BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Dalam rencana penerapan sistem yeng terkomputerisi yakni agar sistem siap dioperasikan, maka perlu diadakan kegiatan-kegiatan dari penerapannya. Adapun langkah-langkah yang perlu dilakukan untuk menerapkan sistem tersebut adalah pembuatan program, testing program, pelatihan serta terima dokumentasi, akan tetapi dalam penulisan tugas akhir ini implementasi sistem yang dilakukan hanya sampai pada tahap pembuatan program.

5.1.1. Implementasi Perangkat Lunak

Untuk mendukung kelancaran sistem informasi yang dirancang, maka sistem ini memerlukan perangkat lunak. Perangkat lunak digunakan untuk mendukung kinerja sistem operasi dan aplikasi *database*. Adapun perangkat lunak yang digunakan adalah sebagai berikut :

PC Server:

a. Sistem operasi minimum Windows 98
b. *Software* Penjualan Dabeda Headwears Company
c. *Software* Microsoft SQL Server 2000, untuk *Database program.*
d. Anti Virus

PC Client:

a. Sistem operasi minimum Windows 98

b. *Software* Penjualan Dabeda Headwears Company

5.1.2. Implementasi Perangkat Keras

Dalam mengimplementasikan program aplikasi yang dibuat, Penulis menyesuaikannya dengan kebutuhan Dabeda Headwears Company terhadap data-data yang diolah oleh program dengan menerapkan hubungan *Client Server*. Adapun jenis topologi yang dipilih adalah Topologi Star. Berikut ini adalah spesifikasi minimum perangkat keras yang diperlukan.

1. *PC Server:*

Hardware :

a. RAM : 128 Ghz

b. *Processor* minimum Pentium III

c. Harddisk : 20 GB

d. NIC / LAN Card : 10/100 Mbps

e. *Drive* : DVD RW

f. Printer

g. Monitor

h. Mouse

i. Keyboard

2. *PC Client:*

Hardware:

a. RAM : 128 Ghz

b. *Processor* minimum Pentium III

c. Harddisk : 20 GB

d. NIC / LAN Card : 10/100 Mbps

e. *Drive* : DVD RW

f. Monitor

g. Mouse

h. Keyboard

5.1.3. Implementasi Basis Data (Sintaks SQL)

Pembangunan basis data dilakukan dengan menggunakan bahasa SQL, dimana aplikasi pemprograman yang digunakan adalah Microsoft SQL Server 2000. Implementasi basis datanya adalah sebagai berikut :

```
IF EXISTS (SELECT name FROM master.dbo.sysdatabases WHERE
name = N'db_dabeda2')
     DROP DATABASE [db_dabeda2]
GO


CREATE DATABASE [db_dabeda2]  ON (NAME =
N'db_dabeda2_dat', FILENAME = N'C:\Program
Files\Microsoft SQL Server\MSSQL\data\db_dabeda2.mdf' ,
SIZE = 2, FILEGROWTH = 10%) LOG ON (NAME =
N'db_dabeda2_log', FILENAME = N'C:\Program
```

```
Files\Microsoft SQL Server\MSSQL\data\db_dabeda2.ldf' ,
SIZE = 2, FILEGROWTH = 10%)

 COLLATE Arabic_CI_AS

GO

exec sp_dboption N'db_dabeda2', N'autoclose', N'true'

GO

exec sp_dboption N'db_dabeda2', N'bulkcopy', N'true'

GO

exec sp_dboption N'db_dabeda2', N'trunc. log', N'true'

GO

exec sp_dboption N'db_dabeda2', N'torn page detection',
N'true'

GO

exec sp_dboption N'db_dabeda2', N'read only', N'false'

GO

exec sp_dboption N'db_dabeda2', N'dbo use', N'false'

GO

exec sp_dboption N'db_dabeda2', N'single', N'false'

GO

exec sp_dboption N'db_dabeda2', N'autoshrink', N'true'

GO

exec sp_dboption N'db_dabeda2', N'ANSI null default',
N'false'

GO

exec sp_dboption N'db_dabeda2', N'recursive triggers',
N'false'

GO
```

```
exec sp_dboption N'db_dabeda2', N'ANSI nulls', N'false'

GO

exec sp_dboption N'db_dabeda2', N'concat null yields
null', N'false'

GO

exec sp_dboption N'db_dabeda2', N'cursor close on
commit', N'false'

GO

exec sp_dboption N'db_dabeda2', N'default to local
cursor', N'false'

GO

exec sp_dboption N'db_dabeda2', N'quoted identifier',
N'false'

GO

exec sp_dboption N'db_dabeda2', N'ANSI warnings',
N'false'

GO

exec sp_dboption N'db_dabeda2', N'auto create
statistics', N'true'

GO

exec sp_dboption N'db_dabeda2', N'auto update
statistics', N'true'

GO

use [db_dabeda2]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_topi]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_topi]
```

```
GO

CREATE TABLE [dbo].[t_topi] (

     [kd_topi] [varchar] (5) COLLATE Arabic_CI_AS NOT
NULL ,

     [nama_topi] [varchar] (50) COLLATE Arabic_CI_AS NULL
,

     [satuan] [varchar] (6) COLLATE Arabic_CI_AS NULL ,

     [stok_topi] [int] NULL ,

     [hrg_satuan] [numeric](10, 0) NULL

) ON [PRIMARY]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_mit_us]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_mit_us]

GO

CREATE TABLE [dbo].[t_mit_us] (

     [kd_mit_us] [varchar] (4) COLLATE Arabic_CI_AS NOT
NULL ,

     [nama_mit_us] [varchar] (30) COLLATE Arabic_CI_AS
NULL ,

     [telp_mit_us] [varchar] (13) COLLATE Arabic_CI_AS
NULL ,

     [alamat_mit_us] [varchar] (50) COLLATE Arabic_CI_AS
NULL

) ON [PRIMARY]

GO
```

```
if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_jual]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_jual]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_jual]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_jual]

GO

CREATE TABLE [dbo].[t_jual] (

     [no_fak_jual] [varchar] (11) COLLATE Arabic_CI_AS
NOT NULL ,

     [tgl_jual] [varchar] (8) COLLATE Arabic_CI_AS NULL ,

     [kd_kons] [varchar] (5) COLLATE Arabic_CI_AS NULL

) ON [PRIMARY]

GOif exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_det_jual]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_det_jual]

GO

CREATE TABLE [dbo].[t_det_jual] (

     [no_fak_jual] [varchar] (11) COLLATE Arabic_CI_AS
NOT NULL ,

     [kd_topi] [varchar] (5) COLLATE Arabic_CI_AS NOT
NULL ,

     [hrg_jual] [numeric](18, 0) NULL ,

     [jml_jual] [numeric](18, 0) NULL

) ON [PRIMARY]
```

```
GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_beli]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_beli]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_kons]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_kons]

GO



CREATE TABLE [dbo].[t_kons] (

     [kd_kons] [varchar] (5) COLLATE Arabic_CI_AS NOT
NULL ,

     [nama_kons] [varchar] (30) COLLATE Arabic_CI_AS NULL
,

     [telp_kons] [varchar] (13) COLLATE Arabic_CI_AS NULL
,

     [alamat_kons] [varchar] (30) COLLATE Arabic_CI_AS
NULL

) ON [PRIMARY]

GO



CREATE TABLE [dbo].[t_beli] (

     [no_fak_beli] [varchar] (11) COLLATE Arabic_CI_AS
NOT NULL ,

     [tgl_beli] [varchar] (10) COLLATE Arabic_CI_AS NULL
,
```

```
     [kd_mit_us] [varchar] (4) COLLATE Arabic_CI_AS NULL

) ON [PRIMARY]

GO

if exists (select * from dbo.sysobjects where id =
object_id(N'[dbo].[t_det_beli]') and OBJECTPROPERTY(id,
N'IsUserTable') = 1)

drop table [dbo].[t_det_beli]

GO

CREATE TABLE [dbo].[t_det_beli] (

     [no_fak_beli] [varchar] (11) COLLATE Arabic_CI_AS
NOT NULL ,

     [kd_topi] [varchar] (5) COLLATE Arabic_CI_AS NOT
NULL ,

     [hrg_beli] [numeric](18, 0) NULL ,

     [jml_beli] [numeric](18, 0) NULL

) ON [PRIMARY]

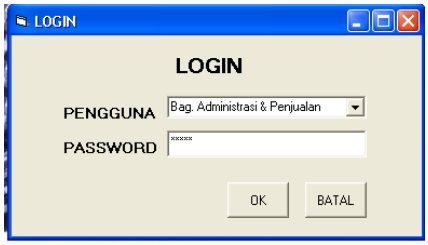
GO
```

