BAB 3

PEMBAHASAN

3.1 Analisis Sistem

Analisis sistem merupakan penguraian dari suatu sistem yang utuh kedalam bagian-bagian komponennya dengan maksud untu mengidentifikasikan dan mengevaluasi permasalahan -permasalahan, kesempatan-kesempatan hambatan-hambatan yang terjadi dan kebutuhan -kebutuhan yang diharapkan sehinnga dapat diusulkan perbaikan perbaikannya. Hal-hal yang dianalisis pada tahap analisi sistem adalah analisis masalah, analisis fungsional, analisis prosedur sistem yang sedang berjalan, analisis aliran informasi, analisis pengkodean, analisis basis *data*, dan analisis kebutuhan non – fungsional.

3.1.1 Analisis Masalah

1. Tidak terkontrolnya efektifitas dan efisiensi operator produksi.
2. Sering terjadinya masalah keterlambatan proses produksi yang diakibatkan tidak terkontrolnya progress produksi dan inefisiensi operator.
3. Dari keterlambatan proses produksi menyebabkan terjadinya penundaan pengiriman ke *customer* yang mengakibatkan komplain terhadap perusahaan.

3.1.2 Analisis Prosedur

Dalam melakukan pengamatan sistem yang sedang berjalan, diperlukan suatu perbaikan proses pelaporan produksi, pengecekan progress dan laporan yang masih berjalan secara manual, dimana *supervisor* mencatat secara manual laporan yang diterima dari operator. Hal ini tentunya kurang efektif, karena dapat menyita waktu dan tenaga yang cukup banyak. Berikut gambaran proses manual dari prosedur yang sedang berjalan:

3.1.3 Analisis Kebutuhan Non-Fungsional

Analisis kebutuhan non fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukkan yang diperlukan sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukkan sehingga menghasilkan suatu keluaran yang diinginkan.

1. Analisis Perangkat Keras

Adapun spesifikasi perangkat keras yang dibutuhkan untuk sistem yang akan dibangun sebagai berikut :

1. *Processor* Dual Core CPU 2.93 GHz
2. RAM 1 GB
3. Free Hard Disk 40GB
4. Monitor 14.1’’
5. Keyboard dan mouse

Perangkat keras yang ada saat ini di perusahaan ***Pt. Technomed Asia*** sudah memenuhi standard untuk menjalankan sistem yang dibangun.

2. Analisi Perangkat Lunak

Adapun spesifikasi perangkat lunak yang digunakan adalah Windows 7 sebagai OS, *Netbeans*-6.9.1-ml-windows, Jdk-6u25-windows-i586, dan *MySQL* sebagai *Database* Perangkat lunak yang ada di perusahaan ***Pt. Technomed Asia*** saat ini sudah memenuhi standar untuk menjalankan sistem yang dibangun.

3. Analisis *User*

Setelah melakukan pengumpulan *data* dan observasi maka yang menggunakan aplikasi ini yaitu *User* / petugas yang bekerja di

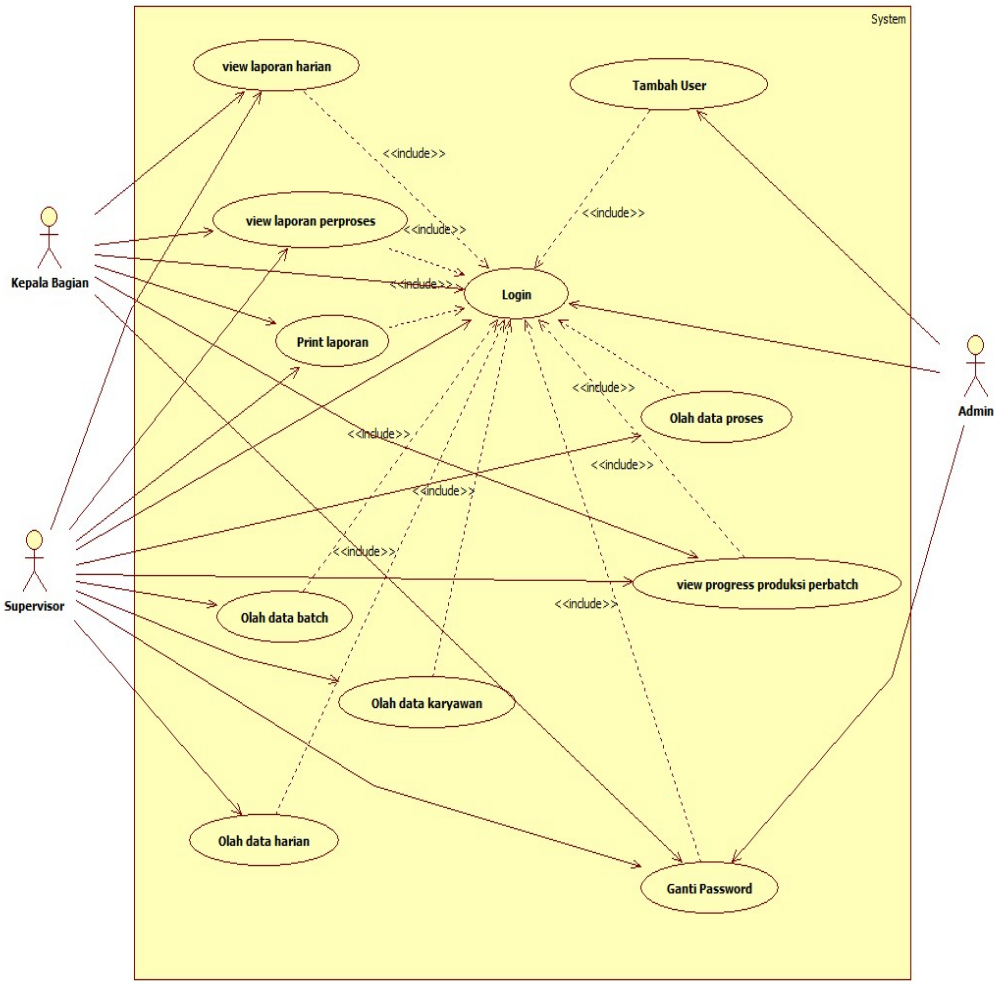
Pt. *Technomed Asia* untuk menggunakan aplikasi ini minimal bisa mengoperasikan komputer.

3.1.4 Analisis Kebutuhan Fungsional

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

3.1.4.1 *Use Case* Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.



Gambar 3.1.1 *Use Case* Diagram

Fungsi ini untuk melakukan autentifikasi terhadap *User* atau *Admin* yang menggunakan aplikasi ini untuk memastikan bahwa orang yang bersangkutan benar-benar *user* yang memiliki hak akses terhadap aplikasi ini.

1. **Skenario *Login***

Identifikasi	
Nomor	1
Nama	*Login*
Tujuan	Memastikan hak akses *user*
Deskripsi	Aplikasi menerima *data password* dan *username* dan

	berdasarkan *data* tersebut memastikan apakah *user* memiliki hak akses terhadap sistem.
Type	Form
Aktor	*Supervisor, admin* dan kepala bagian
Skenario Utama	
Kondisi Awal	*User* belum *Login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Supervisor, admin dan kabag memasukkan *Password* dan *Username*	
	Mengirim *data user* ke aplikasi
	Memeriksa respon aplikasi, jika OK tampilkan menu sesuai hak akses
Skenario Alternatif – Autentifikasi gagal	
Aksi Aktor	**Reaksi Sistem**
user memasukkan lagi *Username* dan *password*	
	Kembali ke *login*
	Hak akses *user* pada menu sesuai hak akses masing - masing

Tabel 3.1.1 Skenario *Login*

2. Skenario *view* laporan harian

Identifikasi	
Nomor	2
Nama	*View* laporan harian
Tujuan	Menampilkan *data* harian yang telah di*input*
Deskripsi	Aplikasi menampilkan *data* harian berdasarkan nama dan periode waktu tertentu yang di*input*kan oleh *user*.
Type	Form

Aktor	Kepala bagian dan super visor
Skenario Utama	
Kondisi Awal	Kepala Bagian dan *supervisor* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Kepala Bagian dan *supervisor* memasukkan nama karyawan dan periode waktu.	
	Menampilkan *data* hasil produksi karyawan berdasarakan periode waktu tertentu yang di*input*kan oleh kabag atau super visor.
Skenario Alternatif – *Data* yang dicari tidak ada	
Aksi Aktor	**Reaksi Sistem**
Kepala bagian memasukkan lagi nama dan periode waktu	
	Kembali ke form
	Menampilkan *data* hasil produksi karyawan berdasarakan periode waktu tertentu yang di*input*kan oleh kabag atau super visor.

Tabel 3.1.2 Skenario *View* Laporan Harian

3. Skenario *view* laporan perproses

Identifikasi	
Nomor	3
Nama	*View* laporan perproses
Tujuan	Menampilkan *data* harian yang telah di*input*kan berdasarkan proses produksi.

Deskripsi	Aplikasi menampilkan *data* harian berdasarkan proses dan periode waktu tertentu yang di*input*kan oleh *user*.
Type	Form
Aktor	Kepala bagian dan Super visor
Skenario Utama	
Kondisi Awal	Kepala bagian dan Super visor sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Kepala Bagian dan *supervisor* memasukkan jenis proses dan periode waktu.	
	Menampilkan *data* hasil produksi karyawan berdasarakan jenis proses dan periode waktu tertentu yang di*input*kan oleh kabag atau supervisor.
Skenario Alternatif – *data* yang dicari tidak ada	
Aksi Aktor	**Reaksi Sistem**
Kepala bagian memasukkan lagi jenis proses dan periode waktu	
	Menampilkan *data* hasil produksi karyawan berdasarakan jenis proses dan periode waktu tertentu yang di*input*kan oleh kabag atau supervisor.

Tabel 3.1.3 Skenario *view laporan per proses*

4. Skenario *view* progress produksi per*batch*

Identifikasi	
Nomor	4

Nama	*View* progress produksi per*batch*
Tujuan	Menampilkan *data* progress produksi yang telah dilakukan berdasarkan no *batch* dan proses .
Deskripsi	Aplikasi menampilkan *data batch* berdasarkan *batch* dan jenis proses yang di*input*kan oleh *user*.
Type	Form
Aktor	Kepala bagian dan Super visor
Skenario Utama	
Kondisi Awal	Kepala bagian dan Super visor sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Kepala Bagian dan *supervisor* memasukkan *batch* no dan jenis proses.	
	Menampilkan *data batch* berdasarakan proses yang di*input*kan oleh kabag atau super visor.
Skenario Alternatif – *data* yang dicari tidak ada	
Aksi Aktor	**Reaksi Sistem**
Kepala bagian memasukkan lagi *batch* no dan jenis proses	
	Menampilkan *data batch* berdasarakan proses yang di*input*kan oleh kabag atau super visor.

Tabel 3.1.4 Skenario view progress produksi per*batch*

5. **Skenario *view summary* efisiensi operator**

Identifikasi	
Nomor	5
Nama	*View summary* efisiensi operator

Tujuan	Menampilkan *data* efisiensi operator .
Deskripsi	Aplikasi menampilkan efisiensi operator berdasarkan periode waktu yang di*input*kan oleh *user*.
Type	Form
Aktor	Kepala bagian dan Super visor
Skenario Utama	
Kondisi Awal	Kepala bagian dan Super visor sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Kepala Bagian dan *supervisor* memasukkan nama dan periode waktu.	
	Menampilkan *data* efisiensi karyawan berdasarakan periode waktu yang di*input*kan oleh kabag atau super visor.
Skenario Alternatif – *data* yang dicari tidak ada	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *view summary* efisiensi operator.
Kepala bagian memasukkan lagi nama dan periode waktu.	
	Data yang dicari sesuai dengan parameter pencarian ditampilkan kedalam tabel dan dicetak atau *export* kedalam bentuk pdf

Tabel 3.1.5 Skenario View summary efisiensi operator

6. Skenario olah *data* karyawan

Identifikasi	
Nomor	6

Nama	Olah *data* karyawan
Tujuan	Mengolah *data* karyawan .
Deskripsi	Aplikasi menerima *input*an *data* karyawan kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Supervisor*
Skenario Utama	
Kondisi Awal	*Supervisor* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Supervisor melakukan pengolahan *data* karyawan	
	Menyimpan *data* hasil pengolahan kedalam *database*.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *input data* karyawan.
supervisor memasukkan lagi *data* karyawan.	
	Aplikasi menerima *input*an *data* karyawan kemudian menyimpannya kedalam *database*.

Tabel 3.1.6 Skeanrio olah *data* karyawan

7. Skenario olah *data* laporan harian

Identifikasi	
Nomor	7
Nama	Olah *data* laporan harian
Tujuan	Mengolah *data* laporan harian .
Deskripsi	Aplikasi menerima *input*an *data* laporan harian

	kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Supervisor*
Skenario Utama	
Kondisi Awal	*Supervisor* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Supervisor melakukan pengolahan *data* karyawan	
	Menyimpan *data* hasil pengolahan kedalam *database*.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *input data* karyawan.
supervisor memasukkan lagi *data* karyawan.	
	Menyimpan *data* hasil pengolahan kedalam *database*.

Tabel 3.1.7 Skenario olah *data* laporan harian

8. Skenario olah *data* jenis proses

Identifikasi	
Nomor	8
Nama	Olah *data* proses
Tujuan	Mengolah *data* jenis proses .
Deskripsi	Aplikasi menerima *input*an *data* jenis proses kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Supervisor*

Skenario Utama	
Kondisi Awal	*Supervisor* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Supervisor melakukan pengolahan *data* proses	
	Menyimpan *data* hasil pengolahan kedalam *database*.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *input data* karyawan.
supervisor memasukkan lagi *data* karyawan.	
	Menyimpan *data* hasil pengolahan kedalam *database*.

Tabel 3.1.8 Skenario olah *data* proses

9. Skenario olah *data batch*

Identifikasi	
Nomor	9
Nama	Olah *data batch*
Tujuan	Mengolah *data batch* nr.
Deskripsi	Aplikasi menerima *input*an *data batch* nr kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Supervisor*
Skenario Utama	
Kondisi Awal	*Supervisor* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Supervisor melakukan pengolahan *data batch*	
	Menyimpan *data* hasil pengolahan kedalam

	database.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *input data* karyawan.
supervisor memasukkan lagi *data* karyawan.	
	Menyimpan *data* hasil pengolahan kedalam *database*.

Tabel 3.1.9 Skenario olah *data batch*

10. Skenario tambah *user*

Identifikasi	
Nomor	10
Nama	Manage *user*
Tujuan	Mengolah *data user*
Deskripsi	Aplikasi menerima *input*an *data user* kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Admin*
Skenario Utama	
Kondisi Awal	*Admin* sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Admin melakukan pengolahan *data batch*.	
	Menyimpan *data* hasil pengolahan kedalam *database*.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form *manage user*.

Admin memasukkan lagi *data user*.	
	Menyimpan *data* hasil pengolahan kedalam *database*.

Tabel 3.1.10 Skenario tambah *user*

11. Skenario ganti *password*

Identifikasi	
Nomor	11
Nama	Ganti *password*
Tujuan	Mengganti *password* dan *username*
Deskripsi	Aplikasi menerima *input*an *data username* dan *password* kemudian menyimpannya kedalam *database*.
Type	Form
Aktor	*Admin, supervisor* dan kepala bagian
Skenario Utama	
Kondisi Awal	*Admin, supervisor* dan kepala bagian sudah *login* kedalam sistem
Aksi Aktor	**Reaksi Sistem**
Admin, supervisor dan kepala bagian mengubah *username* dan *password*.	
	Menyimpan *data* hasil perubahan kedalam *database*.
Skenario Alternatif – *valid*asi *data* gagal	
Aksi Aktor	**Reaksi Sistem**
	Kembali ke form ganti *password*.
Admin, supervisor dan kepala bagian memasukkan lagi *username* dan	

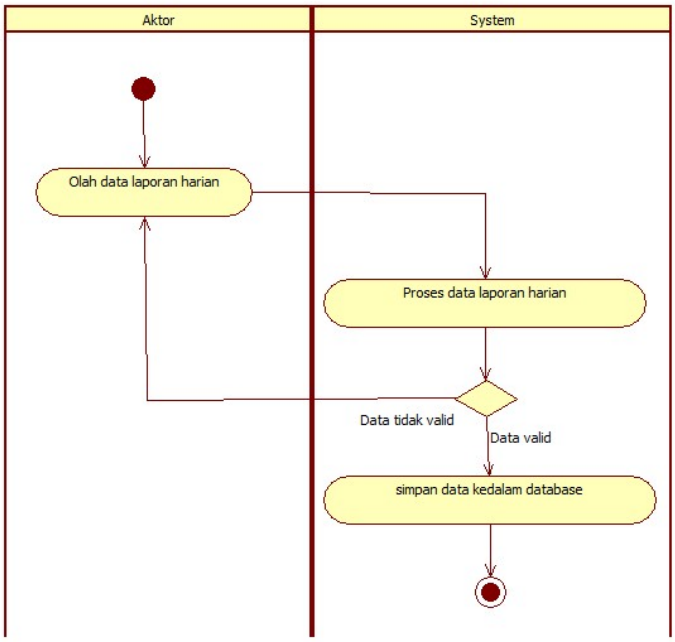
password.	
	Menyimpan *data* hasil perubahan kedalam *database*.

