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya. Dalam analisis sistem ini akan ditemukan beberapa data dan fakta yang akan dijadikan bahan uji dan analisis menuju pengembangan dan penerapan sebuah aplikasi sistem yang diusulkan.

3.1.1 Analisis Masalah

Dari hasil pengamatan terhadap sistem yang sudah ada, berikut adalah masalah yang terdapat dalam penggunaan sistem *existing* seperti masih diperlukan instalasi program aplikasi untuk tiap-tiap PC yang akan digunakan untuk melakukan kegiatan operasional, masih perlu penghapalan untuk tiap-tiap perintah-perintah *mml command* secara detail, diakarenakan ada kemungkinan lupa, maka perlu selalu melihat kembali buku panduan, *logfile* yang harus dibuat secara manual.

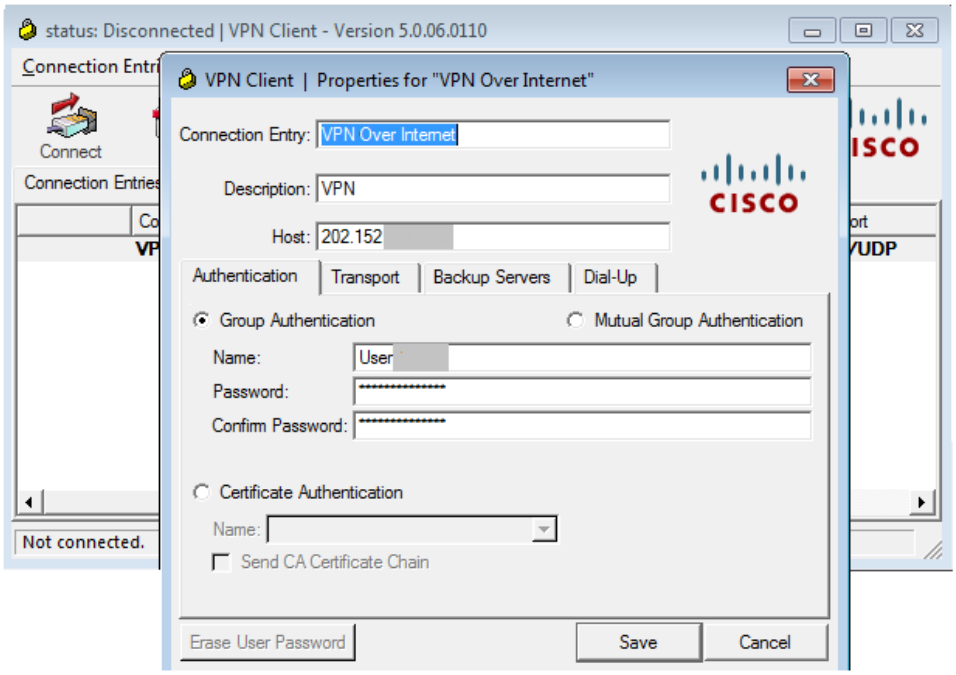
3.1.2 Analisis Sistem Berjalan Untuk Sistem Aplikasi Lama

Berdasarkan hasil pengamatan terhadap sistem aplikasi yang sedang berjalan, berikut pada gambar 3.1 adalah gambaran deskripsi sistem sebelumnya, dimana para *operator* langsung melakukan akses ke perangkat dengan menggunakan program aplikasi *WinFiol* dari masing-masing terminal, baik pada saat berada di LAN *office* ataupun menggunakan *vpn connection* jika berada di luar LAN *office*



Gambar 3.1. Gambaran Sistem Lama

Untuk mengakses *network* pada saat *operator* berada di luar LAN *office* dapat menggunakan program *Cisco VPN client V5.0.06.0110* dengan konfigurasi seperti pada gambar 3.2 :



Gambar 3.2. Cisco VPN *Client*

Keterangan :

Connection Entry : VPN Over Internet

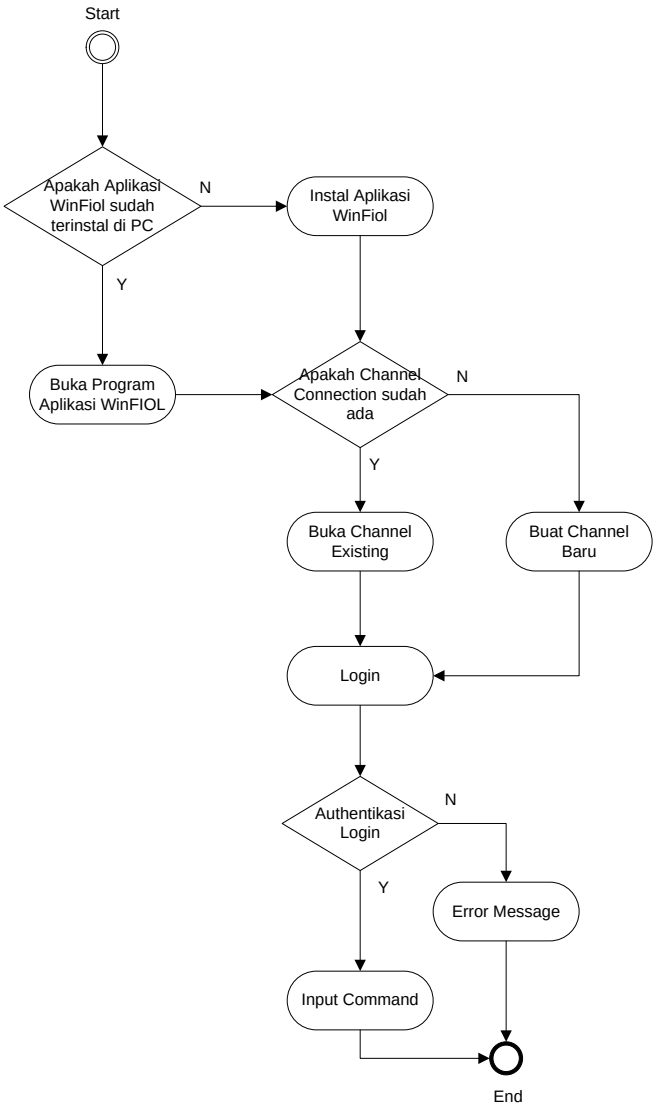
Description *:* VPN

Host *: 202.152.xxx.xxx*

Name *: User*

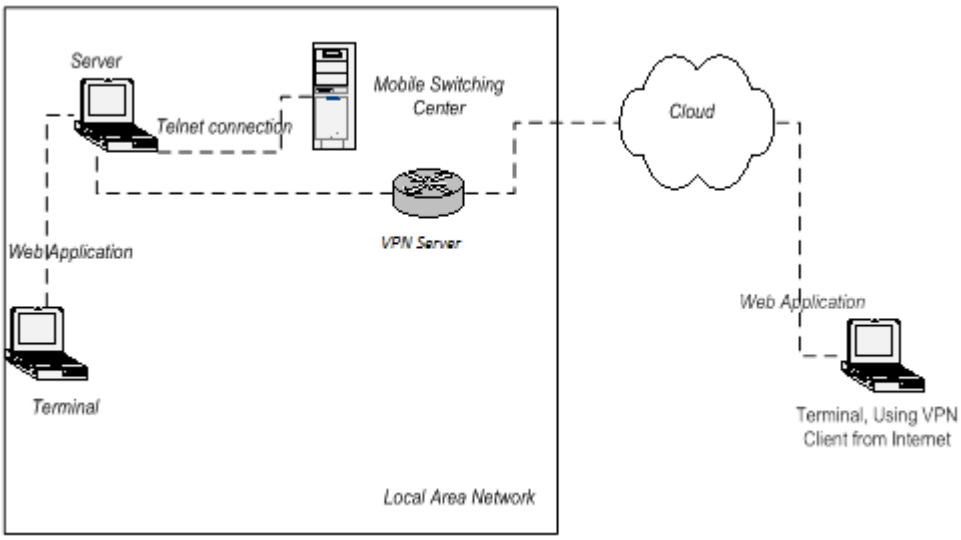
Password *:* **********

Pada gambar 3.3. adalah gambaran *flowchart* dari sistem aplikasi lama yang sedang berjalan :



Gambar 3.3. *Flowchart* Sistem Aplikasi Lama

3.1.3 Deskripsi Sistem Baru



Gambar 3.4. Gambaran Sistem Baru

Pada gambar 3.4. dapat dilihat perbedaan antara sistem awal atau *existing* dengan sistem baru yang akan dibangun, yaitu adanya penambahan perangkat berupa *server* aplikasi dimana program aplikasi ini ditanam. Dalam gambar 3.4 digambarkan dua terminal yang berada di dua jaringan yang berbeda, satu terminal berada di jaringan LAN *office* dan terminal lain berada di luar jaringan LAN *office.* Berikut adalah proses yang terjadi terjadi pada sistem :

a. Terminal

   Proses yang terjadi pada terminal adalah saat terminal berada di luar jaringan *office*, terminal terlebih dahulu harus melakukan koneksi ke *internet*. Setelah koneksi ke *internet* terbangun, maka aplikasi VPN *client* diaktifkan untuk membuat *tunelling* ke VPN *server (*dalam hal ini menggunakan program aplikasi *Cisco VPN Client)* seperti pada gambar 3.2. Setelah koneksi *tunelling* terbangun maka *operator* dapat mulai mengakses aplikasi dengan menggunakan *web browser* dengan cara memasukkan alamat *server* aplikasi pada *web browser*.

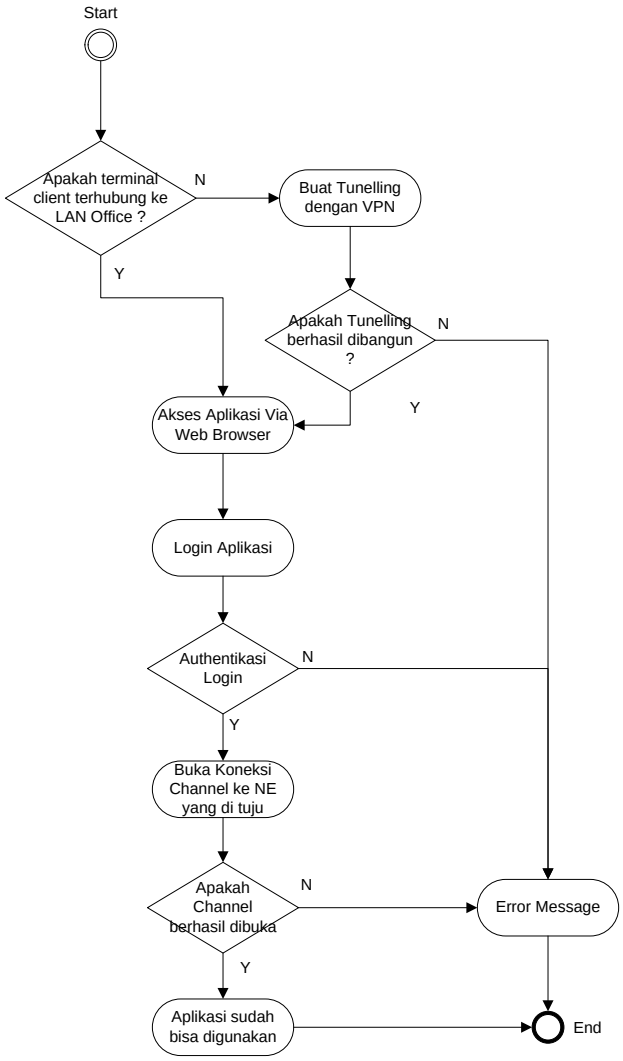
b. VPN *server* dan VPN *client*

   Proses yang terjadi antara VPN *server* dan VPN *client* adalah proes pembangunan *tunelling*, dalam hal ini VPN *server* bersifat pasif yaitu menunggu permintaan koneksi dari VPN *client* melakukan proses autentifikasi dan membangun *tunelling* antara terminal dengan s*erver.*

c. *Server* ke Perangkat.

   Proses komunikasi yang terjadi pada *server* dengan perangkat adalah, setelah koneksi antara *server* dengan perangkat berhasil dibangun, *server* mengirimkan perintah-perintah yang ditujukan ke perangkat, kemudian menampikan *output* kepada *terminal client.*

Berikut pada gambar 3.5. adalah *flowchart* dari sistem aplikasi yang akan dibangun :



Gambar 3.5. *Flowchart* Sistem Aplikasi Baru

3.2 Analisi Kebutuhan Non-Fungsional

Analisa kebutuhan non fungsional menggambarkan keadaan sistem yang akan dibangun, diantaranya adalah analisis pengguna *(User)*, analisis perangkat keras *(Hardware)*, serta analisis perangkat lunak *(Software)* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.2.1 Analisis Pengguna *(User)*

3.2.1.1 *User Existing*

Analisis user *existing* digunakan untuk mengenali dan menjabarkan kembali mengenai keadaan *user* pada saat sistem sebelum dibangun :

Tabel 3.1. Karakteristik *User Existing*

No.	Pengguna	Karakteristik dan Kemampuan
1.	Operator	a. Memahami arsitektur sistem GSM. b. Memahami protokol SS7, SIGTRAN, dan konsep *signalling*. c. Memiliki hak akses ke perangkat MSC. d. Memahami dan menguasai arsitektur perangkat MSC Ericsson dengan *platform* AXE, untuk tingkat lanjut. e. Menguasai *command-command* operasional MSC Ericsson.
2.	Koordinator	a. Memahami arsitektur sistem GSM b. Memahami dan menguasai arsitektur perangkat MSC Ericsson dengan *platform* AXE, untuk tingkat lanjut. c. Bertindak untuk kegiatan koordinasi dan manajerial

3.2.1.2 Kebutuhan *User* untuk Sistem yang akan dibangun

Analisis pengguna untuk sistem yang akan dibangun adalah untuk mengetahui tugas dan karakteristik pengguna yang akan menggunakan sistem aplikasi yang akan di bangun. Dengan mempertimbangkan tingkat pengalaman pengguna dan pemahaman mengenai sistem telekomunikasi dalam rangka penggunaan aplikasi ini.

