

BAB IV

DESKRIPSI KERJA PRAKTEK

Analisis merupakan cara untuk menganalisa permasalahan berdasarkan data yang telah diperoleh dari hasil studi lapangan. Sedangkan desain sistem merupakan langkah yang harus ditempuh untuk menyajikan sebuah sistem informasi agar dapat terorganisir dengan baik dan jelas.

4.1 Analisa Sistem

Dalam pengembangan teknologi informasi saat ini, dibutuhkan analisa dan perancangan sistem pengolahan data yang baik. Sistem pengolahan data tersebut diharapkan mampu meningkatkan kinerja Sistem Informasi Manajemen Aset CV. Padantya Alam Panglungan yang akan dibuat. Untuk mendukung semua ini maka dibutuhkan analisis yang tepat, kebutuhan bisnis dan beberapa teknis analisis untuk menghasilkan perencanaan yang baik.

Data dan informasi yang dibutuhkan ialah berkenaan dengan tujuan dari Sistem Informasi Manajemen Aset CV. Padantya Alam Panglungan. Informasi tentang kebutuhan sistem informasi diperlukan untuk menghasilkan perencanaan yang dapat mendukung kerja aplikasi. Dari hasil penelitian disimpulkan bahwa diperlukan basis data untuk menyimpan data tentang aset perusahaan.

Dalam bab ini disampaikan rancangan sistem yang menjadi landasan dalam pembuatan aplikasi Sistem Informasi Manajemen Aset CV. Padantya Alam Panglungan dengan menggunakan VB.Net dan menggunakan *database* Microsoft SQL Server 2005 Express.

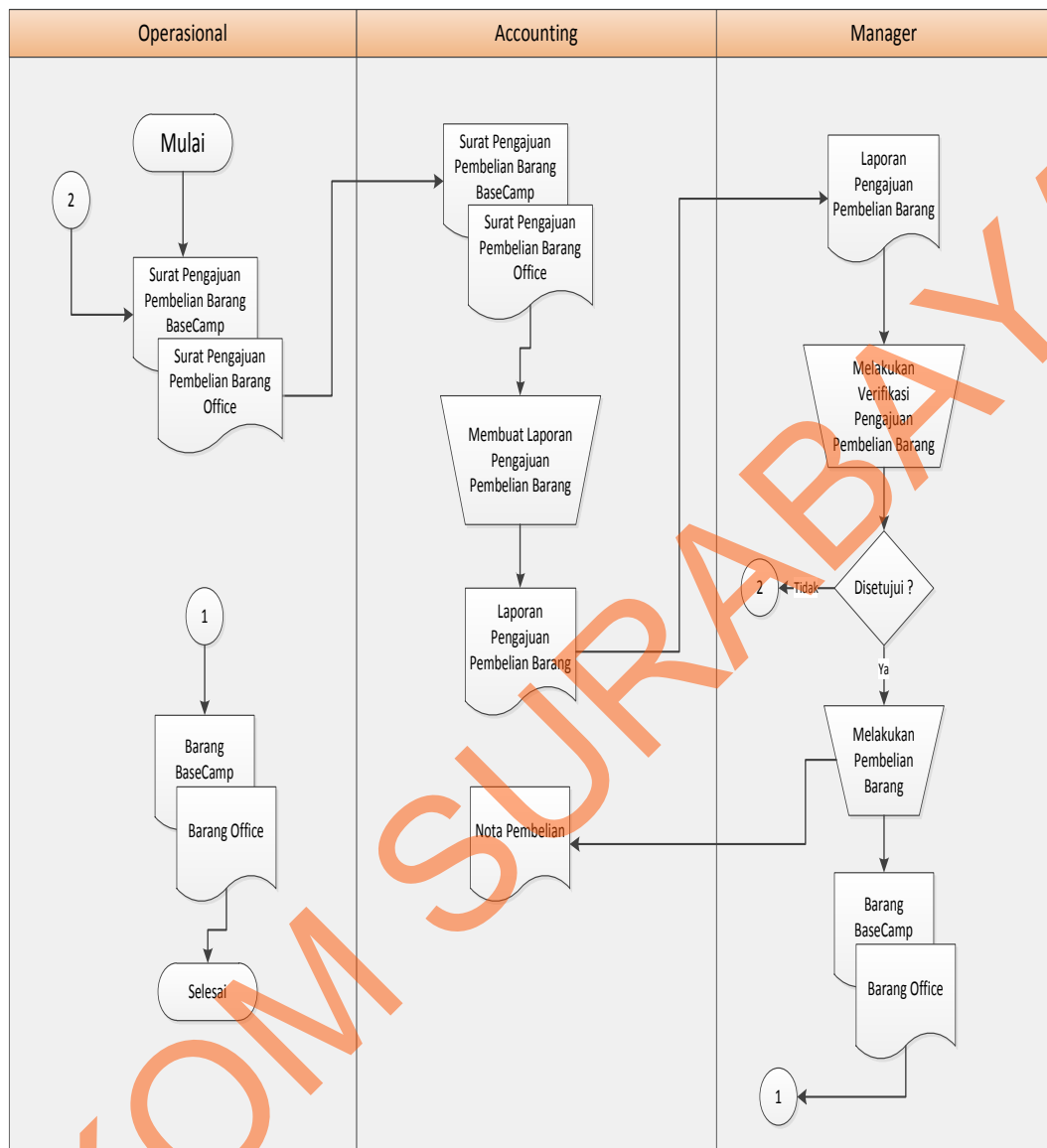
4.2 Perancangan Sistem

Berdasarkan analisis yang telah dilakukan, maka menghasilkan *document flow* hasil analisa sistem yang sedang berjalan dan sistem yang baru akan digambarkan pada *System Flow*, *Data Flow Diagram* (DFD), rancangan *Database* berupa *Entity Relationship Diagram* (ERD), serta desain I/O dari aplikasi yang akan dibuat.

4.2.1 Document Flow

Document Flow sistem informasi manajemen aset pada CV. Padantya Alam Panglungan terdiri atas 2 proses yaitu proses pembelian barang dan proses perbaikan barang. Secara umum *document flow* untuk proses tersebut dapat dilihat pada gambar dibawah ini:

Doc Flow Pembelian Barang

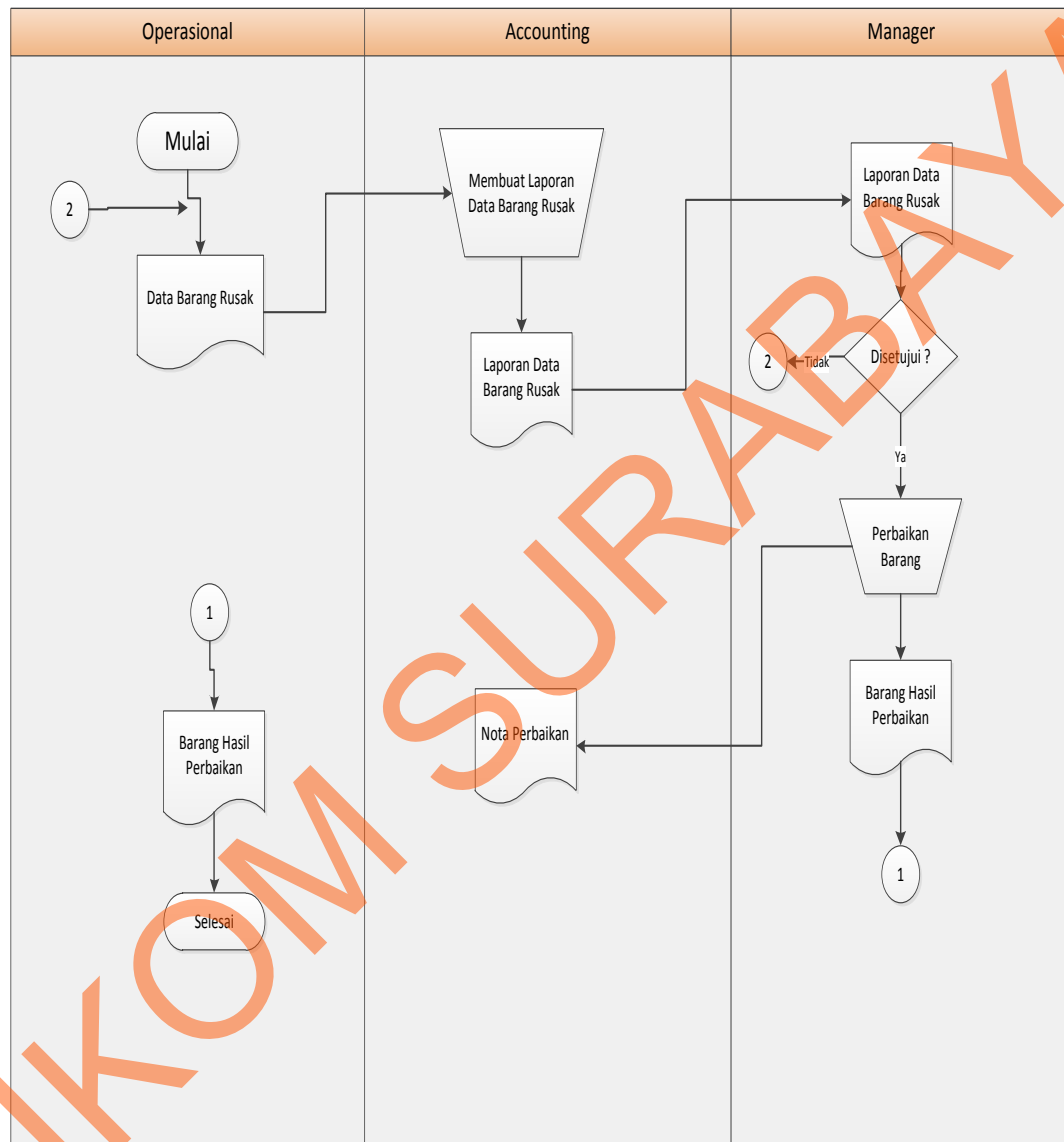


Gambar 4.1 Document Flow Pembelian Barang

Berdasarkan gambar 4.1 dapat dilihat bagaimana proses pembelian barang yang terjadi. Dimana awal dari proses ini yaitu dimulai dari *operasional* yang mengajukan surat pengajuan pembelian barang jika terdapat barang yang tidak tersedia dalam perusahaan. Dan selanjutnya surat pengajuan pembelian barang tersebut diterima oleh *manager*. Jika manager menyetujui pengajuan

pembelian barang dari *operasional* tersebut, maka dilakukan pembelian barang yang dilakukan oleh *manager*.

Doc Flow Perbaikan Barang



Gambar 4.2 Document Flow Perbaikan Barang

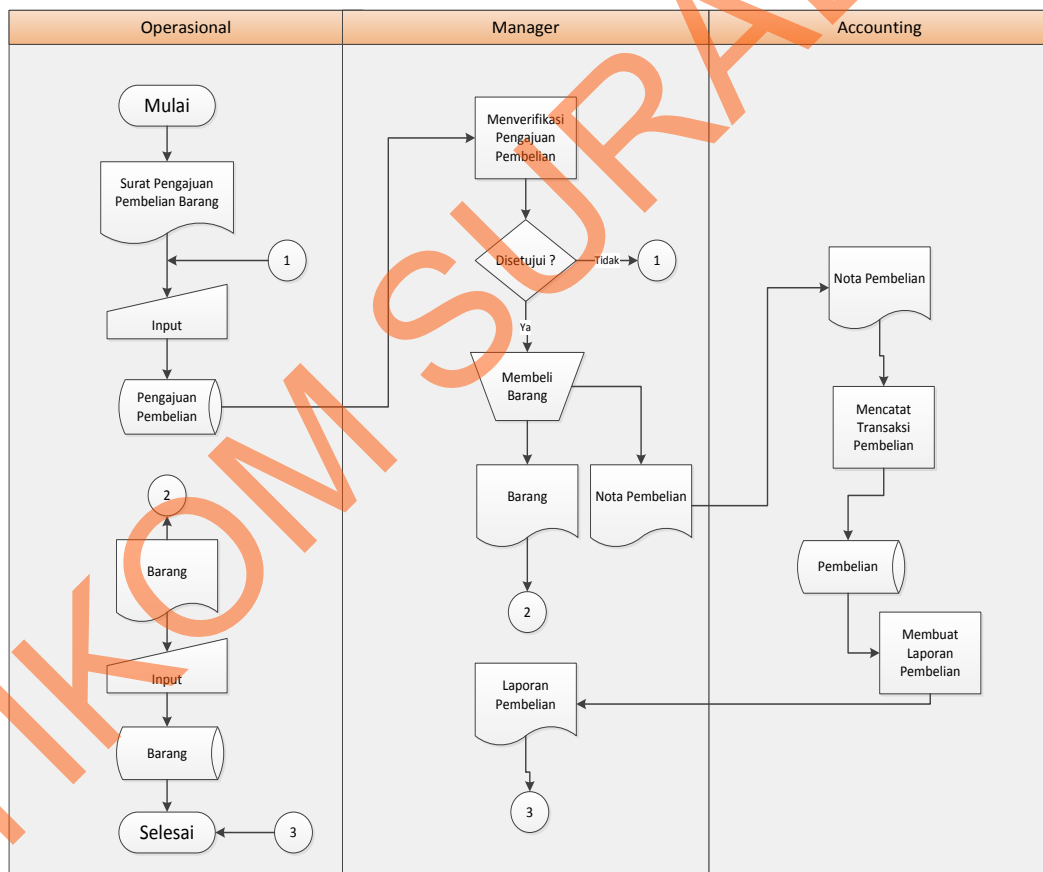
Berdasarkan gambar 4.2 dapat dilihat bagaimana proses reparasi barang yang terjadi. Dimana awal dari proses ini yaitu dimulai dari *operasional* yang memberikan data barang rusak kepada *accounting*, dan selanjutnya *accounting*

membuat laporan barang rusak yang diberikan kepada *manager*. Dan *manager* akan melakukan perbaikan barang-barang yang rusak.