Tabel 3.1.11 Skenario ganti *password*

3.1.4.2 *Activity* Diagram

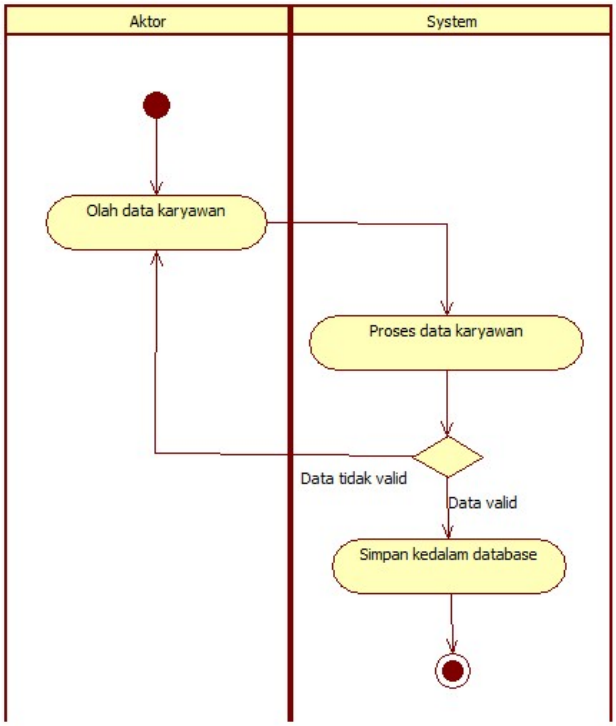
Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi oleh selesainya *state* sebelumnya.

a. *Activity* diagram olah *data* harian



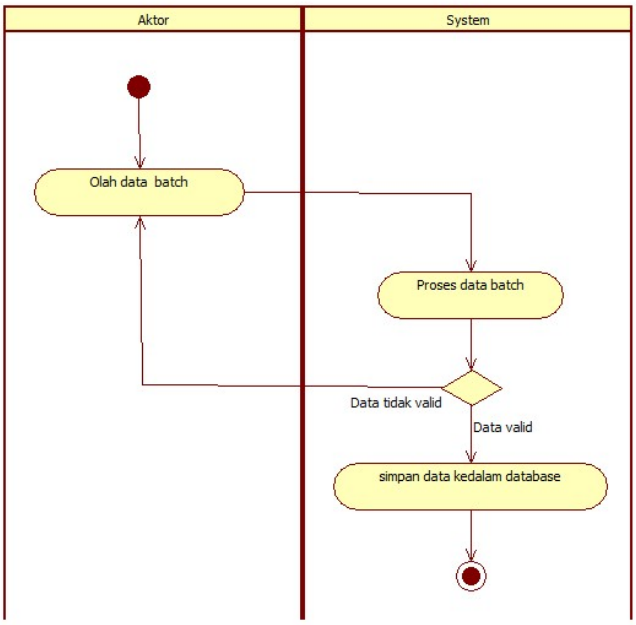
Gambar 3.1.2 *Activity* Diagram olah *data* harian

b. *Activity* diagram olah *data* karyawan



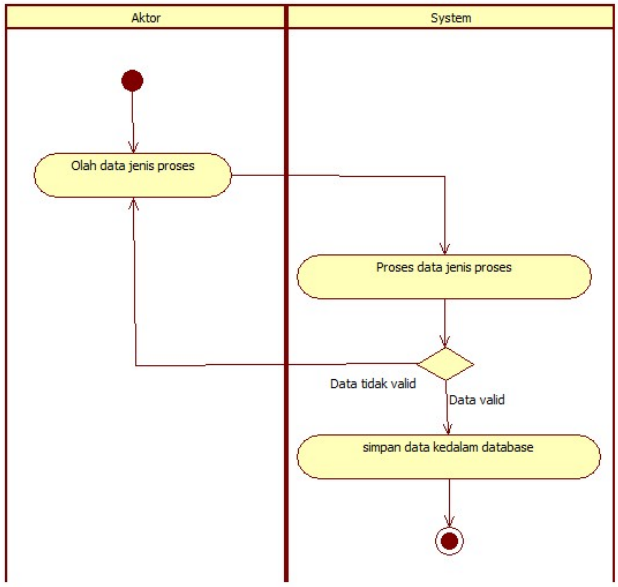
Gambar 3.1.3 *Activity* diagram olah *data* karyawan

c. *Activity* diagram olah *data batch*



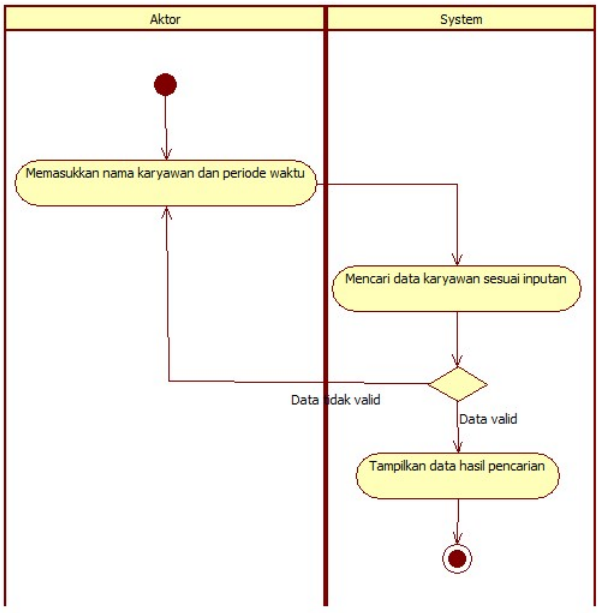
Gambar 3.1.4 *Activity* diagram olah *data batch*

d. *Activity* diagram olah *data* proses



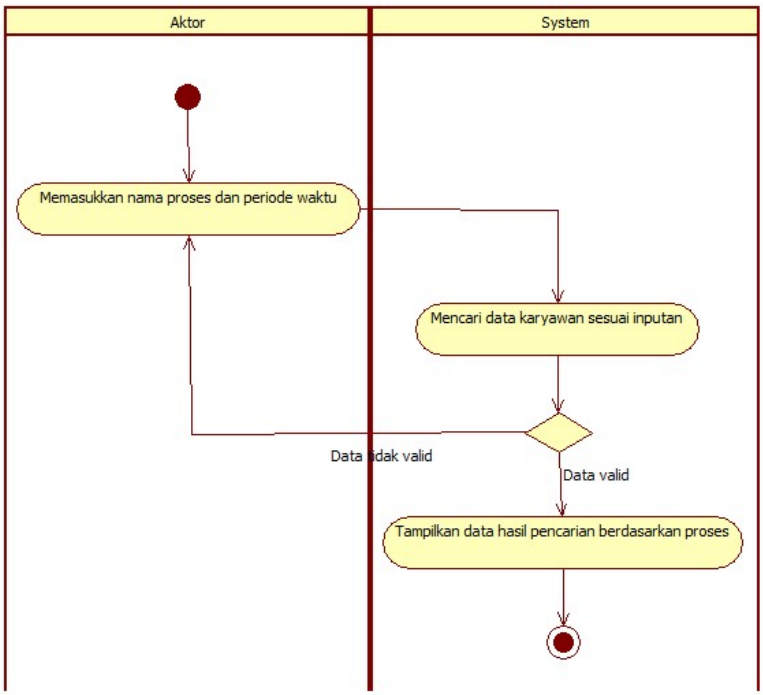
Gambar 3.1.5 *Activity* olah *data* proses

e. *Activity* diagram view laporan harian



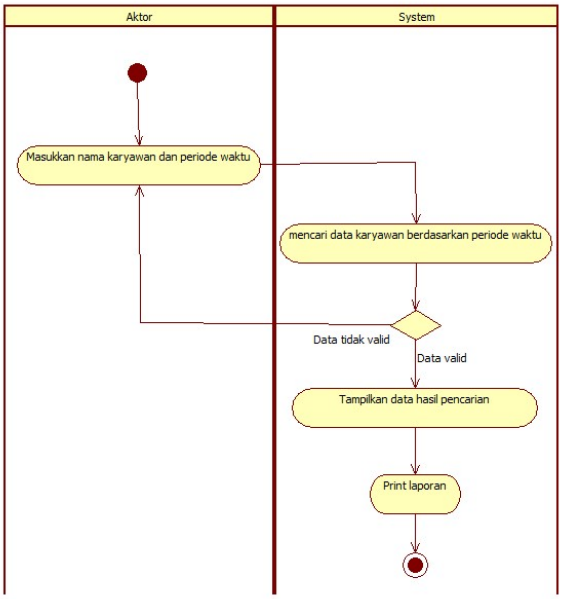
Gambar 3.1.6 *Activity* diagram view laporan harian

f. *Activity* diagram view laporan harian perproses



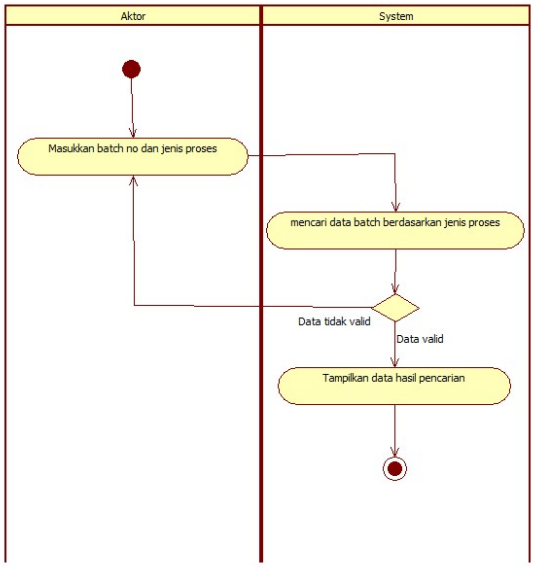
Gambar 3.1.7 *Activity* diagram view laporan harian perproses

g. *Activity* diagram print laporan



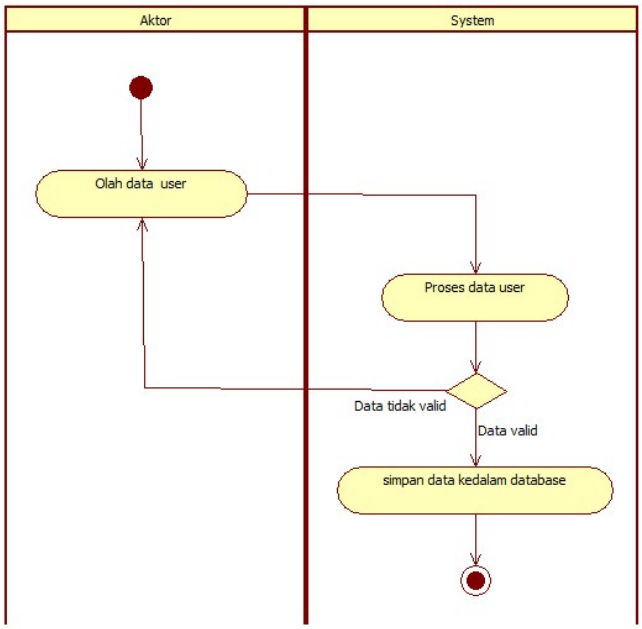
Gambar 3.1.8 Activity diagram print laporan

h. *Activity* diagram view progress produksi per*batch*



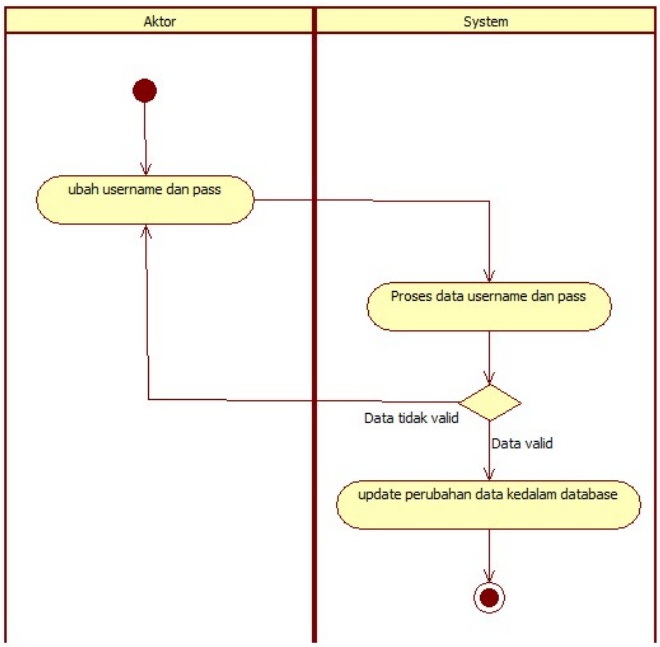
Gambar 3.1.9 *Activity* diagram view progress produksi per*batch*

i. *Activity* diagram tambah *user*



Gambar 3.1.10 *Activity* diagram tambah *user*

j. *Activity* diagram ganti *password*

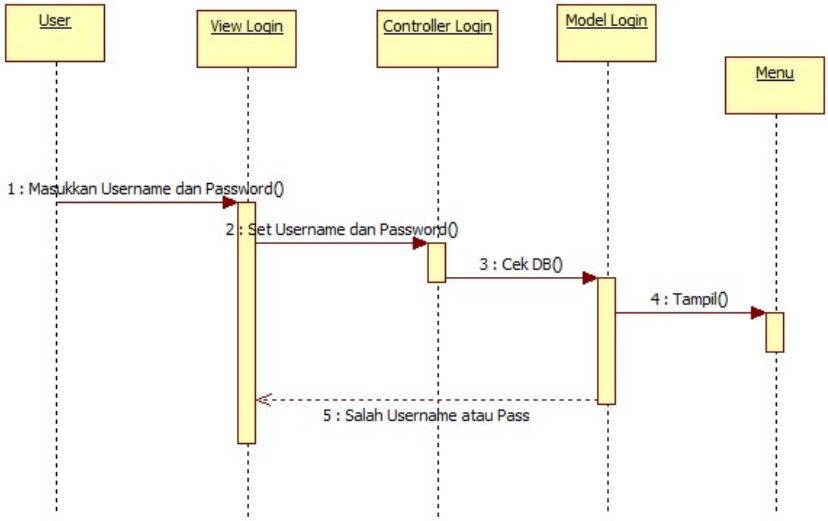


Gambar 3.1.11 *Activity* diagram ganti *password*

3.1.4.3 *Sequence* Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

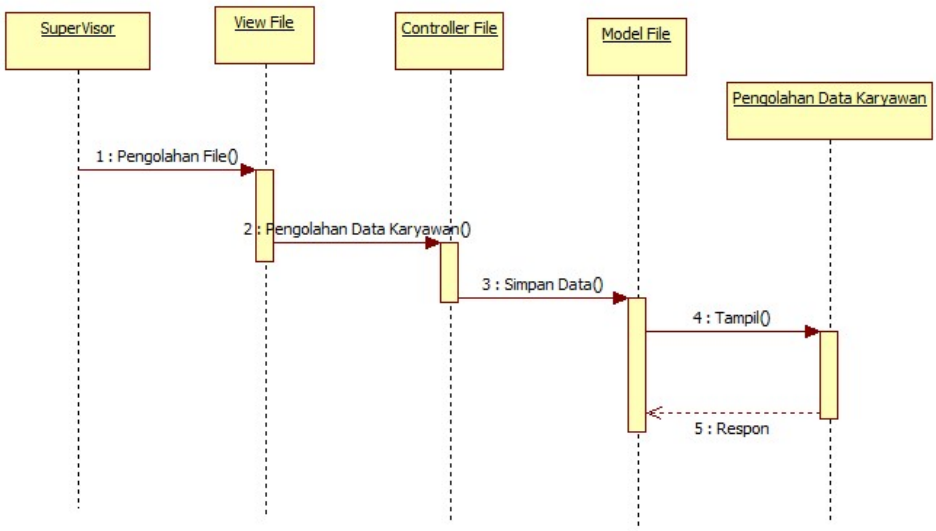
b. Login



Gambar 3.1.12 *Sequence* Diagram

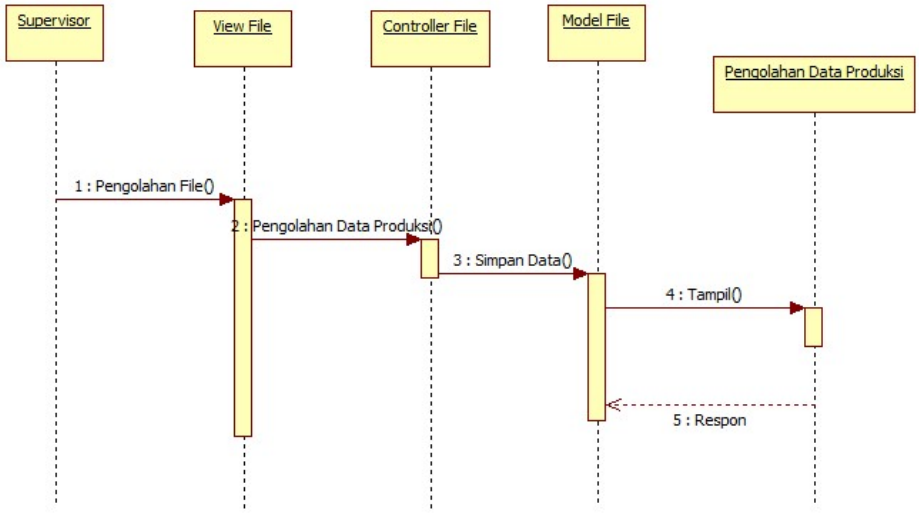
c. File

1. Akses *File* ke pengolahan *Data* Karyawan



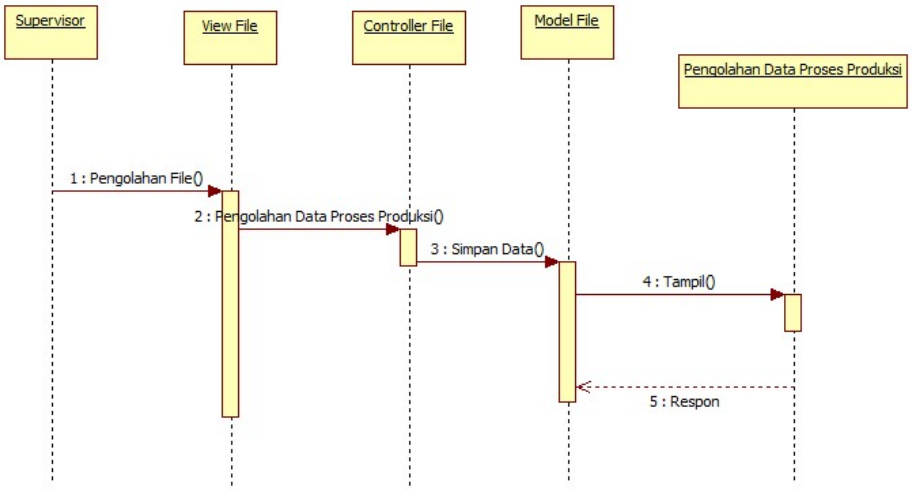
Gambar 3.1.13 *Sequence* Diagram Pengolahan *Data* karyawan

