

BAB IV

IMPLEMENTASI DAN EVALUASI

4.1 Implementasi

Di dalam tahap implementasi ini terdapat 3 sub tahap, yaitu mempersiapkan kebutuhan *system* (baik *hardware* maupun *software*), persiapan instalasi aplikasi, dan hasil implementasi. Adapun penjelasan yang lebih rinci sebagai berikut:

4.1.1 Kebutuhan Sistem

Untuk menjalankan sistem informasi penjualan . pada Toko Agung Jaya membutuhkan perangkat keras (*hardware*) dan perangkat lunak (*software*) dengan kriteria tertentu agar sistem dapat berjalan. Adapun kebutuhan *hardware* dan *software* adalah sebagai berikut:

1. Kebutuhan Perangkat Keras (*Hardware*)

Untuk perangkat keras, minimal pengguna harus mempersiapkan spesifikasi perangkat keras sebagai berikut:

1. *Processor : Core 2 Duo2.8 GHz*
2. *Memori Minimal 1 GB*
3. *Keyboard*
4. *Mouse*
5. *Monitor Syncmaster 4S 15 Inch*
6. *Printer*
7. *Barcode*

2. Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak yang harus dipersiapkan oleh Toko Agung Jaya, yaitu:

1. Microsoft Visual Studio. NET 2005
2. Microsoft SQL Server 2005
3. Sistem operasi yang digunakan minimal Windows XP SP 3

4.1.2 Instalasi Program

Dalam tahap ini, pengguna harus memperhatikan dengan benar terhadap penginstalan perangkat lunak. Berikut langkah-langkah penginstalan:

1. *Install* Microsoft SQL Server 2005
2. *Install* Visual Studio.NET 2005
3. Aplikasi siap dijalankan.

4.1.3 Hasil Implementasi

Setelah kebutuhan sistem terpenuhi, langkah selanjutnya adalah mengimplementasikan rancangan sistem kedalam sebuah rancang bangun aplikasi pada Toko Agung Jaya. Di setiap *user* memiliki hak akses yang dapat diubah secara *dynamic*. *User* yang dapat mengubah hak akses adalah *user* admin dan . Di setiap *form* terdapat *shortcut* tombol yang dibuat secara konsisten. *Shortcut* tersebut antara lain:

Tabel 4.1 Daftar *shortcut* tombol di dalam setiap *form*

No	Shortcut	Tombol
1	P	<i>Print</i> , simpan dan <i>print</i>
2	E	Email
3	<i>Enter</i>	<i>OK</i> , cari laporan, simpan, tambah ke daftar
4	<i>Esc</i>	<i>Refresh</i> , batal
5	<i>Home</i>	Keluar, kembali
6	<i>Delete</i>	Hapus
7	F12	<i>Edit</i>

No	Shortcut	Tombol
8	Alt + A	Ambil foto, cari barang
9	Space	Pembayaran
10	End	Hitung

Dari serangkaian rancangan di atas, implementasi desain form aplikasi akan dijelaskan lebih detail sebagai berikut:

1. Form Login

Form login berguna untuk mengidentifikasi *user* yang akan menggunakan aplikasi, dengan adanya *form* ini maka dapat menentukan *user* tertentu yang kemudian *user* tersebut dapat membuka *form* apa saja. *Role* setiap *user* dapat ditentukan di *form master user*. Didalam *form login* terdapat 2 *textbox* yaitu *username* dan *password*, jika *username* dan *password* benar maka akan masuk ke dalam aplikasi sesuai *role* user yang telah ada, jika salah maka akan keluar *messagebox password* salah kemudian kembali lagi masuk *form login*. Adapun hasil implementasi *form login* dapat dilihat pada gambar di bawah ini.