4.2.2 System Flow

System flow terkomputerisasi merupakan gambaran dari sistem yang akan dibuat, beberapa proses yang masih dilakukan secara sederhana dikembangkan menjadi proses komputer. Proses yang dibuat adalah pencatatan barang pada saat pembelian dan perbaikan barang rusak. Untuk proses tersebut dapat dilihat pada gambar dibawah ini:

Sis Flow Pembelian Barang



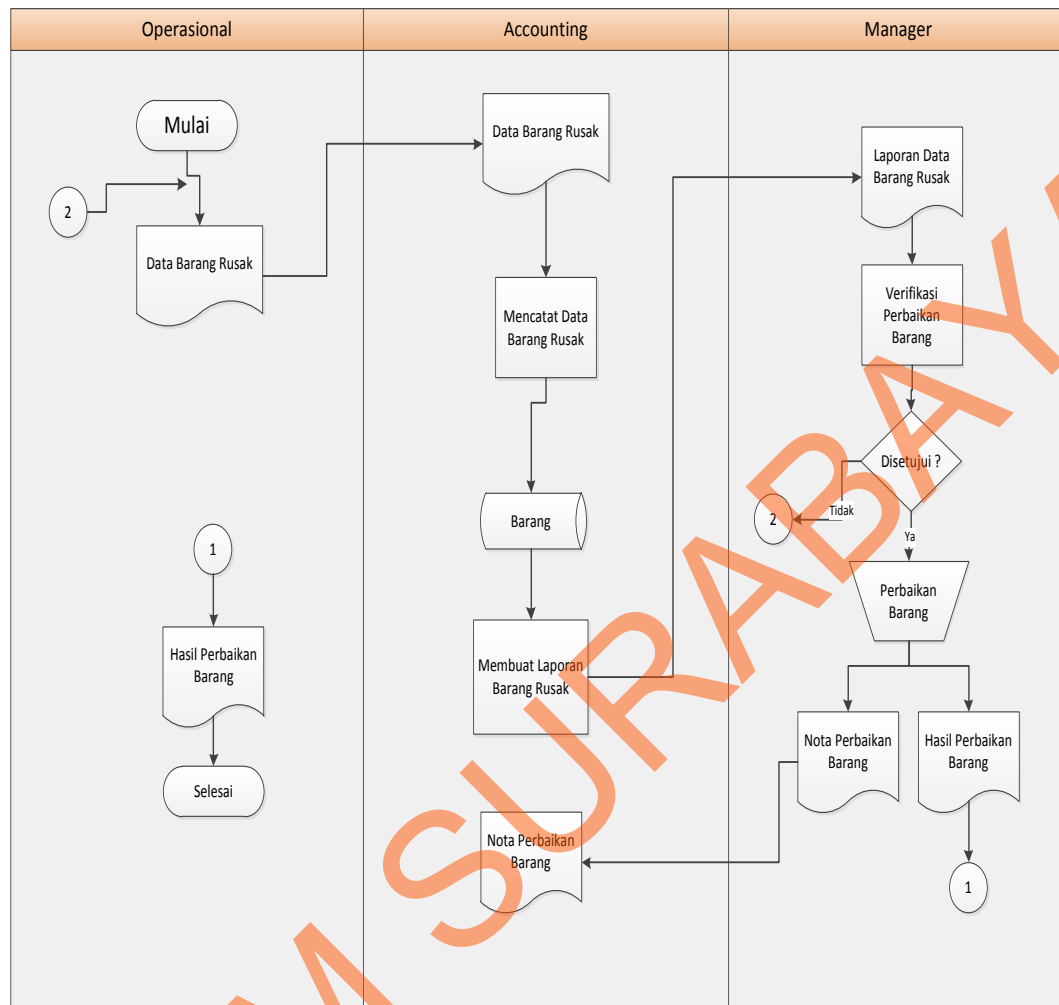
Gambar 4.3 *System Flow* Pembelian Barang

Berdasarkan gambar 4.3 dapat dilihat bagaimana proses pembelian barang yang terjadi. Dimana awal dari proses ini yaitu dimulai dari *operasional* yang mengajukan surat pengajuan pembelian barang jika terdapat barang yang

tidak tersedia dalam perusahaan. Dan selanjutnya surat pengajuan pembelian barang tersebut diterima oleh *manager*. Jika *manager* menyetujui pengajuan pembelian barang dari *operasional* tersebut, maka dilakukan pembelian barang yang dilakukan oleh *manager*. Nota dari pembelian barang yang dilakukan oleh *manager* akan diserahkan kepada *accounting* dan dibuatkan laporan pembelian barang. Laporan pembelian tersebut akan disimpan ke dalam *database* pembelian barang dan data barang yang dibeli akan disimpan ke dalam *database* barang.

STIKOM SURABAYA

Sis Flow Perbaikan Barang



Gambar 4.4 System Flow Perbaikan Barang

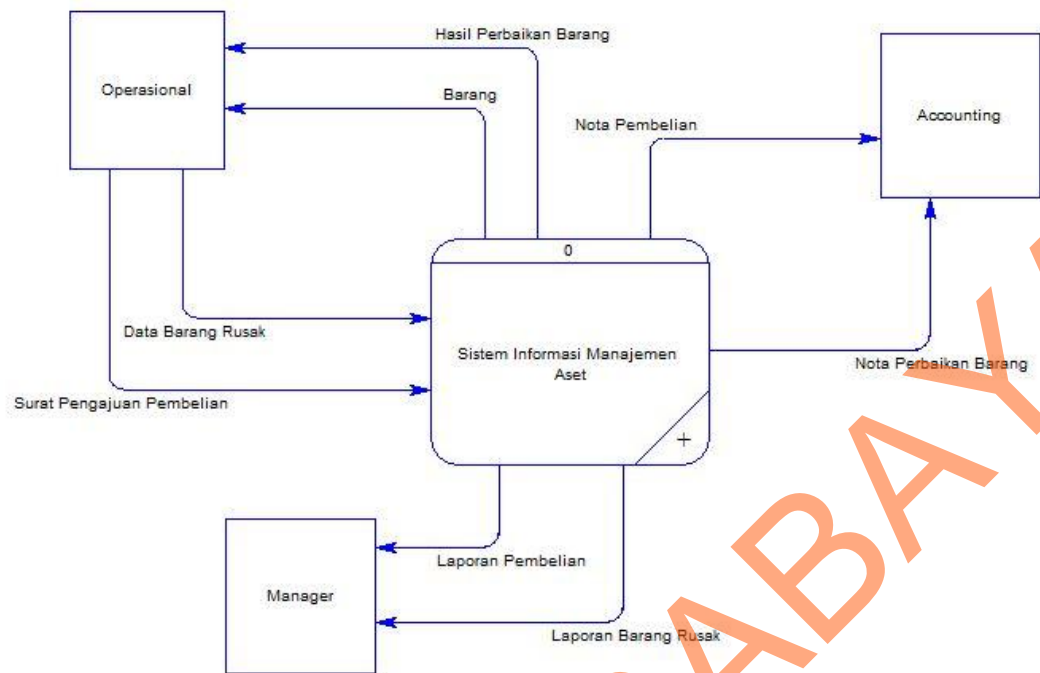
Berdasarkan gambar 4.4 dapat dilihat bagaimana proses reparasi barang yang terjadi. Dimana awal dari proses ini yaitu dimulai dari *operasional* yang memberikan data barang rusak kepada *manager*, dan *manager* akan melakukan perbaikan barang-barang yang rusak atau *manager* akan melakukan penyusutan barang untuk beberapa tahun mendatang. Jika terjadi perbaikan barang maka nota perbaikan barang diserahkan kepada *accounting*, dan selanjutnya *accounting* akan membuat laporan perbaikan barang. Dan data barang akan disimpan didalam database barang.

4.2.3 DFD (Data Flow Diagram)

Data flow Diagram (DFD) adalah suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, dimana data disimpan, proses apa yang dihasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir atau dimana data tersebut akan disimpan. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur.

a. Context Diagram

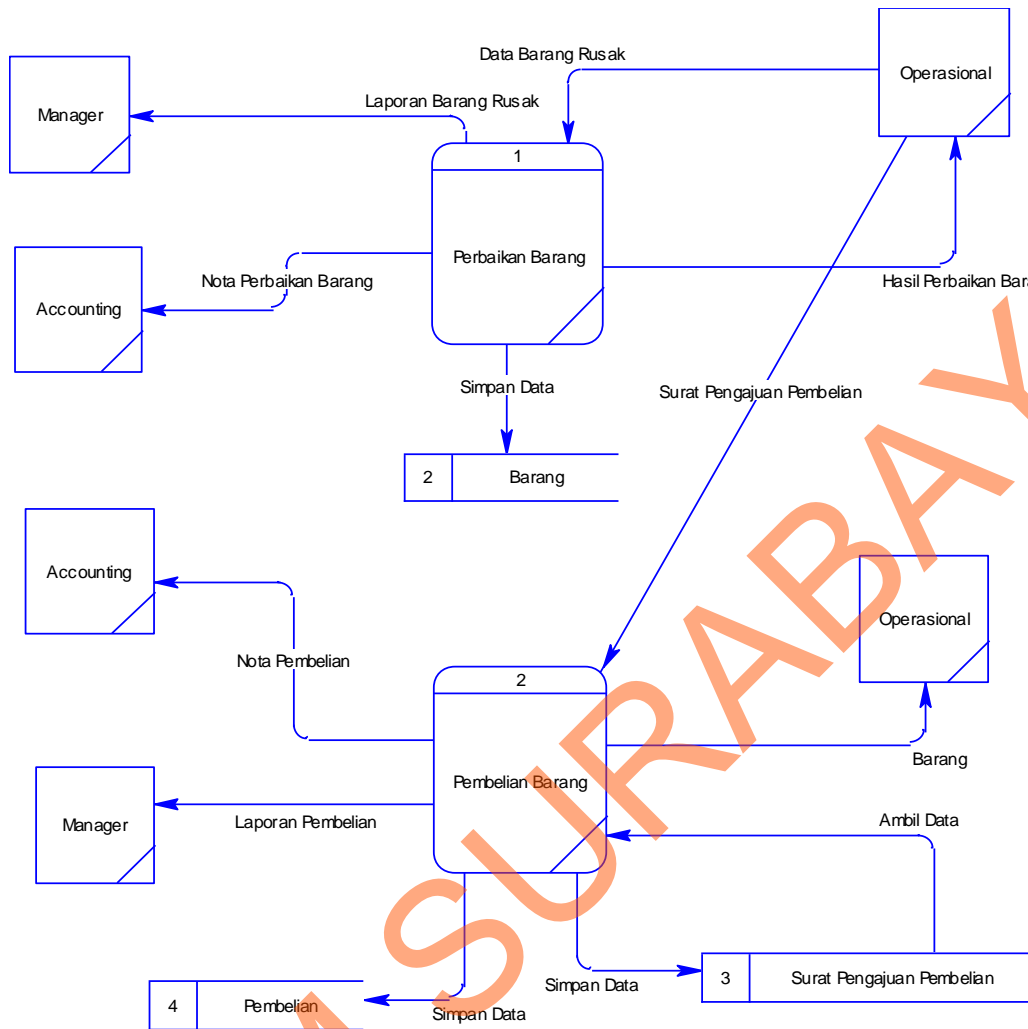
Context diagram dari sistem informasi manajemen aset menggambarkan secara umum yang terjadi pada CV. Padantya Alam Panglungan Surabaya. Pada context diagram tersebut melibatkan 3 *entity*, yaitu *operasional*, *accounting*, dan *manager*.