Tabel 3.2. Karakteristik Pengguna Sistem Aplikasi Baru

No.	Pengguna	Keterangan
1.	Operator	Karakteristik : a. Memahami arsitektur sistem GSM b. Memahami protokol SS7, SIGTRAN, dan konsep *signalling*. c. Memahami arsitektur perangkat MSC Ericsson dengan *platform* AXE, untuk tingkat pemula Hak Akses : a. Melakukan koneksi terhadap *connection-connection* yang telah ada dan meenggunakan MML *command menu* yang sudah disediakan. b. Merubah *password* pribadi untuk *login* ke aplikasi.
2.	*Admin*	Karakteristik : a. Memahami arsitektur sistem GSM. b. Memahami protokol SS7, SIGTRAN, dan konsep *signalling*. c. Memiliki hak akses ke perangkat MSC. d. Memiliki dasar-dasar pengetahuan basis data. e. Memahami dan menguasai arsitektur perangkat MSC Ericsson dengan *platform* AXE, untuk tingkat lanjut. f. Menguasai *command-command* operasional MSC Ericsson. Hak Akses : a. Melakukan pengolahan data *operator* b. Melakukan pengolahan data *MML command*

		c. Melakukan pengolahan data *channel connection* d. Melakukan pengolahan data kordinator
3.	Koordinator	Karakteristik : a. Memahami arsitektur sistem GSM b. Memahami dan menguasai arsitektur perangkat MSC Ericsson dengan *platform* AXE, untuk tingkat lanjut. c. Bertindak untuk kegiatan koordinasi dan manajerial Hak Akses : -Mendapatkan *report logfile* dari setiap aktifitas *operator* dan *admin*

3.2.1.3 *Gap* Analisis

Dari penjelasan-penjelasan sebelummnya dapat diketahui *gap* atau perbedaan karakteristik antara *user existing* dengan *user* sistem aplikasi yang akan dibangun, adalah mengenai cara penggunaan aplikasi dan juga hak akses ke dalam aplikasi yang akan dibangun ini, oleh karena itu diperlukan pelatihan untuk setiap *operator* dan *admin* untuk menggunakan sistem aplikasi baru ini.

3.2.2 Analisis Perangkat Keras (*Hardware*)

3.2.2.1 *Hardware Existing*

Analisis *Hardware existing* digunakan untuk mengenali dan menjabarkan kembali mengenai keadaan *hardware* pada saat sistem sebelum dibangun, dalam hal ini dikarenakan sistem yang akan dibangun adalah betul-betul sistem baru, maka untuk *hardware existing* adalah belum ada, sehingga diperlukan penyediaan *hardware* yang baru secara menyeluruh.

3.2.2.2 Kebutuhan *Hardware* untuk Sistem yang akan Dibangun

Analisis *hardware* untuk sistem yang akan dibangun adalah untuk mengetahui kebutuhan-kebutuhan perangkat keras yang diperlukan dalam pembangunan dan pengujian sistem yang akan dibangun. Dengan mempertimbangkan tingkat kebutuhan spesifikasi minimum bagi pembangunan dan pengujian sistem yang akan dibangun, Berikut pada Tabel 3.3. dan Tabel 3.4. dapat diketahui mengenai kebutuhan minimal perangkat keras untuk pengujian sistem yang dibangun.

Tabel 3.3. Perangkat Keras *Client*

No.	Perangkat keras	Spesifikasi
1.	*Prosessor*	Dengan kecepatan 1,8 GHz
2.	Monitor	LCD 12”
3.	VGA	VGA Card On-Board 64 Mb
4.	Memori	DDR2 2GB
5.	*Keyboard*	*Keyboard*
6.	*Mouse*	*Mouse*
7.	*Hard disk*	350 GB

Tabel 3.4. Perangkat Keras *Server*

No.	Perangkat keras	Spesifikasi
1.	*Prosessor*	Dengan kecepatan 1,6 GHz
2.	Monitor	LCD 14”
3.	VGA	VGA Card On-Board 1,5Gb
4.	Memori	4 GB DDR3
5.	*Keyboard*	*Keyboard*
6.	*Mouse*	*Mouse*
7.	*Hard disk*	360 GB

3.2.2.3 *Gap* Analisis

Dikarenakan pembangunan sistem aplikasi ini adalah baru maka, dari penjelasan-penjelasan diatas dapat diketahui kebutuhan-kebutuhan *hardware* yang diperlukan dalam pembangunan sistem ini.

3.2.3 Analisis Perangkat Lunak *(Software)*

3.2.3.1 *Software Existing*

Analisis *Software existing* dilakukan untuk memaparkan *software-software* apa saja yang sudah ada sebelum sistem dibangun. Berikut pada Tabel 3.5. adalah *software-software* yang sudah ada sebelum sistem dibangun.

Tabel 3.5. Perangkat Lunak *Existing*

No.	Perangkat Lunak
1.	Sistem Operasi *Windows-7 Ultimate*
2.	*Microsoft Office 2010*
3.	*Web browser* yang digunakan *Mozilla Firefox 17.0.1*

3.2.3.2 Kebutuhan *Software* untuk Sistem yang akan Dibangun

Analisis *software* untuk sistem yang akan dibangun dilakukan untuk mengetahui dan memaparkan *software-software* apa saja yang diperlukan dan terlibat dalam rangka proses pembangunan sistem Berikut pada Tabel 3.6. adalah *software-software* yang terlibat dalam proses pembangunan sistem yang akan dibangun.

Tabel 3.6. Kebutuhan Perangkat Lunak untuk Pembangunan Sistem

No.	Perangkat Lunak
1.	*Macromedia Dreamweaver 8.0* Sebagai teks editor
2.	*MySQL 5.5.25a* Sebagai Database
3.	*XAMPP 1.8.0 for Windows*
4.	*PHP 5.4.4*

5.	*Apache 2.4.2 (Win32)*

3.2.3.3 *Gap* Analisis

Dari hasil analisa terhadap perangkat lunak yang telah ada dan dimiliki dan juga analisa kebutuhan-kebutuhan perangkat lunak yang diperlukan dalam rangka pembangunan sistem aplikasi ini, maka dapat diketahui *gap* atau perbedaan antara perangkat lunak yang ada dengan kebutuhan perangkat lunak, oleh karena itu diperlukan pemenuhan kebutuhan untuk kebutuhan perangkat lunak tersebut.

3.2.4 Model Bisnis Sistem Baru.

Kebutuhan-kebutuhan yang ada pada aplikasi ini adalah sebagai berikut :

a. Sistem Aplikasi harus menyediakan fasilitas *login* yang digunakan *admin* dan *operator*. *Form login* berisi *username* dan *password* yang harus di isi oleh *admin* atau *operator*.
b. Sistem harus menyediakan fasilitas lupa *password* yang digunakan oleh *admin* dan *operator* apabila lupa dengan *password*.
c. Sistem harus dapat menyediakan fasilitas pencarian *command* berdasarkan nama-nama *command* yang ingin diketahui oleh *operator*.
d. Sistem harus dapat melakukan atau menjalankan perintah-perintah yang diinputkan oleh *operator* ke tiap-tiap *network element* yang dituju.
e. Sistem harus dapat melakukan *loging file* secara otomatis untuk tiap-tiap *operator* setiap kali *operator* tersebut *login*.
f. *Admin* dapat mengelola data-data *user*, pengolahan-pengolahan tersebut antara lain:
    1. Sistem harus menyediakan fasilitas ubah *password* yang digunakan *admin* dan *operator* untuk mengubah *password.*
    2. Penambahan dan ubah status data *operator* bagi *admin*.

3. Sistem harus menyediakan fasilitas *update profile* yang digunakan *operator* untuk mengubah data pribadi.

g. *Admin* dapat mengelola data-data *channel* dan *command-command*, pengolahan-pengolahan tersebut antara lain:
   1. Menambah *command-command* baru.
   2. Mengurangi atau menghapus *command-command* lama yang sudah tidak di *support* oleh perangkat.
   3. Mengedit *command* yang telah ada.
   4. Menambah atau menghapus koneksi *channel*.
   5. Mengubah data koneksi *channel* yang sudah ada.

h. Sistem harus menyediakan fasilitas pengolahan data laporan bagi *admin*, baik itu laporan statistik *login* ataupun data *logging file operator*.

i. Fungsi *disconnected channel* untuk menutup *channel* yang aktif tanpa *logout* dari sistem aplikasi untuk kemudian memilih *channel connection* yang baru.

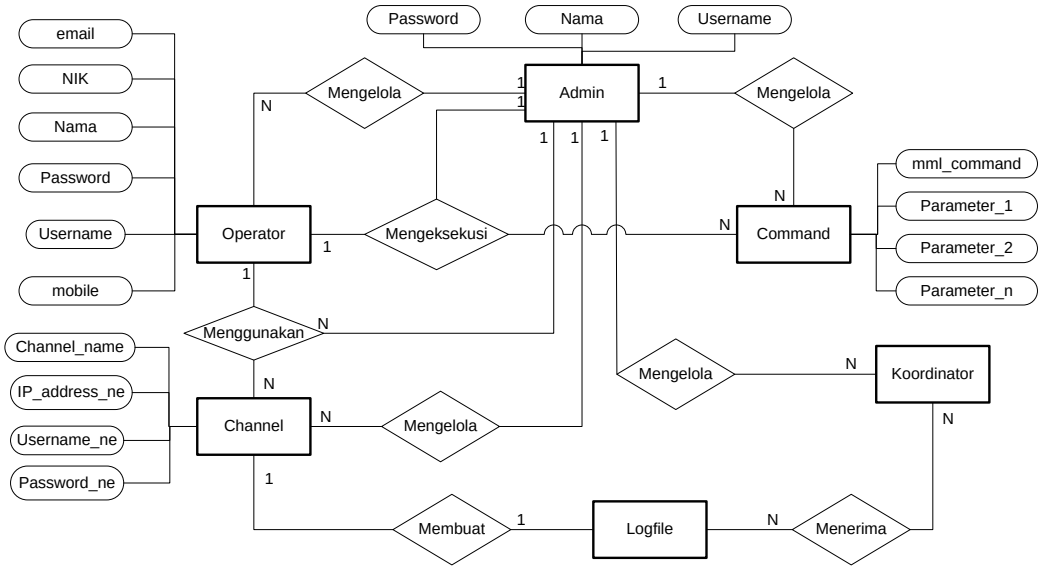
j. *Logout* digunakan *admin* dan *operator* untuk keluar dari sistem.

3.2.5 Analisis Basis Data

Setiap aplikasi membutuhkan data untuk menghasilkan informasi yang dibutuhkan. Data yang ada saling terhubung atau berelasi antara satu dengan yang lain untuk menyajikan informasi yang benar. Apabila data yang ada tidak berelasi, maka tidak akan terbentuk suatu pengolahan data menjadi informasi untuk memodelkan relasi data dalam model analisis perangkat lunak terstruktur digunakan sebuah alat bantu berupa sebuah diagram yang disebut diagram E-R (*Entity-Relationship*).