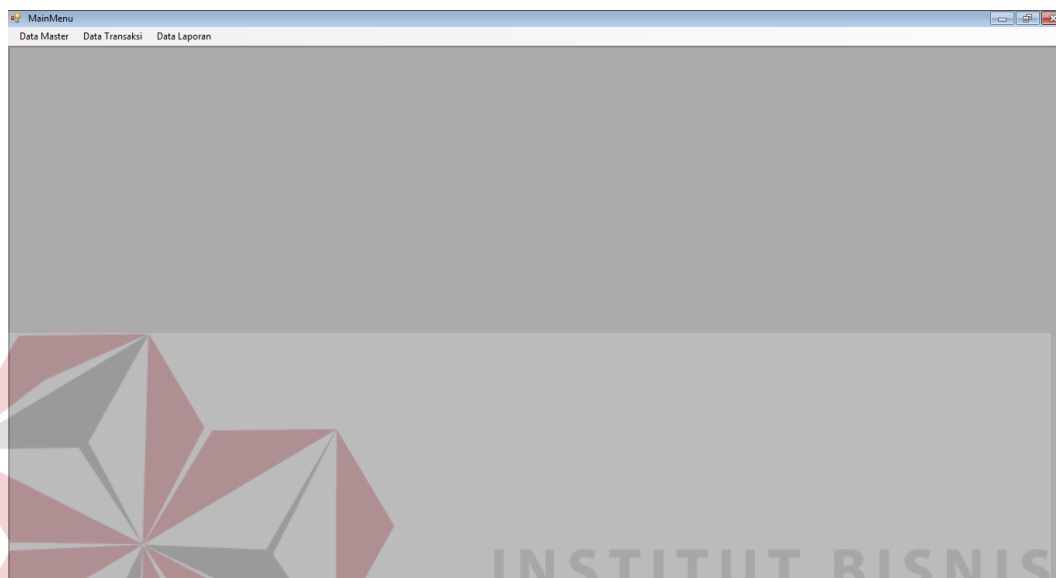


Gambar 4.1 *Form Login*

2. Form Master

Form master merupakan *group* menu yang menyediakan menu untuk maintenance dan *setting* tabel master. Di dalam master ini terdapat *form*, antara lain: form master pelanggan, , master pengguna, master jabatan

pengguna, master *supplier*, master setting potongan *supplier*, master barang, master jenis barang, master kategori barang, master *user*, master menu *form*, dan master *setting* ganti *password*. Adapun master dapat dilihat pada gambar dibawah ini:



Gambar 4.2 *Form* Master

3. *Form* Master Pelanggan

Form master pelanggan digunakan untuk *maintenance* (*input*, *edit*, *delete*) data pelanggan yang dilengkapi dengan foto pelanggan. Untuk mencari foto pelanggan yang diinginkan, *user* dapat menekan tombol ambil foto yang ada di bawah gambar foto. *Form* ini menyediakan fasilitas cari pelanggan pada *textbox* sisi kanan yang ada di atas data *gridview*. Jika *user* ingin *edit* data, maka *user* harus melakukan klik pada data *gridview*. Setelah diklik, data pelanggan akan mengisi di setiap *field* yang ada di sebelah kiri. *Form* ini telah disempurnakan dengan fasilitas cek data *input*, misalnya *field* yang membutuhkan *input* data angka hanya dapat diisi dengan angka, jika *user* mengisi dengan huruf maka sistem akan menolak dan memberikan peringatan

dengan simbol merah di sebelah kiri *field*. Seluruh *form* yang tersedia di aplikasi ini telah disempurnakan dengan fasilitas cek data *input* sama seperti *form* pelanggan. Seluruh *form* juga disempurnakan dengan fasilitas *shortcut* yang konsisten. *Shortcut* ini disediakan untuk mempermudah *user* melakukan proses. Adapun *form* master pelanggan dapat dilihat pada gambar di bawah ini:

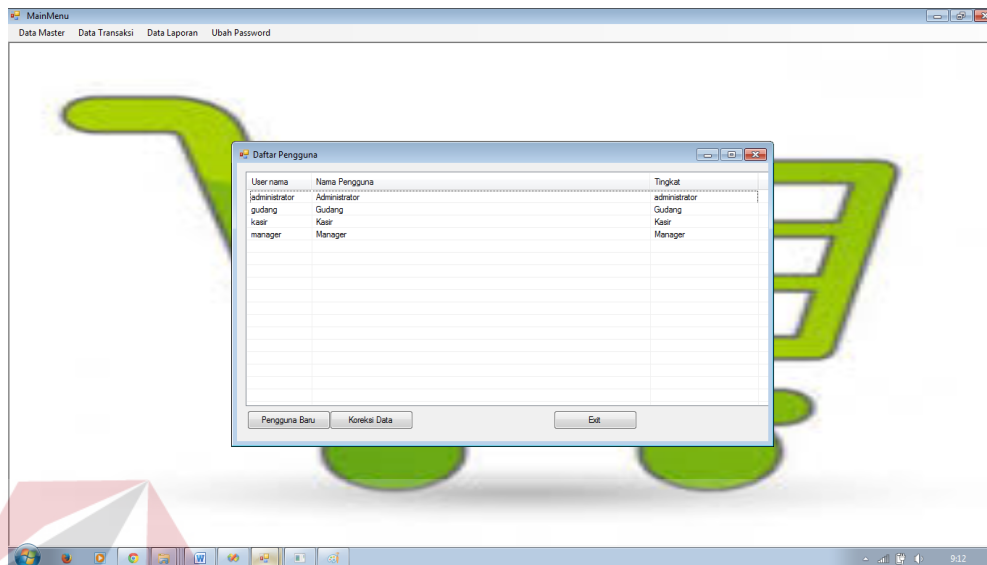
Kode Pelang...	Nama Pelanggan	Alamat	Kota	Propinsi	Telp
C1	Jono				

Gambar 4.3 *Form* Master Pelanggan

4. *Form* Master Pengguna

Form master pengguna digunakan untuk *maintenance* (*input, edit, delete*) data pengguna. Jika *user* ingin *edit* data, maka *user* harus melakukan klik pada data *gridview*. Setelah diklik, data pengguna akan mengisi di setiap *field* yang ada di sebelah kiri. Di setiap *field* telah disempurnakan dengan fasilitas cek data *input*, *field* yang harus diisi dengan huruf harus diisi dengan huruf, begitupun juga *field* yang harus diisi dengan angka harus diisi dengan angka. *Form* master program Studi juga disempurnakan dengan fasilitas *shortcut* yang konsisten. *Shortcut* ini disediakan untuk mempermudah *user* melakukan

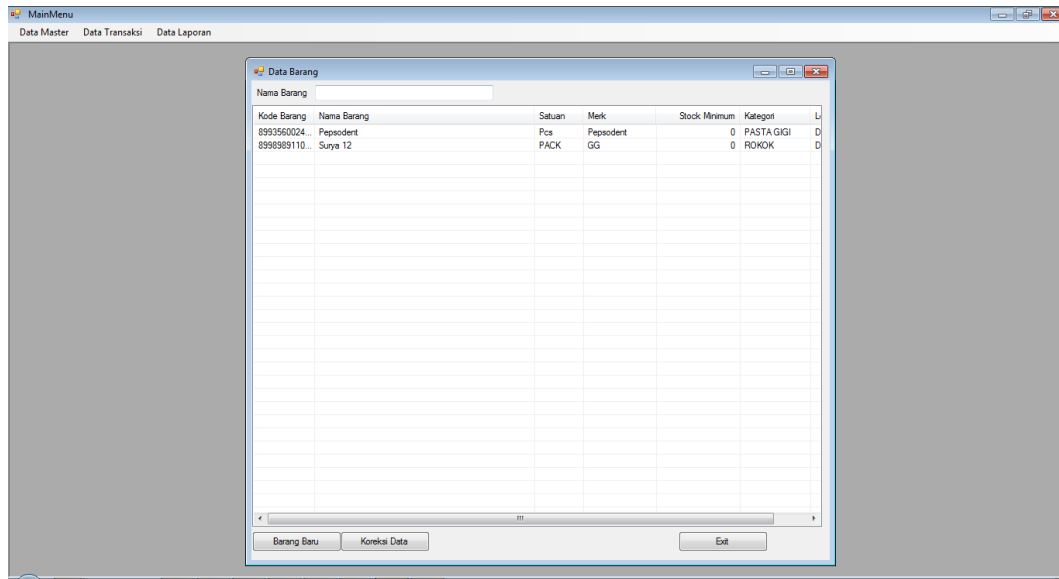
proses. Adapun *form* master program studi dapat dilihat pada gambar di bawah ini:



Gambar 4.4 *Form* Master Pengguna

5. *Form* Master Barang

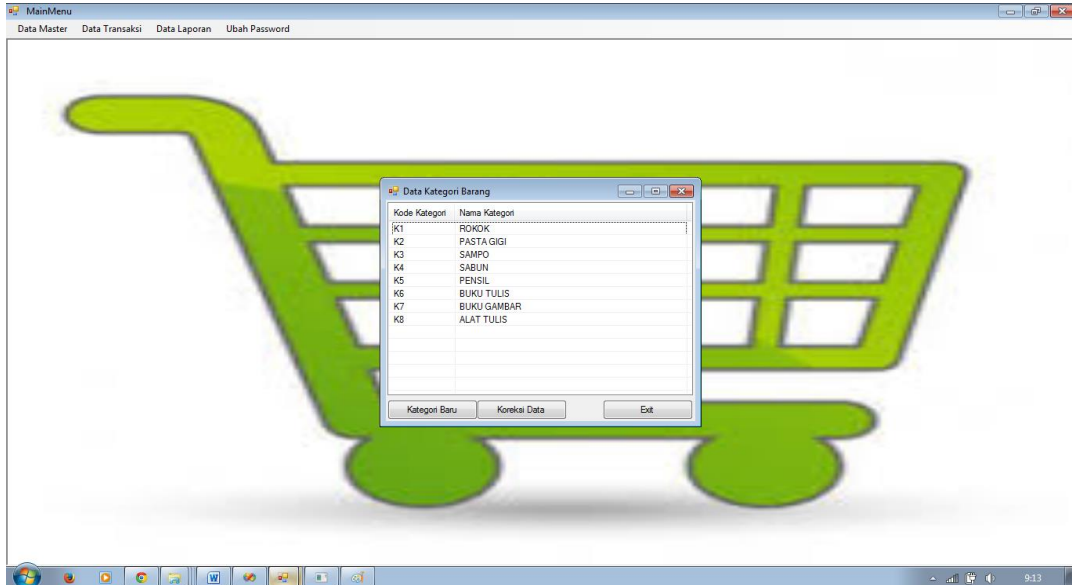
Form master barang digunakan untuk *maintenance* (*input, edit, delete*) data barang. *Form* master barang ini dilengkapi dengan tampilan sisa stok barang dan HPP barang. Jika *user* ingin *edit* data, maka *user* harus melakukan klik pada data *gridview*. Setelah diklik, data barang akan mengisi di setiap *field* yang ada di sebelah kiri. Di setiap *field* telah disempurnakan dengan fasilitas cek data *input*, *field* yang harus diisi dengan huruf harus diisi dengan huruf, begitupun juga *field* yang harus diisi dengan angka harus diisi dengan angka. *Form* master barang juga disempurnakan dengan fasilitas *shortcut* yang konsisten. *Shortcut* ini disediakan untuk mempermudah *user* melakukan proses. Adapun *form* master barang dapat dilihat pada gambar di bawah ini:



Gambar 4.5 Form Master Barang

6. Form Kategori Barang

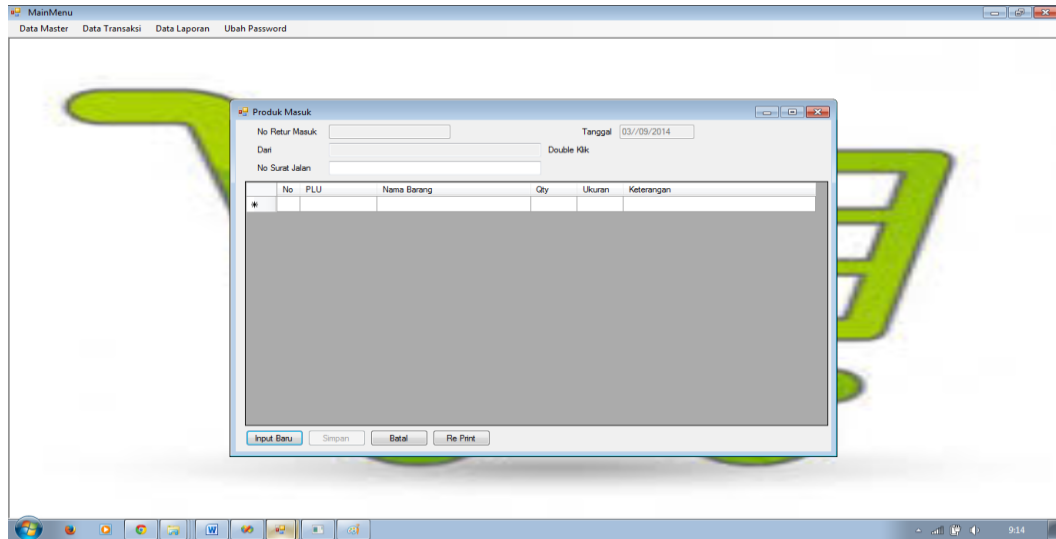
Form master kategori barang digunakan untuk *maintenance* (*input, edit, delete*) data kategori barang. Jika user ingin edit data, maka user harus melakukan klik pada data *gridview*. Setelah diklik, data kategori barang akan mengisi di setiap *field* yang ada di sebelah kiri. Terdapat 2 *field* di dalam form kategori barang yaitu *field* id kategori barang dan *field* nama kategori barang. Di setiap *field* telah disempurnakan dengan fasilitas cek data input, *field* yang harus diisi dengan huruf harus diisi dengan huruf, begitupun juga *field* yang harus diisi dengan angka harus diisi dengan angka. Form master kategori barang juga disempurnakan dengan fasilitas *shortcut* yang konsisten. *Shortcut* ini disediakan untuk mempermudah user melakukan proses. Adapun form master kategori barang dapat dilihat pada gambar di bawah ini:



Gambar 4.6 *Form* kategori Barang

8. *Form* Barang Datang

Form barang datang digunakan untuk (*input*, *edit*, *delete*) data kategori barang datang . Jika user ingin edit data, maka user harus melakukan klik pada data *gridview*. Setelah diklik, data kategori barang akan mengisi di setiap *field* yang ada di sebelah kiri. Terdapat 2 *field* di dalam form kategori barang yaitu *field* id kategori barang dan *field* nama kategori barang. Disetiap *field* telah disempurnakan dengan fasilitas cek data input, *field* yang harus diisi dengan huruf harus diisi dengan huruf, begitupun juga *field* yang harus diisi dengan angka harus diisi dengan angka. Form master kategori barang juga disempurnakan dengan fasilitas *shortcut* yang konsisten. *Shortcut* ini disediakan untuk mempermudah user melakukan proses. Adapun form master kategori barang dapat dilihat pada gambar di bawah ini:

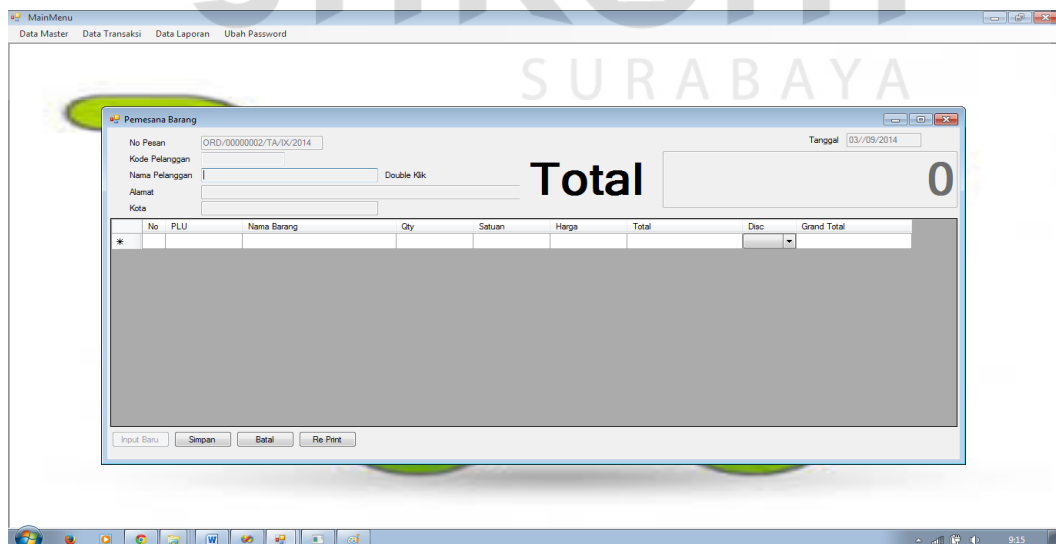


Gambar 4.7 *Form* Barang Datang

9. *Form* Pemesanan

Form Pemesanan Barang berisikan data dari pelanggan dan pesanan dari pelanggan pada nama pelanggan bisa double click untuk mengambil data pelanggan yang sudah didaftarkan sebelumnya, lalu pada data grid view tinggal memasukan barang-barang yang akan dipesan oleh pelanggan.