Gambar 4.5 *Context Diagram* Sistem Informasi Manajemen Aset

b. DFD Level 0 Sistem Informasi Manajemen Aset

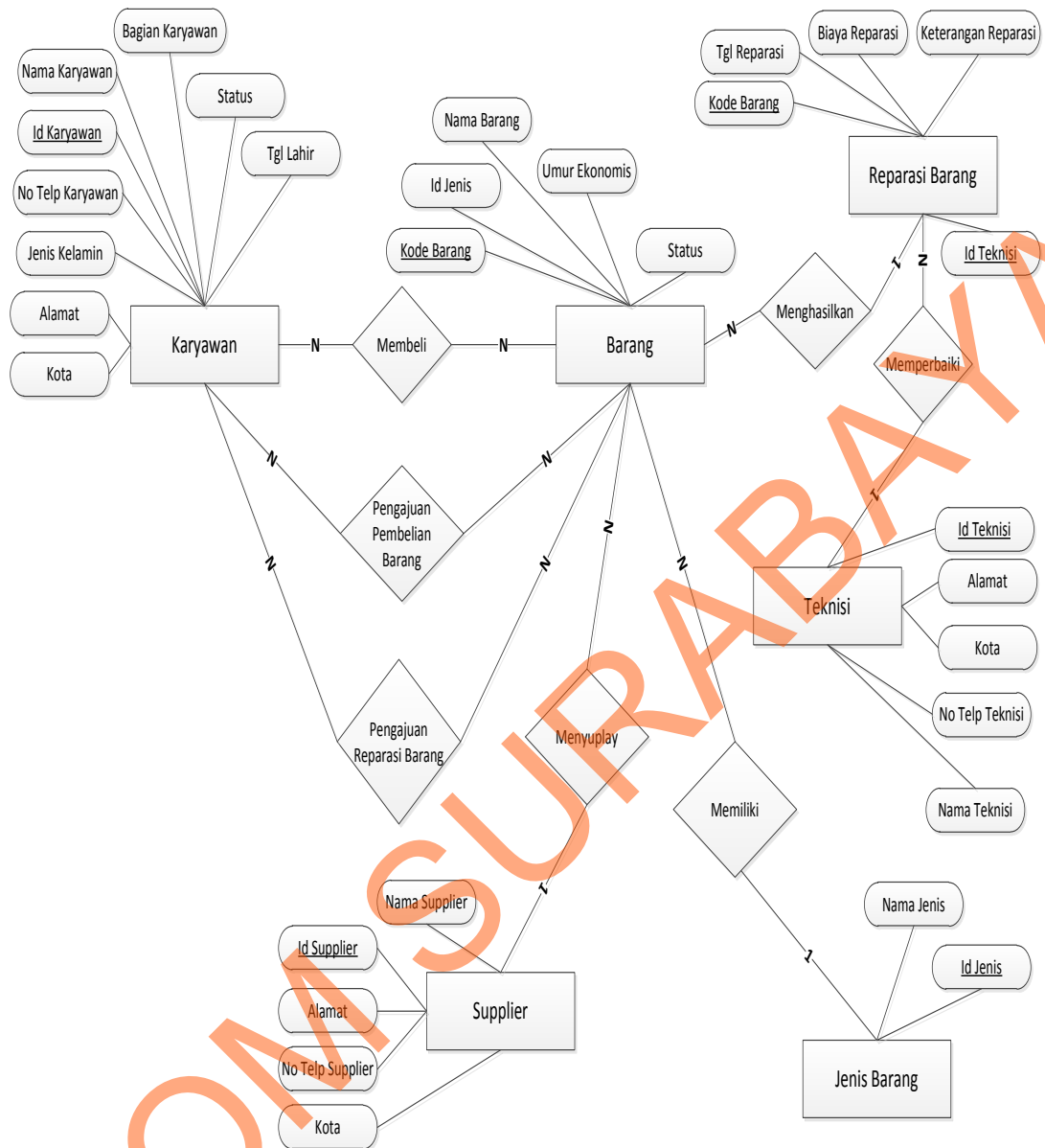
Setelah membuat *context diagram* dari sistem informasi manajemen aset di CV. Padantya Alam Panglungan, untuk selanjutnya *context diagram* tersebut akan dibagi menjadi sub proses yang lebih detail.



Gambar 4.6 DFD Level 0 Sistem Informasi Manajemen Aset

4.2.4 ERD (Entity Relationship Diagram)

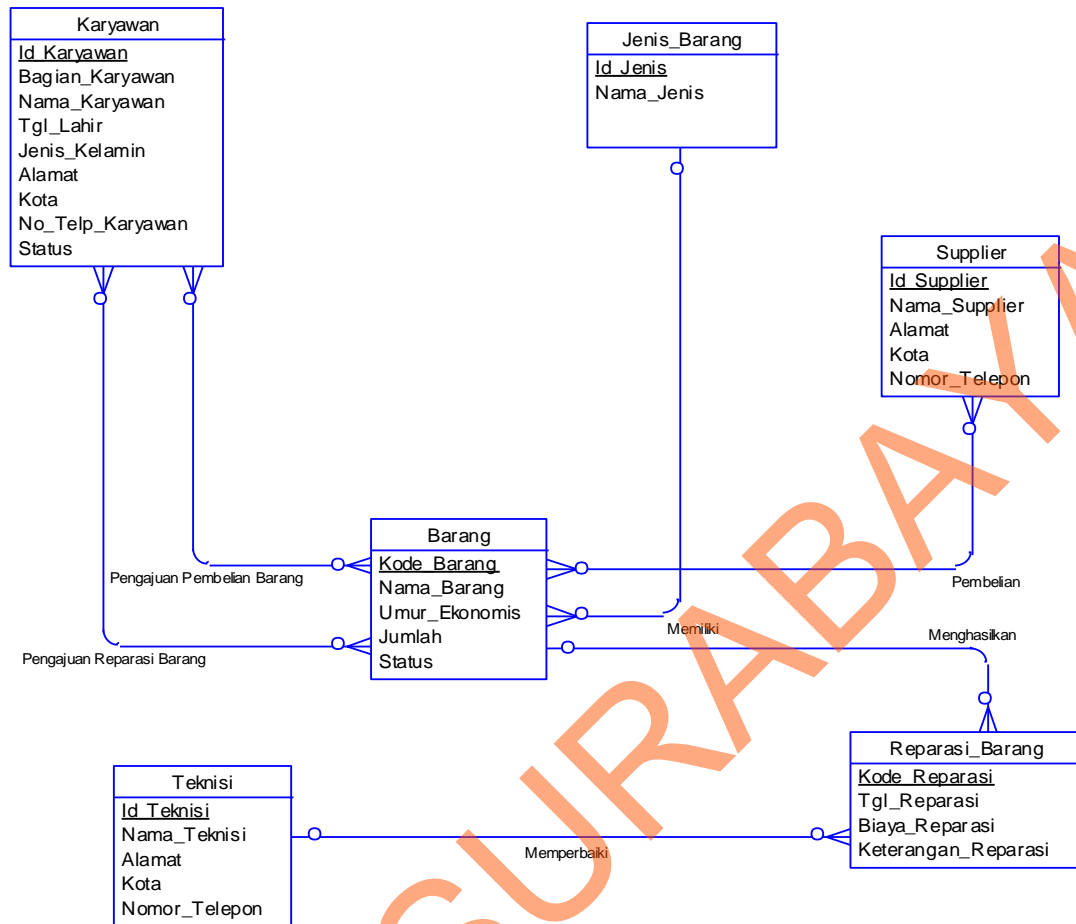
Entity Relationship Diagram merupakan gambaran struktur database dari sistem informasi manajemen aset yang telah dikembangkan. ERD dibagi menjadi dua, yaitu *Conceptual Data Model* atau secara logik dan *Physical Data Model* atau secara fisik.



Gambar 4.7 Entity Relationship Diagram

a. Conceptual Data Model (CDM)

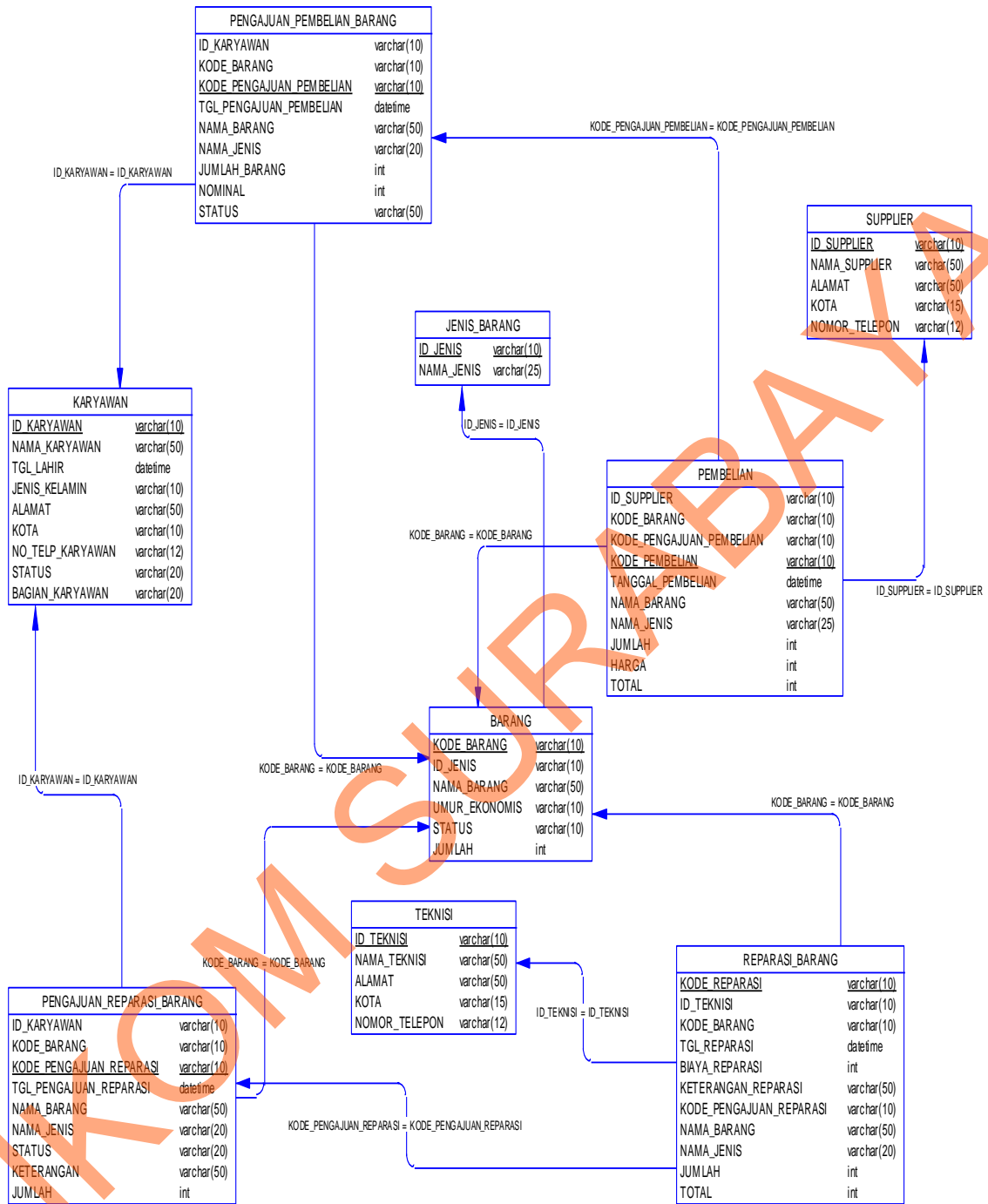
Terdapat 6 tabel yang terintegrasi secara logik. Berikut gambar *Conceptual Data Model*.



Gambar 4.8 *Conceptual Data Model* Sistem Informasi Manajemen Aset

b. Physical Data Model (PDM)

Physical Data Model (PDM) telah menunjukkan adanya relasi antar tabel. Berikut gambar Physical Data Model.