2. Akses *File* ke pengolahan *Data* Produksi



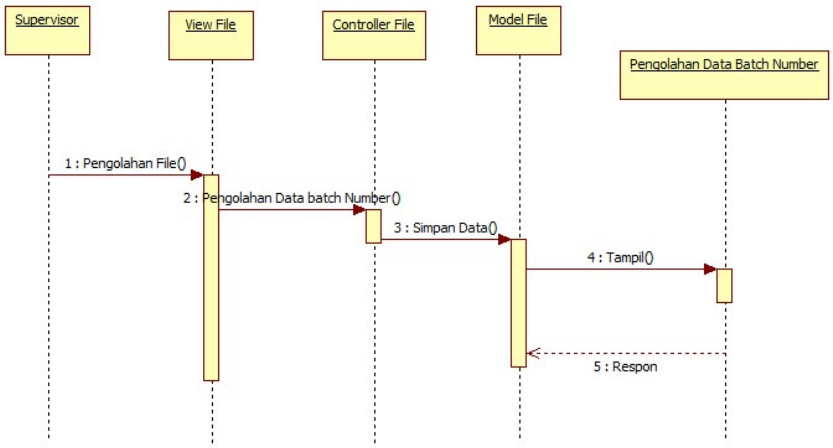
Gambar 3.1.14 *Sequence* Diagram Pengolahan *data* Produksi

3. Akses *File* ke pengolahan *Data* Proses Produksi



Gambar 3.1.15 *Sequence* Diagram Pengolahan *Data* Proses Produksi

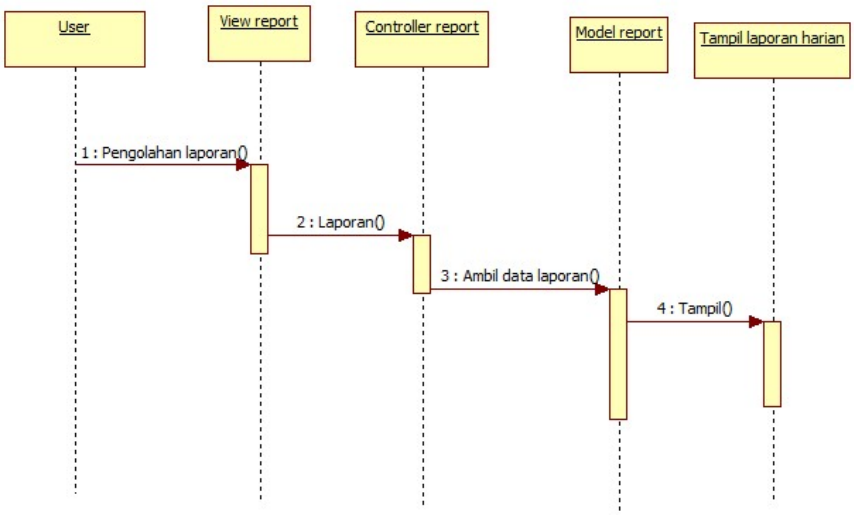
4. Akses *File* ke pengolahan *data Batch* number



Gambar 3.1.16 *Sequence* Diagram pengolahan *Data Batch* Number

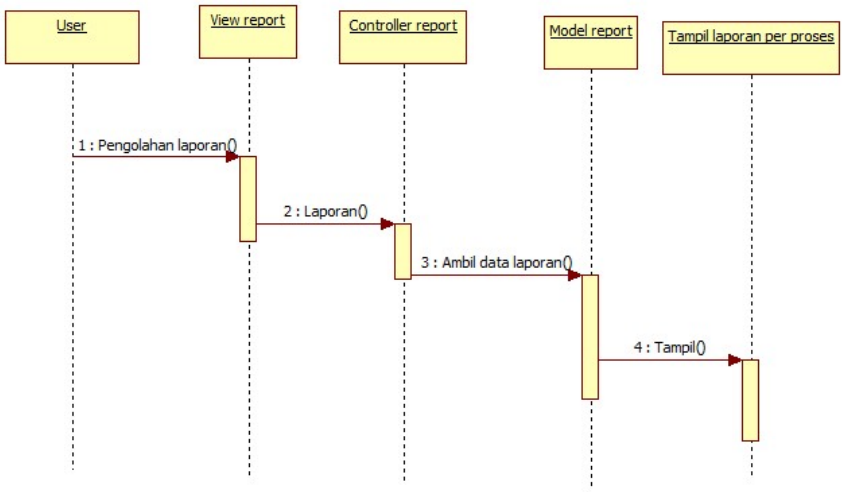
d. Report

1. *View* laporan harian



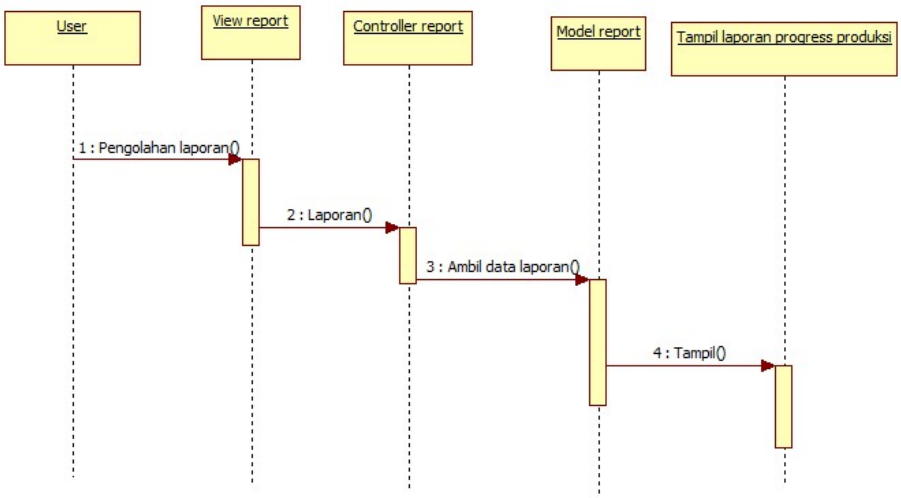
Gambar 3.1.17 *Sequence* Diagram *View* Laporan Harian

2. *View* laporan harian per proses



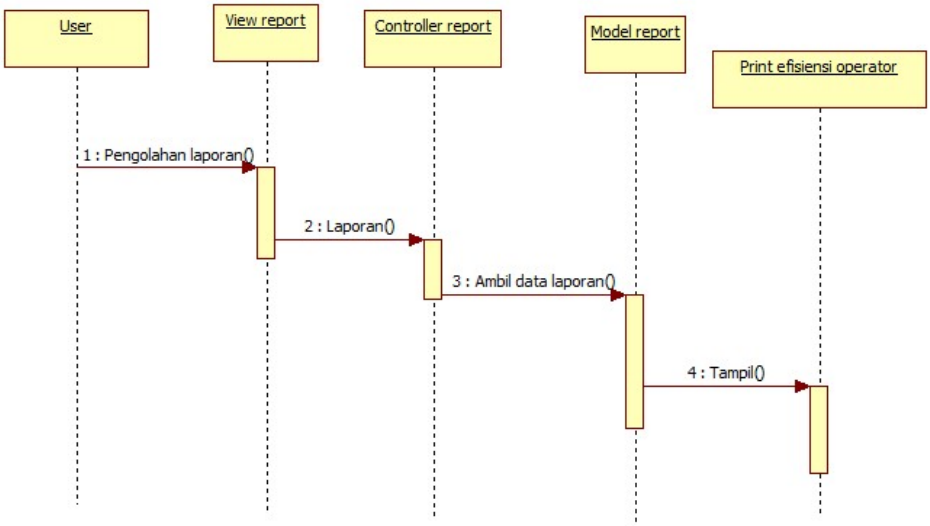
Gambar 3.1.18 *Sequence* Diagram *View* Laporan Harian Perproses

3. *View* progress produksi



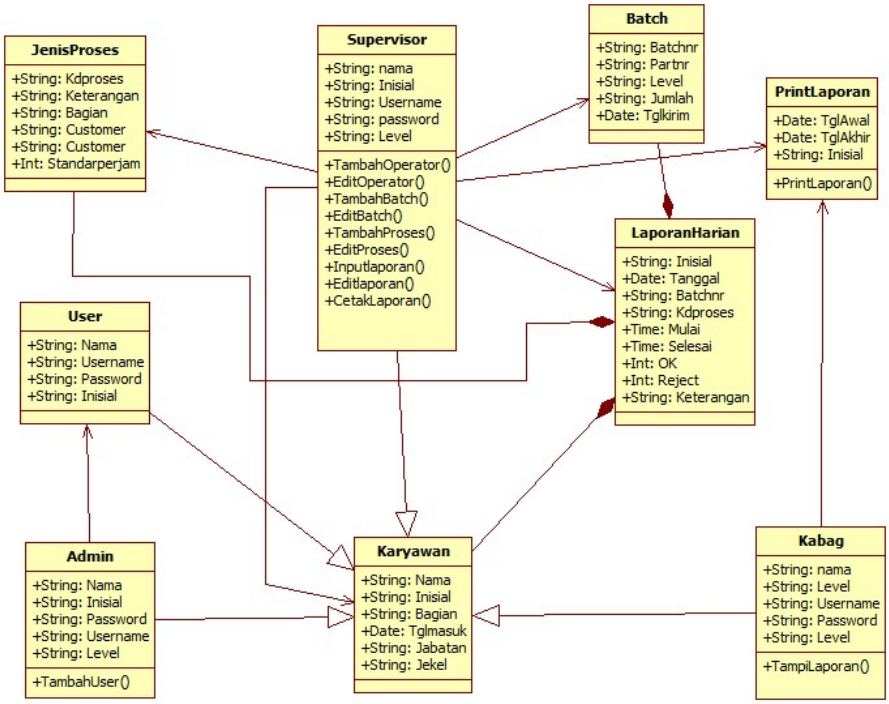
Gambar 3.1.19 *Sequence* Diagram *View* Progress Produksi

4. Print efisiensi operator



Gambar 3.1.20 *Sequence* Diagram Print Efisiensi Operator

3.1.4.4 *Class* Diagram



Gambar 3.1.21 *Class* Diagram

3.2 **Perancangan Sistem**

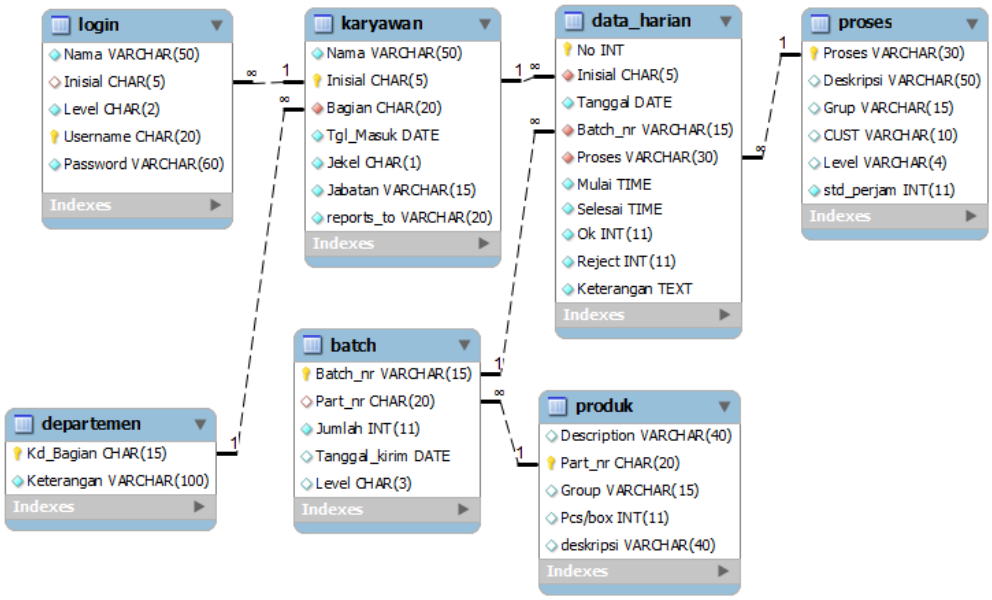
Perancangan akan dimulai setelah tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya. Perancangan digambarkan sebagai proses multi-langkah dimana representasi perancangan *data*, skema relasi, struktur *file*, perancangan antar muka, dan perancangan prosedural dari persyaratan informasi.

3.2.1 Perancangan *Data*

Perancangan *data* atau lebih dikenal dengan perancangan basis *data* yaitu menciptakan atau merancang *data* yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah skema relasi dan struktur *file*. Dari dua hasil tersebut, implementasi basis *data* akan bisa dikerjakan.

a. Skema Relasi

Model *data* relasional merupakan model *data* di mana hubungan antar *data,*arti *data* dan batasannya dijelaskan dengan baris dan kolom. Secara formal, kesemuanya itu digambarkan ke dalam skema relasi dan diagram skema. Adapun skema relasi yang terdapat didalam aplikasi ini adalah sebagai berikut:



Gambar 3.2.1 Skema Relasi

1. ***Batch***

   {#*Batch*_nr, Part_nr, jumlah, Tanggal_kirim, level}

2. ***Data*_Harian**

   {#No, Inisial, Tanggal, *Batch*, Proses, Mulai, Selesai, Ok, Reject, Keterangan}

3. **Karyawan**

   { Nama, #Inisial, Bagian, Tgl_Masuk, Jekel, Jabatan, *Report*s_to}

4. ***Login***

   {#*Username*, inisial, Nama, *Password*, Level}

5. **Produk**

   {Description,# Part, Group, Pcs/box, Deskripsi}

6. **Proses**

   {#Proses, Deskripsi, Grup, Cust, Level, Std_Perjam}

7. **Departemen**

   {#Kd_bagian, Keterangan**}**

b. Struktur *File*

Struktur *file* tempat penyimpanan *data* pada Aplikasi ini adalah sebagai berikut:

1. Tabel *Batch*

Field	Type *Data*	Panjang	Null	Key
*Batch*_nr	VARCHAR	15	No	PK
Part_Nr	CHAR	20	Yes	-
Jumlah	INT	11	No	-
Tanggal_kirim	DATE		Yes	-
Level	CHAR	3	Yes	-

Tabel 3.2.1 Struktur Tabel *batch*

Tabel *Batch*

Nama *File*: *Batch*

2. Tabel *Data* Harian

Field	Type *Data*	Panjang	Null	Key
No	INT	11		PK
Inisial	CHAR	5		FK
Tanggal	Date			-
Batch	VARCHAR	15		FK
Proses	VARCHAR	30		FK
Mulai	TIME			-
Selesai	TIME			-
OK	INT	11		-

Reject	INT	11		-
Keterangan	TEXT			-

Tabel 3.2.2 Struktur Tabel *Data* Harian

Tabel *Data*_Harian

Nama *File*: *Data*_Harian

3. Tabel Karyawan

Field	Type *Data*	Panjang	Null	Key
Nama	VARCHAR	50		-
Inisial	CHAR	5		UQ
Bagian	CHAR	20		-
Tgl_Masuk	DATE			-
Jekel	CHAR	1		-
Jabatan	VARCHAR	15		-
*Report*s_to	VARCHAR	20		-

Tabel 3.2.3 Struktur Tabel Karyawan

Tabel Karyawan

Nama *File*: Karyawan

4. Tabel *Login*

Field	Type *Data*	Panjang	Null	Key
Nama	VARCHAR	50		-
Inisial				FK
Username	CHAR	20		PK
Password	VARCHAR	60		-
Level	CHAR	2		-

Tabel 3.2.4 Struktur Tabel *Login*

Tabel *Login*

Nama *File*: *Login*

5. Tabel Proses

Field	Type *Data*	Panjang	Null	Key
Proses	VARCHAR	30		PK
Deskripsi	VARCHAR	50		-
Grup	VARCHAR	15		-
Cust	VARCHAR	10		-
Level	VARCHAR	4		-
Std_perjam	INT	11		-

Tabel 3.2.5 Struktur Tabel Proses

Tabel Proses

Nama *File*: Proses

6. Tabel Produk

Field	Type *Data*	Panjang	Null	Key
Description	VARCHAR	30		-
Part_nr	VARCHAR	50		PK
Group	VARCHAR	15		-
Pcs/box	VARCHAR	10		-
Deskripsi	VARCHAR	4		-

Tabel 3.2.6 Struktur Tabel produk

7. Tabel Departemen

Field	Type *Data*	Panjang	Null	Key
Kd_bagian	CHAR	15		PK
keterangan	VARCHAR	100		-

Tabel 3.2.7 Struktur Tabel Departemen

3.2.2 Perancangan Antar Muka

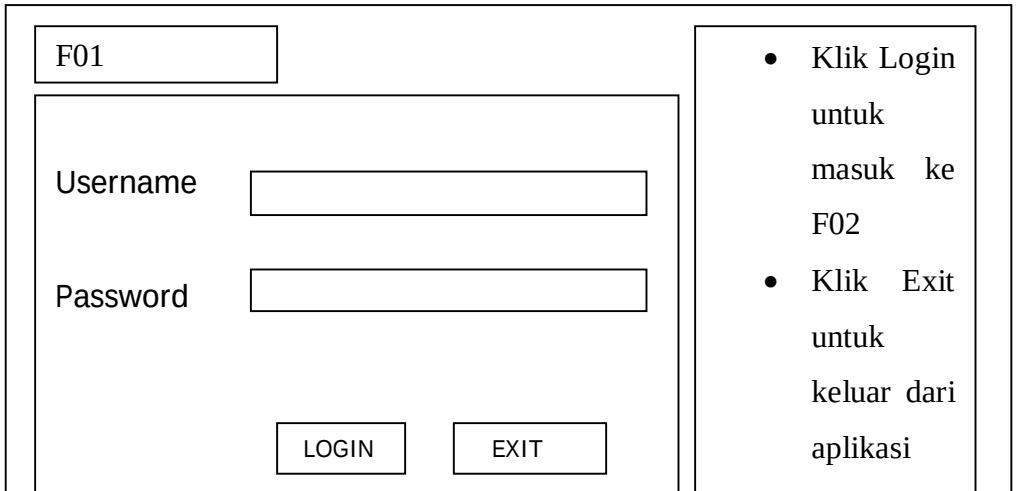
Perancangan antar muka dilakukan untuk mempermudah dalam mengimplementasikan sistem yang akan dibangun sebagai tempat tampilan *data*.