5.1.4. Implementasi Antar Muka dan Penggunaan Program

Pada tahapan ini akan ditampilkan *form-form* yang terdapat pada program yang dikembangkan oleh penulis antara lain sebagai berikut:

a. Form Login

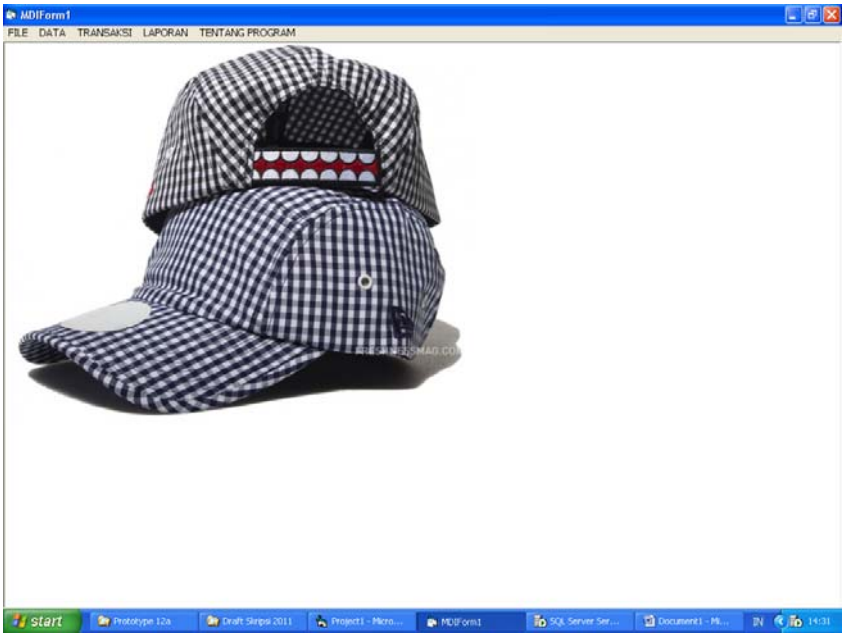
Form login berfungsi sebagai penentu bagi pengguna program aplikasi, sehingga tidak sembarang orang bisa mengakses program ini. Pada *form login* ini terdapat dua tombol, yaitu tombol ”OK” yang berfungsi untuk masuk ke tampilan Menu Utama dengan menu yang aktif disesuaikan dengan hak akses dari pengguna program ini, serta tombol ”BATAL” yang berfungsi untuk membatalkan proses login. Adapun tampilannya seperti di bawah ini:



Gambar 5.1
Form Login

b. Form Menu Utama

Form menu utama adalah *form* yang bisa digunakan oleh pengguna untuk mengakses berbagai fasilitas pengolahan data perusahaan. Tampilan *form* menu utama bisa dilihat pada gambar di bawah ini:



Gambar 5.2
Form Menu Utama

Adapun penjelasan dari menu-menu yang terdapat pada *form* menu utama adalah sebagai berikut:

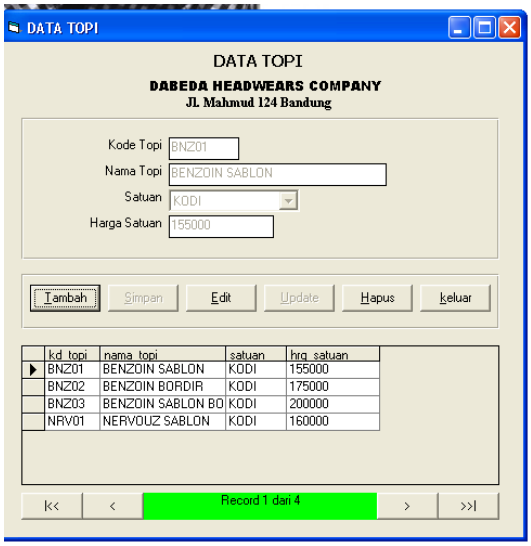
Tabel 5.1 Menu Utama

Menu	Deskripsi
File	Akses untuk melakukan *login*, *logout,* dan keluar (dari aplikasi)
Data	Akses untuk memasukkan data topi, data konsumen dan data mitra usaha

Transaksi	Akses untuk memasukkan data transaksi penjualan dan data transaksi pembelian
Laporan	Akses untuk mencetak laporan penjualan, pembelian, persediaan barang, data konsumen dan data mitra usaha
Tentang Program	Akses yang berisi profil pembuat program

c. Form Data Topi

Form data topi merupakan *form* yang berfungsi untuk memasukkan data topi ke dalam *database*. Tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.3
Form Data Topi

Adapun fungsi dari tombol-tombol yang ada pada form data topi di atas adalah sebagai berikut:

Tabel 5.2
Fungsi Tombol-Tombol *Form* Data Topi

NO	TOMBOL	KETERANGAN
1	Tambah	Untuk menambah *record* baru
2	Simpan	Untuk menyimpan *record*
3	Edit	Untuk mengedit *record* yang telah ada
4	Hapus	Untuk menghapus *record* yang telah ada
5	Update	Untuk merefresh *database* ketika dilakukan proses edit.
6	Keluar	Untuk keluar dari *Form*
7	|<	Penunjuk arah bergerak ke *record* paling pertama
8	<	Penunjuk arah bergerak ke *record* sebelumnya
9	>	Penunjuk arah bergerak ke *record* berikutnya
10	>|	Penunjuk arah bergerak ke *record* paling akhir

Penjelasan mendetailnya sebagai berikut:

Untuk menambah data baru, maka Anda tekan tombol “TAMBAH” lalu isi semua *field* yang ada. Jika anda ingin mengurungkan niat untuk menambah data baru tersebut ke dalam *database,* maka tekan tombol “BATAL” (yang muncul otomatis ketika tombol “TAMBAH” ditekan), tapi jika Anda ingin melanjutkan proses penambahan data, maka tekan tombol “SIMPAN”. Jika input data tersebut berhasil

dioperasikan oleh program, maka akan ada pesan bahwa data baru telah tersimpan ke dalam *database.*

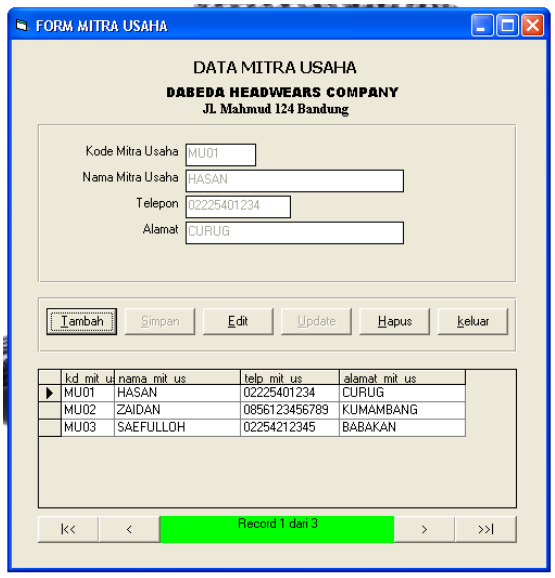
Untuk mengedit data, maka Anda pilih dulu data yang akan diedit dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Setelah itu, Anda tekan tombol "EDIT", lalu rubahlah data tersebut pada *field* yang Anda inginkan, kemudian tekan tombol "UPDATE", maka data telah berhasil Anda rubah.

Untuk menghapus data, langkah awalnya sama dengan langkah untuk mengedit data, yakni pilih dulu data yang akan dihapus dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Kemudian klik tombol "HAPUS", maka Anda akan mendapat pesan "Anda yakin akan menghapus data tersebut?", jika mau melanjutkan proses penghapusan data, maka tekan tombol "YES", jika tidak, maka pilih tombol "NO".