Gambar 4.9 Physical Data Model Sistem Informasi Manajemen Aset

4.2.5 Struktur Tabel

Struktur tabel pada sistem informasi manajemen aset CV. Padantya Alam

Panglungan Surabaya adalah sebagai berikut:

a. Tabel Barang

Nama Tabel : Barang

Primary Key : Kode_Barang

Foreign Key : Id_Jenis

Fungsi : Tabel untuk memasukkan data-data barang

Tabel 4.1 Barang

No.	Field	Type	Length	Key
1.	Kode_Barang	Varchar	10	Primary Key
2.	Nama_Barang	Varchar	50	
3.	Id_Jenis	Varchar	10	Foreign Key
4.	Umur_Ekonomis	Varchar	10	
5.	Jumlah	Int		
6.	Status	Varchar	10	

b. Tabel Jenis Barang

Nama Tabel : Jenis_Barang

Primary Key : Id_Jenis

Foreign Key : -

Fungsi : Tabel untuk memasukkan data-data jenis barang

Tabel 4.2 Jenis Barang

No.	Field	Type	Length	Key
1.	Id_Jenis	Varchar	10	Primary Key
2.	Nama_Jenis	Varchar	25	

c. Tabel Karyawan

Nama Tabel : Karyawan

Primary Key : Id_Karyawan

Foreign Key : -

Fungsi : Tabel untuk memasukkan data-data karyawan

Tabel 4.3 Karyawan

No.	Field	Type	Length	Key
1.	Id_Karyawan	Varchar	10	Primary Key
2.	Bagian_Karyawan	Varchar	20	
3.	Nama_Karyawan	Varchar	50	
4.	Tanggal_Lahir	Datetime		
5.	Jenis_Kelamin	Varchar	10	
6.	Alamat	Varchar	50	
7.	Kota	Varchar	10	
8.	Nomor_Telepon	Varchar	12	
9.	Status	Varchar	20	

d. Tabel Supplier

Nama Tabel : Supplier

Primary Key : Id_Supplier

Foreign Key : -

Fungsi : Tabel untuk memasukkan data-data supplier

Tabel 4.4 Supplier

No.	Field	Type	Length	Key
1.	Id_Supplier	Varchar	10	Primary Key
2.	Nama_Supplier	Varchar	50	
3.	Alamat	Varchar	50	
4.	Kota	Varchar	15	
5.	Nomor_Telepon	Varchar	12	

e. Tabel Teknisi

Nama Tabel : Teknisi

Primary Key : Id_Teknisi

Foreign Key : -

Fungsi : Tabel untuk memasukkan data-data teknisi

Tabel 4.5 Teknisi

No.	Field	Type	Length	Key
1.	Id_Teknisi	Varchar	10	Primary Key
2.	Nama_Teknisi	Varchar	50	
3.	Alamat	Varchar	50	
4.	Kota	Varchar	15	
5.	Nomor_Telepon	Varchar	12	

f. Tabel Pengajuan Pembelian Barang

Nama Tabel : Pengajuan_Pembelian

Primary Key : Kode_Pengajuan_Pembelian

Foreign Key : Kode_Barang, Id_Karyawan

Fungsi : Tabel untuk memasukkan data-data pengajuan pembelian barang

Tabel 4.6 Pengajuan Pembelian

No.	Field	Type	Length	Key
1.	Kode_Pengajuan_Pembelian	Varchar	10	Primary Key
2.	Kode_Barang	Varchar	10	Foreign Key
3.	Id_Karyawan	Varchar	10	Foreign Key
4.	Tanggal_Pengajuan_Pembelian	Datetime		
5.	Nama_Barang	Varchar	50	
6.	Nama_Jenis	Varchar	20	
7.	Jumlah_Barang	Int		
8.	Nominal	Int		
9.	Status	Varchar	50	

g. Tabel Pembelian

Nama Tabel : Pembelian

Primary Key : Kode_Pembelian

Foreign Key : Kode_Barang, Kode_Pengajuan_Pembelian, Id_Supplier

Fungsi : Tabel untuk memasukkan data-data pembelian

Tabel 4.7 Pembelian

No.	Field	Type	Length	Key
1.	Kode_Pembelian	Varchar	10	Primary Key
2.	Kode_Pengajuan_Pembelian	Varchar	10	Foreign Key
3.	Tanggal_Pembelian	Datetime		
4.	Id_Supplier	Varchar	10	Foreign Key
5.	Kode_Barang	Varchar	10	Foreign Key
6.	Nama_Barang	Varchar	50	
7.	Nama_Jenis	Varchar	25	
8.	Jumlah	Int		
9.	Harga	Int		
10.	Total	Int		

h. Tabel Pengajuan Reparasi Barang

Nama Tabel : Pengajuan_Reparasi_Barang

Primary Key : Kode_Pengajuan_Reparasi

Foreign Key : Kode_Barang, Id_Karyawan

Fungsi : Tabel untuk memasukkan data-data pengajuan reparasi barang

Tabel 4.8 Pengajuan Reparasi Barang

No.	Field	Type	Length	Key
1.	Kode_Pengajuan_Reparasi	Varchar	10	Primary Key
2.	Tanggal_Pengajuan_Reparasi	Datetime		
3.	Id_Karyawan	Varchar	10	Foreign Key
4.	Kode_Barang	Varchar	10	Foreign Key
5.	Nama_Barang	Varchar	50	
6.	Nama_Jenis	Varchar	20	
7.	Jumlah	Int		
8.	Keterangan	Varchar	50	
9.	Status	Varchar	20	

i. Tabel Reparasi Barang

Nama Tabel : Reparasi_Barang

Primary Key : Kode_Reparasi

Foreign Key : Kode_Pengajuan_Reparasi, Kode_Barang, Id_Teknisi

Fungsi : Tabel untuk memasukkan data-data reparasi barang

Tabel 4.9 Reparasi

No.	Field	Type	Length	Key
1.	Kode_Reparasi	Varchar	10	Primary Key
2.	Kode_Pengajuan_Reparasi	Varchar	10	Foreign Key
3.	Tanggal_Reparasi	Datetime		
4.	Id_Teknisi	Varchar	10	Foreign Key
5.	Kode_Barang	Varchar	10	Foreign Key
6.	Nama_Barang	Varchar	50	
7.	Nama_Jenis	Varchar	20	
8.	Jumlah	Int		
9.	Biaya_Reparasi	Int		
10.	Total	Int		
11.	Keterangan_Reparasi	Varchar	50	

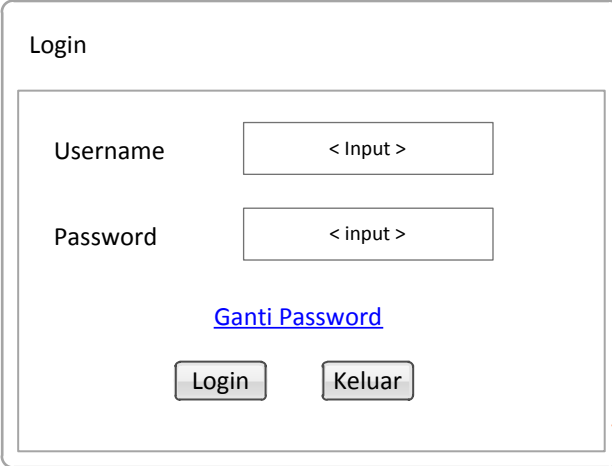
4.2.6 Desain I/O (Input/Output)

Desain *input/output* Sistem Informasi Manajemen Aset CV. Padantya

Alam Panglungan adalah sebagai berikut :

1. Form Login

Berikut ini merupakan tampilan dari *form login* yang berfungsi untuk *validasi user* yang menggunakan aplikasi.




The image shows a login form titled "Login". It contains two input fields: "Username" and "Password", each with a placeholder text "< Input >". Below the password field is a blue hyperlink labeled "Ganti Password". At the bottom of the form are two buttons: "Login" and "Keluar".

Gambar 4.10 Desain Form Login

2. Form Menu Utama

Berikut ini merupakan tampilan dari *form* menu utama yang berfungsi untuk masuk sebagai *username*.



The image shows a main menu form titled "Menu Utama". It contains a table with four columns: "Menu", "Master", "Transaksi", and "Keluar".

Gambar 4.11 Desain Form Menu Utama

3. Form Master Barang

Berikut ini merupakan tampilan dari *form* barang yang berfungsi untuk *input* data barang.

Gambar 4.12 Desain Form Master Barang

4. Form Master Jenis Barang

Berikut ini merupakan tampilan dari *form* jenis barang yang berfungsi untuk *input* data jenis barang.

Gambar 4.13 Desain Form Master Jenis Barang

5. Form Master Karyawan

Berikut ini merupakan tampilan dari *form* karyawan yang berfungsi untuk *input* data karyawan.

Form Karyawan

Id_Karyawan

Bagian_Karyawan

Nama_Karyawan

Tanggal_Lahir

Jenis_Kelamin

Alamat

Kota

Nomor_Telepon

Status

Password

Gambar 4.14 Desain Form Master Karyawan

6. Form Master Supplier

Berikut ini merupakan tampilan dari *form supplier* yang berfungsi untuk *input data supplier*.

Form Supplier

Id_Supplier

Nama_Supplier

Alamat

Kota

Nomor_Telepon

Gambar 4.15 Desain Form Master Supplier

7. Form Master Teknisi

Berikut ini merupakan tampilan dari *form* teknisi yang berfungsi untuk *input* data teknisi.

Gambar 4.16 Desain Form Master Teknisi


8. Form Transaksi Pengajuan Pembelian

Berikut ini merupakan tampilan dari *form* transaksi pengajuan pembelian yang berfungsi untuk *input* data pengajuan pembelian barang.

Gambar 4.17 Desain Form Transaksi Pengajuan Pembelian

9. Form Transaksi Pembelian

Berikut ini merupakan tampilan dari *form* transaksi pembelian yang berfungsi untuk *input* data pembelian barang.

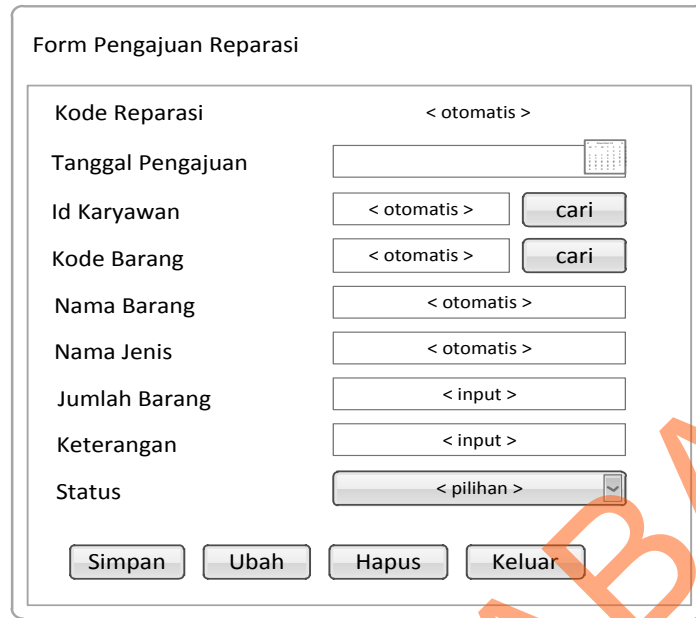


Form Pembelian	
Kode Pengajuan	< otomatis > <input type="button" value="cari"/>
Kode Pembelian	< otomatis >
Tanggal Pembelian	<input type="text"/> <input type="button" value="calendar"/>
Id Supplier	< otomatis > <input type="button" value="cari"/>
Kode Barang	< otomatis >
Nama Barang	< otomatis >
Nama Jenis	< otomatis >
Jumlah Barang	< otomatis >
Harga	< input >
Total	< otomatis >
<input type="button" value="Simpan"/> <input type="button" value="Ubah"/> <input type="button" value="Hapus"/> <input type="button" value="Keluar"/>	

Gambar 4.18 Desain Form Transaksi Pembelian

10. Form Transaksi PengajuanReparasi

Berikut ini merupakan tampilan dari *form* transaksi pengajuan reparasi yang berfungsi untuk *input* data pengajuan reparasi barang.



Form Pengajuan Reparasi

Kode Reparasi	< otomatis >
Tanggal Pengajuan	<input type="text"/>
Id Karyawan	< otomatis > <input type="button" value="cari"/>
Kode Barang	< otomatis > <input type="button" value="cari"/>
Nama Barang	< otomatis >
Nama Jenis	< otomatis >
Jumlah Barang	< input >
Keterangan	< input >
Status	< pilihan >

Gambar 4.19 Desain Form Transaksi Pengajuan Reparasi Barang

11. Form Transaksi Reparasi Barang

Berikut ini merupakan tampilan dari *form* transaksi reparasi barang yang berfungsi untuk *input* data reparasi barang.



Form Reparasi

Kode Pengajuan	< otomatis > <input type="button" value="cari"/>
Kode Reparasi	< otomatis >
Tanggal Reparasi	<input type="text"/>
Id Teknisi	< otomatis > <input type="button" value="cari"/>
Kode Barang	< otomatis >
Nama Barang	< otomatis >
Nama Jenis	< otomatis >
Jumlah Barang	< otomatis >
Biaya Reparasi	< input >
Total	< otomatis >
Keterangan	< input >

Gambar 4.20 Desain Form Transaksi Reparasi Barang

12. Laporan Barang

Berikut ini merupakan desain dari *output* laporan barang yang berfungsi untuk memberikan informasi tentang barang yang ada dan keadaan barang saat ini, baik inventaris *office* maupun inventaris lapangan.