3.2.5.1 *Entity Relationship Diagram* (ERD)

Dalam menggambarkan hubungan antar data pada sistem yang akan dibangun, digunakan diagram ERD (*Entity Relation Diagram*) sebagai pemodelan sistem yang akan dibangun. Berikut adalah ERD untuk sistem aplikasi yang penulis bangun :



Gambar 3.6. ERD Aplikasi *Routing Mapping* berbasis *Web*

3.2.5.2 Atribut Dalam Setiap Entitas

Tabel 3.7. Atribut Dalam Tiap Entitas

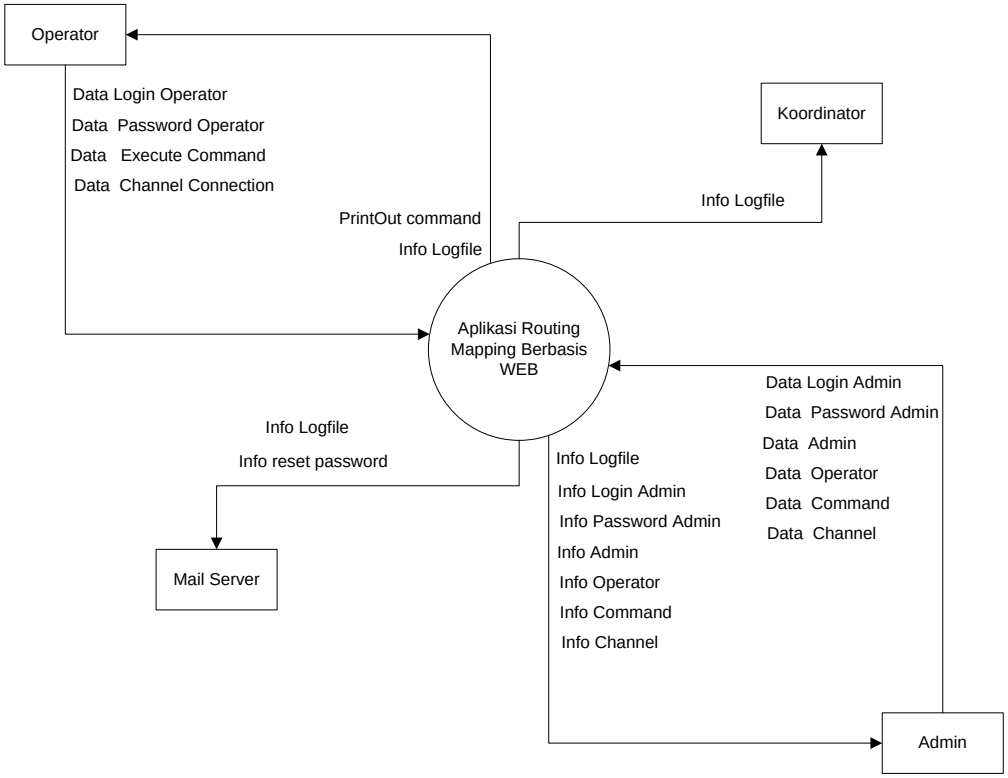
No	Entitas	Atribut
1.	*Admin*	{*username*, *password*, nama, nik, *email*, *mobile*}
2.	*Operator*	{*username*, *password*, nama, nik, *email*, *mobile*}
4.	*Channel*	{*ipaddress*, *username*_ne, *password*_ne, *channel_name*}
5.	*Command*	{id, keterangan, parameter1, parameter2, parameter3, parameter4, parameter5, parameter6, parameter7}
6.	*Logfile*	{*logfile* berupa file txt untuk tiap-tiap user yang

		login}
7.	Koordinator	{nama, nik, *email, mobile*}

3.3 Analisis Kebutuhan Fungsional

3.3.1 Diagram Konteks

Diagram konteks ini menggambarkan keterkaitan aliran-aliran data antara sistem dengan bagian-bagian luar. Pada *diagram konteks* ini, sistem informasi yang dibuat akan menghasilkan sumber informasi yang dibutuhkan dan tujuan yang ingin dihasilkan.

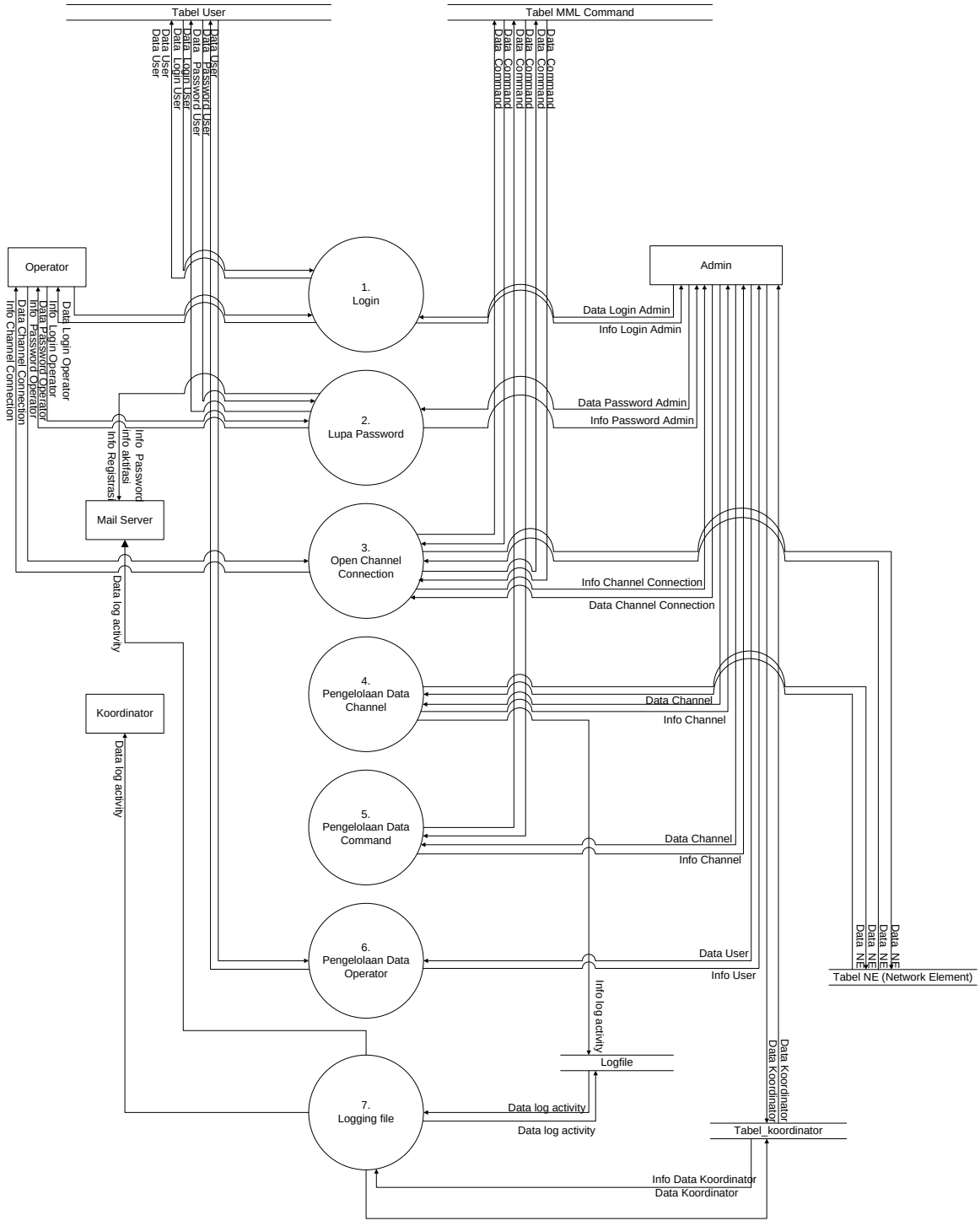


Gambar 3.7. Diagram Konteks Aplikasi *Routing Mapping* berbasis *Web*

3.3.2 *Data Flow Diagram* (DFD)

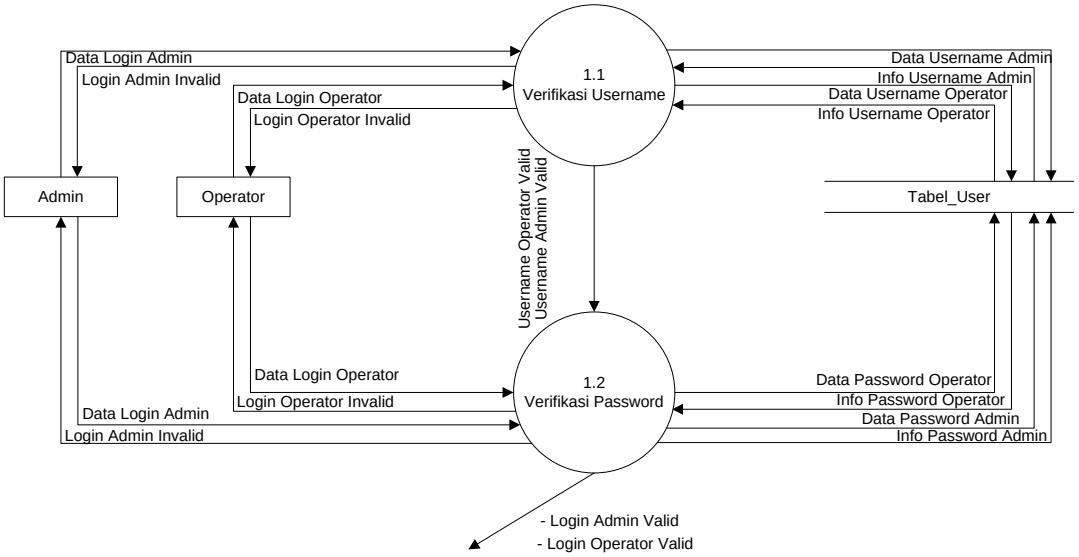
Data Flow Diagram digunakan untuk menggambarkan aliran data yang mengalir pada suatu sistem informasi. Dalam *data flow diagram* (DFD) terdiri dari entitas luar, aliran data, proses, dan penyimpanan data. Salah satu keuntungan menggunakan *data flow diagram* memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan. Adapun *data flow diagram* dari Pembangunan Aplikasi ini adalah sebagai berikut :

1. DFD *level* 1



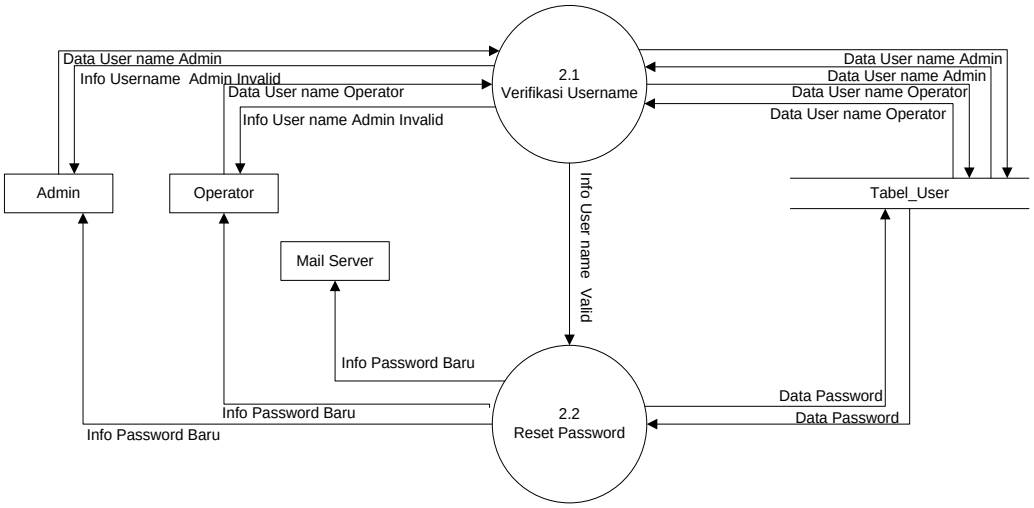
Gambar 3.8. DFD *Level* 1 Aplikasi *Routing Mapping* berbasis *Web*

2. DFD *level* 2 proses 1 *(login)*



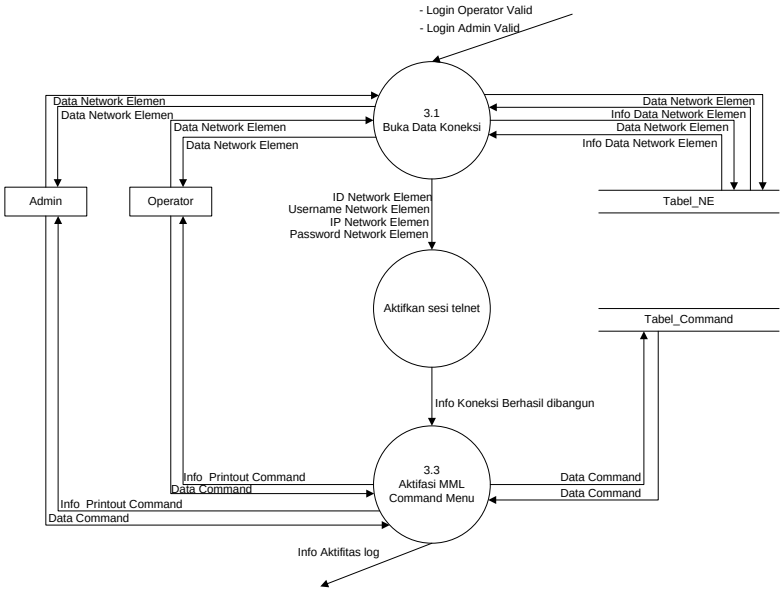
Gambar 3.9 DFD *Level* 2 Proses 1 *(Login)*

3. DFD *Level* 2 proses 2 (lupa *password*)



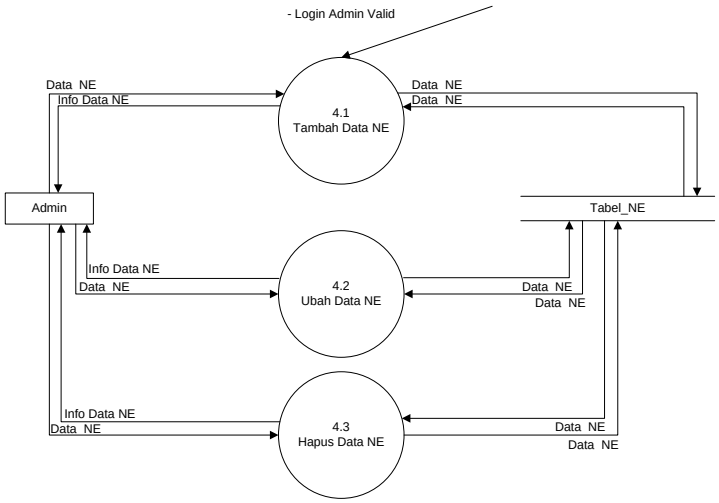
Gambar 3.10 DFD *Level* 2 Proses 2 (Lupa *Password*)