3.2.2.1 Perancangan *Form*

Dalam perancangan *form* di sini dilakukan perancangan baik terhadap *form input* dan *form output*. Adapun *form-form* yang terdapat dalam Aplikasi ini adalah sebagai berikut:

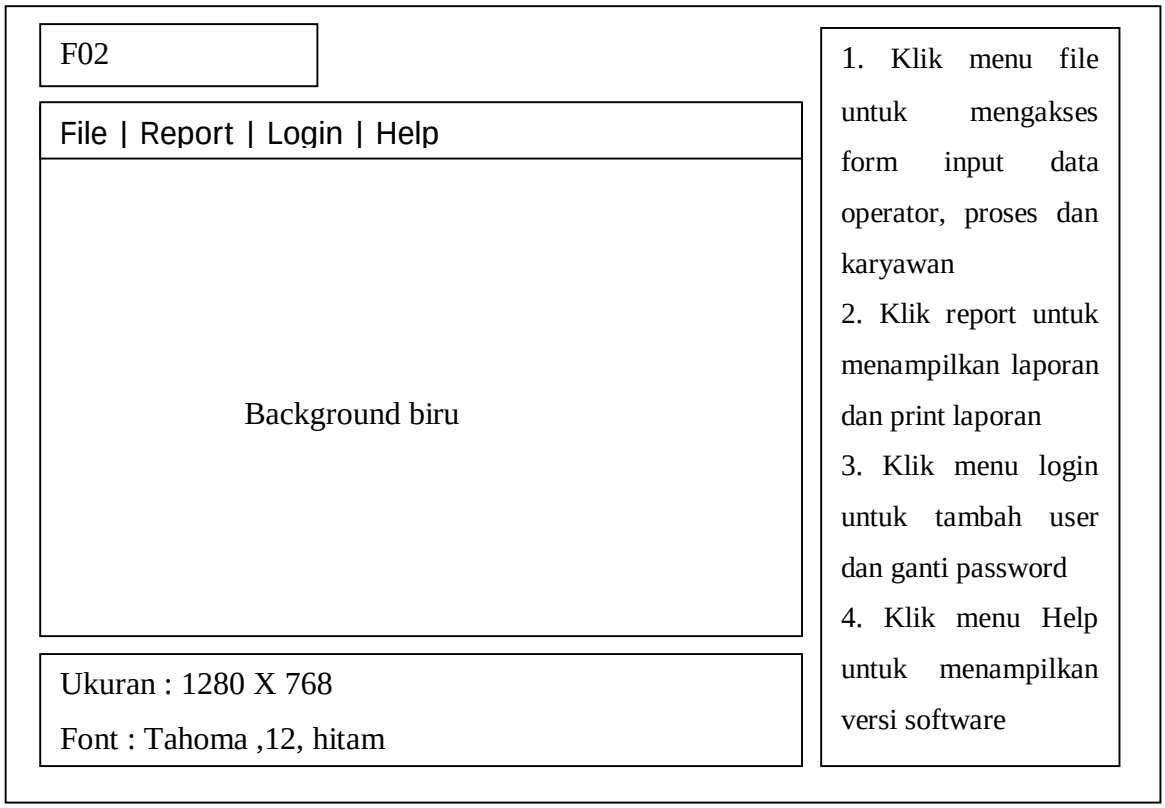
1. Perancangan antar muka *Login*

Gambar di bawah ini merupakan perancangan tampilan awal sistem sekaligus tampilan *Login* yang berfungsi untuk mendapatkan hak akses bagi *user* yang akan mengolah aplikasi ini sesuai dengan hak aksesnya.



2. Perancangan antar muka Menu Utama

Di bawah ini adalah perancangan antar muka menu utama yang berfungsi sebagai menu untuk petugas melakukan pengolahan *data* sesuai dengan hak akses masing –masing.

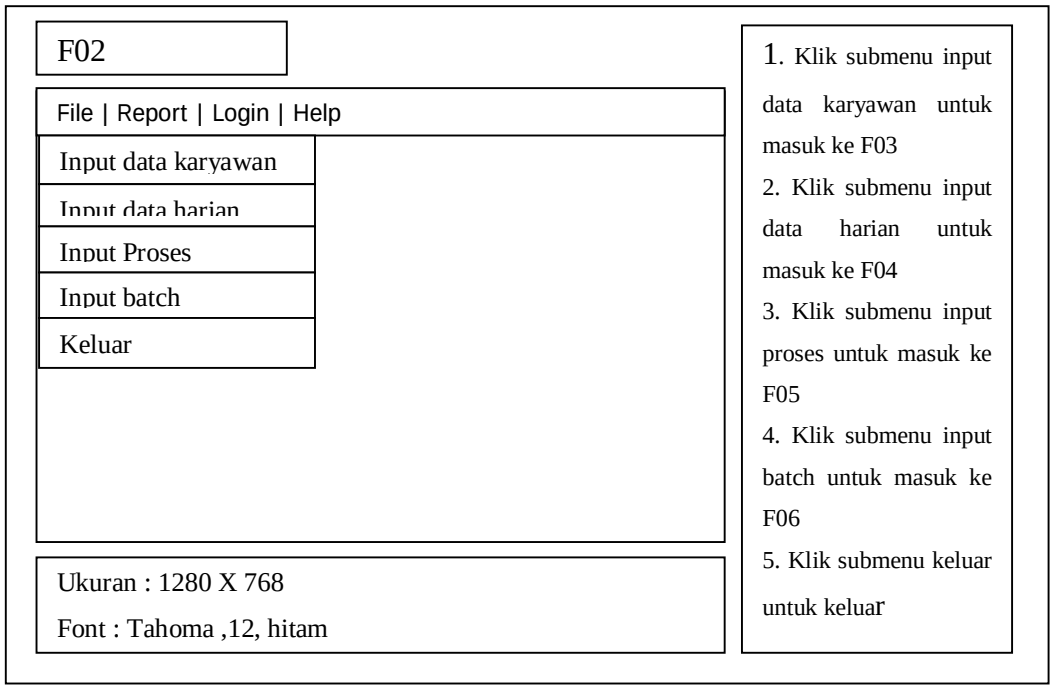


Gambar 3.2.3 Perancangan Antar Muka Menu Utama

4. Perancangan antar muka *file*

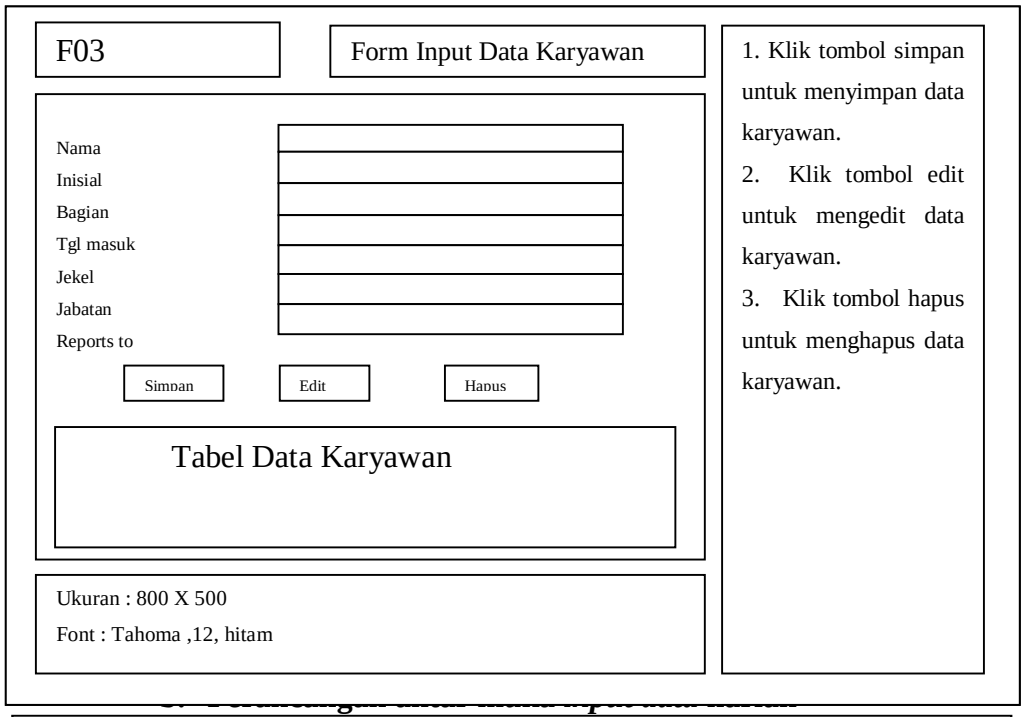
Gambar di bawah ini merupakan perancangan antar muka menu *File* yang berfungsi sebagai menu untuk *user* melakukan pengolahan *data* karyawan, *data* laporan harian, *data* jenis proses, *data batch* dan keluar dari aplikasi.

Akses terhadap menu *file* dibatasi hanya untuk *user* dengan tingkat akses '2'. Hal ini dimaksudkan agar *data* terlindungi dari campur tangan pihak lain yang tidak berwenang untuk melakukan perubahan.

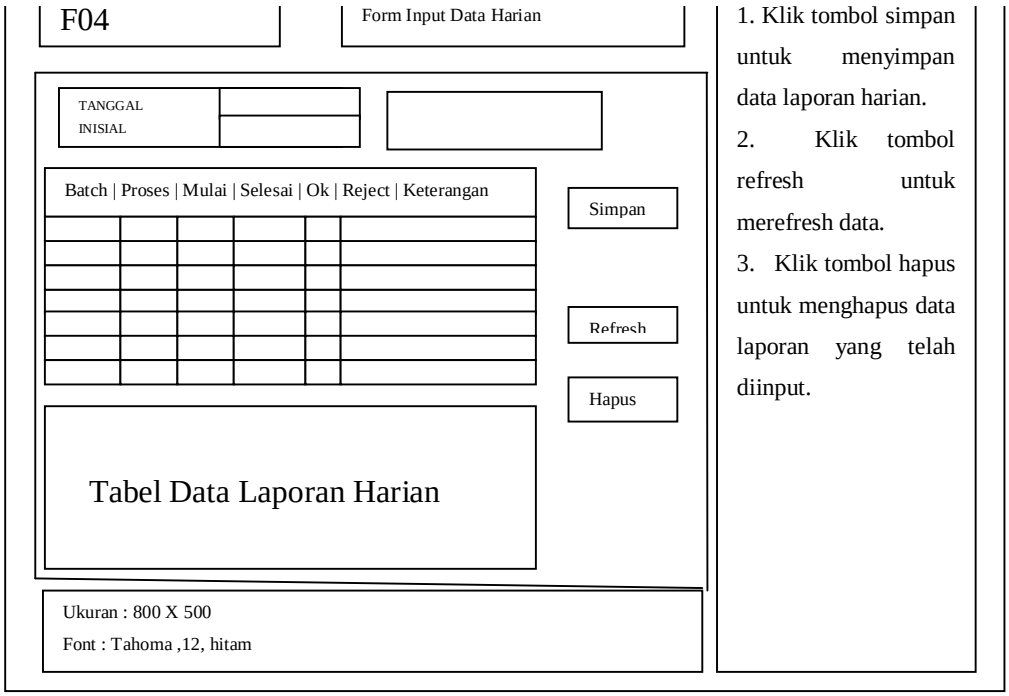


Gambar 3.2.4 Perancangan Antar Muka Menu File

a. Perancangan antar muka *input data* karyawan

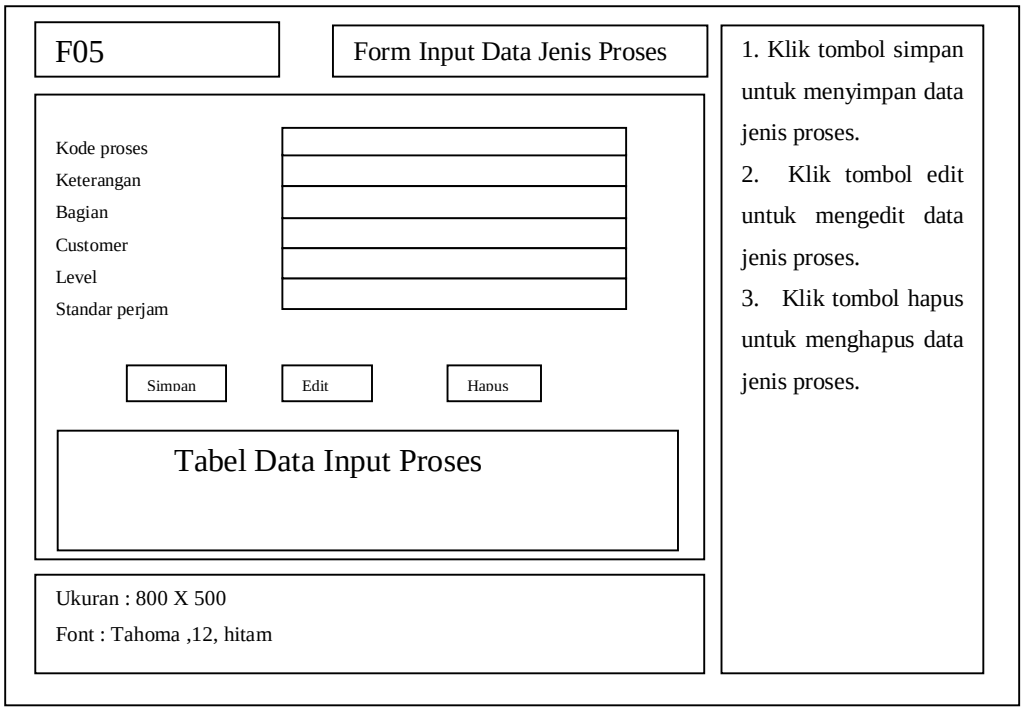


Gambar 3.2.5 Perancangan Antar Muka Input Data Karyawan



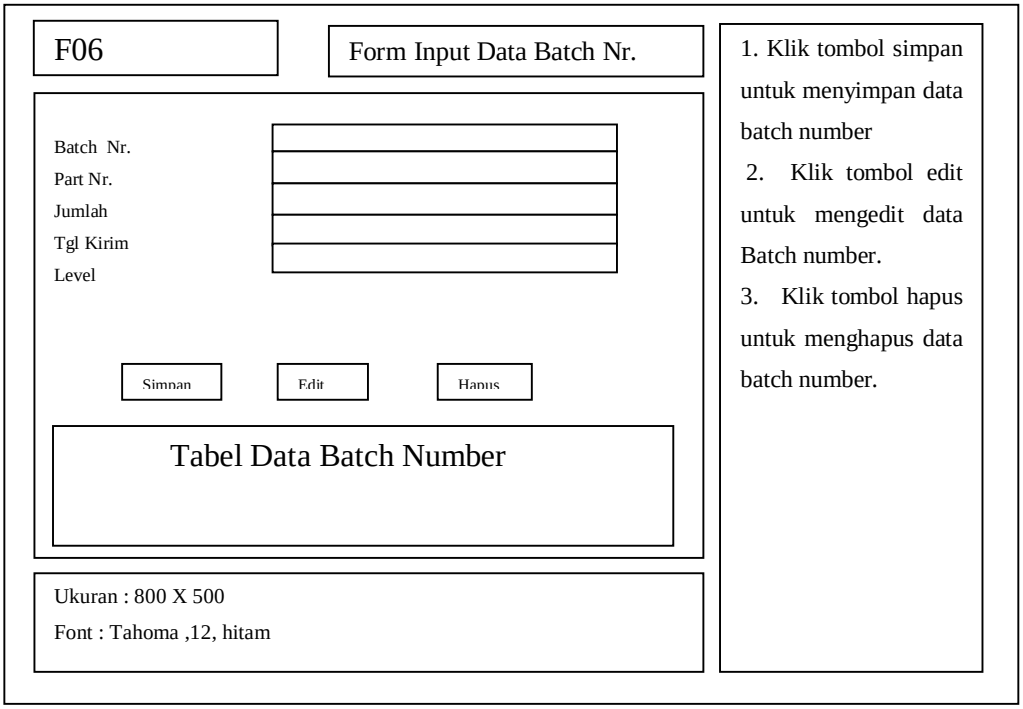
Gambar 3.2.6 Perancangan Antar Muka Input Data harian