Laporan Barang				
Kode Barang	Nama Barang	Umur Ekonomis	Jumlah	Status

Gambar 4.21 Desain Laporan Barang

13. Laporan Pembelian

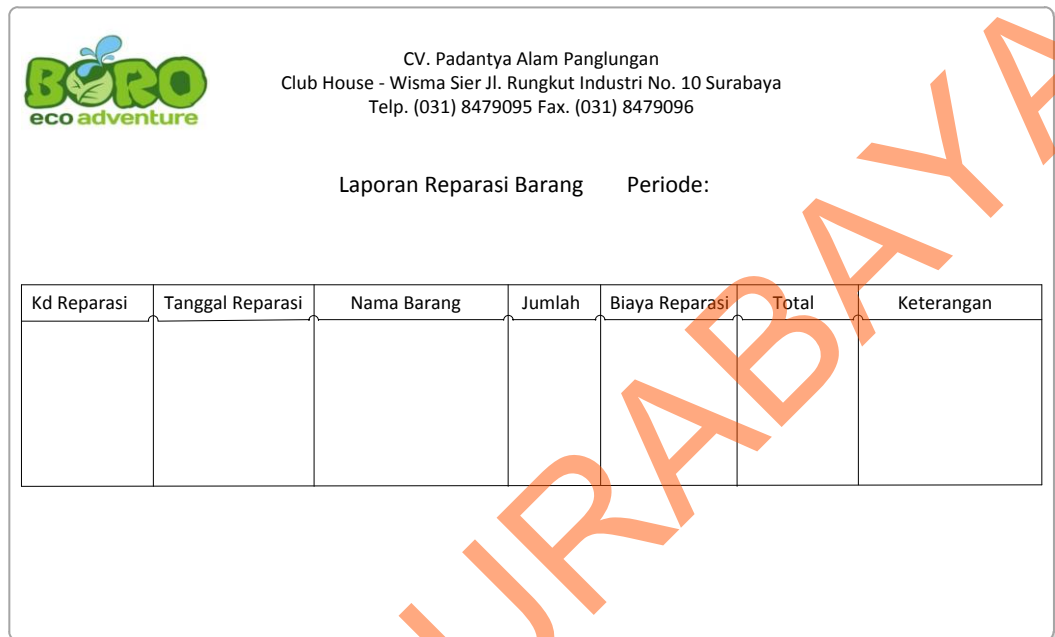
Berikut ini merupakan desain output laporan pembelian barang yang berfungsi untuk mengetahui pembelian dari setiap supplier.

Laporan Pembelian Barang Periode:						
Kd Pembelian	Tanggal Pembelian	Kode Barang	Nama Barang	Jumlah	Harga Barang	Total

Gambar 4.22 Desain Laporan Pembelian

14. Laporan Reparasi Barang

Berikut ini merupakan desain output laporan reparasi barang yang berfungsi untuk mengetahui reparasi barang dari setiap teknisi.



CV. Padantya Alam Panglungan
Club House - Wisma Sier Jl. Rungkut Industri No. 10 Surabaya
Telp. (031) 8479095 Fax. (031) 8479096

Laporan Reparasi Barang Periode:

Kd Reparasi	Tanggal Reparasi	Nama Barang	Jumlah	Biaya Reparasi	Total	Keterangan

Gambar 4.23 Desain Laporan Reparasi Barang

4.3 Implementasi Sistem

Implementasi sistem merupakan tahap pengujian dimana desain sistem dapat berjalan dengan baik. Desain form yang telah dibuat cukup sesuai untuk mengimplementasikan sistem, sehingga tidak membutuhkan banyak perubahan.

4.3.1 Kebutuhan Sistem

Pada tahap ini dijelaskan mengenai implementasi dari perangkat keras dan lunak yang harus dipersiapkan oleh pengguna. Untuk perangkat keras, minimal pengguna harus mempersiapkan spesifikasi sebagai berikut:

1. Intel Pentium 4 CPU 2.00 Ghz
2. Memory 512 MB RAM
3. VGA 64 GB

Kebutuhan minimum perangkat lunak untuk aplikasi ini adalah sebagai berikut:

1. Microsoft Windows 7
2. Microsoft SQL Server 2005
3. Visual Basic. Net 2005
4. Microsoft Office 2010

4.3.2 Penjelasan Pemakaian Aplikasi

Tahap ini merupakan langkah-langkah dari pemakaian aplikasi sistem informasi manajemen aset pada CV. Padantya Alam Panglungan Surabaya.

Berikut sub-sub pembahasan pemakaian aplikasi ini:

1. Form Menu

Form menu merupakan tampilan awal dari program ini. Pada form ini terdapat menu-menu yang bermanfaat dalam menjalankan sistem informasi manajemen aset oleh masing-masing karyawan.



Gambar 4.24 Tampilan *Form* Menu Utama

Pada menu utama terdapat beberapa sub menu seperti berikut ini:

- a. Home :berisi sub menu *login* yang berfungsi untuk masuk ke dalam aplikasi sesuai dengan *username* dan *password* masing-masing, *logout* yang berfungsi untuk keluar dari aplikasi sesuai dengan *username* dan *password* masing-masing.
- b. Menu :berisi master barang, master jenis barang, master karyawan, master supplier, dan master teknisi.
- c. Transaksi :berisi form-form transaksi yang meliputi transaksi pengajuan pembelian, transaksi pengajuan reparasi barang, transaksi pembelian, dan transaksi reparasi barang.
- d. Laporan :berisi form-form untuk menampilkan laporan yang disediakan oleh aplikasi ini.
- e. Keluar :untuk keluar membatalkan masuk kedalam aplikasi.

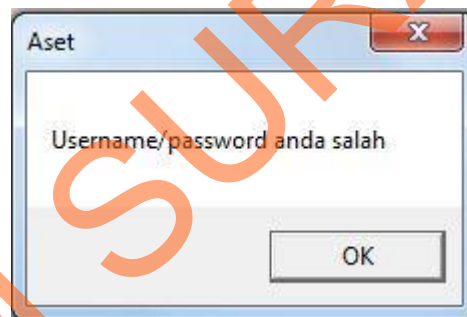
2. Form Login

Form login ini adalah *form* pertama yang muncul ketika program dijalankan. *User* harus menginputkan *username* dan *password* yang sesuai agar dapat masuk ke menu utama dari aplikasi ini. *Form Login* ini untuk mengontrol agar hanya orang yang berhak saja yang dapat mengakses aplikasi ini. Jika orang tersebut tidak memiliki wewenang, maka ia tidak akan dapat membuka aplikasi ini. Hal ini untuk menjaga keamanan data. Tampilan dari *form login* ini dapat dilihat sebagai pada gambar 4.25.



Gambar 4.25 Tampilan *Form Login*

Jika *username* dan *password* salah yang diinputkan *user* salah atau tidak sesuai maka akan muncul *message box* seperti pada gambar 4.26.



Gambar 4.26 *Message Box* pada *Form Login* Salah

Klik tombol OK dan user harus menginputkan kembali *username* dan *password* yang sesuai. Dan jika *username* dan *password* yang diinputkan *user* benar, maka akan muncul *message box* seperti pada gambar 4.27.



Gambar 4.27 *Message Box* pada *Form Login* Benar

Klik tombol OK maka akan muncul *message box* seperti pada gambar 4.28. Pada *message box* tersebut akan muncul *username* yang sesuai dengan *user* yang *login*.



Gambar 4.28 *Message Box* pada *Login Sebagai User Operasional*

Jika user ingin mengganti *password* maka klik pada tulis ganti *password* pada *form login* maka akan muncul *form* seperti gambar 4.29.



Gambar 4.29 Tampilan pada *Form Ganti Password*

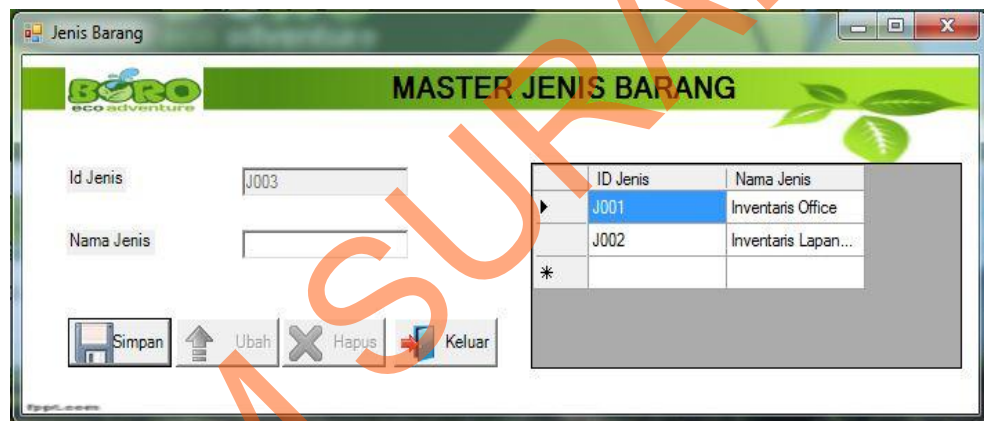
Klik tombol *login* maka *password* lama akan diganti dengan *password* baru sehingga *user* harus menggunakan *password* baru ketika *login* agar dapat masuk ke menu utama. Dan akan muncul *message box* seperti gambar 4.30.



Gambar 4.30 *Message Box* pada *Form Password* Sudah Diganti

3. Form Jenis Barang

Implementasi pada form jenis barang ini dapat dilihat pada gambar 4.31.



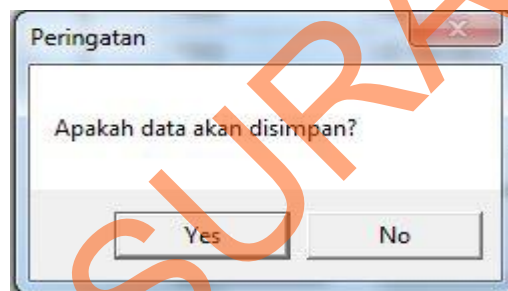
Gambar 4.31 Tampilan *Form Master* Jenis Barang

Pada form ini *user* dapat *maintenance* data jenis barang. Pada *textbox* *id_jenis* akan terisi secara otomatis sesuai dengan jumlah yang ada di *database* dengan penambahan dua karakter bertipe *string*. Kemudian *textbox* *nama_jenis* diinputkan sesuai jenis barang. Dalam *form* ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.

2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data jenis barang yang tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form data barang.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar yang aktif saja. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.32 yang akan menanyakan kembali apakah data akan disimpan.



Gambar 4.32 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.33 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.



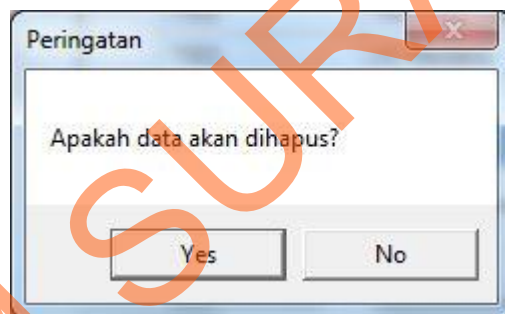
Gambar 4.33 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data jenis barang yang ada, maka klik data jenis barang yang ada dikolom *gridview* kemudian

button ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.34 dan *message box* hapus seperti gambar 4.35.



Gambar 4.34 *Message Box* pada *Button* Ubah



Gambar 4.35 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.36 dan *message box* hapus seperti gambar 4.37 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.36 *Message Box* pada *Button* Ubah



Gambar 4.37 Message Box pada Button Hapus

4. Form Barang

Implementasi pada form barang ini dapat dilihat pada gambar 4.38.

Kode Barang	Nama Barang	Id Jenis
B001	White Board	J001
B002	Meja Kaca	J001
B003	Meja Receptionist	J001
B004	Kursi	J001
B005	Laptop	J001
B006	Komputer	J001

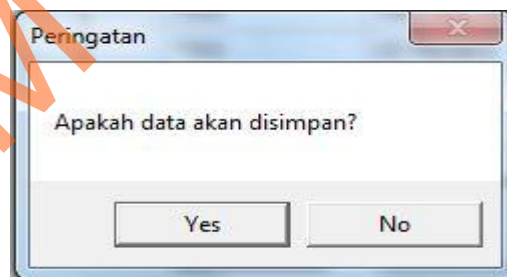
Gambar 4.38 Tampilan Form Master Barang

Pada form ini *user* dapat *maintenance* data barang. Pada *textbox* kode_barang akan terisi secara otomatis sesuai dengan jumlah yang ada di *database* dengan penambahan dua karakter bertipe *string*. Kemudian *textbox* nama_barang, *textbox* umur_ekonomis, dan *textbox* Jumlah harus diisi sesuai dengan data yang dibutuhkan. Untuk *combobox* id_jenis dipilih dari klik *button* cari dan akan muncul data jenis barang sesuai data yang ada

pada tabel jenis barang, *combobox* Status dipilih berdasarkan kondisi barang yang diinputkan. Dalam form ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data barang yang tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form data barang.
5. *Button* cari, berfungsi untuk menampilkan data yang akan dicari.