4. DFD *Level* 2 proses 3 (*open connection*)



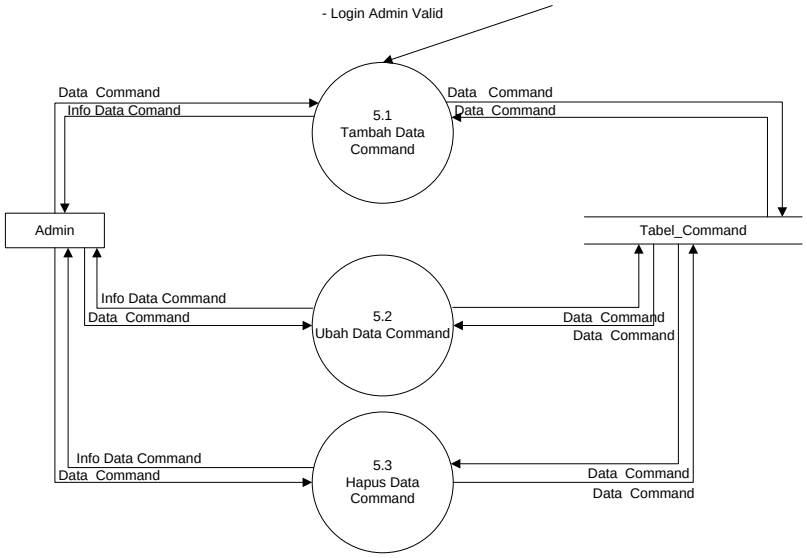
Gambar 3.11 DFD *Level* 2 Proses 3 (*Open Connection*)

5. DFD *Level* 2 proses 4 (pengelolaan data *channel*)



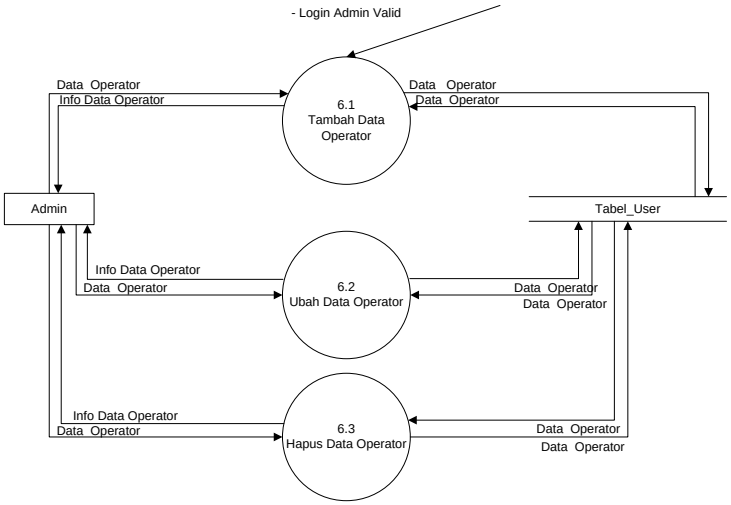
Gambar 3.12 DFD *Level* 2 Proses 4 (Pengelolaan Data *Channel*)

6. DFD *Level* 2 Proses 5 (Pengelolaan Data *Command*)



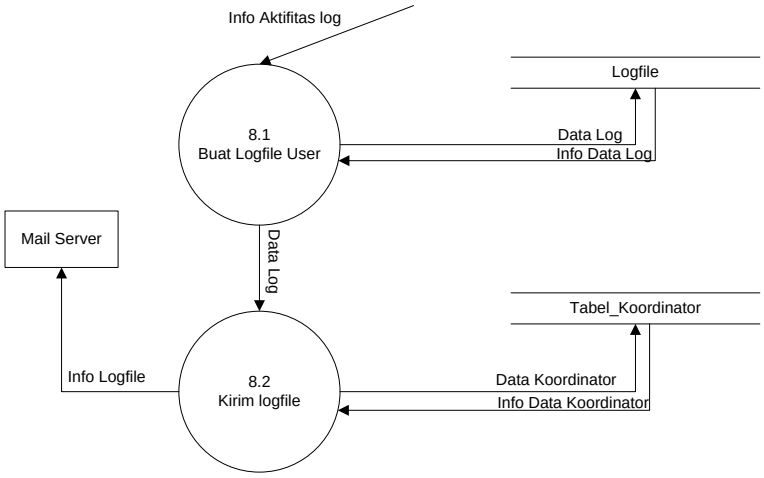
Gambar 3.12 DFD *Level* 2 Proses 5 (Pengelolaan Data *Command*)

7. DFD *Level* 2 Proses 6 (Pengelolaan Data *Operator*)



Gambar 3.13 DFD *Level* 2 Proses 6 (Pengelolaan Data *Operator*)

8. DFD *Level* 2 Proses 7 (*Logging File*)



Gambar 3.14 DFD *Level* 2 Proses 7 *(Logging File)*

3.3.3 Spesifikasi Proses

Spesifikasi proses digunakan untuk menggambarkan proses model yang terdapat pada DFD. Spesifikasi proses dari gambaran DFD di atas akan dijelaskan pada Tabel 3.8. :

Tabel 3.8. Spesifikasi Proses

No.	Detail	Keterangan
1	Nomor Proses	1
	Nama Proses	*Login*
	Source	*Admin, Operator*
	Input	Data *Login Admin*, Data *Login Operator*
	Output	- Info *Login Admin* - Info *Login Operator* - Info *Login Admin Invalid* Info *Login Operator Invalid*
	Destination	Tabel *User*
	Deskripsi	Proses *login* digunakan oleh *admin* dan *operator* untuk mengakses aplikasi

	Logika Proses	1. *Admin* dan *operator* memasukkan data *login admin* dan *operator* 2. Jika data *login admin* sesuai maka *login admin* sukses 3. Jika data *login operator* sesuai maka *login operator* sukses 4. Jika data *login admin* tidak sesuai maka sistem menampilkan info *login invalid.* 5. Jika data *login operator* tidak sesuai maka sistem menampilkan info *login invalid*.
2	Nomor Proses	1.1
	Nama Proses	Verifikasi *Username*
	Source	*Admin* dan *Operator*
	Input	Data *Login Admin*, Data *Login Operator*
	Output	- Info *Login Admin* - Info *Login Operator* - Info *Login Admin Invalid* - Info *Login Operator Invalid*
	Destination	Tabel_*User*
	Deskripsi	Proses ini digunakan untuk verifikasi *admin* dan *operator*
	Logika Proses	1. *Admin* dan *operator* memasukkan data *login* 2. Apabila *username* yang diinputkan benar maka *username valid* 3. Apabila *username* yang diinputkan tidak benar maka akan ada informasi *login invalid*
3	Nomor Proses	1.2
	Nama Proses	Verifikasi *Password*

	Source	*Admin* dan *Operator*
	Input	Data *Login Admin*, Data *Login Operator*
	Output	- Info *Login Admin* - Info *Login Operator* - Info *Login Admin Invalid* - Info *Login Operator Invalid*
	Destination	Tabel_*User*
	Deskripsi	Proses ini digunakan untuk memeriksa *password admin* dan *operator*
	Logika Proses	1. *Admin* dan *operator* memasukkan data *login* 2. Apabila *password* yang dimasukkan benar maka *password valid* 3. Apabila *password* yang diinputkan tidak benar maka akan ada informasi *login invalid*
4.	Nomor Proses	2
	Nama Proses	Lupa *Password*
	Source	*Admin* dan *Operator*
	Input	Data *Login Admin*, Data *Login Operator*
	Output	- Info *Login Admin* - Info *Login Operator*
	Destination	Tabel_*User*, *Mail Server*
	Deskripsi	Proses ini digunakan jika lupa *password*
	Logika Proses	1. *Admin* dan *operator* mengisi data lupa *password* 2. Jika data *username admin* dan atau *operator* sesuai dan terdaftar di sistem maka sistem secara otomatis akan me-*reset password login* kemudian mengirimkan data *password* ke *mail server* 6. Jika data tidak benar atau *username* tidak terdaftar maka akan ada informasi data *invalid*.

5.	Nomor Proses	2.1
	Nama Proses	Verifikasi *Username*
	Source	*Operator, Admin*
	Input	Data *Username, Admin, Operator*
	Output	- Info *Username Admin, Operator invalid*
	Destination	Tabel_*User*
	Deskripsi	Proses ini di gunakan untuk menguji data *username* yang lupa *password*
	Logika Proses	1. *Operator*, *admin* memasukkan data *username* 2. Jika data *username* sesuai maka sistem akan melanjutkan ketahap selanjutnya yaitu *reset password operator*, *admin* 3. Jika data *username operator*, *admin* tidak sesuai maka sistem mengeluarkan info *username invalid*
6.	Nomor Proses	2.2
	Nama Proses	*Reset Password*
	Source	Tabel *User*
	Input	Data *Username*
	Output	Info *Password Default* dari Sistem
	Destination	*Admin, Operator, Mail server*
	Deskripsi	Sistem ini akan me-*reset password* sesuai dengan *username* yang dimaksud
	Logika Proses	1. Sistem ini akan bekerja secara otomatis me-*reset password operator, admin, operator* dan menggantinya dengan *password default*. 2. Kemudian mengirimkan info *password* baru tersebut melalui *mail server*
7.	Nomor Proses	3
	Nama Proses	*Open Channel Connection*

	Source	*Admin, Operator*
	Input	ID_NE
	Output	Buka Sesi *Telnet, MML Command Menu* Aktif
	Destination	Tabel_NE, Tabel_*Command*
	Deskripsi	Proses ini di gunakan untuk membuka sesi *telnet channel* dan mengaktifkan MML *command menu*
	Logika Proses	1. *Operator* atau *admin* memilih *channel* mana yang ingin dihubungi. 2. Kemudian sistem akan mencari data NE yang dimaksud. 3. Kemudian sesi *telnet* diaktifkan. 4. Lalu sistem akan mengaktifkan MML *command menu*.
8.	Nomor Proses	3.1
	Nama Proses	Buka Data Koneksi
	Source	Tabel_NE
	Input	ID_NE
	Output	ID_NE, IP_NE, *Username*_NE, *Password*_NE
	Destination	Proses 3.2
	Deskripsi	Proses ini digunakan untuk pengambilan data NE
	Logika Proses	1. *Operator*, *admin* memasukkan ID_NE yang ingin di koneksikan oleh aplikasi. 2. Sistem akan memeriksa ke dalam *database* untuk mendapatkan data-data dari NE yang dimaksud untuk kemudian menggunakan data-data tersebut untuk melanjutkan ke proses selanjutnya yaitu Proses 3.2
9.	Nomor Proses	3.2
	Nama Proses	Aktifkan sesi *telnet*

	Source	Tabel_NE
	Input	ID_NE IP_NE *Username*_NE *Password*_NE
	Output	Sesi *Telnet* ke NE Aktif
	Destination	Proses 3.3
	Deskripsi	Proses ini digunakan untuk membuka koneksi *telnet* ke NE
	Logika Proses	Aktifkan sesi *telnet* untuk *operator* atau *admin* ke NE yang dimaksud, dengan data-data sesuai dengan informasi yang didapat dari Proses 3.2
10.	Nomor Proses	3.3
	Nama Proses	Aktivasi *MML Command Menu*
	Source	Tabel_*Command*
	Input	Info Koneksi *Telnet* Berhasil dibangun
	Output	Menampilkan Data *Command*, Deskripsi *Command* Beserta Parameter-parameter *Command MML*
	Destination	*Operator, Admin*
	Deskripsi	Proses ini digunakan untuk me-*loading* semua *command-command mml* dari *database* Tabel_Command.
	Logika Proses	1. Sistem akan mengakses tabel_*command* 2. Kemudian sistem me-*loading* deskripsi-deskripsi *command* untuk ditampilkan di *MML Command Menu* 3. *Operator* atau *admin* menggunakan *command-command* yang ada untuk mengakses NE
11.	Nomor Proses	4

	Nama Proses	*Manage Channel*
	Source	*Admin*
	Input	Data_NE
	Output	Info Data_NE
	Destination	Tabel_NE
	Deskripsi	Proses ini digunakan untuk mengolah data *channel connection*
	Logika Proses	1. Sistem akan mengakses Tabel Data_NE 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *Admin* dapat memulai mengolahnya, baik menambah, menghapus atau mengubah
12.	Nomor Proses	4.1
	Nama Proses	Tambah data NE
	Source	*Admin*
	Input	Data_NE
	Output	Info Data_NE
	Destination	Tabel Data_NE
	Deskripsi	Proses ini digunakan untuk menambahkan data *channel connection*
	Logika Proses	1. Sistem akan mengakses Tabel Data_NE 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *Admin* akan menambah data-data *channel connection*, dengan cara mengisi semua data-data pada *filed-filed* yang tersedia 4. Lalu sistem akan mengakses kembali ke Tabel_NE untuk kemudian menambahkan data *connection* tersebut ke dalam Tabel_NE