Untuk keluar dari form ini, maka Anda tinggal tekan tombol "KELUAR".

d. Form Data Mitra Usaha

Form Data Mitra Usaha merupakan *form* yang berfungsi untuk memasukkan data Mitra Usaha ke dalam *database*. Tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.4
Form Data Mitra Usaha

Pada *form* tersebut ada beberapa tombol yang bisa dijalankan oleh pengguna sebagaimana dijelaskan pada tabel di bawah:

Tabel 5.3
Fungsi Tombol-Tombol *Form* Data Mitra Usaha

NO	TOMBOL	KETERANGAN
1	Tambah	Untuk menambah *record* baru
2	Simpan	Untuk menyimpan *record*

3	Edit	Untuk mengedit *record* yang telah ada
4	Hapus	Untuk menghapus *record* yang telah ada
5	Update	Untuk merefresh *database* ketika dilakukan proses edit.
6	Keluar	Untuk keluar dari *Form*
7	|<	Penunjuk arah bergerak ke *record* paling pertama
8	<	Penunjuk arah bergerak ke *record* sebelumnya
9	>	Penunjuk arah bergerak ke *record* berikutnya
10	>|	Penunjuk arah bergerak ke *record* paling akhir

Penjelasan mendetailnya sebagai berikut:

Untuk menambah data baru, maka Anda tekan tombol “TAMBAH” lalu isi semua *field* yang ada. Jika anda ingin mengurungkan niat untuk menambah data baru tersebut ke dalam *database,* maka tekan tombol “BATAL” (yang muncul otomatis ketika tombol “TAMBAH” ditekan), tapi jika Anda ingin melanjutkan proses penambahan data, maka tekan tombol “SIMPAN”. Jika input data tersebut berhasil dioperasikan oleh program, maka akan ada pesan bahwa data baru telah tersimpan ke dalam *database.*

Untuk mengedit data, maka Anda pilih dulu data yang akan diedit dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Setelah itu, Anda tekan tombol “EDIT”, lalu rubahlah data

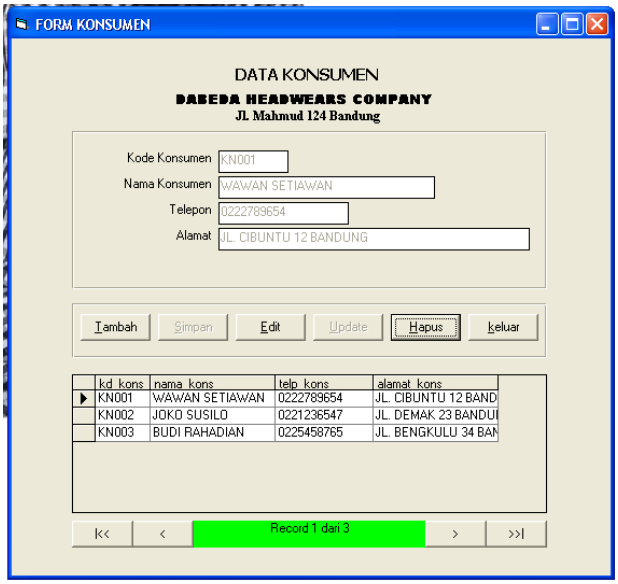
tersebut pada *field* yang Anda inginkan, kemudian tekan tombol “UPDATE”, maka data telah berhasil Anda rubah.

Untuk menghapus data, langkah awalnya sama dengan langkah untuk mengedit data, yakni pilih dulu data yang akan dihapus dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Kemudian klik tombol “HAPUS”, maka Anda akan mendapat pesan “Anda yakin akan menghapus data tersebut?”, jika mau melanjutkan proses penghapusan data, maka tekan tombol “YES”, jika tidak, maka pilih tombol “NO”.

Untuk keluar dari form ini, maka Anda tinggal tekan tombol “KELUAR”.

e. *Form* Data Konsumen

Form Data Konsumen merupakan *form* yang berfungsi untuk memasukkan data Konsumen ke dalam *database*. Tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.5
Form Data Konsumen

Pada *form* tersebut ada beberapa tombol yang bisa dijalankan oleh pengguna sebagaimana dijelaskan pada tabel di bawah:

Tabel 5.4
Fungsi Tombol-Tombol *Form* Data Konsumen

NO	TOMBOL	KETERANGAN
1	Tambah	Untuk menambah *record* baru
2	Simpan	Untuk menyimpan *record*
3	Edit	Untuk mengedit *record* yang telah ada

4	Hapus	Untuk menghapus *record* yang telah ada
5	Update	Untuk merefresh *database* ketika dilakukan proses edit.
6	Keluar	Untuk keluar dari *Form*
7	|<	Penunjuk arah bergerak ke *record* paling pertama
8	<	Penunjuk arah bergerak ke *record* sebelumnya
9	>	Penunjuk arah bergerak ke *record* berikutnya
10	>|	Penunjuk arah bergerak ke *record* paling akhir

Penjelasan mendetailnya sebagai berikut:

Untuk menambah data baru, maka Anda tekan tombol "TAMBAH" lalu isi semua *field* yang ada. Jika anda ingin mengurungkan niat untuk menambah data baru tersebut ke dalam *database,* maka tekan tombol "BATAL" (yang muncul otomatis ketika tombol "TAMBAH" ditekan), tapi jika Anda ingin melanjutkan proses penambahan data, maka tekan tombol "SIMPAN". Jika input data tersebut berhasil dioperasikan oleh program, maka akan ada pesan bahwa data baru telah tersimpan ke dalam *database.*