Pada saat pertama kali *form* ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.39 yang akan menanyakan kembali apakah data akan disimpan.



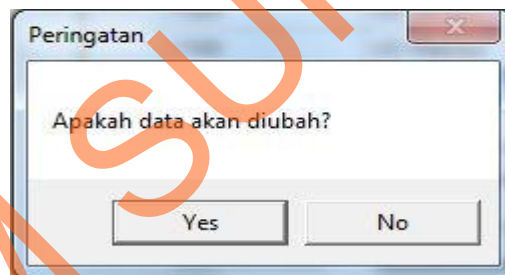
Gambar 4.39 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.40 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.

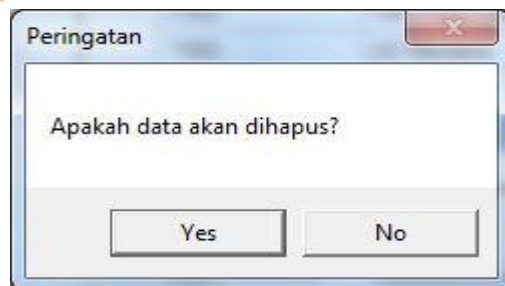


Gambar 4.40 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.41 dan *message box* hapus seperti gambar 4.42.



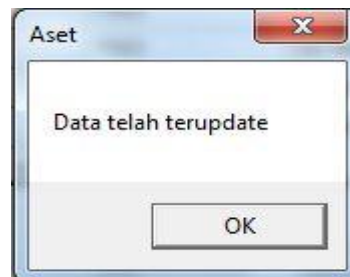
Gambar 4.41 *Message Box* pada *Button* Ubah



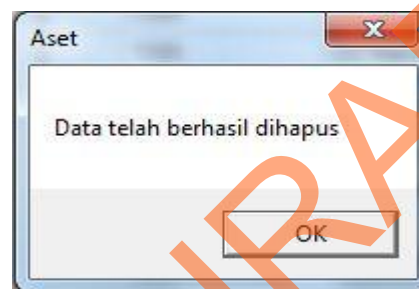
Gambar 4.42 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.43 dan *message box*

hapus seperti gambar 4.44 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.43 *Message Box* pada *Button* Ubah



Gambar 4.44 *Message Box* pada *Button* hapus

5. Form Karyawan

Implementasi pada form karyawan ini dapat dilihat pada gambar 4.45.

The screenshot shows a web application window titled 'Karyawan' with a green header 'MASTER KARYAWAN'. The form contains the following fields:

- Id Karyawan:** K004
- Bagian Karyawan:** Accounting
- Nama Karyawan:** Nada
- Tanggal Lahir:** 22 Februari 1991
- Jenis Kelamin:** Perempuan
- Alamat:** Kedungbaruk
- Kota:** Surabaya
- Nomor Telepon:** 087852749792
- Status:** Aktif
- Password:** [Redacted]

On the right, there is a table with the following data:

	Id Karyawan	Bagian Karyawan	Nama Karyawan	Tanggal
▶	K001	Manager	Agus Sugiarto	01/03/19
	K002	Accounting	Wulan	01/01/19
	K003	Operasional	Wartib	01/01/19
*				

At the bottom of the form, there are four buttons: Simpan, Ubah, Hapus, and Keluar.

Gambar 4.45 Tampilan *Form Master* Karyawan

Pada form ini user *maintenance* data karyawan. Pada *textbox* *id_karyawan* akan terisi secara otomatis sesuai dengan jumlah yang ada di database dengan penambahan dua karakter bertipe string. Untuk *Datetimepicker* *tanggal_lahir* diisi berdasarkan tanggal lahir karyawan tersebut. Kemudian *textbox* *nama_karyawan*, *textbox* *alamat*, *textbox* *kota*, dan *textbox* *nomor_telepon*, *textbox* *password* harus diisi sesuai dengan data yang dibutuhkan. Untuk *combobox* *bagian_karyawan* dipilih berdasarkan posisi karyawan diperusahaan tersebut, *combobox* *jenis_kelamin* dipilih dari jenis kelamin masing-masing karyawan, *combobox* *status* dipilih berdasarkan status karyawan saat ini, aktif atau tidak aktif. Dalam form ini terdapat beberapa *button*, yaitu :

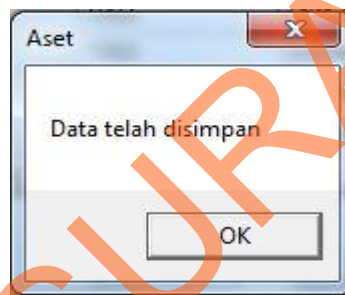
1. *Button* *simpan*, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* *ubah*, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* *hapus*, berfungsi untuk menghapus data karyawan yang sudah tidak diperlukan lagi.
4. *Button* *keluar*, berfungsi untuk keluar dari form data karyawan.

Pada saat pertama kali form ini dijalankan, maka hanya *button* *simpan* dan *button* *keluar* saja yang aktif. Pada saat tekan *button* *simpan* maka muncul *message box* seperti gambar 4.46 yang akan menanyakan kembali apakah data akan disimpan.



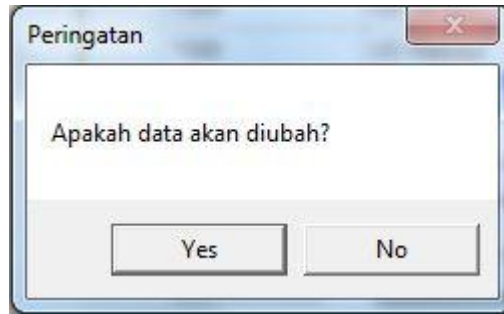
Gambar 4.46 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.47 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.

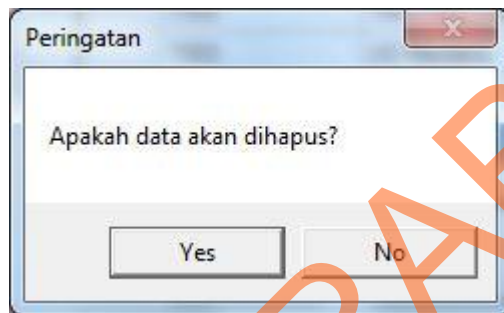


Gambar 4.47 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.48 dan *message box* hapus seperti gambar 4.49.



Gambar 4.48 *Message Box* pada *Button* Ubah



Gambar 4.49 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.50 dan *message box* hapus seperti gambar 4.51 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.50 *Message Box* pada *Button* Ubah



Gambar 4.51 Message Box pada Button hapus

6. Form Supplier

Implementasi pada *form supplier* ini dapat dilihat pada gambar 4.52.

Id Supplier	Nama Supplier	Alamat	Kota
S001	Toko Sampoema	Jl. Nginden Timur	Surabaya
S002	UD Maju	Jl. Kali Rungkut	Surabaya
S003	Toko Merpati	Jl. Ahmad Yani	Surabaya

Gambar 4.52 Tampilan Form Master Supplier

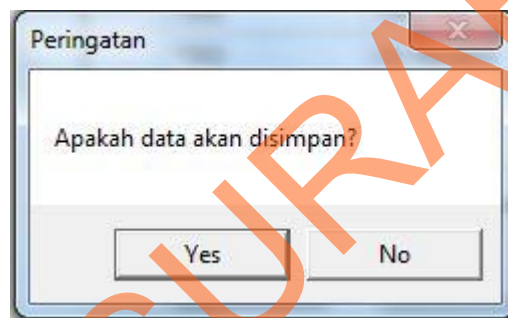
Pada form ini user *maintenance data supplier*. Pada *textbox id_supplier* akan terisi secara otomatis sesuai dengan jumlah yang ada di database dengan penambahan dua karakter bertipe string. Kemudian *textbox nama_supplier*, *textbox alamat*, *textbox kota*, dan *textbox nomor_telepon*, harus diisi sesuai dengan data yang dibutuhkan. Dalam form ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* ubah, berfungsi untuk menyimpan data yang telah *diupdate* oleh

user.

3. *Button* hapus, berfungsi untuk menghapus data *supplier* yang sudah tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form data *supplier*.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.53 yang akan menanyakan kembali apakah data akan disimpan.



Gambar 4.53 *Message Box* pada *Button* Simpan

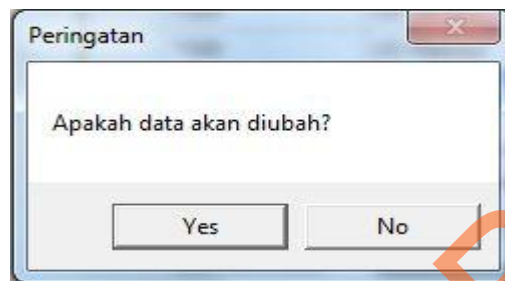
Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.54 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.



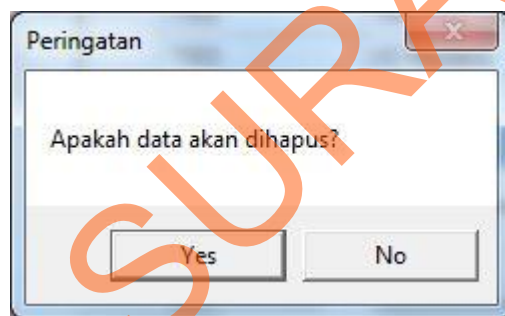
Gambar 4.54 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan

button hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. Message box ubah seperti gambar 4.55 dan *message box* hapus seperti gambar 4.56.

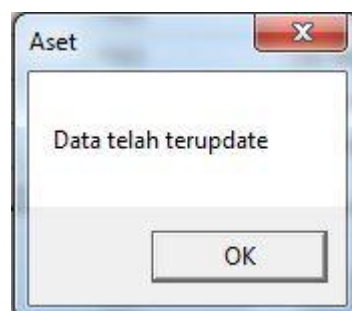


Gambar 4.55 Message Box pada Button Ubah



Gambar 4.56 Message Box pada Button Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.57 dan *message box* hapus seperti gambar 4.58 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.57 Message Box pada Button Ubah



Gambar 4.58 Message Box pada Button hapus

7. Form Teknisi

Implementasi pada form teknisi ini dapat dilihat pada gambar 4.59.

Id_Teknisi	Nama_Teknisi	Alamat	Kota
T001	UD Sejahtera	Jl. Jendral Sudim...	Sidoarjo
T002	Toko Maju Mundur	Jl. Delta Sari	Surabaya
T003	UD Merdeka	Jl. Basuki Rahmad	Surabaya
T004	UD Semeru	Jl. Hasanuddin	Surabaya
T005	UD Makmur	Blauran	Surabaya

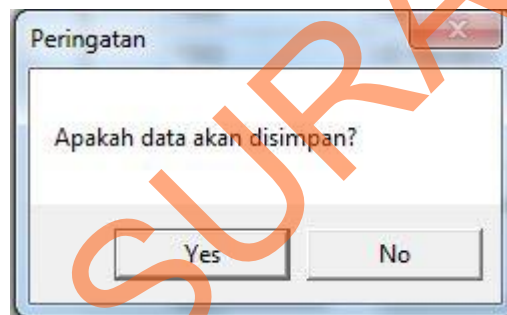
Gambar 4.59 Tampilan Form Master Teknisi

Pada form ini user *maintenance* data teknisi. Pada *textbox* id teknisi akan terisi secara otomatis sesuai dengan jumlah yang ada di database dengan penambahan dua karakter bertipe *string*. Kemudian *textbox* nama_teknisi, *textbox* alamat, *textbox* kota, dan *textbox* nomor_telepon, harus diisi sesuai dengan data yang dibutuhkan. Dalam form ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.

2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data teknisi yang sudah tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form data teknisi.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.60 yang akan menanyakan kembali apakah data akan disimpan.



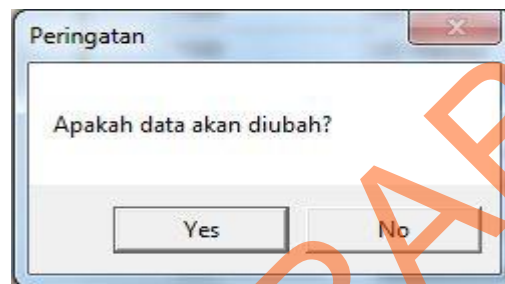
Gambar 4.60 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.61 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.