13.	Nomor Proses	4.2
	Nama Proses	Ubah Data NE
	Source	*Admin*
	Input	Data_NE
	Output	Info Data_NE
	Destination	Tabel_NE
	Deskripsi	Proses ini digunakan untuk merubah data *channel connection* yang sudah ada
	Logika Proses	1. Sistem akan mengakses Tabel Data_NE 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan mengubah data-data *channel connection*, dengan cara merubah data-data pada *filed-filed* yang dikehendaki. 4. Lalu sistem akan mengakses kembali ke Tabel_NE untuk kemudian menyimpan perubahan yang telah dilakukan tersebut ke dalam Tabel_NE
14.	Nomor Proses	4.3
	Nama Proses	Hapus Data NE
	Source	*Admin*
	Input	Data_NE
	Output	Info Data_NE
	Destination	Tabel_NE
	Deskripsi	Proses ini digunakan untuk menghapus data *channel connection* yang sudah ada
	Logika Proses	1. Sistem akan mengakses Tabel Data_NE 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan menghapus data-data

		channel connection yang diinginkan. 4. Lalu sistem akan mengakses kembali ke Tabel_NE untuk kemudian menghapus data *connection* yang dikehendaki tersebut dari Tabel_NE.
15.	Nomor Proses	5
	Nama Proses	*Manage Command*
	Source	*Admin*
	Input	Data_*Command*
	Output	Info Data_*Command*
	Destination	Tabel_*Command*
	Deskripsi	Proses ini digunakan untuk mengolah data *MML Command*
	Logika Proses	1. Sistem akan mengakses Tabel Data_*Command* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* dapat memulai mengolahnya, baik menambah, menghapus atau mengubah.
16.	Nomor Proses	5.1
	Nama Proses	Tambah data *Command*
	Source	*Admin*
	Input	Data_*Command*
	Output	Info Data_*Command*
	Destination	Tabel Data_*Command*
	Deskripsi	Proses ini digunakan untuk menambahkan data *MML Command* pada aplikasi
	Logika Proses	1. Sistem akan mengakses Tabel Data_*Command.* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan menambah data-data

		command, dengan cara mengisi semua data-data pada *filed-filed* yang tersedia. 4. Lalu sistem akan mengakses kembali ke Tabel_*Command* untuk kemudian menambahkan data *connection* tersebut ke dalam Tabel_*Command*.
17.	Nomor Proses	5.2
	Nama Proses	Ubah atau *Edit* Data *Command*
	Source	*Admin*
	Input	Data_*Command*
	Output	Info Data_*Command*
	Destination	Tabel_*Command*
	Deskripsi	Proses ini digunakan untuk merubah data *command* menu yang sudah ada.
	Logika Proses	1. Sistem akan mengakses Tabel Data_*Command.* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan mengubah data-data *channel connection*, dengan cara merubah data-data pada *filed-filed* yang dikehendaki. 4. Lalu sistem akan mengakses kembali ke Tabel_*Command* untuk kemudian menyimpan perubahan yang telah dilakukan tersebut ke dalam Tabel_*Command.*
18.	Nomor Proses	5.3
	Nama Proses	Hapus Data *Command*
	Source	*Admin*
	Input	Data_*Command*
	Output	Info Data_*Command*
	Destination	Tabel_*Command*

	Deskripsi	Proses ini digunakan untuk menghapus data mml *command* yang sudah ada.
	Logika Proses	1. Sistem akan mengakses Tabel Data_*Command.* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan menghapus data-data *command* yang diinginkan. 4. Lalu sistem akan mengakses kembali ke Tabel_*command* untuk kemudian menghapus data *command* yang dikehendaki tersebut dari Tabel_*command.*
19.	Nomor Proses	6
	Nama Proses	Manage Operator
	Source	*Admin*
	Input	Data_*User*
	Output	Info Data_*User*
	Destination	Tabel_*User*
	Deskripsi	Proses ini digunakan untuk mengolah data *User*
	Logika Proses	1. Sistem akan mengakses Tabel Data_*User* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* dapat memulai mengolahnya, baik mengubah status, merubah isi, ataupun menghapus
20.	Nomor Proses	6.1
	Nama Proses	*Edit Operator*
	Source	*Admin*
	Input	Data_*User*
	Output	Info Data_*User*
	Destination	Tabel Data_*User*

	Deskripsi	Proses ini digunakan untuk merubah termasuk mengaktifkan status hak akses *operator*
	Logika Proses	1. Sistem akan mengakses Tabel Data_*User* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* dapat mengubah data *operator*, termasuk mengaktifkan *operator* pada saat pertama kali daftar 4. Lalu sistem akan mengakses kembali ke Tabel_*User* untuk kemudian menyimpan perubahan data yang telah dibuat
21.	Nomor Proses	6.2
	Nama Proses	Hapus Data *Operator*
	Source	*Admin*
	Input	Data_*User*
	Output	Info Data_*User*
	Destination	Tabel_*User*
	Deskripsi	Proses ini digunakan untuk menghapus data *operator* yang sudah ada
	Logika Proses	1. Sistem akan mengakses Tabel Data_*User* 2. Me-*loading* dan menginformasikannya ke dalam aplikasi. 3. Kemudian *admin* akan menghapus data-data *operator* yang diinginkan. 4. Lalu sistem akan mengakses kembali ke Tabel_*user* untuk kemudian menghapus data *operator* yang dikehendaki tersebut dari Tabel_*User*
22.	Nomor Proses	7
	Nama Proses	*Logfile*

	Source	Info aktifitas *log* dari Proses 4.3 Aktifasi *mml command*, Tabel Koordinator
	Input	Info *logfile*
	Output	Data *logfile*
	Destination	*Mail Server*, Koordinator
	Deskripsi	Proses ini melakukan proses *logfile* untuk semua aktifitas untuk tiap-tiap *user* baik *operator* maupun *admin* dan mengirimkannya kepada koordinator
	Logika Proses	1. Proses ini akan mencatat semua aktifitas *user*. 2. Mengelompokkan data *log* berdasarkan *username* dan tanggal. 4. Kemudian mengirimkan data *logfile* kepada *mail server*.
23.	Nomor Proses	7.1
	Nama Proses	Buat *logfile user*
	Source	Data aktifitas *logfile*
	Input	Data aktifitas *logfile*
	Output	*Logfile*
	Destination	Proses 7.2 Kirim *logfile*
	Deskripsi	Proses ini melakukan proses *logfile* untuk semua aktifitas untuk tiap-tiap *user* baik *operator* maupun *admin*
	Logika Proses	1. Proses ini akan mencatat semua aktifitas *user*. 2. Mengelompokan data *log* berdasarkan *username* dan tanggal. 5. Kemudian menyimpannya ke dalam sebuah *logfile*
24.	Nomor Proses	7.2
	Nama Proses	Kirim *Logfile*
	Source	Data *logfile*, Tabel_Koordinator

	Input	Data *logfile*
	Output	Info *Logfile*
	Destination	*Mail Server*
	Deskripsi	Proses ini melakukan proses pengiriman *logfile* untuk tiap-tiap koordinator
	Logika Proses	1. Proses ini akan mengambil info data *logfile.* 2. Mengecek data koordinator yang perlu dirimkan *logfile.* Menghubungi *mail server* untuk pengiriman.

3.4 Perancangan Basis Data

3.4.1 Kamus Data

Kamus data merupakan sebuah daftar yang tersusun dari elemen data yang berhubungan dengan sistem. Kamus data dibuat berdasarkan arus data yang ada di *diagram flow data.* Berikut ini adalah kamus data yang di ambil dari *data flow diagram* pembangunan aplikasi *routing mapping* berbasis *web* yang diusulkan :

Tabel 3.9. Kamus Data

Nama Aliran data	Data *Login Admin*
Where used / how used	*Admin* Proses 1, Proses 1.1, Proses 1.2
Deskripsi	Berisi data *login Admin*
Struktur data	*username+password*
Email *Password*	[A-Z | a-z | 0-9] [A-Z | a-z | 0-9|*symbol*]
Nama Aliran data	Data *Login Operator*
Where used / how used	*Operator* Proses 1, Proses 1.1, Proses 1.2
Deskripsi	Berisi data *login operator*

Struktur data	*username+password*
username *password*	[A-Z | a-z | 0-9] [A-Z | a-z |0-9|*symbol*]
Nama Aliran data	Data *Operator*
Where used / how used	*Admin* Proses 6, Proses 6.1, Proses 6.2
Deskripsi	Berisi data *operator* yang akan di simpan di dalam *database*
Struktur data	*username*+*password*+nama+nik+*email*+*mobile*+tipe+status
username *password* nama nik *email* *mobile* tipe status	[A-Z | a-z | 0-9] [A-Z | a-z | 0-9|*symbol*] [A-Z | a-z] [0-9] [A-Z | a-z|0-9|*symbol*] [0-9] [*admin*|*operator*] [Aktif|Belum Aktif]
Nama Aliran data	Data *Command*
Where used / how used	*Admin, Operator* Proses 3.3, Proses 5, Proses 5.1, Proses 5.2, Proses 5.3
Deskripsi	Berisi data *mml command* yang disimpan dalam *database*
Struktur data	*Id_command*, *command*, penjelasan, parameter1, parameter2, parameter3, parameter4, parameter5, parameter6, parameter7
Id_command Command	[0-9] [A-Z|a-z]

Penjelasan Parameter1 Parameter2 Parameter3 Parameter4 Parameter5 Parameter6 Parameter7	[A-Z|a-z] [A-Z|a-z] [A-Z|a-z] [A-Z|a-z] [A-Z|a-z] [A-Z|a-z] [A-Z|a-z] [A-Z|a-z]
Nama Aliran data	Data NE
Where used / how used	*Admin*, *Operator* Proses 3, Proses 3.1, Proses 3.2, Proses 4, Proses 4.1, Proses 4.2, Proses 4.3
Deskripsi	Berisi data NE yang disimpan dalam *database* yang digunakan untuk melakukan sesi *telnet*
Struktur data	Id_ne, ne, ip, *username*, *password*
Id_ne Ne Ip *Username* *Password*	[0-9] [a-z|A-Z] [0-9|*symbol*] [a-z|A-Z|0-9] [a-z|A-Z|0-9|*symbol*]
Nama Aliran data	*Logfile*
Where used / how used	*Mail server*, Koordinator Proses 7, Proses 7.1, Proses 7.2
Deskripsi	Berisi file *text* tentang aktifitas sesi *telnet* untuk tiap-tiap *Operator* dan *Admin*
Struktur data	Berisi *logfile* berbentuk *text*
Nama file	Tanggal-*Username*

3.4.2 Skema Relasi

3.4.3 Perancangan Struktur Tabel

Tabel-tabel yang terdapat dalam basis data yang digunakan dalam sistem ini adalah sebagai berikut :

1. Tabel *User*

Tabel ini adalah perancangan untuk tabel_*user* yang akan digunakan pada pembangunan sistem. Struktur tabel *user* ini adalah sebagai berikut:

Tabel 3.10. Tabel *User*

Nama Filed	Type	Size	Keterangan
id_user	Int	11	*primary key, auto increment, not null*
username	varchar	20	*primary key, not null*
password	Varchar	40	*not null*
Nama	varchar	30	*not null*
Nik	varchar	12	*not null*
email	varchar	30	*not null*
mobile	varchar	15	*not null*
Tipe	enum	-	*admin*,operator
aktif_sesi	enum	-	0,1

2. Tabel NE (*Network Elemen)*

Tabel ini adalah perancangan untuk tabel_NE yang akan digunakan pada pembangunan sistem. Struktur tabel NE ini adalah sebagai berikut:

Tabel 3.11. Tabel NE

Nama Filed	Type	Size	Keterangan
id_ne	Int	11	*primary key, auto increment, not null*
Ne	varchar	8	*not null*
Ip	varchar	15	*not null*
Username	varchar	10	*not null*

Password	varchar	50	*not null*

3. Tabel *Command*

Tabel ini adalah perancangan untuk tabel_*command* yang akan digunakan pada pembangunan sistem. Struktur tabel *command* ini adalah sebagai berikut:

Tabel 3.12. Tabel *Command*

Nama Filed	Type	Size	Keterangan
Id_command	int	11	*primary key, auto increment, not null*
Command	varchar	5	*not null*
Penjelasan	varchar	50	*not null*
parameter1	varchar	5	*Null*
parameter2	varchar	5	*Null*
parameter3	varchar	5	*Null*
parameter4	varchar	5	*Null*
parameter5	varchar	5	*Null*
parameter6	varchar	5	*Null*
parameter7	varchar	5	Null

3.4.4 Perancangan Antarmuka

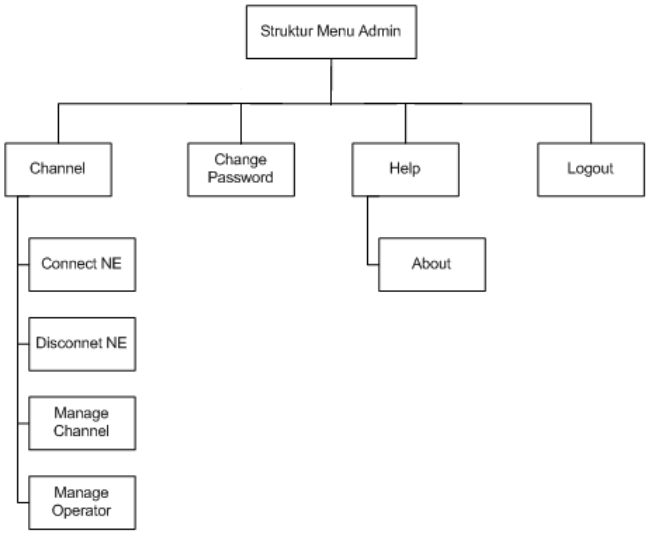
Interface atau antarmuka merupakan tampilan dari suau program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *user*. Sistem yang akan di bangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *user*.