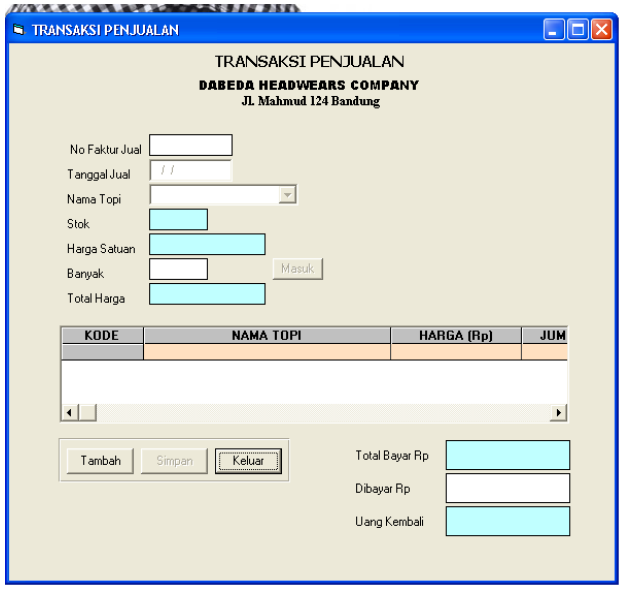
Untuk mengedit data, maka Anda pilih dulu data yang akan diedit dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Setelah itu, Anda tekan tombol "EDIT", lalu rubahlah data tersebut pada *field* yang Anda inginkan, kemudian tekan tombol "UPDATE", maka data telah berhasil Anda rubah.

Untuk menghapus data, langkah awalnya sama dengan langkah untuk mengedit data, yakni pilih dulu data yang akan dihapus dengan menekan tombol navigasi (tombol tanda panah) yang ada di kiri dan kanan bawah dari tampilan data. Kemudian klik tombol "HAPUS", maka Anda akan mendapat pesan "Anda yakin akan menghapus data tersebut?", jika mau melanjutkan proses penghapusan data, maka tekan tombol "YES", jika tidak, maka pilih tombol "NO".

Untuk keluar dari form ini, maka Anda tinggal tekan tombol "KELUAR".

f. *Form* Data Transaksi Penjualan

Form data transaksi penjualan merupakan *form* yang berfungsi untuk memasukkan data transaksi penjualan ke dalam *database*. Adapun tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.6
Form Data Transaksi Penjualan

Pada *form* data transaksi penjualan tersebut ada beberapa tombol yang bisa dijalankan oleh pengguna, sebagaimana dijelaskan pada tabel di bawah ini:

Tabel 5.5
Fungsi Tombol-Tombol *Form* Data Transaksi Penjualan

NO	TOMBOL	KETERANGAN
1	Tambah	Untuk menambah *record* baru
2	Simpan	Untuk menyimpan *record*
3	Keluar	Untuk keluar dari *form* ini
4	Batal	Untuk membatalkan data transaksi

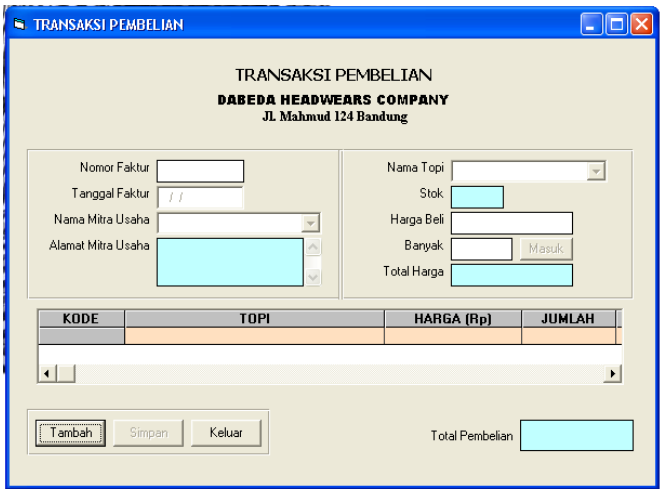
Penjelasan mendetailnya sebagai berikut:

Untuk menambah data transaksi baru, maka Anda tekan tombol "TAMBAH" lalu isi semua *field* yang ada. Lalu tekan tombol "MASUK" agar data ditampilkan pada *Grid*. Selanjutnya masukkan jumlah pembayaran pada *field* "dibayar Rp". Jika anda ingin mengurungkan niat untuk menambah data transaksi baru tersebut ke dalam *database*, maka tekan tombol "BATAL" (yang muncul otomatis ketika tombol "TAMBAH" ditekan), tapi jika Anda ingin melanjutkan proses penambahan data, maka tekan tombol "SIMPAN". Jika input data tersebut berhasil dioperasikan oleh program, maka akan ada pesan bahwa data baru telah tersimpan ke dalam *database.* Setelah itu akan muncul form cetak faktur. Untuk mencetak faktur tersebut, maka tekan tombol "CETAK".

Untuk keluar dari form, tekan tombol "KELUAR"

g. *Form* Data Transaksi Pembelian

Form data transaksi pembelian merupakan *form* yang berfungsi untuk memasukkan data transaksi pembelian ke dalam *database*. Adapun tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.7
Form Data Transaksi Pembelian

Pada *form* data transaksi pembelian tersebut ada beberapa tombol yang bisa dijalankan oleh pengguna, sebagaimana dijelaskan pada tabel di bawah ini:

Tabel 5.6
Fungsi Tombol-Tombol *Form* Data Transaksi Pembelian

NO	TOMBOL	KETERANGAN
1	Tambah	Untuk menambah *record* baru
2	Simpan	Untuk menyimpan *record*
3	Keluar	Untuk keluar dari *form* ini
4	Batal	Untuk membatalkan data transaksi