c. Perancangan antar muka *input data* jenis proses



Gambar 3.2.7 Perancangan Antar Muka Input Data Jenis Proses

d. Perancangan antar muka *input data batch* number



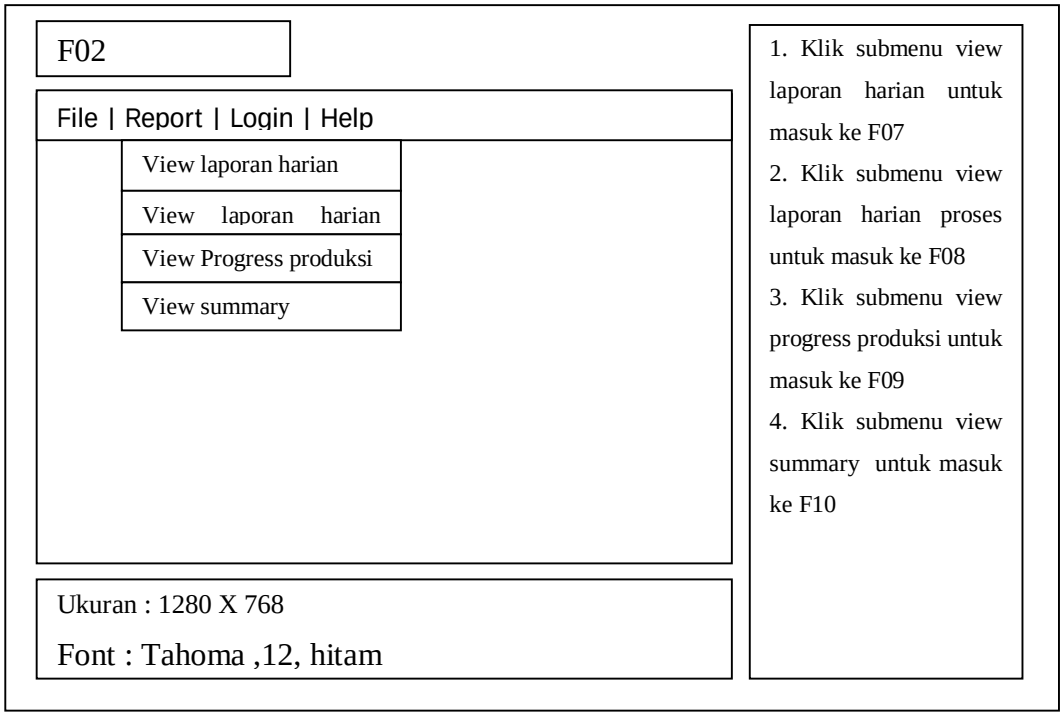
Gambar 3.2.8 Perancangan Antar Muka Input Data Batch Number

e. Exit

Jika menu klik ditekan maka aplikasi akan ditutup

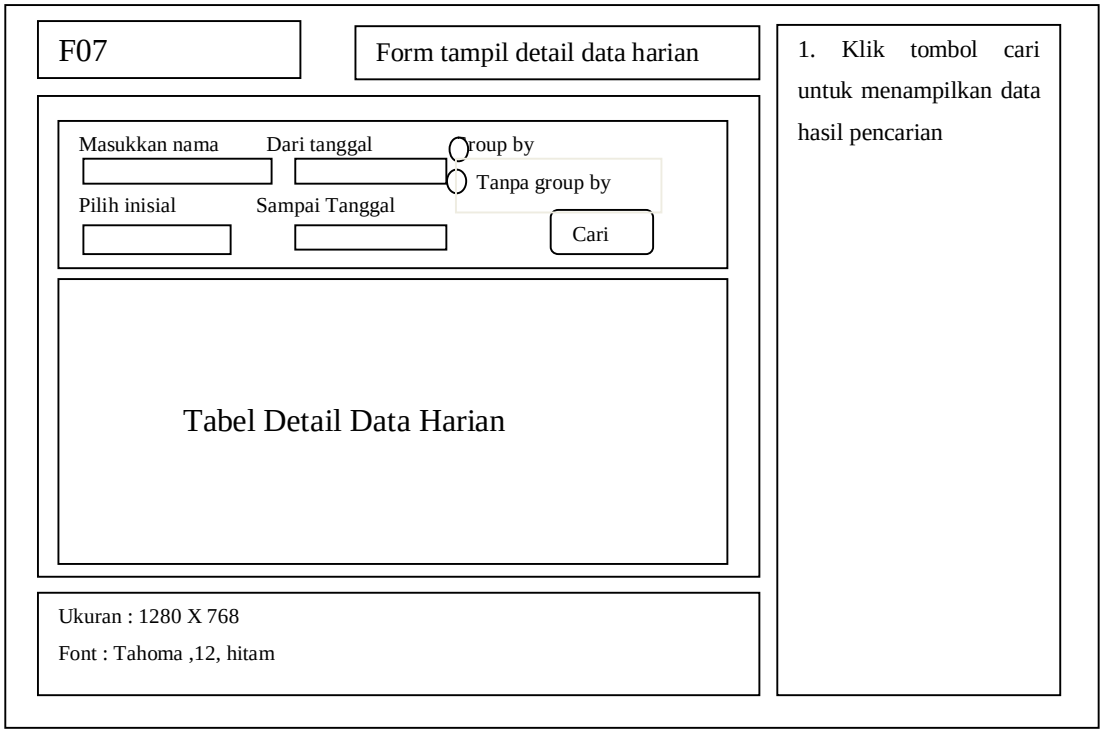
5. Perancangan antar muka *report*

Gambar dibawah ini merupakan perancangan antar muka menu *report* yang berfuingsi untuk menampilkan berbagai laporan terkait.



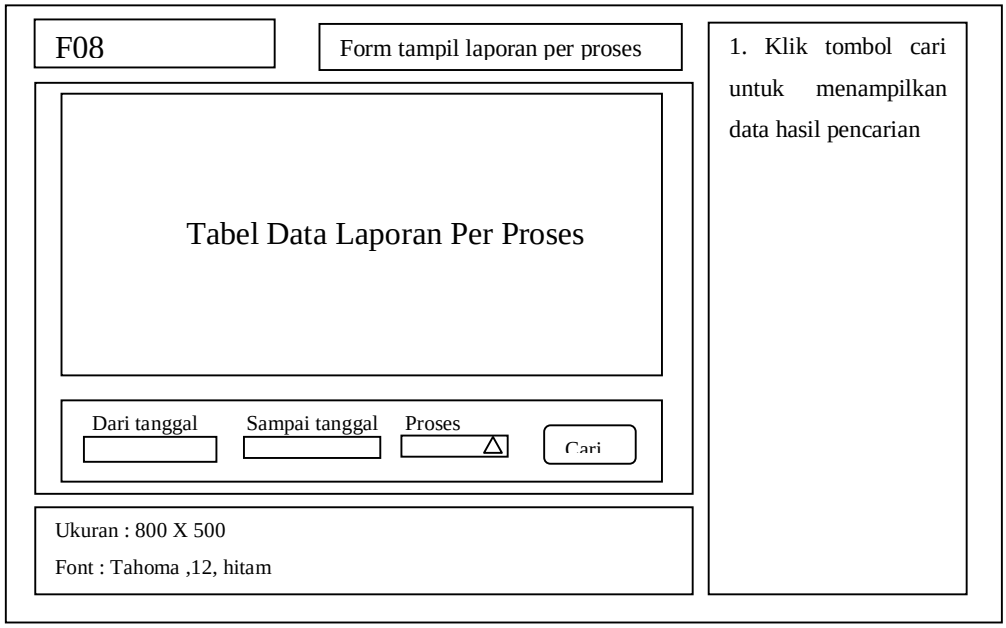
Gambar 3.2.9 Perancangan Antar Muka Menu Report

a. Perancangan antar muka laporan harian



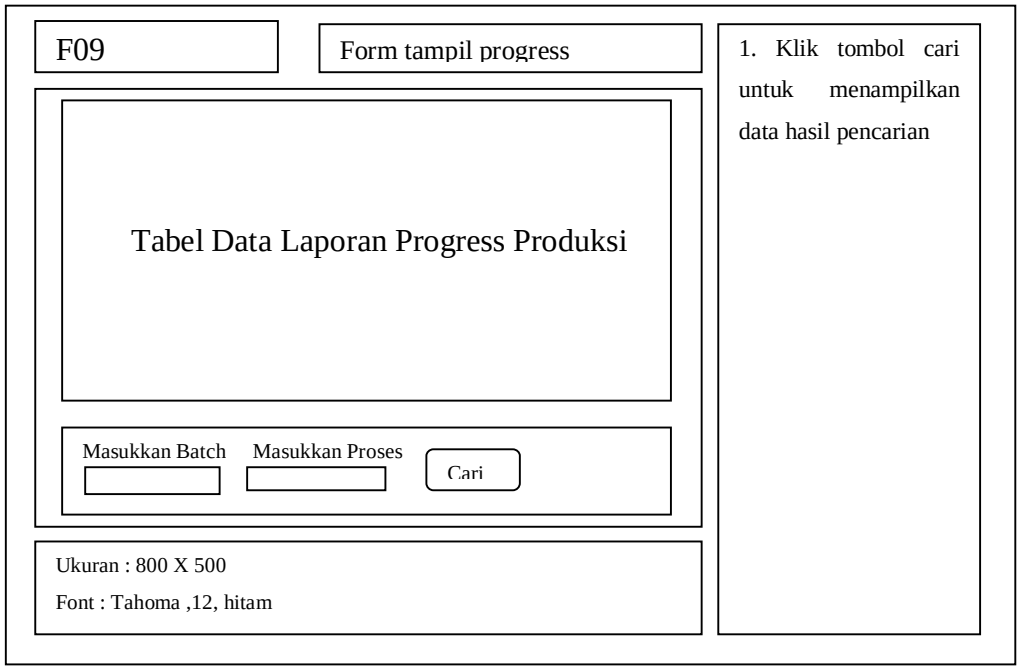
Gambar 3.2.10 Perancangan Antar Muka Submenu Detail Laporan Harian

b. Perancangan antar muka *view* laporan per proses



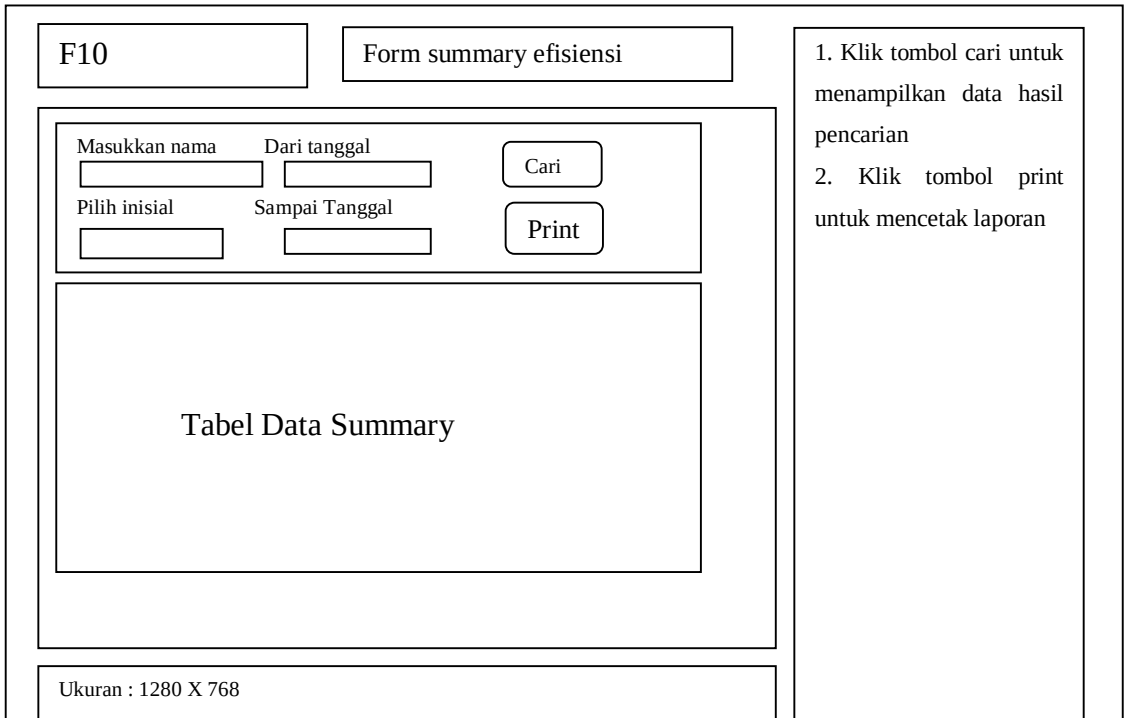
Gambar 3.2.11 Perancangan Antar Muka View Laporan Perproses

c. Perancangan antar muka *view* laporan progres produksi

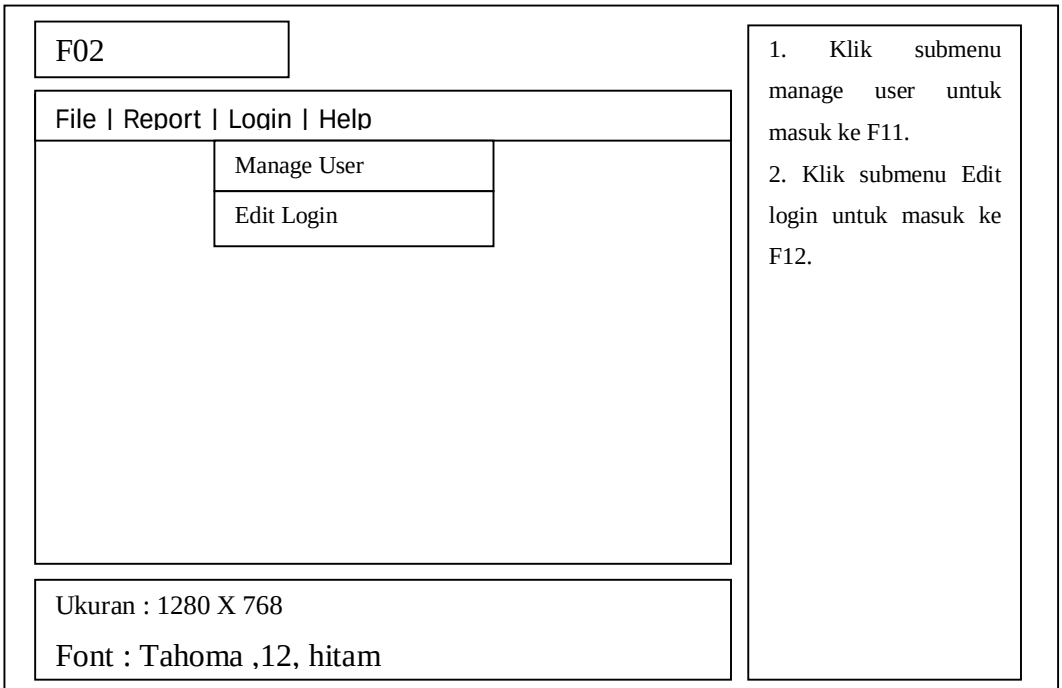


Gambar 3.2.12 Perancangan Antar Muka View Progress Poduksi Perbatch

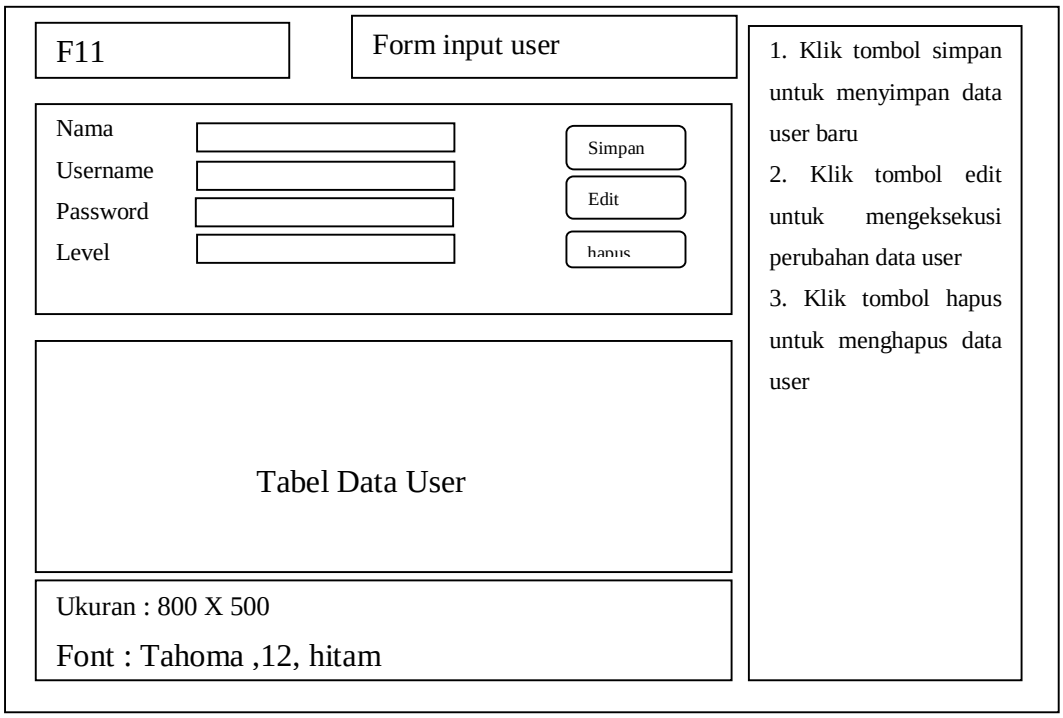
d. Perancangan antar muka *summary* efisiensi operator