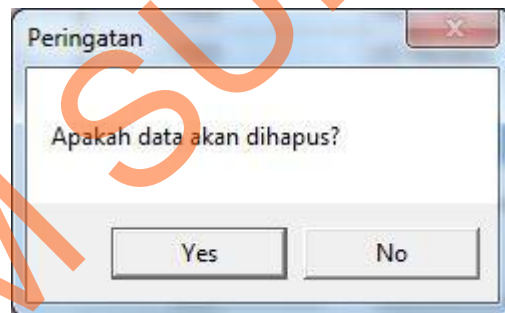


Gambar 4.61 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.62 dan *message hapus* seperti gambar 4.63.



Gambar 4.62 *Message Box* pada *Button* Ubah

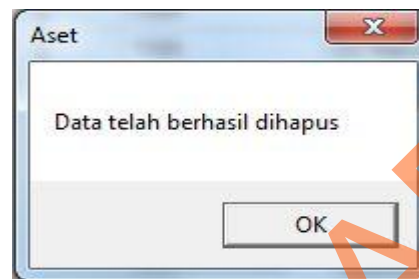


Gambar 4.63 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.64 dan *message box* hapus seperti gambar 4.65 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.64 *Message Box* pada *Button* Ubah



Gambar 4.65 *Message Box* pada *Button* hapus

8. Form Transaksi Pengajuan Pembelian

Implementasi pada form transaksi pengajuan pembelian ini dapat dilihat pada gambar 4.66.

Kode Pengajuan Pembelian	Tanggal Pengajuan Pembelian	Id Karyawan
PP001	03/11/2012	K003
PP002	03/11/2012	K002
PP003	05/11/2012	K003
PP004	05/11/2012	K003
PP005	05/11/2012	K002

Gambar 4.66 Tampilan *Form* Transaksi Pengajuan Pembelian

Pada form ini user *maintenance* data pengajuan pembelian. Pada *label* kode_pengajuan_pembelian akan terisi secara otomatis sesuai dengan jumlah yang ada di database dengan penambahan dua karakter bertipe string. *Textbox* id_karyawan juga terisi secara otomatis sesuai dengan yang login sebagai user tersebut. Kemudian *textbox* kode_barang akan muncul secara otomatis jika *button* cari dipilih dan *textbox* nama_barang, *textbox* nama_jenis akan muncul secara otomatis mengikuti isi dari kode_barang yang dipilih. *Textbox* jumlah dan *textbox* nominal diinputkan sesuai data yang dibutuhkan. *Combobox* status dipilih dari pilihan data tersedia atau data tidak tersedia/habis. Dalam form ini terdapat beberapa *button*, yaitu:

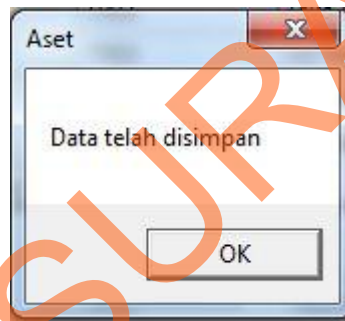
1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data transaksi pengajuan pembelian barang yang sudah tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form transaksi pengajuan pembelian.
5. *Button* cari, berfungsi untuk menampilkan data yang akan dicari.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.67 yang akan menanyakan kembali apakah data akan disimpan.



Gambar 4.67 *Message Box* pada *Button Simpan*

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.68 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.

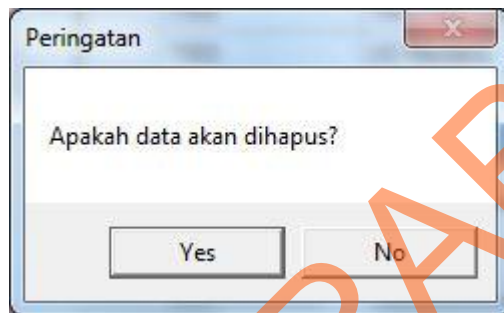


Gambar 4.68 *Message Box* pada *Button Simpan*

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.69 dan *message box* hapus seperti gambar 4.70.

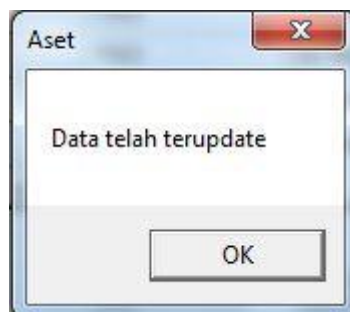


Gambar 4.69 *Message Box* pada *Button* Ubah



Gambar 4.70 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.71 dan *message box* seperti gambar 4.72 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.71 *Message Box* pada *Button* Ubah



Gambar 4.72 Message Box pada Button Hapus

9. Form Transaksi Pembelian

Implementasi pada form transaksi pengajuan pembelian ini dapat dilihat pada gambar 4.73.

Kode_Pembelian	Kode_Pengajuan_f	Tanggal_Pembelian	Id_S
P001	PP001	03/11/2012	S001
P002	PP002	05/11/2012	S002
P003	PP003	05/11/2012	S003
P004	PP004	05/11/2012	S003
P005	PP005	05/11/2012	S004
*			

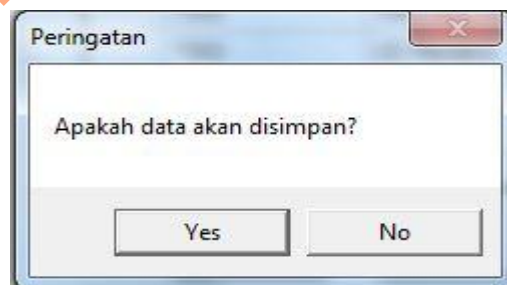
Gambar 4.73 Tampilan Form Transaksi Pembelian

Pada form ini user *maintenance* data transaksi pembelian. Pada *textbox* kode_pengajuan_pembelian akan muncul secara otomatis jika *button* cari dipilih dan *textbox* kode_barang, *textbox* nama_barang, *textbox* nama_jenis dan *textbox* jumlah akan muncul secara otomatis mengikuti isi dari kode_pengajuan_pembelian yang dipilih. *Textbox* id_supplier terisi jika

button cari dipilih. Untuk *textbox* harga diinputkan sesuai yang dibutuhkan dan *textbox* total akan terisi sesuai dengan total jumlah yang dikalikan dengan *textbox* harga. Dalam form ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data transaksi pembelian barang yang sudah tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form data transaksi pembelian.
5. *Button* cari, berfungsi untuk menampilkan data yang akan dicari.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.74 yang akan menanyakan kembali apakah data akan disimpan.



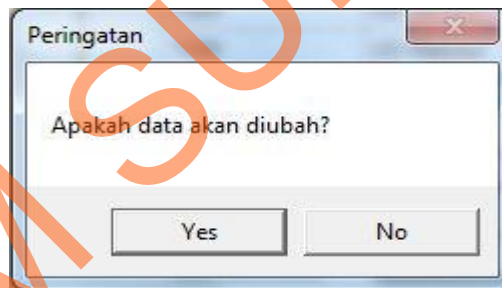
Gambar 4.74 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.75 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.



Gambar 4.75 *Message Box* pada *Button Simpan*

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.76 dan *message box* hapus seperti gambar 4.77.



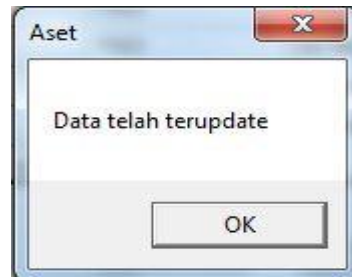
Gambar 4.76 *Message Box* pada *Button Ubah*



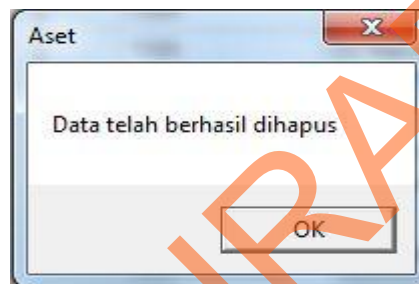
Gambar 4.77 *Message Box* pada *Button Hapus*

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.78 dan *message box*

hapus seperti gambar 4.79 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.78 Message Box pada Button Ubah



Gambar 4.79 Message Box pada Button Hapus

10. Form Transaksi Pengajuan Reparasi

Implementasi pada form transaksi pengajuan pembelian ini dapat dilihat pada gambar 4.80.

Kode Pengajuan Reparasi	Tanggal Pengajuan Reparasi	Id_Karyawan
PR001	03/11/2012	K003
PR002	03/11/2012	K003
PR003	05/11/2012	K003
PR004	05/11/2012	K003

Gambar 4.80 Tampilan Form Transaksi Pengajuan Reparasi

Pada form ini user *maintenance* data pengajuan reparasi. Pada *label* *kode_pengajuan_reparasi* akan terisi secara otomatis sesuai dengan jumlah yang ada di database dengan penambahan dua karakter bertipe string. *Textbox* *id_karyawan* juga terisi secara otomatis sesuai dengan yang login sebagai user tersebut. Kemudian *textbox* *kode_barang* akan muncul secara otomatis jika *button* *cari* dipilih dan *textbox* *nama_barang*, *textbox* *nama_jenis* akan muncul secara otomatis mengikuti isi dari *kode_barang* yang dipilih. *Textbox* *jumlah* dan *textbox* *keterangan* diinputkan sesuai data yang dibutuhkan. *Combobox* *status* dipilih dari pilihan data rusak atau baik. Dalam form ini terdapat beberapa *button*, yaitu:

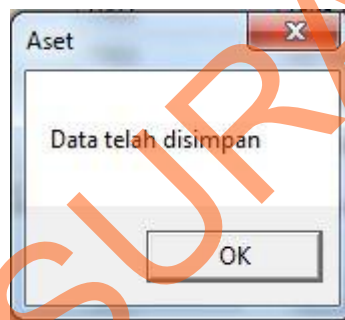
1. *Button* *simpan*, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* *ubah*, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* *hapus*, berfungsi untuk menghapus data transaksi pengajuan reparasi barang yang sudah tidak diperlukan.
4. *Button* *keluar*, berfungsi untuk keluar dari form transaksi pengajuan reparasi.
5. *Button* *cari*, berfungsi untuk menampilkan data yang akan dicari.

Pada saat pertama kali form ini dijalankan, maka hanya *button* *simpan* dan *button* *keluar* saja yang aktif. Pada saat tekan *button* *simpan* maka muncul *message box* seperti gambar 4.81 yang akan menanyakan kembali apakah data akan disimpan.



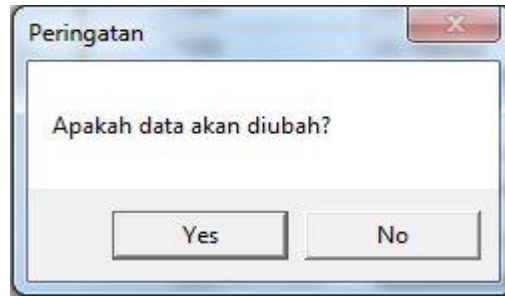
Gambar 4.81 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.82 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.



Gambar 4.82 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.83 dan *message box* hapus seperti gambar 4.84.

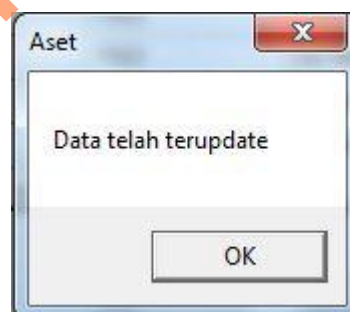


Gambar 4.83 *Message Box* pada *Button* Ubah



Gambar 4.84 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.85 dan *message box* hapus seperti gambar 4.86 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.85 *Message Box* pada *Button* Ubah



Gambar 4.86 Message Box pada Button Hapus

11. Form Transaksi Reparasi

Implementasi pada form transaksi pengajuan pembelian ini dapat dilihat pada gambar 4.87.