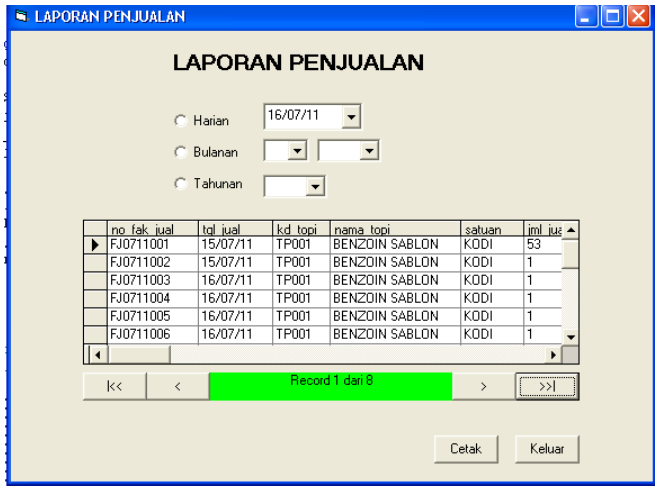
Penjelasan mendetailnya sebagai berikut:

Untuk menambah data transaksi baru, maka Anda tekan tombol “TAMBAH” lalu isi semua *field* yang ada. Lalu tekan tombol “MASUK” agar data ditampilkan pada *Grid*. Selanjutnya masukkan jumlah pembayaran pada *field* “dibayar Rp”. Jika anda ingin mengurungkan niat untuk menambah data transaksi baru tersebut ke dalam *database*, maka tekan tombol “BATAL” (yang muncul otomatis ketika tombol “TAMBAH” ditekan), tapi jika Anda ingin melanjutkan proses penambahan data, maka tekan tombol “SIMPAN”. Jika input data tersebut berhasil dioperasikan oleh program, maka akan ada pesan bahwa data baru telah tersimpan ke dalam *database.* Setelah itu akan muncul form cetak faktur. Untuk mencetak faktur tersebut, maka tekan tombol “CETAK”.

Untuk keluar dari form, tekan tombol “KELUAR”

h. Form Laporan Penjualan

Form laporan penjualan merupakan informasi tentang penjualan yang telah terjadi di perusahaan. Pilih terlebih dahulu apakah Anda ingin mencetak laporan harian, bulanan, atau tahunan, lalu tekan tombol ”CETAK” untuk mencetak informasi penjualan untuk selanjutnya dijadikan arsip, dan tombol ”KELUAR” yang berfungsi untuk keluar dari *form*.



Gambar 5.8
Form Laporan Penjualan

i. *Form* Laporan Pembelian

Form laporan pembelian merupakan informasi tentang pembelian yang telah terjadi di perusahaan. Pilih terlebih dahulu apakah Anda ingin mencetak laporan harian, bulanan, atau tahunan, lalu tekan tombol ”CETAK” untuk mencetak informasi penjualan untuk selanjutnya dijadikan arsip, dan tombol ”KELUAR” yang berfungsi untuk keluar dari *form*.



Gambar 5.9
Form Laporan Pembelian

j. *Form* Laporan Persediaan Barang (Topi)

Form laporan persediaan barang (topi) merupakan informasi tentang persediaan barang (topi) yang telah terjadi di perusahaan. Pada *Form* ini terdapat dua tombol, yaitu tombol ”CETAK” untuk mencetak informasi pembelian untuk dijadikan arsip, dan yang kedua tombol ”KELUAR” yang berfungsi untuk keluar dari *form*.

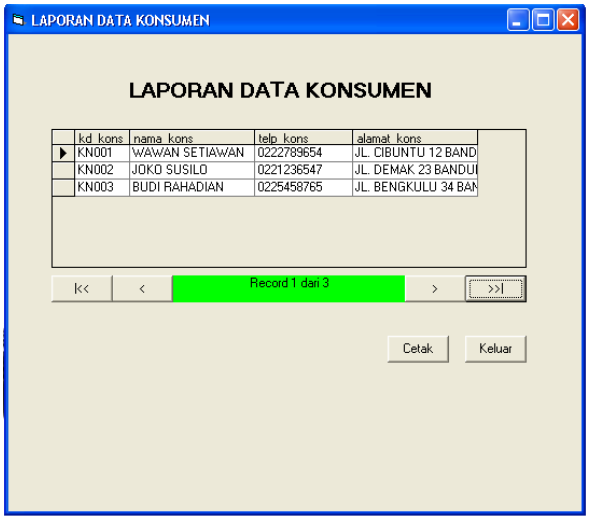


Gambar 5.10
Form Laporan Persediaan Barang (Topi)

k. Form laporan Data Konsumen

Form laporan data konsumen merupakan informasi tentang data konsumen yang dimiliki oleh perusahaan. Pada *Form* ini terdapat dua tombol, yaitu tombol ”CETAK” untuk mencetak informasi konsumen untuk dijadikan arsip, dan yang kedua tombol ”KELUAR” yang berfungsi untuk keluar dari *form*.

Tampilannya bisa dilihat pada gambar di bawah ini:

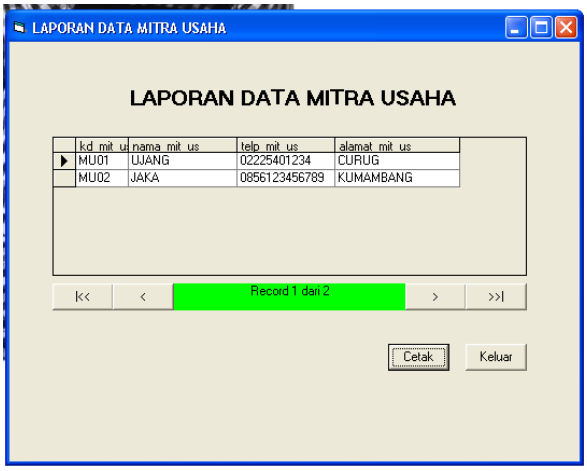


Gambar 5.11
Form Laporan Data Konsumen

l. *Form* Laporan Data Mitra Usaha

Form laporan data Mitra Usaha merupakan informasi tentang data Mitra Usaha yang dimiliki oleh perusahaan. Pada *Form* ini terdapat dua tombol, yaitu tombol ”CETAK” untuk mencetak informasi Mitra Usaha untuk dijadikan arsip, dan yang kedua tombol ”KELUAR” yang berfungsi untuk keluar dari *form*.