6. Perancangan antar muka *login*

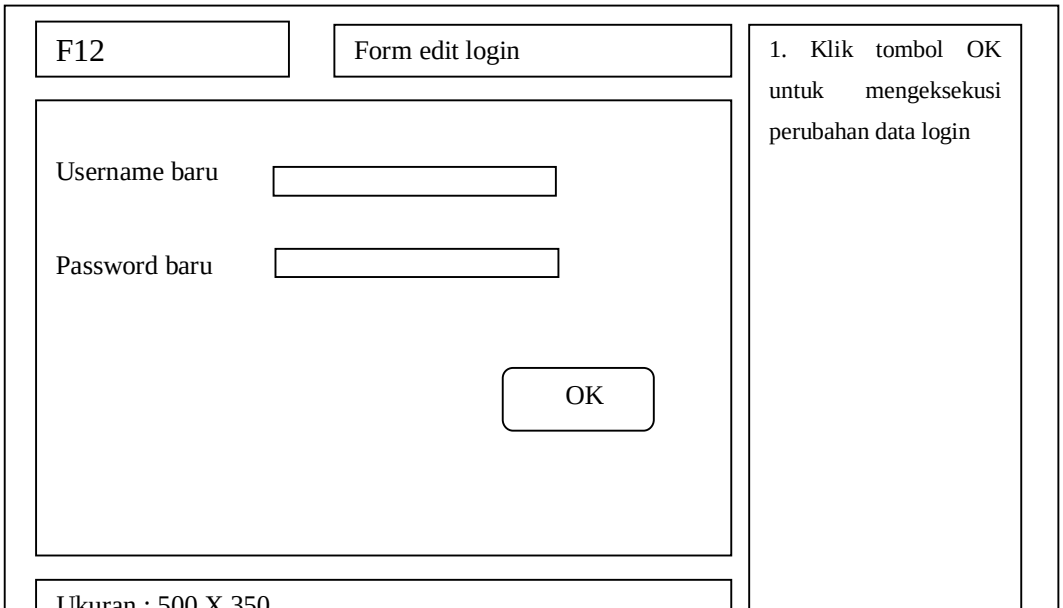


a. Perancangan antar muka sub menu *manage user*



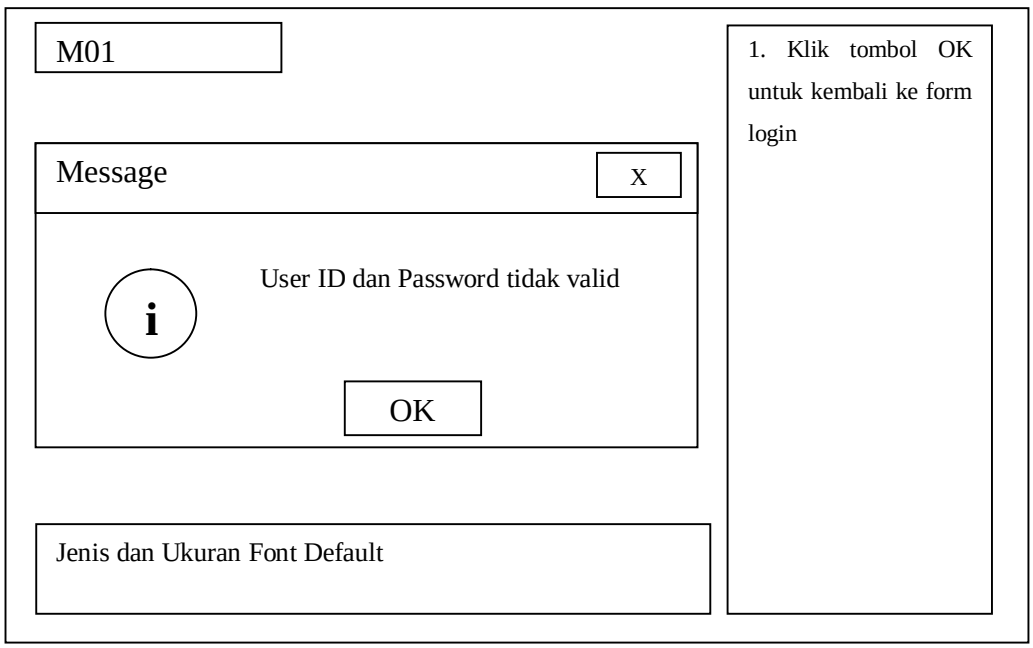
Gambar 3.2.15 Perancangan Antar Muka Manage User

b. Perancangan antar sub menu *edit login*



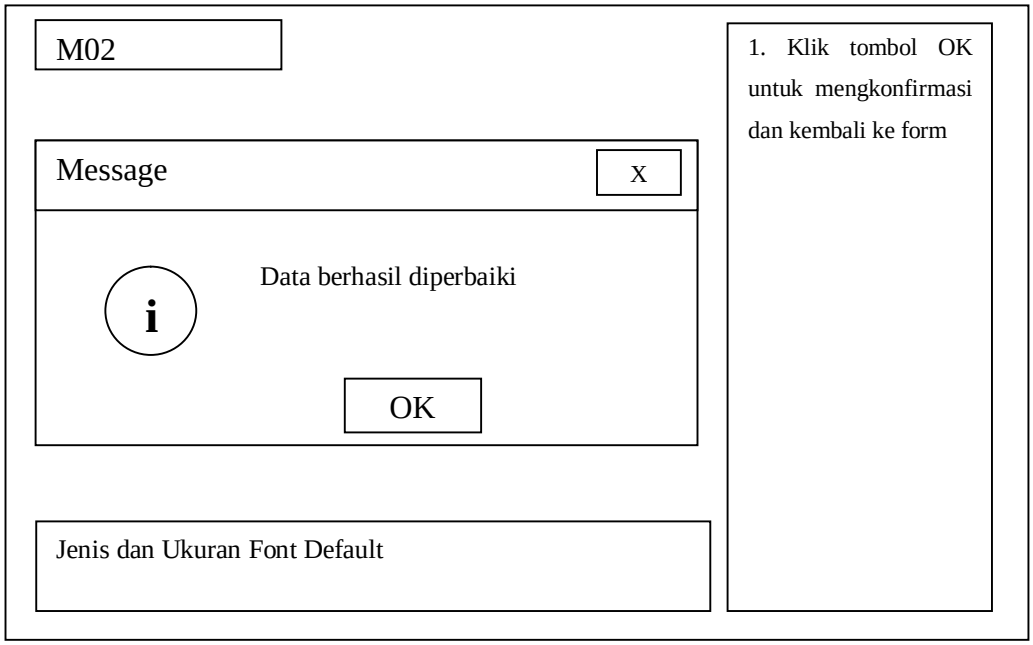
3.2.2.2 Perancangan Pesan

1. Pesan Gagal *Login*



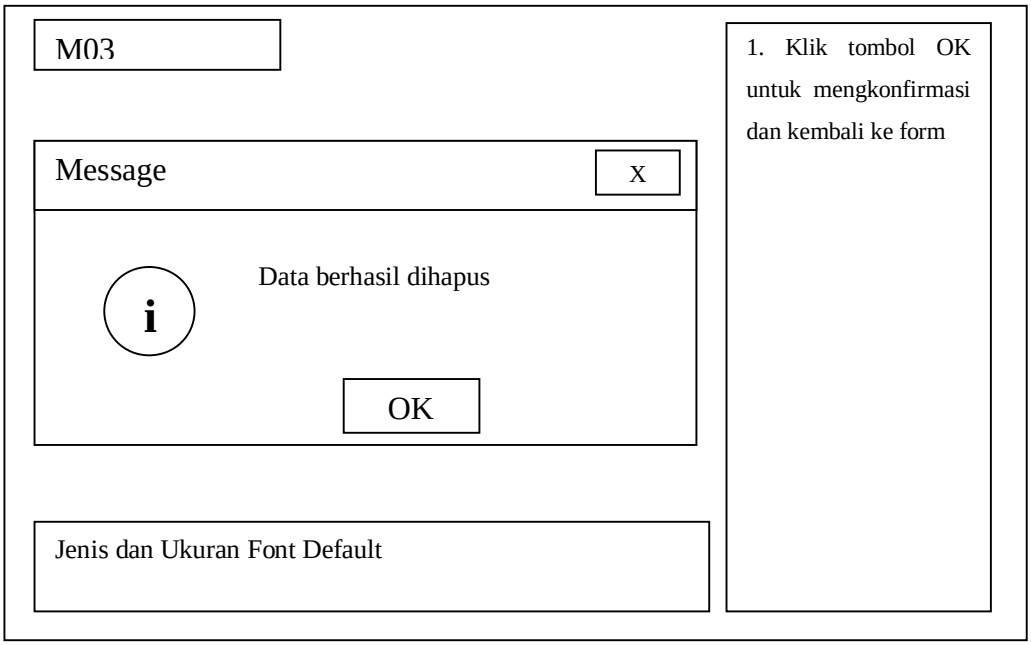
Gambar 3.2.17 Perancangan Antar Muka Pesan Gagal Login

2. Pesan berhasil *edit data*



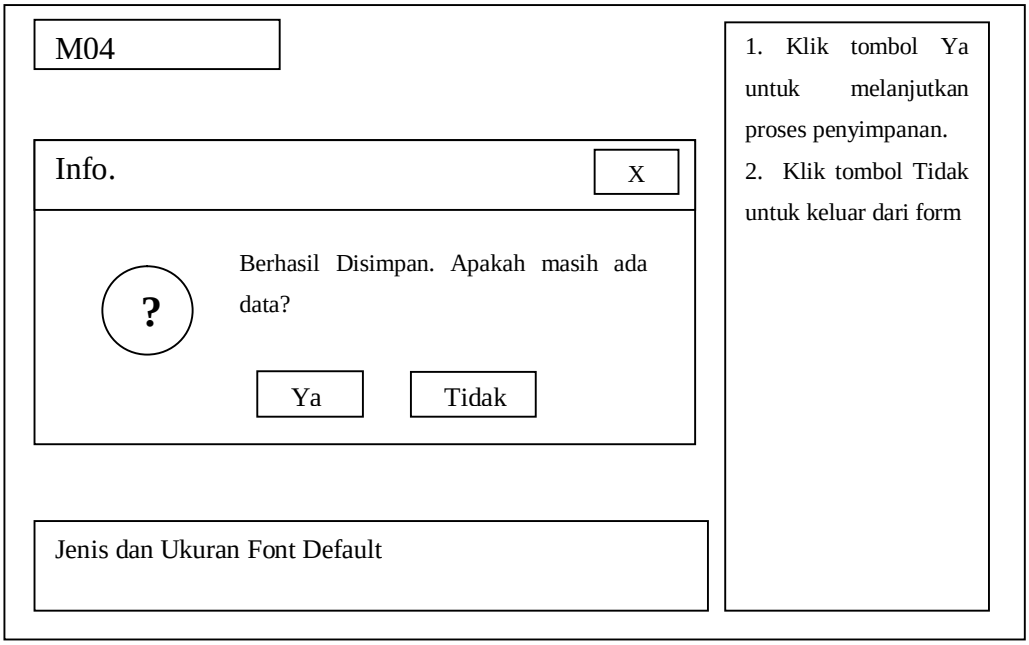
Gambar 3.2.18 Perancangan Antar Muka Pesan Berhasil Edit Data

3. Pesan berhasil hapus *data*



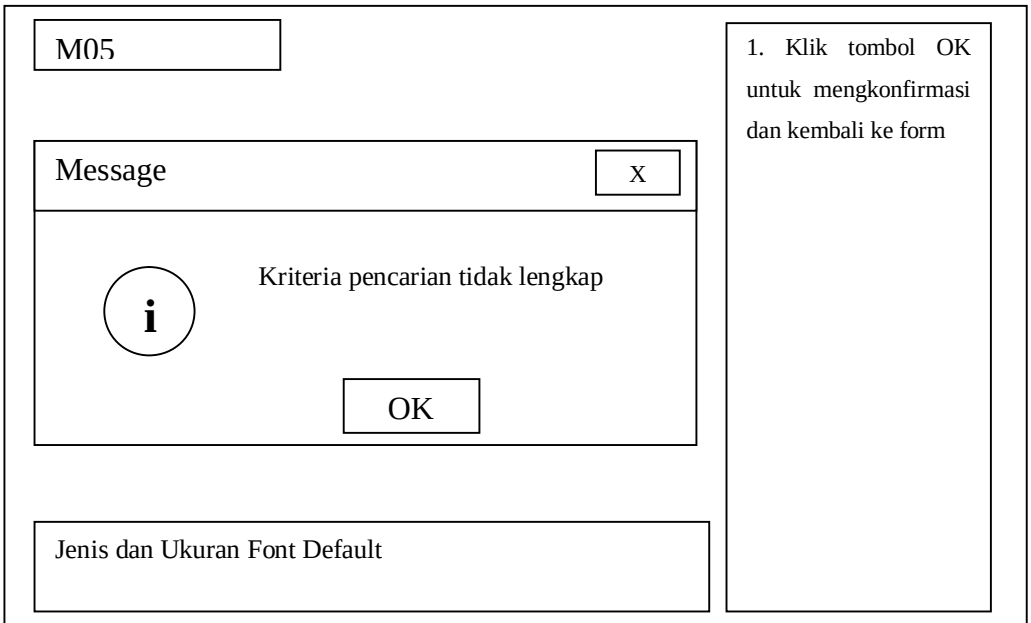
Gambar 3.2.19 Perancangan Antar Muka Pesan Berhasil Hapus Data