Begitu juga dengan perancangan *interface* untuk pembangunan aplikasi *routing mapping* berbasis *web* ini, penulis ingin mendesain sebuah *interface* yang semudah mungkin untuk digunakan, agar tercapai tujuannya yaitu membantu proses kegiatan operasional dengan lebih baik.

3.4.5 Perancangan Struktur Menu

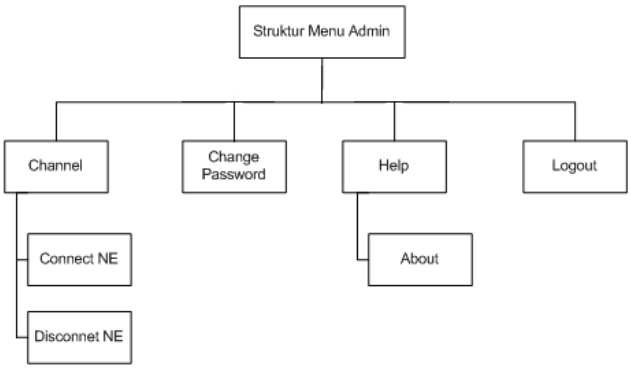
Perancangan struktur menu menggambarkan keterkaitan setiap menu yang bisa di akses oleh *user* yang digambarkan sebagai berikut:

1. Struktur menu *admin*



Gambar 3.15. Struktur Menu *Admin*

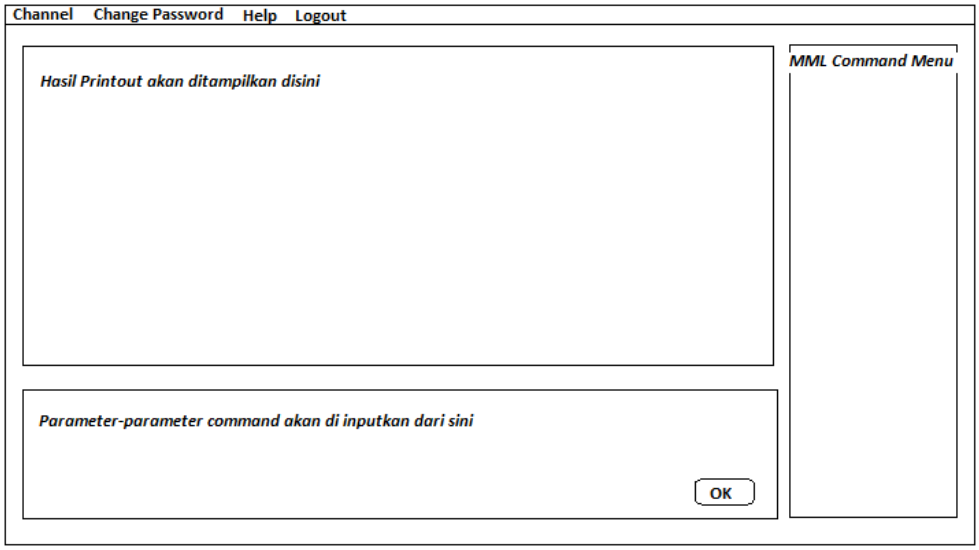
2. Struktur menu *operator*



Gambar 3.16. Struktur Menu *Operator*

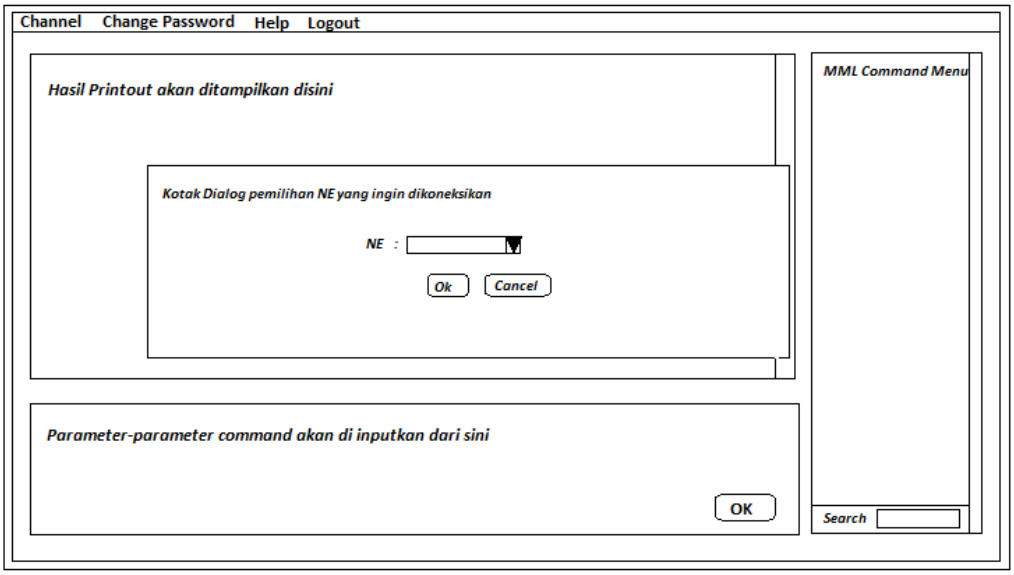
3.4.6 Perancangan Antarmuka

1. Perancangan antarmuka menu utama



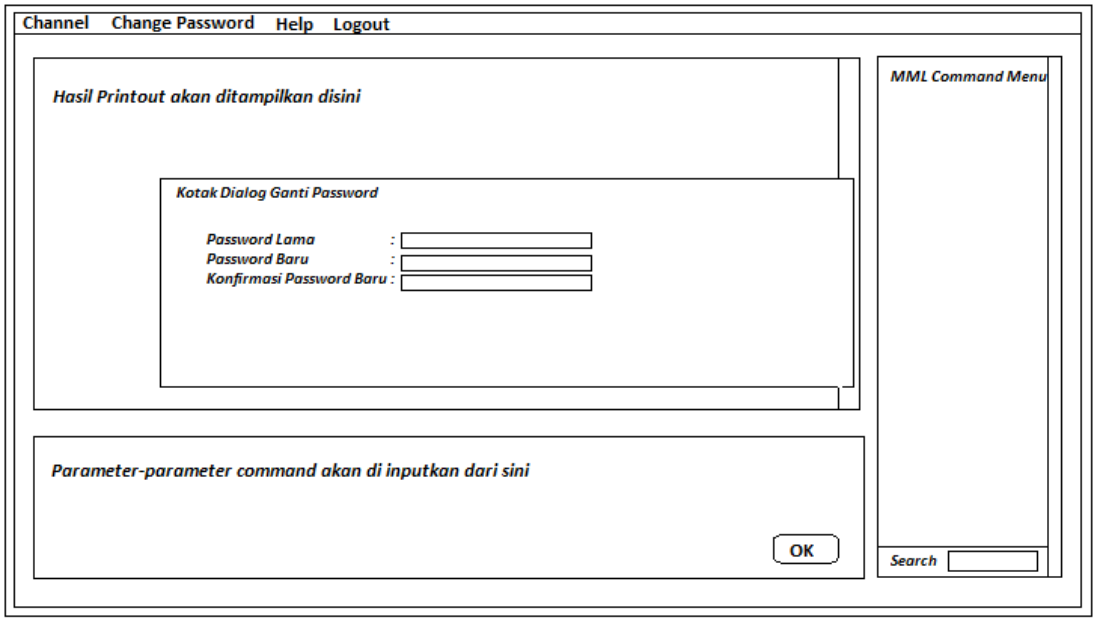
Gambar 3.17. Perancangan Antarmuka Menu Utama

2. Perancangan antarmuka menu *channel connection*



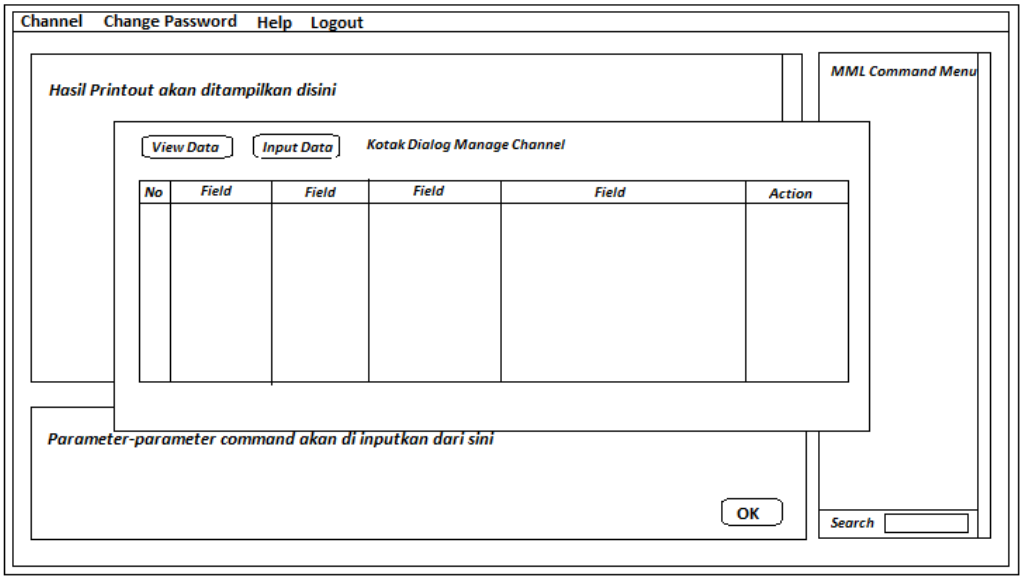
Gambar 3.18. Perancangan Antarmuka Menu *Channel Connection*

3. Perancangan antarmuka menu ganti *password*



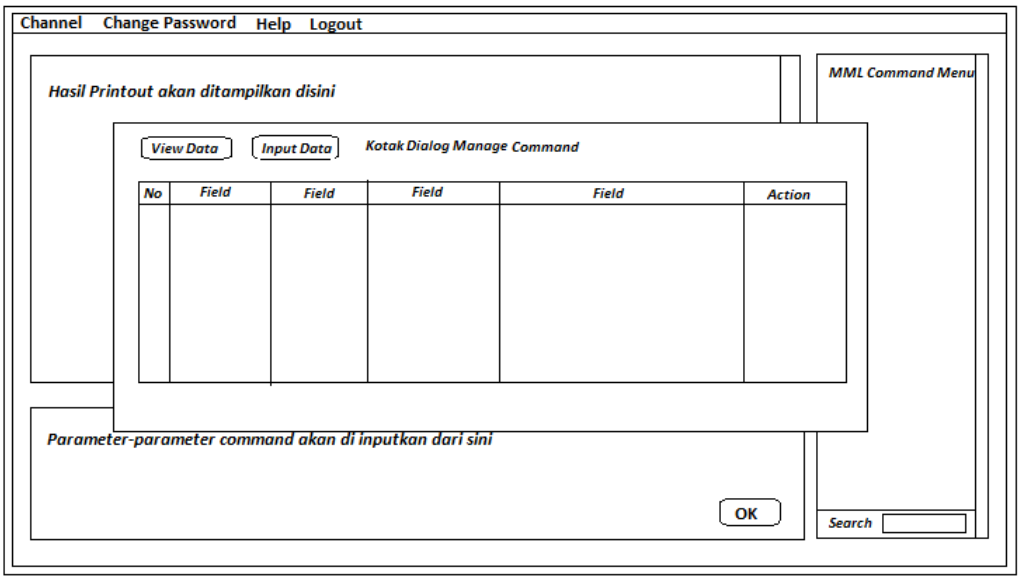
Gambar 3.19. Perancangan Antarmuka Menu Ganti *Password*

4. Perancangan antarmuka menu *manage channel*



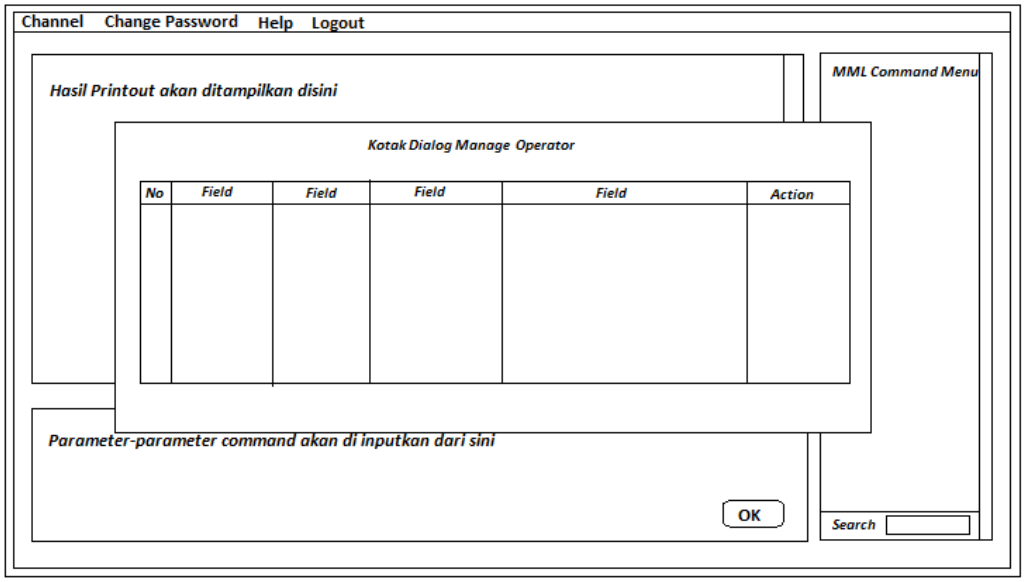
Gambar 3.20. Perancangan Antarmuka Menu *Manage Channel*