Tampilannya bisa dilihat pada gambar di bawah ini:



Gambar 5.12
Form Laporan Data Mitra Usaha

m. Form Tentang Program

Form tentang program ini berfungsi untuk mengetahui versi dan pembuat program aplikasi Dabeda Headwears Company.



Gambar 5.13 Form Tentang Program

Di dalam *Form* tersebut terdapat dua tombol, yakni tombol “OK” untuk keluar dari form dan tombol “system info...” untuk mengetahui spesifikasi sistem komputer.

5.1.5. Implementasi Instalasi Program

Langkah-langkah dalam proses instalasi aplikasi penjualan pembelian dan persediaan barang Dabeda Headwears Company adalah sebagai berikut:

1. klik 2 kali atau klik kanan > *open* pada *file* setup.exe master aplikasi penjualan pembelian dan persediaan barang Dabeda Headwears Company seperti pada gambar 5.10 di bawah ini :



Gambar 5.14
Tahap Pertama Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

2. Maka akan terlihat tampilan awal proses instalasi seperti pada gambar berikut:



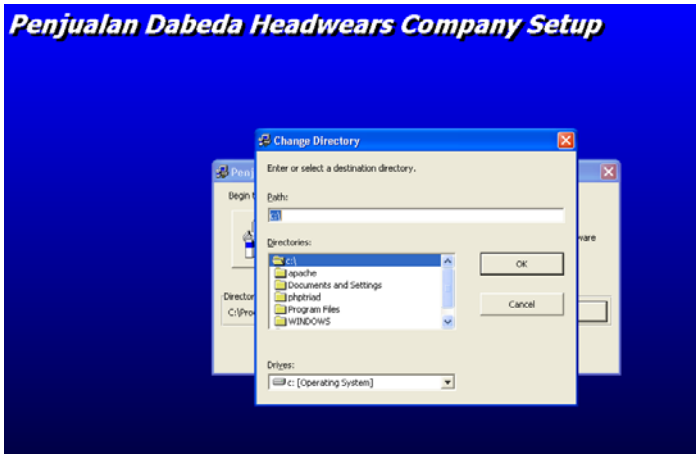
Gambar 5.15
Tahap Kedua Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

3. Klik tombol **OK**, maka akan terlihat tampilan sebagai berikut:



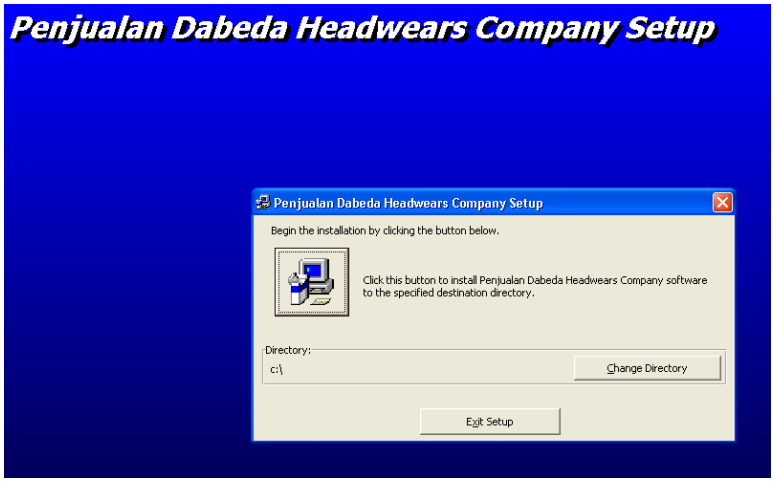
Gambar 5.16
Tahap Ketiga Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

4. kemudian klik tombol **change directory** untuk menentukan drive dan direktori di mana Anda ingin menginstal Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company. Akan terlihat tampilan sebagai berikut:



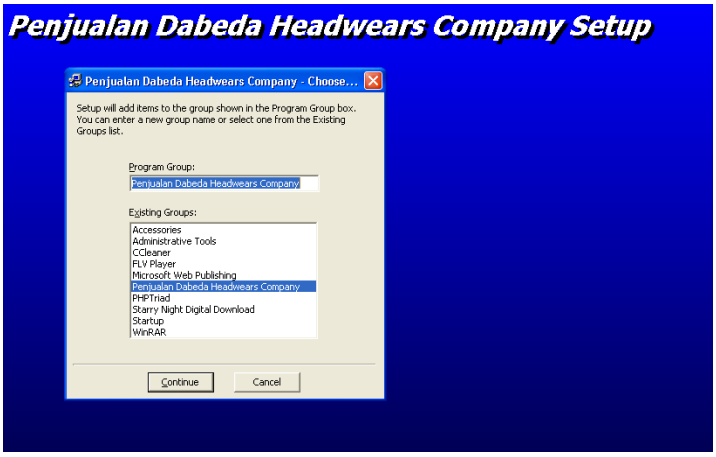
Gambar 5.17
Tahap Keempat Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

5. setelah menentukan direktori instalasi, klik gambar komputer untuk memulai instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company.