4. Pesan berhasil simpan *data*

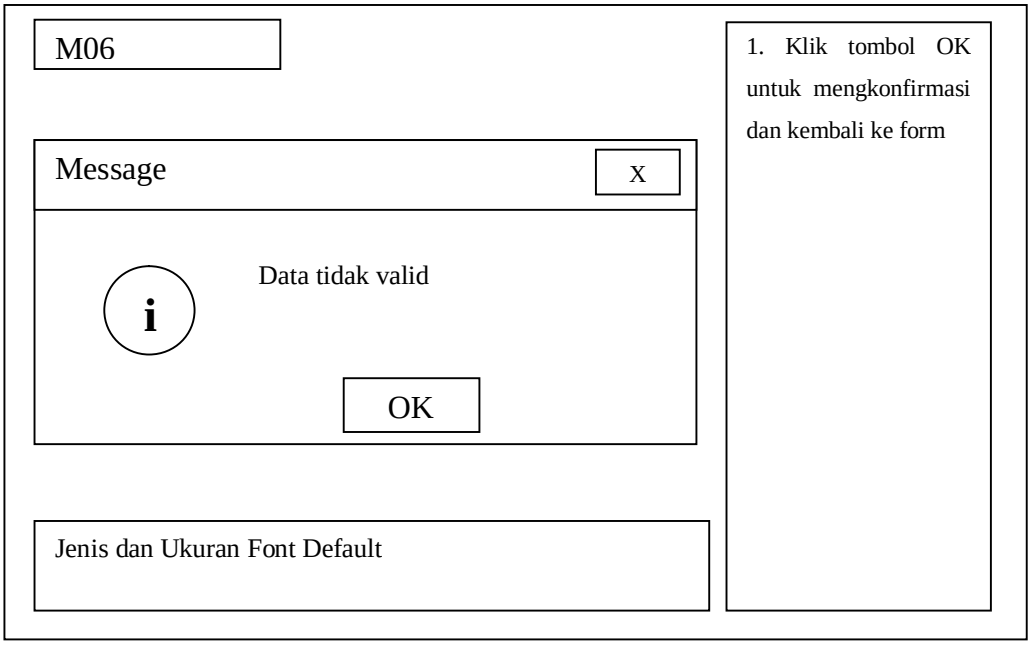


Gambar 3.2.20 Perancangan Antar Muka Pesan Berhasil Simpan Data

5. Pesan kriteria pencarian tidak lengkap

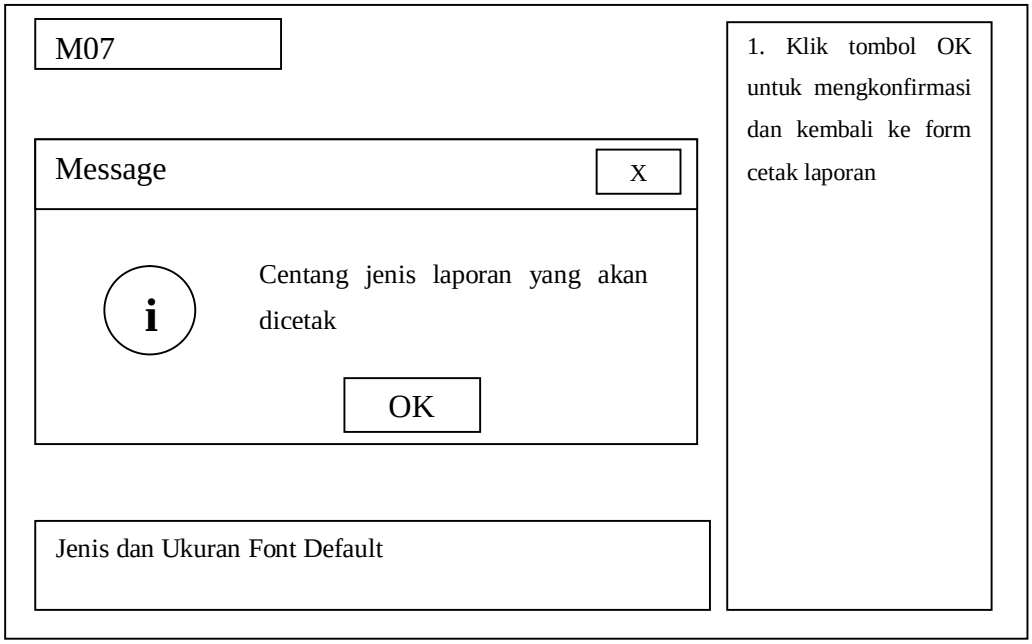


6. Pesan *input data* tidak *valid*



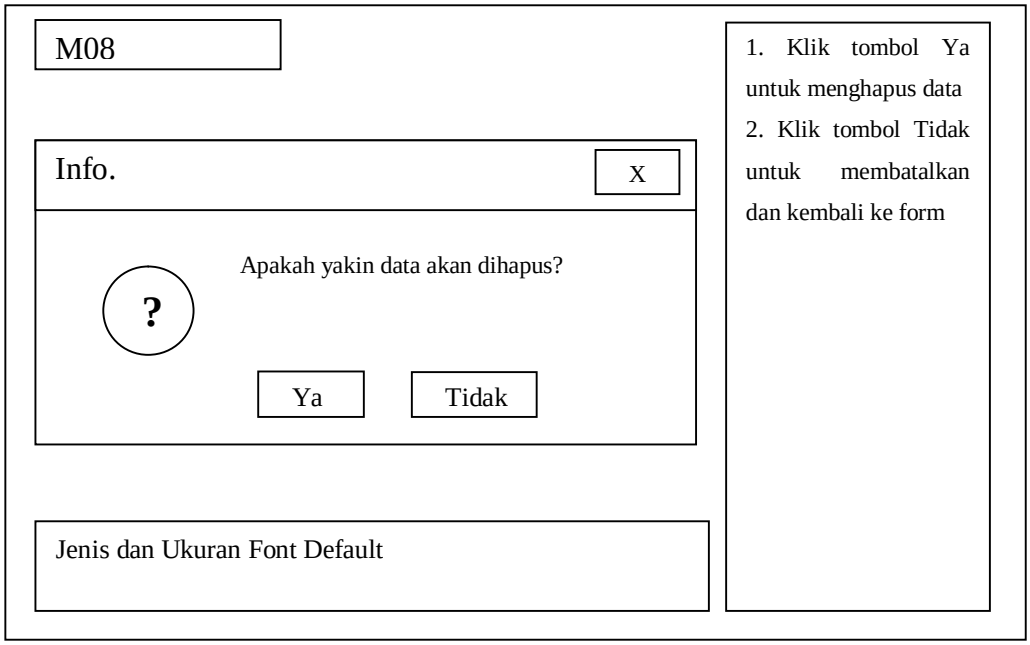
Gambar 3.2.22 Perancangan Antar Muka Pesan Input Data Tidak Valid

7. Pesan perintah pilih jenis laporan



Gambar 3.2.23 Perancangan Antar Muka Pesan Pilih Jenis Laporan

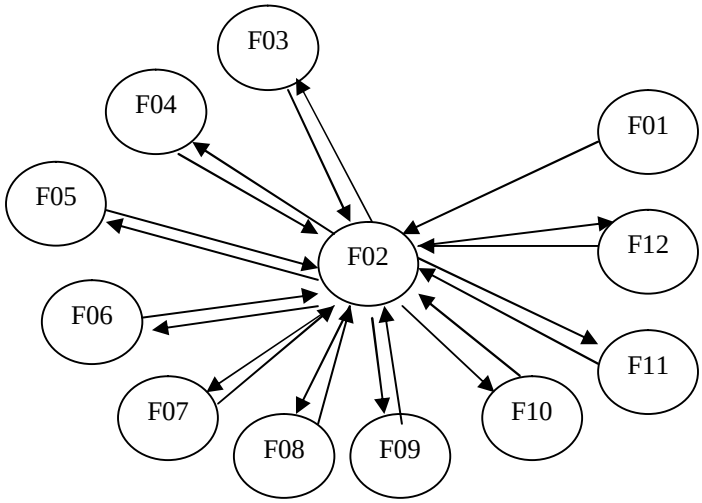
8. Pesan konfirmasi hapus *data*



Gambar 3.2.24 Perancangan Antar Muka Pesan Konfirmasi Hapus Data

3.2.2.3 Jaringan Semantik

Setelah melakukan perancangan antar muka maka dilakukanlah perancangan terhadap hubungan antar *form*. Untuk menggambarkan hubungannya digambarkan dalam sebuah jaringan semantik.



Gambar 3.2.25 Jaringan Semantik

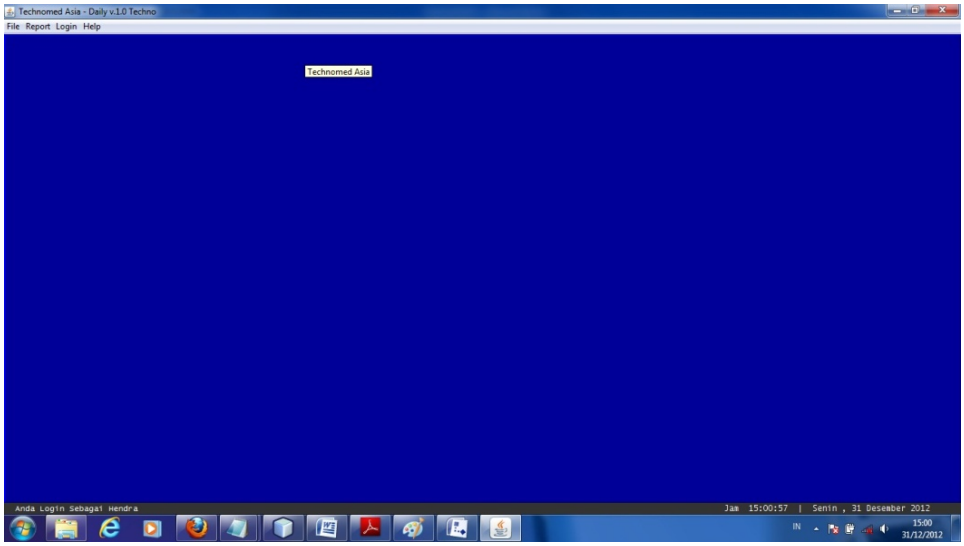
3.2.3 Implementasi perancangan antar muka

A. Implementasi *login*



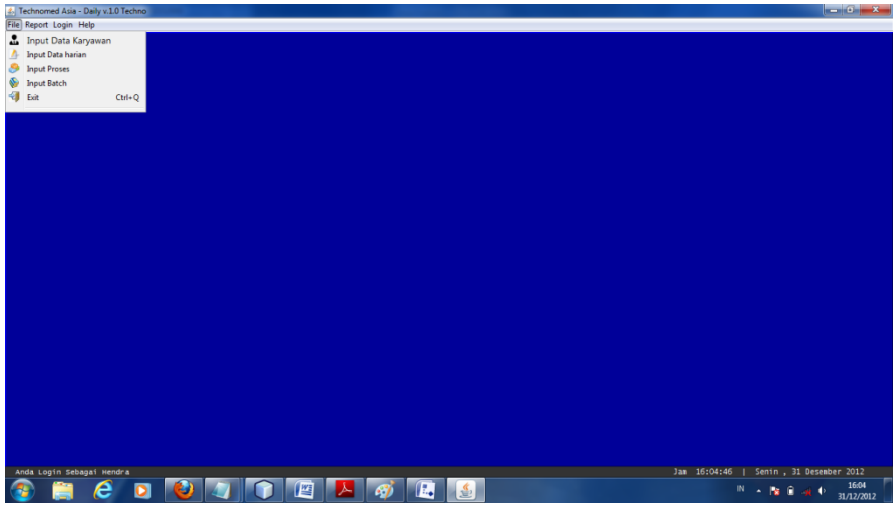
Gambar 3.2.26 Implementasi Form *Login*

B. Implementasi menu utama



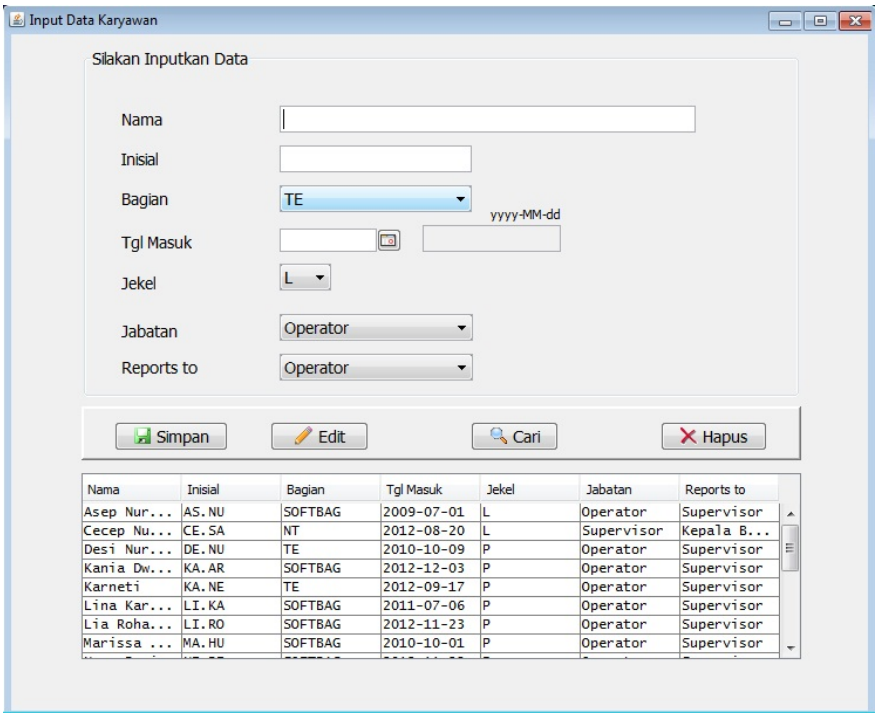
Gambar 3.2.27 Implementasi Menu Utama

1. Implementasi *File*



Gambar 3.2.28 Implementasi Menu *View*

a. *Input data* karyawan

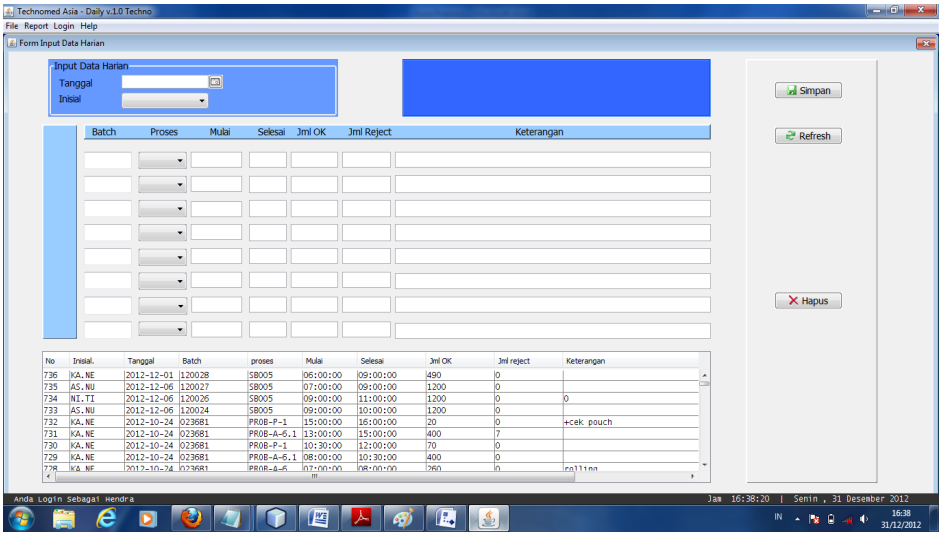


Gambar 3.2.29 Implementasi *Input Data* Karyawan

Petunjuk penggunaan:

1. Tambah *data*
   - Isi semua field . Jangan dikosongkan.
   - Jika *data* yang dimasukkan sudah benar, tekan tombol ‘Simpan’.
   - *Data* akan tersimpan di *database*.
2. *Edit data*
   - Klik dua kali data yang akan diedit yang terdapat dalam tabel.
   - *Data* akan ditampilkan dalam field isian.
   - Lakukan perubahan pada *data* yang dirasa perlu.
   - Tekan tombol ‘*Edit*’ untuk mengeksekusi perubahan
3. Hapus *data*
   - Klik satu kali *data* yang akan dihapus dalam tabel.
   - Tekan tombol ‘Hapus’.
   - Akan muncul dialog box untuk mengkonfirmasi penghapusan *data*.
   - Tekan tombol ‘Ya’ untuk menghapus, atau ‘Tidak’ untuk membatalkan penghapusan.

b. *Input data* harian



Gambar 3.2.30 Implementasi *Input Data* Harian

Petunjuk penggunaan:

1. Tambah *data*

- Isi semua field . Jangan dikosongkan.
- Jika *data* yang dimasukkan sudah benar, tekan tombol ‘Simpan’.
- *Data* akan tersimpan di *database*.

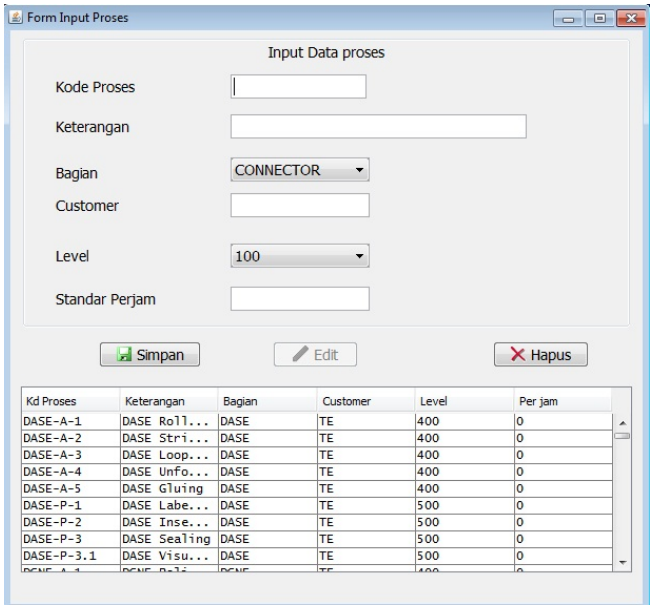
2. *Edit data*

- Klik dua kali *data* yang akan di*edit* yang terdapat dalam tabel.
- *Data* akan ditampilkan dalam field isian.
- Lakukan perubahan pada *data* yang dirasa perlu.
- Tekan tombol ‘*Edit*’ untuk mengeksekusi perubahan

3. Hapus *data*

- Klik satu kali *data* yang akan dihapus dalam tabel.
- Tekan tombol ‘Hapus’.
- Akan muncul dialog box untuk mengkonfirmasi penghapusan *data*.
- Tekan tombol ‘Ya’ untuk menghapus, atau ‘Tidak’ untuk membatalkan penghapusan.

c. *Input* proses



Gambar 3.2.31 Implementasi Input Proses

Petunjuk penggunaan:

1. Tambah *data*

- Isi semua field . Jangan dikosongkan.
- Jika *data* yang dimasukkan sudah benar, tekan tombol ‘Simpan’.
- *Data* akan tersimpan di *database*.

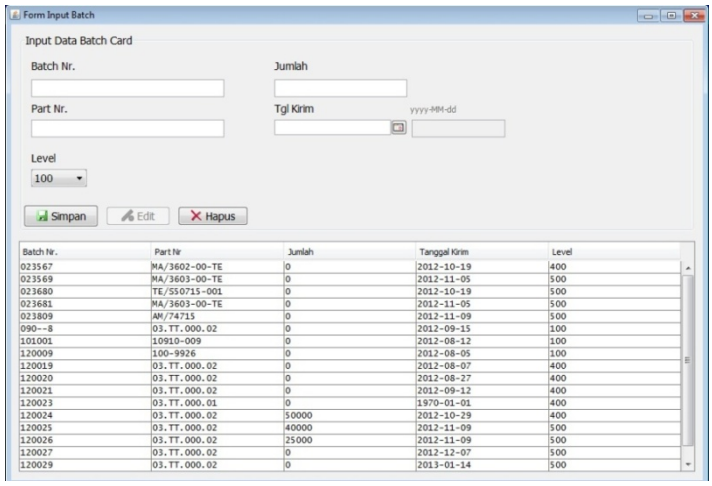
2. *Edit data*

- Klik dua kali *data* yang akan di*edit* yang terdapat dalam tabel.
- *Data* akan ditampilkan dalam field isian.
- Lakukan perubahan pada *data* yang dirasa perlu.
- Tekan tombol ‘*Edit*’ untuk mengeksekusi perubahan

3. Hapus *data*

- Klik satu kali baris *data* yang akan dihapus dalam tabel.
- Tekan tombol ‘Hapus’.
- Akan muncul dialog box untuk mengkonfirmasi penghapusan *data*.
- Tekan tombol ‘Ya’ untuk menghapus, atau ‘Tidak’ untuk membatalkan penghapusan.

d. *Input batch* number



Gambar 3.2.32 Implementasi *Input Batch* Number

Petunjuk penggunaan:

1. Tambah *data*

- Isi semua field . Jangan dikosongkan.
- Jika *data* yang dimasukkan sudah benar, tekan tombol ‘Simpan’.
- *Data* akan tersimpan di *database*.