5. Perancangan antarmuka menu *manage command*



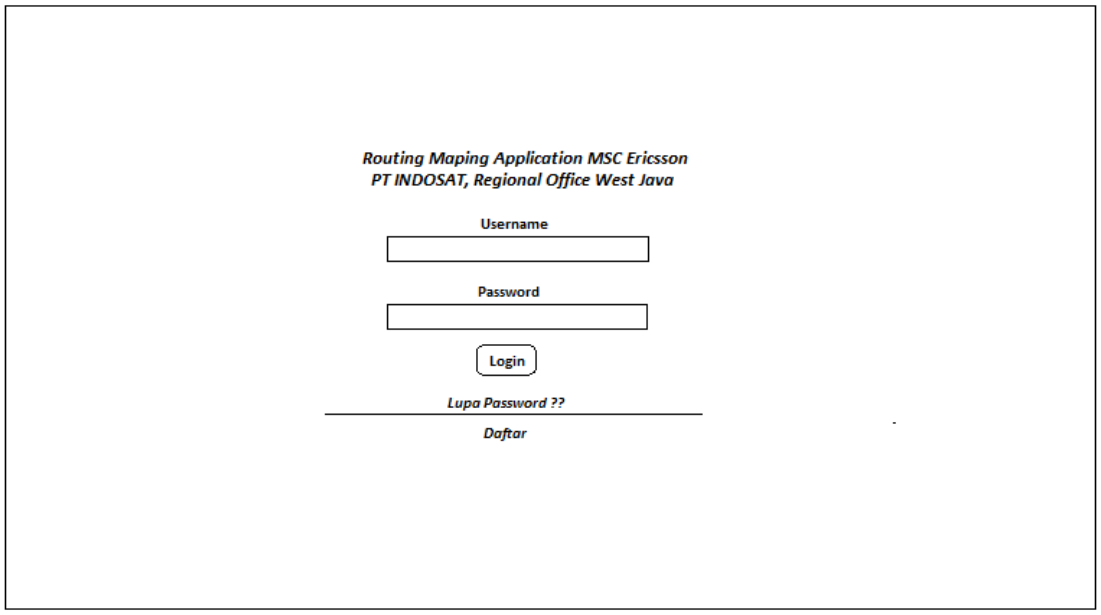
Gambar 3.21. Perancangan Antarmuka Menu *Manage Command*

6. Perancangan antarmuka menu *manage Operator*



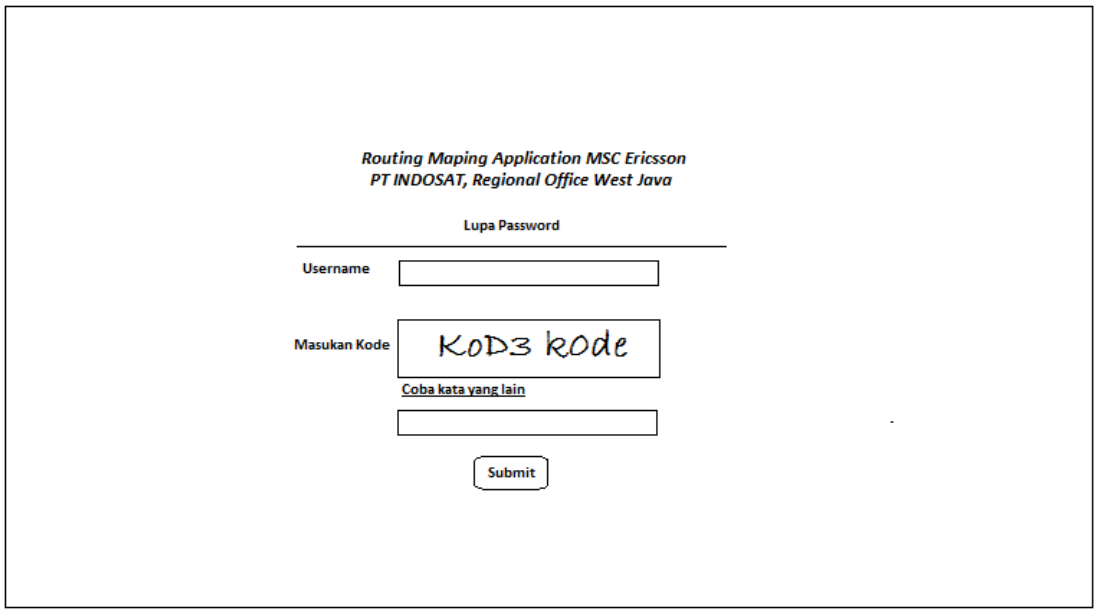
Gambar 3.22. Perancangan Antarmuka Menu *Manage Operator*

7. Perancangan antamuka *form login*



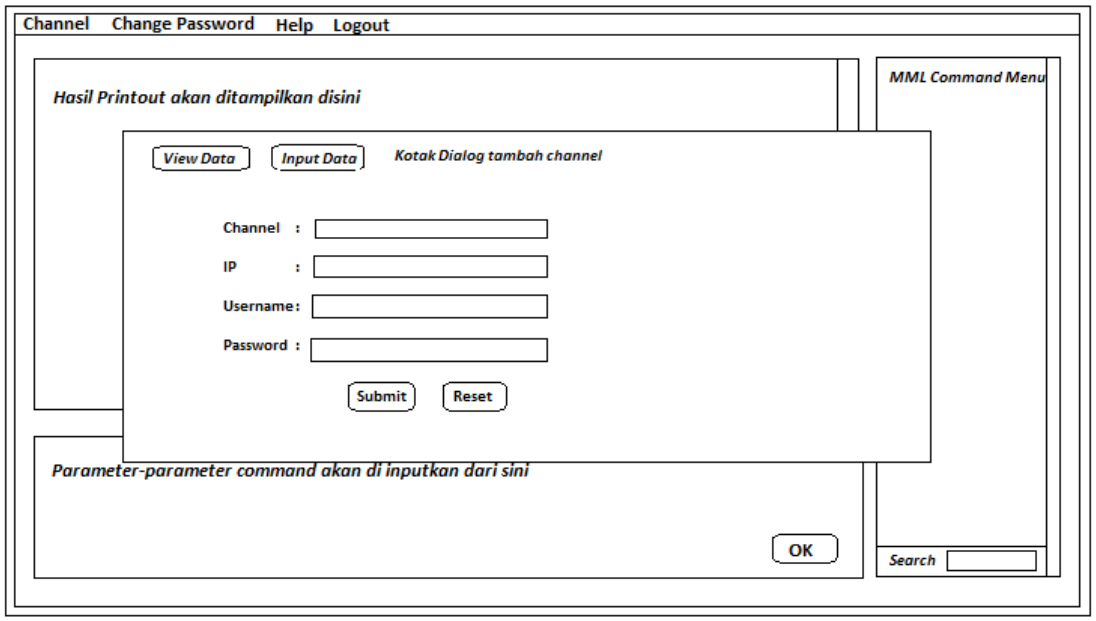
Gambar 3.23. Perancangan Antamuka *Form Login*

8. Perancangan antamuka lupa *password*



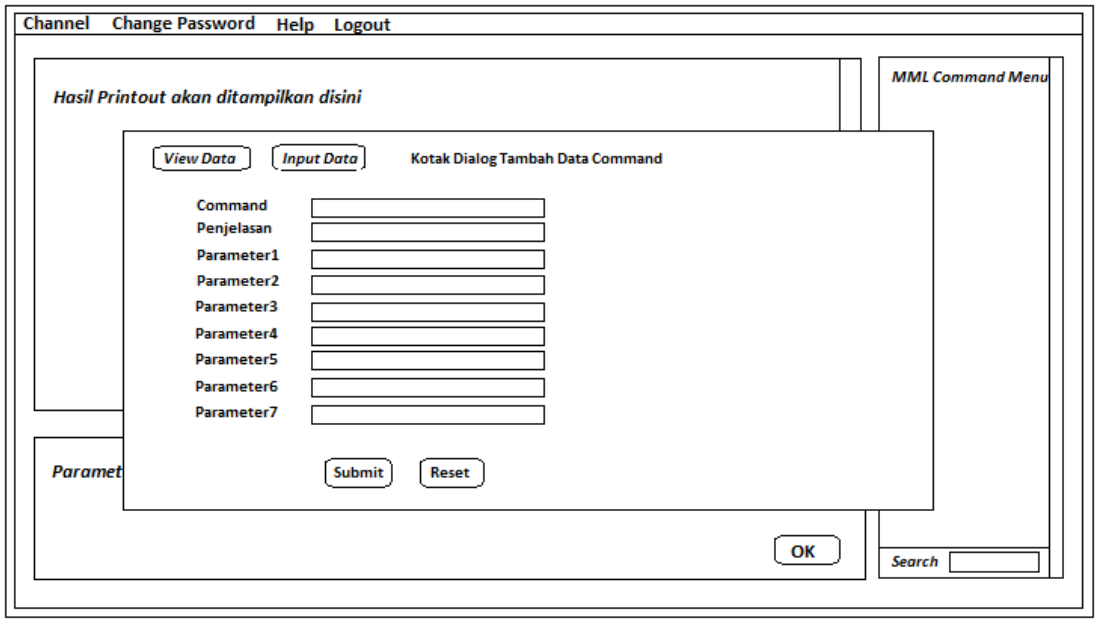
Gambar 3.24. Perancangan Antarmuka Lupa *Password*

9. Perancangan antarmuka tambah *channel*



Gambar 3.25. Perancangan Antarmuka Tambah *Channel*

10. Perancangan antarmuka tambah *command*

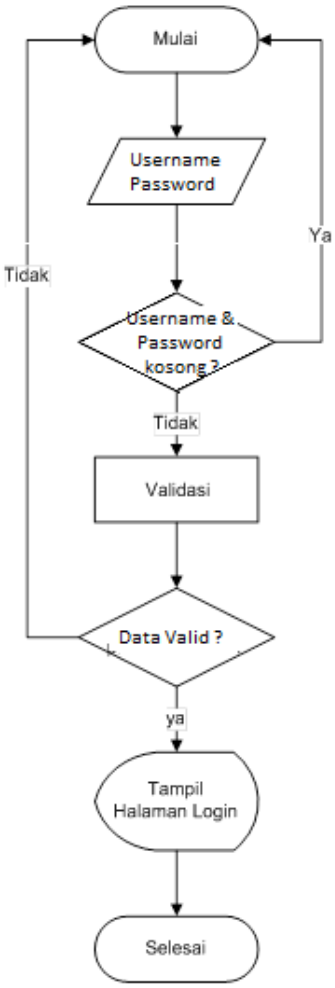


Gambar 3.26. Perancangan Antarmuka Tambah *Command*

3.4.7 Perancangan Prosedural

Berikut merupakan perancangan prosedural yang terdapat dalam sistem aplikasi:

3.4.7.1 Prosedur *Login*



Gambar 3.27. Prosedur *Login*

Algoritma prosedur *login* :

Di masukan data *username* dan *password*, jika *username* terdaftar di *database* tabel data_*user* dan *password* yang dimasukan sesuai maka *user* diijinkan untuk manggunakan aplikasi dan halaman utama di buka.

Deskripsi :

Baca *username* dan *password* dari data masukan

Jika (*username*_masukan = *username*.data_*user*)

Dan Jika(*password* = *password*.data_*user*)

{

Autentifikasi = Berhasil_*Login*

Loading Halaman Utama

}

Jika tidak

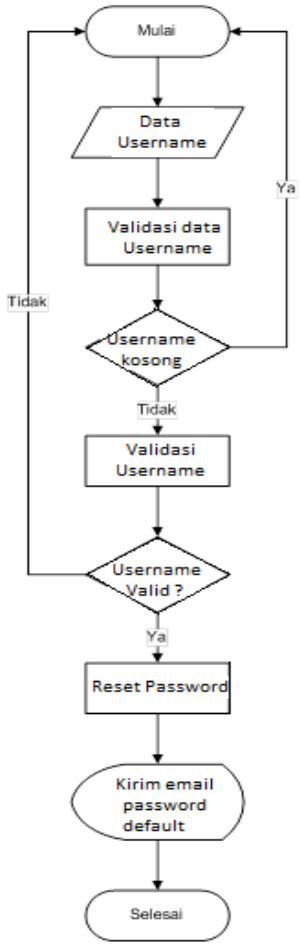
{

Munculkan kotak dialog "*Username* / *Password* anda salah"

Kembali ke halaman *login*

}

3.4.7.2 Prosedur Lupa *Password*



Gambar 3.28. Prosedur Lupa Password

Algoritma prosedur lupa *password* :

Prosedur ini untuk me-*reset password user*, jika *user* mengalami lupa *password*. Sebagai data masukan adalah data *username* dan karakter *captcha* sebagai pembukti bahwa peminta *reset* adalah manusia. Setelah autentifikasi sukses maka sistem akan me-*reset password* dengan *password* baru, untuk kemudian mengirim data *password* baru tersebut kepada *user* melalui *email* yang dimiliki *user* seperti yang terdapat pada *database* data_*user*.