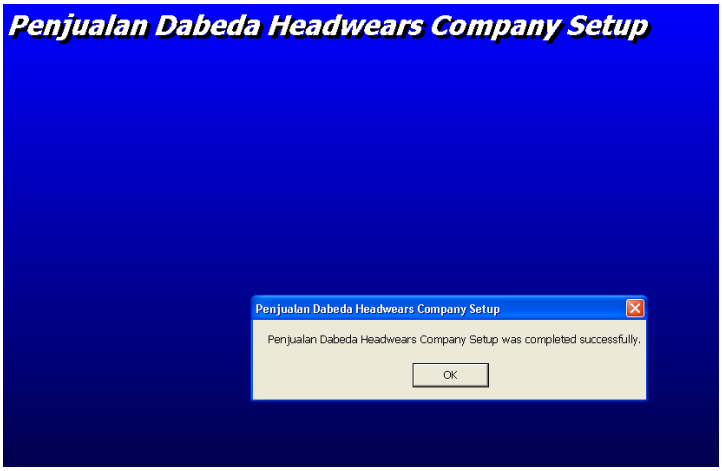
Gambar 5.18
Tahap Kelima Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

6. selanjutnya, Anda dapat menentukan grup tempat aplikasi akan ditampilkan pada menu windows (di dalam start menu windows). Setelah itu klik tombol **continue** untuk melanjutkan proses instalasi.



Gambar 5.19
Tahap Keenam Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

7. setelah selesai menginstal Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company, klik **OK.**



Gambar 5.20
Tahap Terakhir Instalasi Aplikasi Penjualan Pembelian dan Persediaan barang Dabeda Headwears Company

5.2. Pengujian

Pengujian merupakan bagian yang tidak kalah pentingnya dalam siklus pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangakat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang handal, yaitu mampu merepresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

Pengujian perangkat lunak ini menggunakan metode pengujian *black box*. Dimana pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

5.2.1. Rencana Pengujian

Pengujian sistem informasi penjualan pembelian dan persediaan barang menggunakan data uji berupa sebuah data dan masukan dari pengguna.

Tabel 5.7
Rencana Pengujian Sistem Informasi Penjualan Pembelian persediaan Barang

Kelas Uji	Butir Uji	Tingkat Pengujian	Jenis Pengujian
Pengujian Login Pengguna	Pengecekan pengguna yang telah terdaftar	Sistem	*Black Box*
Pengujian Pengisian Data	Pengisian Data transaksi penjualan	Modul	*Black Box*
	Pengisian Data transaksi pembelian	Modul	*Black Box*

5.2.2. Kasus dan Hasil Pengujian

a. Pengujian Login

Pengujian login yang dilakukan hanya untuk pengecekan pengguna yang telah terdaftar, sedangkan pengujian pendaftaran pengguna baru dilakukan dalam pengujian data pengguna.

Tabel 5.8
Pengujian *Login* 1

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pengguna dan *password* terdaftar Pengguna : Bag. Produksi Password : produksi	Tercantum pada *combobox* pengguna dan *textbox password*	Dapat mengisi login pengguna sesuai yang diharapkan	[√] diterima [ ] ditolak
Klik LOGIN	Dapat masuk ke form menu utama untuk pengguna yang terdaftar	Tombol LOGIN dapat berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak

Tabel 5.9
Pengujian *Login* 2

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pengguna tidak terdaftar. Pengguna : Bag. Produksi *Password* : 123	Tidak dapat login dan menampilkan pesan peringatan.	pengguna tidak dapat login dan menampilkan pesan ”*password* salah.”	[√] diterima [ ] ditolak
Klik ”LOGIN”			

b. Pengujian Pengisian Data

Pengujian pengisian data terbagi menjadi 2 bagian, yaitu:

1. Pengujian Pengisian Data transaksi penjualan

Berikut adalah tabel pengujian pengisian data transaksi penjualan:

Tabel 5.10
Pengujian Pengisian Data Transaksi Penjualan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik "Tambah"	Tombol yang aktif hanya tombol "Simpan" dan "Batal"	Dapat mengisi tiap field sesuai yang diharapkan	[√] diterima [ ] ditolak
Mengisi textbox tiap field. Klik "Simpan"	Data tersimpan di data transaksi penjualan.	Tombol "Simpan" dapat berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak
Klik "Batal"	Pengisian data dibatalkan	Tombol "Batal" dapat berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pengisian *textbox* banyak dengan selain angka	Muncul pesan bahwa data harus diisi dengan angka	*Tekxtbox* berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak

2. Pengujian Pengisian Data transaksi pembelian

Berikut adalah tabel pengujian pengisian data transaksi pembelian

:

Tabel 5.11
Pengujian Pengisian Data Transaksi pembelian

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Klik "Tambah"	Tombol yang aktif hanya tombol "Simpan" dan "Batal"	Dapat mengisi tiap field sesuai yang diharapkan	[√] diterima [ ] ditolak
Mengisi *textbox* tiap *field.* Klik "Simpan"	Data tersimpan di data transaksi pembelian.	Tombol "Simpan" dapat berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak
Klik "Batal"	Pengisian data dibatalkan	Tombol "Batal" dapat berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pengisian *textbox* banyak dengan selain angka	Muncul pesan bahwa data harus diisi dengan angka	*Tekxtbox* berfungsi sesuai yang diharapkan	[√] diterima [ ] ditolak

5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus dan hasil uji, maka dapat diambil kesimpulan bahwa proses pengisian data pada setiap *form* dapat dilakukan dengan sukses serta secara fungsional sistem dapat menghasilkan *output* yang diharapkan.