Kode_Reparasi	Kode_Pengajuan_f	Tanggal_Reparasi	Id_Te
R001	PR001	03/11/2012	T001
R002	PR002	05/11/2012	T002
R003	PR003	05/11/2012	T004
R004	PR004	05/11/2012	T004
R005	PR005	06/11/2012	T002
R006	PR006	06/11/2012	T003
R007	PR007	07/11/2012	T005

Gambar 4.87 Tampilan Form Transaksi Reparasi

Pada form ini user *maintenance* data transaksi reparasi. Pada *textbox* kode_pengajuan_reparasi akan muncul secara otomatis jika *button* cari dipilih dan *textbox* kode_barang, *textbox* nama_barang, *textbox* nama_jenis dan *textbox* jumlah akan muncul secara otomatis mengikuti isi dari kode_pengajuan_reparasi yang dipilih. *Textbox* id_teknisi terisi jika *button* cari dipilih. Untuk *textbox* biaya diinputkan sesuai yang dibutuhkan

dan *textbox* total akan terisi sesuai dengan total jumlah yang dikalikan dengan *textbox* biaya. Dalam form ini terdapat beberapa *button*, yaitu:

1. *Button* simpan, berfungsi untuk menyimpan data yang dientry oleh *user*.
2. *Button* ubah, berfungsi untuk menyimpan data yang telah diupdate oleh *user*.
3. *Button* hapus, berfungsi untuk menghapus data transaksi reparasi barang yang sudah tidak diperlukan lagi.
4. *Button* keluar, berfungsi untuk keluar dari form transaksi reparasi.
5. *Button* cari, berfungsi untuk menampilkan data yang akan dicari.

Pada saat pertama kali form ini dijalankan, maka hanya *button* simpan dan *button* keluar saja yang aktif. Pada saat tekan *button* simpan maka muncul *message box* seperti gambar 4.88 yang akan menanyakan kembali apakah data akan disimpan.



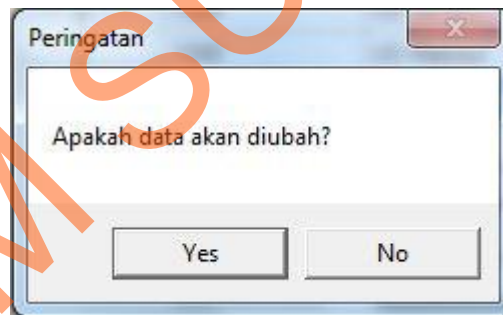
Gambar 4.88 *Message Box* pada *Button* Simpan

Jika menyetujui pesan tersebut akan disimpan, maka tekan OK dan akan muncul *message box* seperti gambar 4.89 yang akan memberitahukan bahwa data telah disimpan, dan tekan OK.



Gambar 4.89 *Message Box* pada *Button* Simpan

Jika *username* ingin mengubah atau menghapus data barang yang ada, maka klik data barang yang ada dikolom *gridview* kemudian *button* ubah dan *button* hapus akan aktif. Pada saat tekan *button* ubah atau *button* hapus maka muncul *message box* yang menanyakan apakah data benar-benar ingin diubah atau dihapus. *Message box* ubah seperti gambar 4.90 dan *message box* hapus seperti gambar 4.91.

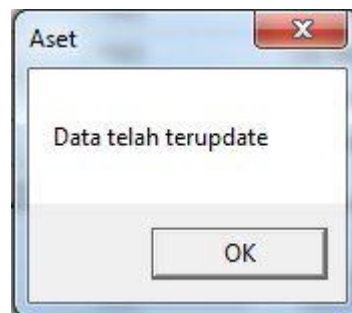


Gambar 4.90 *Message Box* pada *Button* Ubah

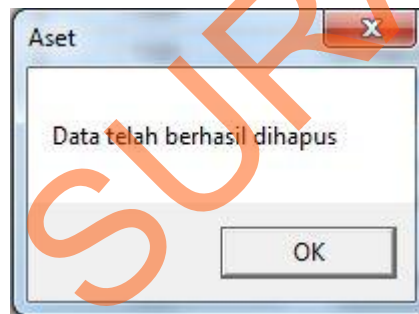


Gambar 4.91 *Message Box* pada *Button* Hapus

Jika menyetujui pesan tersebut akan diubah atau dihapus, maka tekan OK dan akan muncul *message box* ubah seperti gambar 4.92 dan *message box* hapus seperti gambar 4.93 yang akan memberitahukan bahwa data telah diubah atau dihapus, dan tekan OK.



Gambar 4.92 *Message Box* pada *Button* Ubah



Gambar 4.93 *Message Box* pada *Button* Hapus

12. Laporan Barang

Laporan ini berfungsi untuk memberikan informasi mengenai barang yang tersedia dan kondisi barang tersebut. Didalam laporan barang ini, terdapat kode_barang, nama_barang, umur_ekonomis, jumlah_barang, dan status barang dalam kondisi baik atau rusak yang dikelompokkan berdasarkan nama jenis barang yaitu inventaris office dan inventaris lapangan. Untuk laporan barang dapat dilihat pada gambar 4.94.



CV. Padantya Alam Panglungan
 Club House Lt.2 Wisma Sier Jl. Rungkut Industri No. 10
 Telp. (031) 8479095 Fax. (031) 8479096 Email: info@bororafting.com
 SURABAYA

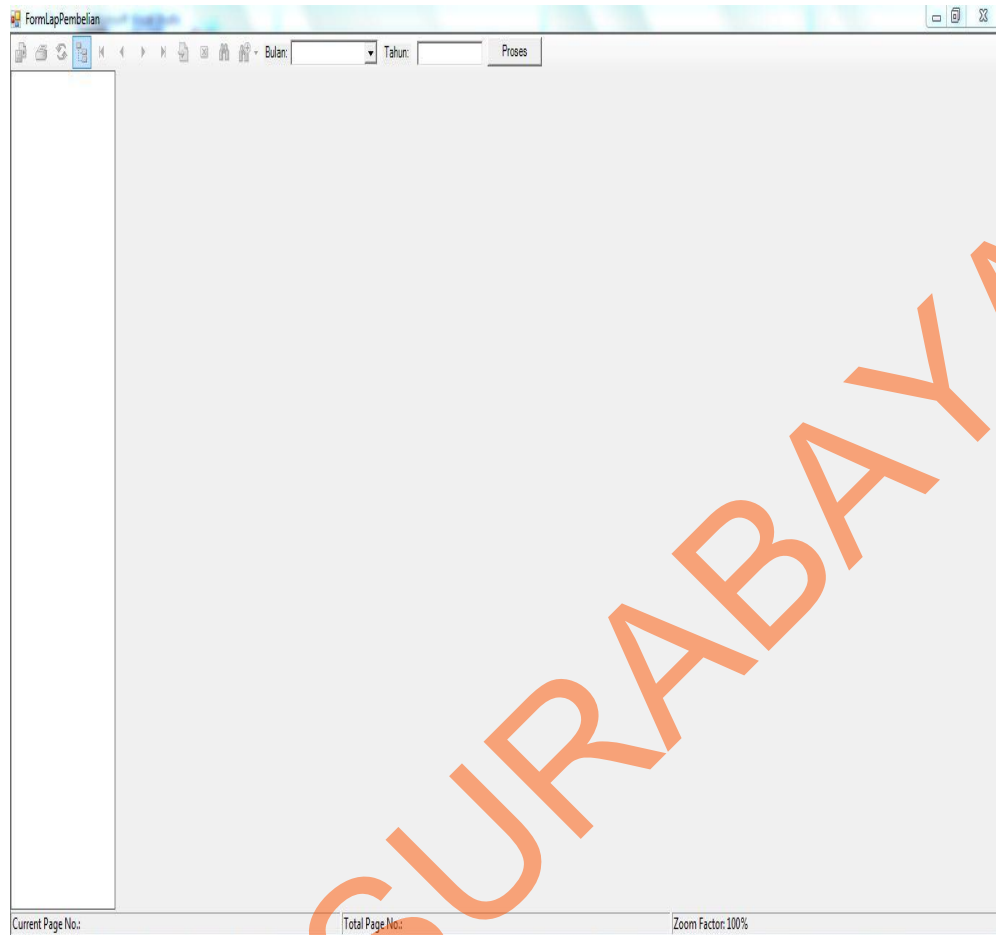
Laporan Barang

Kode Barang	Nama Barang	Umur Ekonomis	Jumlah	Status
Inventaris Lapangan				
B013	Perahu	2 Tahun	11	Baik
B014	Kayak	1 Tahun	1	Baik
B015	Dayung	1 Tahun	62	Baik
B016	Pelampung Tamu	1 Tahun	71	Baik
B017	Pelampung Guide	1 Tahun	14	Baik
B018	Helm Tamu	2 Tahun	48	Baik
B019	Helm Guide	2 Tahun	8	Baik
B020	Drybag Hitam	1 Tahun	6	Baik
B021	Jetset	2 Tahun	3	Baik
Inventaris Office				
B001	White Board	5 Tahun	1	Baik
B002	Meja Kaca	1 tahun	1	Baik
B003	Meja Receptionist	2 Tahun	1	Baik
B004	Kursi	2 Tahun	10	Baik
B005	Laptop	3 Tahun	1	Baik

Gambar 4.94 Laporan Barang

13. Laporan Pembelian Barang

Laporan ini berfungsi untuk memberikan informasi mengenai barang apa saja yang dibeli oleh perusahaan dari pengajuan barang yang tidak tersedia saat ini atau dalam kondisi perusahaan yang kekurangan barang. Didalam laporan pembelian terdapat kode_pembelian, tanggal_pembelian, kode_barang, nama_barang, jumlah, dan harga_barang yang telah dibeli oleh perusahaan dari supplier. Laporan ini dibuat berdasarkan periode bulan dan tahun dari setiap laporan pembelian barang yang terjadi. Gambar 4.95 adalah proses menampilkan laporan pembelian barang.



Gambar 4.95 *Form* Laporan Pembelian Barang

Pada *combobox* bulan dipilih berdasarkan bulan untuk laporan yang ada dalam data laporan pembelian barang. *Textbox* tahun diisi secara manual sesuai dengan tahun yang diinginkan. Untuk *textbox* tahun jika diinputkan selain angka maka laporan tidak bisa menampilkan data, karena *textbox* tahun hanya bisa diinputkan berupa angka. Untuk laporan pembelian barang dapat dilihat pada gambar 4.96.



CV. Padantya Alam Panglungan
 Club House Lt.2 Wisma Sier Jl. Rungkut Industri No. 10
 Telp. (031) 8479095 Fax. (031) 8479096 Email: info@bororafting.com
 SURABAYA

Laporan Pembelian Barang Periode: November 2012

Kd_Pembelian	Tgl_Pembelian	Kode Barang	nama_barang	jumlah	Harga Barang	total
P001	03-Nop-2012	B020	Drybag Hitam	2	3.000	6.000
P002	05-Nop-2012	B012	White Board	1	3.000	3.000
P003	05-Nop-2012	B018	Helm Tamu	6	85.000	510.000
P004	05-Nop-2012	B021	Jetset	3	60.000	180.000
P005	05-Nop-2012	B006	Komputer	1	4.000.000	4.000.000
				13		4.699.000

Gambar 4.96 Laporan Pembelian Barang

13. Laporan Reparasi

Laporan ini berfungsi untuk memberikan informasi mengenai barang apa saja yang rusak dan telah diperbaiki. Didalam laporan ini terdapat kode_reparasi, tanggal_reparasi, nama_barang yang direparasi, jumlah_barang yang telah direparasi, biaya reparasi, dan keterangan_reparasi barang berdasarkan periode bulan dan tahun dari setiap laporan reparasi barang yang terjadi. Gambar 4.97 adalah proses menampilkan laporan reparasi barang.

Gambar 4.97 Form Laporan Reparasi Barang

Pada *combobox* bulan dipilih berdasarkan bulan untuk laporan yang ada dalam data laporan pembelian barang. *Textbox* tahun diisi secara manual sesuai dengan tahun yang diinginkan. Untuk *textbox* tahun jika diinputkan selain angka maka laporan tidak bisa menampilkan data, karena *textbox* tahun hanya bisa diinputkan berupa angka. Untuk laporan reparasi barang dapat dilihat pada gambar 4.98.



CV. Padantya Alam Panglungan
 Club House Lt.2 Wisma Sier Jl. Rungkut Industri No. 10
 Telp. (031) 8479095 Fax. (031) 8479096 Email: info@bororafting.com
 SURABAYA

Laporan Reparasi Barang Periode: November 2012

kode_reparasi	tanggal_reparasi	nama_barang	jumlah	Biaya	total	Keterangan
R001	03-Nop-2012	Perahu	2	2.500	5.000	Bocor
R002	05-Nop-2012	Pelampung Tamu	3	6.000	18.000	Sobek
R003	05-Nop-2012	Helm Guide	7	45.000	315.000	Rusak Karena Pecah
R004	05-Nop-2012	Dayung	4	25.000	100.000	Rusak karena patah
R005	06-Nop-2012	Pelampung Guide	1	50.000	50.000	Sobek
R006	06-Nop-2012	Printer	1	180.000	180.000	Ganti Catrid
R007	07-Nop-2012	Kayak	1	75.000	75.000	Reparasi karena patah
R008	07-Nop-2012	Kursi	2	100.000	200.000	Diperbaiki Kursi yang
			21		943.000	

Gambar 4.98 Laporan Reparasi Barang