Deskripsi :

Baca *username* dan ka rakter_*captcha* dari masukan

Jika (*username*_masukan = *username*.data_*user*)

Dan Jika(*captcha*_masukan = *captcha.system)*

{

Autentifikasi = Berhasil_*Username*Terdaftar

huruf=str_shuffle("qwertyuiopasdfghjkzxcvbnmQWERTYUIOPASDFGHJKLZXCVBNM1234567890")

kode = substr(huruf,1,8)

kodes = encpass(kode)

UPDATE `tabel_*user*` set `*password*` = 'kodes' *where* `*username*` = '*username*' "

email = data_*user*('*email'*)

kirim *email* ("Kami telah me-*reset password* Anda, *password* sementara Anda adalah $kode, Segera lakukan *login* dan lakukan pergantian *password* Anda, Terima Kasih ...")

Munculkan kotak dialog ("*reset password* berhasil, silahkan cek *email* Anda")

}

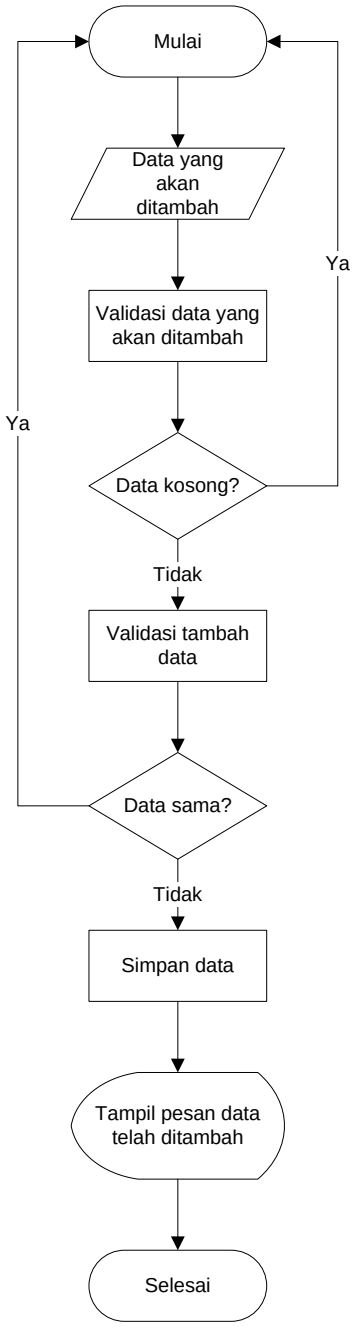
Jika tidak

{

Munculkan kotak dialog "*reset password* gagal"

}

3.4.7.3 Prosedur Tambah



Gambar 3.29. Prosedur Tambah

Algoritma prosedur tambah :

Prosedur ini untuk menambahkan data, baik data *user*, data *channel*, data *command*, dan data koordinator. Sebagai data masukan adalah data-data yang terkait dengan masing-masing tabel.

Deskripsi :

```
Baca kolom_1 dari data masukan
Baca kolom_2 dari data masukan
Baca kolom_3 dari data masukan
Baca kolom_4 dari data masukan
Baca kolom_n dari data masukan
Jika (kolom_1.masukan =””){
    Munculkan kotak dialog (“kolom_1 harus di isi”)
    }
Atau Jika (kolom_2.masukan =””){
    Munculkan kotak dialog (“kolom_2 harus di isi”)
    }
Atau Jika (kolom_3.masukan =””){
    Munculkan kotak dialog (“kolom_3 harus di isi”)
}
Atau Jika (kolom_4.masukan =””){
    Munculkan kotak dialog (“kolom_4 harus di isi”)
}
Atau Jika (kolom_n.masukan =””){
    Munculkan kotak dialog (“kolom_n harus di isi”)
}
Jika Tidak
    {
    num1 = mysql_num_rows(mysql_query("select * from tabel where
    kolom_n=”kolom_n.masukan”));
    Jika (num1>1)
```

{

Munculkan kotak dialog (“Nilai kolom_n sudah ada”)

}

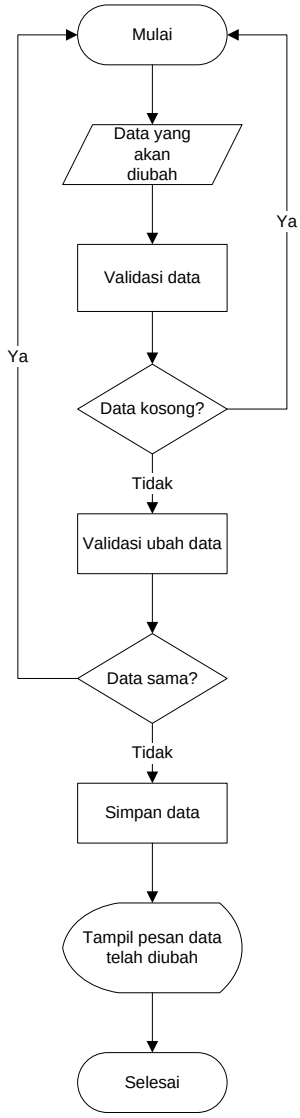
Jika Tidak

{

Mysql_query = "*insert into* tabel (kolom_1, kolom_2, kolom_3, kolom_4, kolom_n) values('kolom_1.masukan', ' kolom_2.masukan ', ' kolom_3.masukan ', ' kolom_4.masukan ', ' kolom_n.masukan')"

Munculkan kotak dialog (“Data berhasil ditambah”)

3.4.7.4 Prosedur Ubah



Gambar 3.30. Prosedur Ubah

Algoritma prosedur ubah :

Prosedur ini untuk merubah data yang sudah ada, baik data *user*, data *channel*, data *command*, dan data koordinator. Sebagai data masukan adalah data-data yang terkait dengan masing-masing tabel.

Deskripsi :

Baca kolom_1 dari data masukan
Baca kolom_2 dari data masukan
Baca kolom_3 dari data masukan
Baca kolom_4 dari data masukan
Baca kolom_n dari data masukan
Jika (kolom_1.masukan =””){
    Munculkan kotak dialog (“kolom_1 harus di isi”)
    }
Atau Jika (kolom_2.masukan =””){
    Munculkan kotak dialog (“kolom_2 harus di isi”)
    }
Atau Jika (kolom_3.masukan =””){
    Munculkan kotak dialog (“kolom_3 harus di isi”)
}
Atau Jika (kolom_4.masukan =””){
    Munculkan kotak dialog (“kolom_4 harus di isi”)
}
Atau Jika (kolom_n.masukan =””){
    Munculkan kotak dialog (“kolom_n harus di isi”)
}
Jika Tidak
    {
    num1 = mysql_num_rows(mysql_query("*select* * *from* tabel *where* kolom_n=”kolom_n.masukan”));
    Jika (num1>1)

{

Munculkan kotak dialog (“Nilai kolom_n sudah ada”)

}

Jika Tidak

{

Mysql_query = "UPDATE `tabel` SET

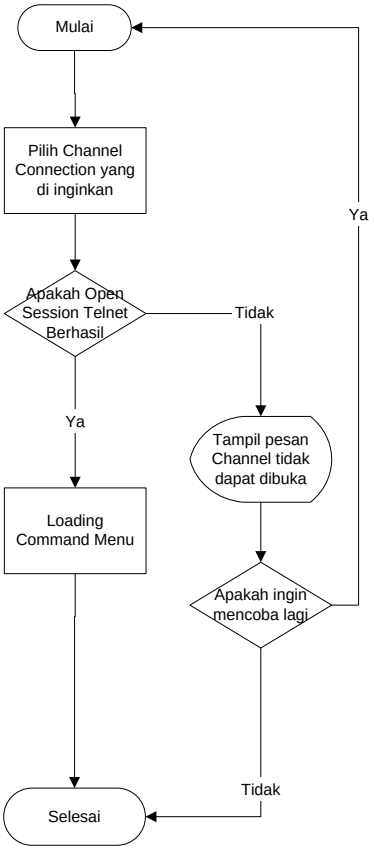
`kolom_1`='kolom_1.masukan', `kolom_2`='kolom_2.masukan',

`kolom_3`='kolom_3.masukan', `kolom_4`='kolom_4.masukan',

`kolom_n`='kolom_n.masukan'”

Munculkan kotak dialog (“Data berhasil diubah”)

3.4.7.5 Prosedur *Connection*



Gambar 3.31. Prosedur *Connection*

Algoritma prosedur *connection* :

Prosedur ini untuk mermbangun koneksi *telnet* ke perangkat dengan menggunakan *channel* koneksi yang sudah ada.

Deskripsi :

Baca *channel_id* dari data masukan *drop down* menu

Jika (*channel_id*=””){

Munculkan kotak dialog (“Silahkan pilih *channel*_id terlebih dahulu ”)

}

Jika Tidak

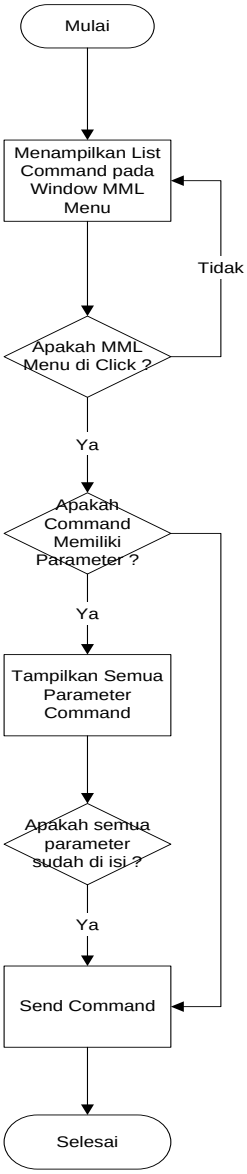
{

Jalankan sesi *telnet*

Aktifkan *command* menu

}

3.4.7.6 Prosedur *send command*



Gambar 3.32. Prosedur *Send Command*

Algoritma prosedur *send command* :

Prosedur ini untuk menunjukan proses eksekusi dan pengecekan *command* ke perangkat. Sebagai data masukan adalah data-data *command* beserta parameter-parameter nya.

Deskripsi :

Tampilkan data_*command* pada *mml command* menu

```
Jika command_menu_onclick()
{
    Tampilkan parameter command dari command yang terkait
    Jika ( parameter.tabel_command >= 1)
    {
        Baca param_1 dari data masukan
        Baca param_2 dari data masukan
        Baca param_3 dari data masukan
        Baca param_4 dari data masukan
        Baca param_n dari data masukan
        command_send=
        'command.tabel_command:'+param1.tabel_command='+'param_1.m
        asukan'+','+param2.tabel_command='+'param_2.masukan'+','param
        3.tabel_command='+'param_3.masukan'+','
        param4.tabel_command='+'param_4.masukan'+','
        param_n.tabel_command='+'param_n.masukan'+', '
        jika(param_1.masukan == "")
        {
            Munculkan kotak dialog ("Nilai parameter 1 harus diisi")
        }
        atau jika (param_2.masukan == "")
        {
            Munculkan kotak dialog ("Nilai parameter 2 harus diisi")
        }
        atau jika (param_3.masukan  == "")
        {
            Munculkan kotak dialog ("Nilai parameter 3 harus diisi")
        }
        atau jika (param_4.masukan == "")
        {
```

Munculkan kotak dialog ("Nilai parameter 4 harus diisi")

}

atau jika (param_n.masukan == "")

{

Munculkan kotak dialog ("Nilai parameter n harus diisi")

}

atau{

Lakukan *telnet* dan kirim *command_send*

}

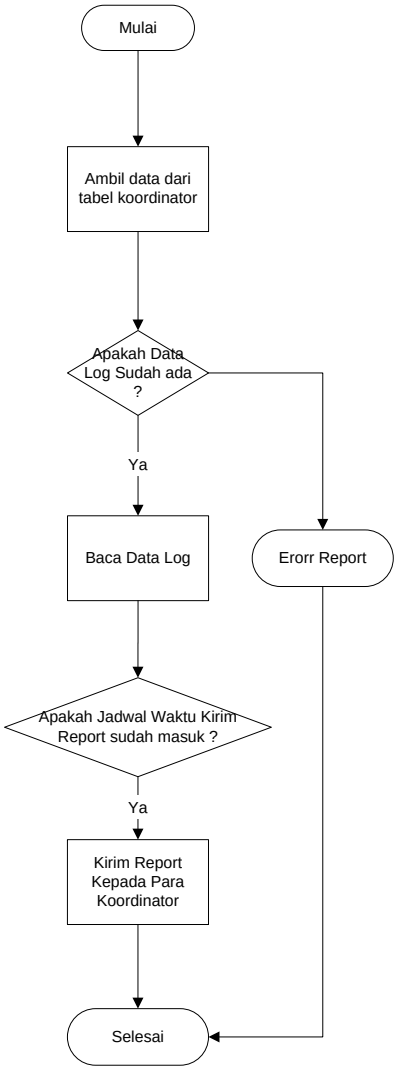
Jika tidak

{

Lakukan *telnet* dan kirim *command_send*

}

3.4.7.7 Prosedur *Reporting*



Gambar 3.33. Prosedur *Reporting*

Algoritma prosedur *report* :

Prosedur ini untuk menjalankan *reporting* data *logfile*. *file report log* untuk tiap-tiap *user* dikirimkan kepada koordinator setiap hari.

Deskripsi :

Baca data koordinator dari tabel_koordinator

Jika (alamat *directory log* = benar)

```
{
  Jika (directory bisa diakses = benar)
  {
    Jika ( file log bisa terbaca = benar)
    {
      Jika ( waktu == 02:00)
      {
        Kirim logfile ke koordinator via email
        }
      }
    }
  }
```