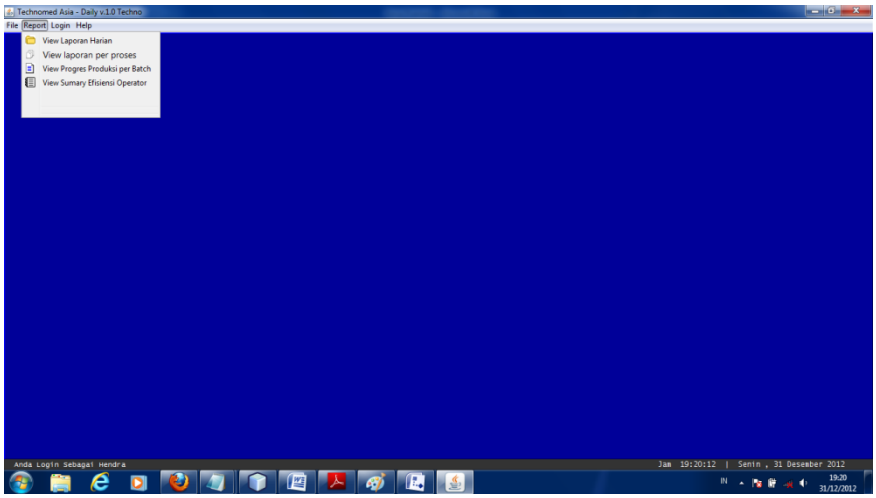
2. *Edit data*

- Klik dua kali *data* yang akan di*edit* yang terdapat dalam tabel.
- *Data* akan ditampilkan dalam field isian.
- Lakukan perubahan pada *data* yang dirasa perlu.
- Tekan tombol ‘*Edit*’ untuk mengeksekusi perubahan

3. Hapus *data*

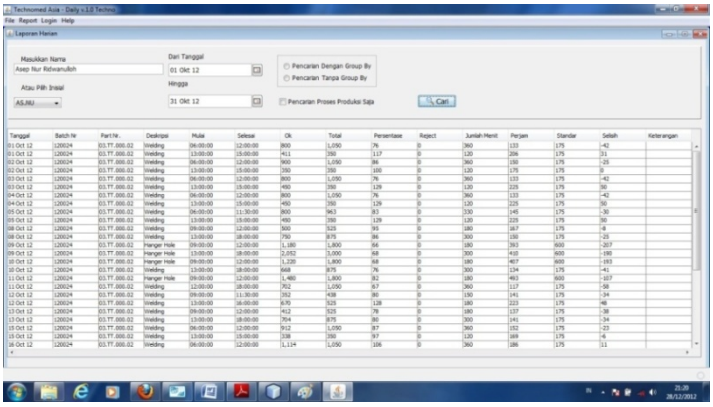
- Klik satu kali baris *data* yang akan dihapus dalam tabel.
- Tekan tombol ‘Hapus’.
- Akan muncul dialog box untuk mengkonfirmasi penghapusan *data*.
- Tekan tombol ‘Ya’ untuk menghapus, atau ‘Tidak’ untuk membatalkan penghapusan.

2. Implementasi Menu *Report*



Gambar 3.2.33 Implementasi Menu *Report*

a. *View* laporan harian



Gambar 3.2.34 Implementasi *View* laporan Harian

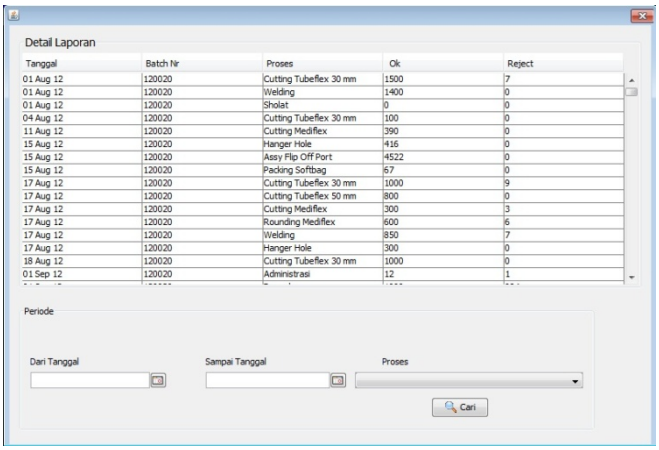
Petunjuk penggunaan:

1. Menampilkan *data* operator berdasarkan periode waktu:

- Isi field tanggal awal .
- Isi field tanggal akhir.
- Pilih nama operator yang akan ditampilkan.
- Pilih radio button group by untuk mengelompokkan *data* berdasarakan proses dan sebaliknya.
- Tekan tombol ‘Cari’ untuk menampilkan *data* .

- *Data* akan ditampilkan dalam tabel.

b. *View* laporan perproses



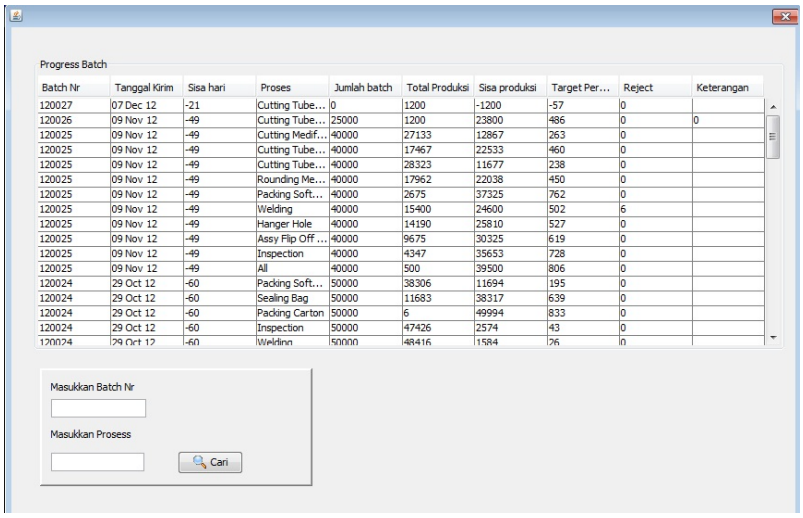
Gambar 3.2.35 Implementasi *View* Laporan Perproses

Petunjuk penggunaan:

1. Menampilkan *data* berdasarkan tanggal dan jenis proses:

- Isi field tanggal awal .
- Isi field tanggal akhir.
- Pilih jenis porses yang akan ditampilkan
- Tekan tombol ‘Cari’ untuk menampilkan *data* hasil pencarian
- *Data* akan ditampilkan dalam tabel

c. *View* progress produksi per*batch*



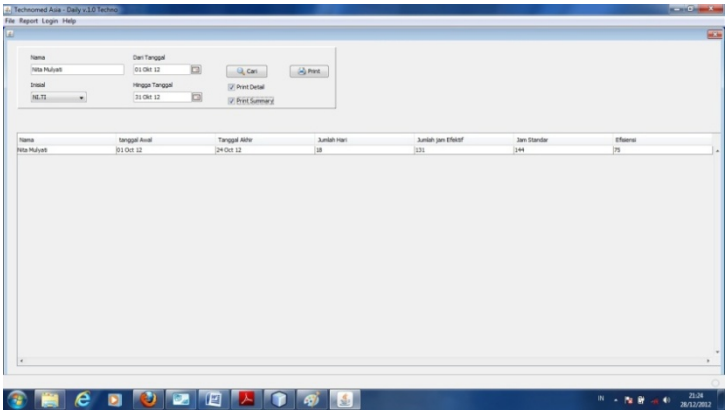
Gambar 3.2.36 Implementasi *View* Progress Produksi Per*batch*

Petunjuk penggunaan:

1. Menampilkan *data* berdasarkan *Batch* Nr. dan jenis proses:

- Isi field *batch* nr dengan *batch* nr yang akan dicari .
- Isi field proses dengan proses yang ingin ditampilkan.
- Tekan tombol ‘Cari’ untuk menampilkan *data* hasil pencarian .
- *Data* akan ditampilkan dalam tabel.

d. *View Summary* efisiensi operator



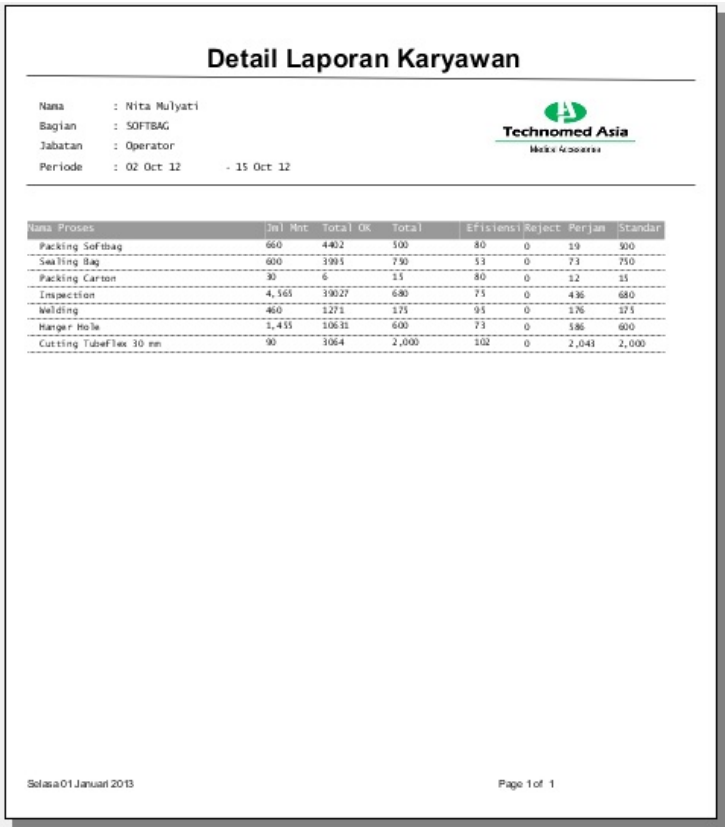
Gambar 3.2.37 Implementasi *View Summary* Efisiensi Operator

Petunjuk penggunaan:

1. Menampilkan *data* operator dan efisiensi kerjanya:
- Isi field nama dengan nama yang akan dicari atau pilih inisial dari combobox.
- Isi field tanggal awal.
- Isi field tanggal akhir.
- Tekan tombol 'Cari' untuk menampilkan *data* hasil pencarian .
- *Data* akan ditampilkan dalam tabel.

2. Cetak laporan:
- Setelah langkah 1 dikerjakan, kemudian centang pilihan jenis laporan yang akan dicetak pada check box yang disediakan.
- Tekan tombol print.
- Kemudian akan muncul jendela print pre*view* (gambar 3.2.43 untuk print detail dan gambar 3.2.44 untuk print *summary*).

e. Implementasi print detail



Detail Laporan Karyawan

Nama : Nita Mulyati
Bagian : SOFTBAG
Jabatan : Operator
Periode : 02 Oct 12 - 15 Oct 12

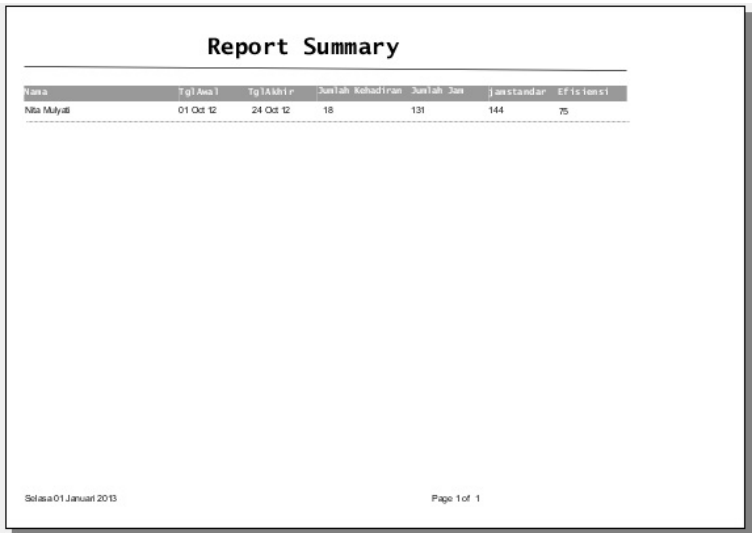
Technomed Asia

Nama Proses	Jml Mnt	Total OK	Total	Efisiensi	Reject	Perjam	Standar
Packing Softbag	660	4402	500	80	0	19	500
Sealing Bag	600	3995	750	53	0	73	750
Packing Carton	30	6	15	80	0	12	15
Inspection	4,565	39027	680	75	0	436	680
Welding	460	1271	175	95	0	176	175
Hanger Hole	1,455	10631	600	73	0	586	600
Cutting Tubeflex 30 mm	90	3064	2,000	102	0	2,043	2,000

Selasa 01 Januari 2013 Page 1 of 1

Gambar 3.2.38 Implementasi Print Detail

f. Implementasi print *summary*



Report Summary

Nama	TglAwal	TglAkhir	Jumlah Kehadiran	Jumlah Jam	jamstandar	Efisiensi
Nita Mulyati	01 Oct 12	24 Oct 12	18	131	144	75

Selasa 01 Januari 2013 Page 1 of 1

Gambar 3.2.39 Implementasi Print *Summary*

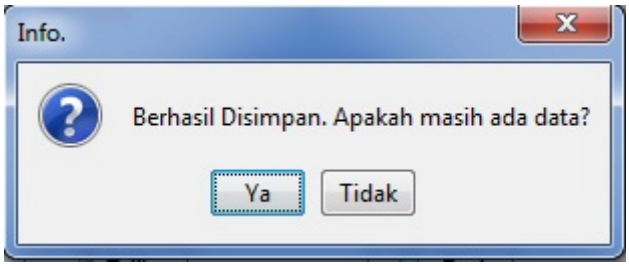
3.2.4 Implementasi perancangan pesan

A. Pesan gagal *login*



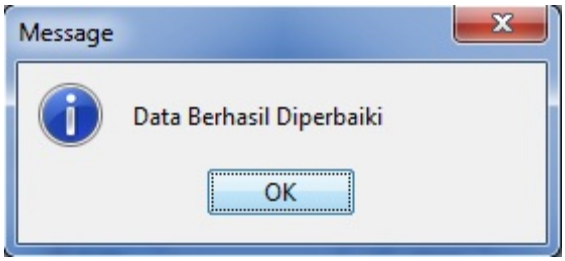
Gambar 3.2.40 Pesan Gagal *Login*

B. Pesan berhasil simpan *data*



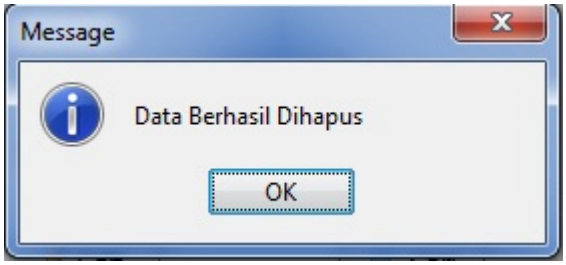
Gambar 3.2.41 Pesan berhasil Simpan *Data*

C. Pesan berhasil *edit data*



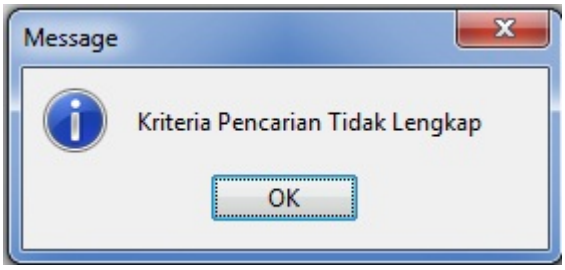
Gambar 3.2.42 Pesan Berhasil Edit Data

D. Pesan berhasil hapus *data*



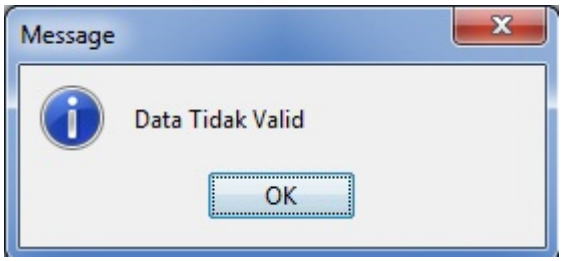
Gambar 3.2.43 Pesan Berhasil Hapus *Data*

E. Pesan kriteria pencarian tidak lengkap



Gambar 3.2.44 Pesan Kriteria Pencarian Tidak Lengkap

F. Pesan *input data* tidak *valid*



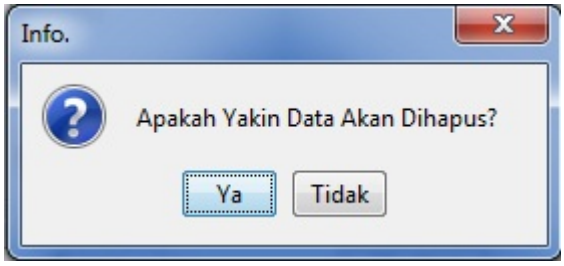
Gambar 3.2.45 Pesan *Input Data* Tidak *Valid*

G. Perintah pilih jenis laporan



Gambar 3.2.46 Perintah Pilih Jenis Laporan

H. Pesan konfirmasi hapus *data*



Gambar 3.2.47 Pesan Konfirmasi Hapus *Data*