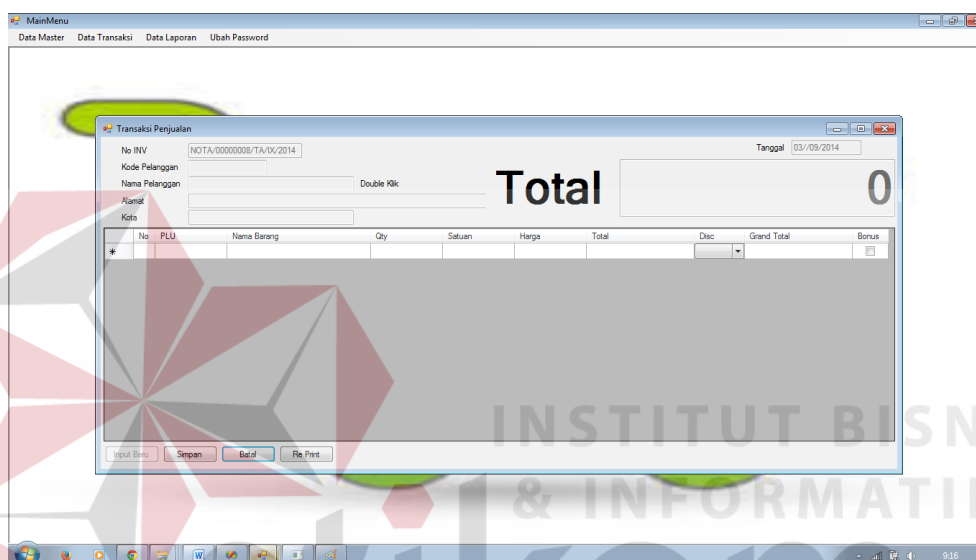
Adapun form tersebut dapat dilihat pada gambar di bawah ini:



Gambar 4.8 *Form* Pemesanan Barang

10. *Form* Penjualan

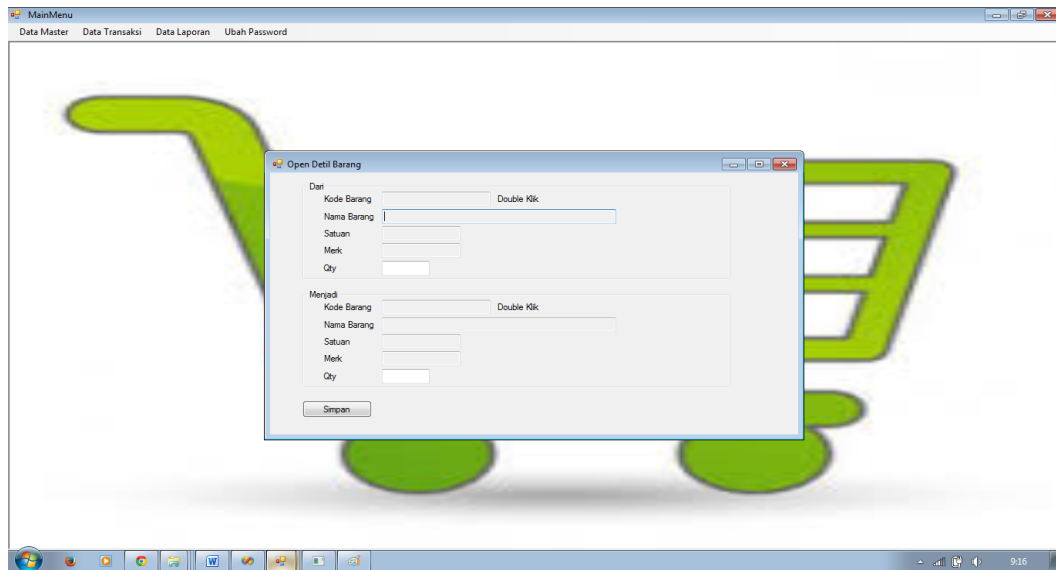
Form Penjualan berisikan data yang dimasukan oleh petugas kasir dengan menggunakan barcode scanner, lalu barang-barang yang sudah di scan akan masuk pada data grid view beserta harga dan total yang harus dibayar oleh pembeli. Button yang digunakan adalah simpan, print, dan batal, Adapun form tersebut dapat dilihat pada gambar di bawah ini:



Gambar 4.9 *Form* Penjualan

11. *Form* Transfer Barang dari Gudang ke Toko

Form Transfer Antar Barang digunakan untuk merubah barang satu pack menjadi eceran sehingga bisa dijual dengan harga eceran, hal tersebut dikarenakan Toko Agung Jaya yang menjual barang dengan eceran maupun grosir, cara menggunakannya tinggal double click di nama barang lalu convert, Adapun form tersebut dapat dilihat pada gambar di bawah ini:



Gambar 4.10 *Form* Transfer Antar Barang

12. *Form* Laporan Penjualan

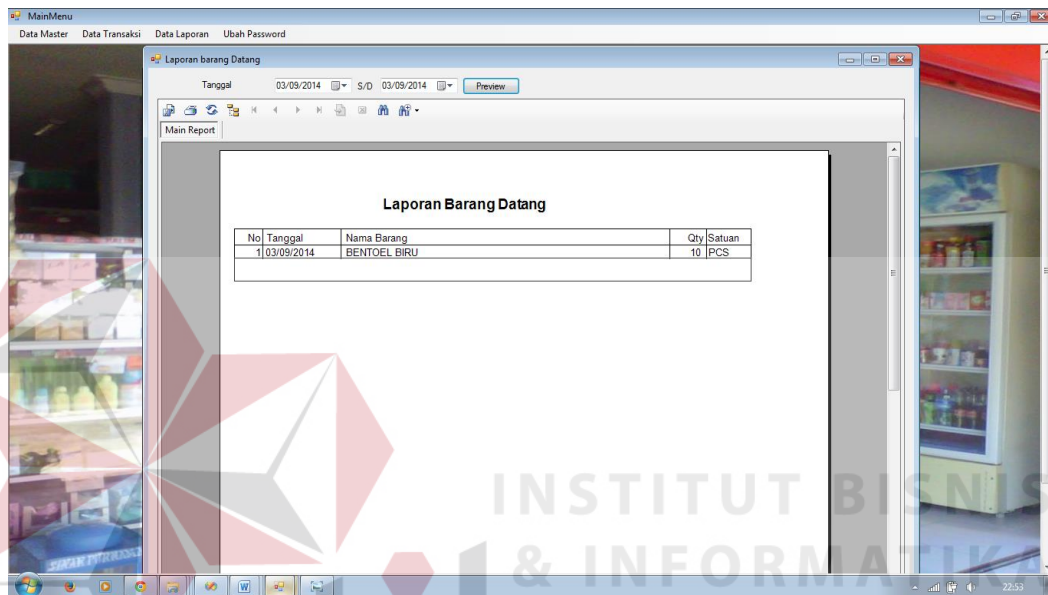
Form Penjualan Penjualan berisikan tampilan list dari penjualan pada Toko Agung Jaya selama hari, bulan, atau tahun, dalam form tersebut terdapat button print untuk mengeprint laporan penjualan yang ditampilkan. Adapun form tersebut dapat dilihat pada gambar di bawah ini:

No	Tanggal	Nama Barang	Qty	Harga	Total	Disc	Grand Total
1	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	0	25.000
2	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
3	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	5	23.750
4	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	0	25.000
5	03/09/2014	Surya 12	1	20.000	20.000	0	20.000
6	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
7	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	0	25.000
8	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
9	03/09/2014	Surya 12	1	20.000	20.000	0	20.000
10	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
11	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	0	25.000
12	03/09/2014	ROKOK BOKOR MAS	1	25.000	25.000	0	25.000
13	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
14	03/09/2014	Pepsodent white	1	10.000	10.000	0	10.000
15	03/09/2014	Surya 12	1	20.000	20.000	0	20.000
						Grand Total	267,750

Gambar 4.11 *Form* Laporan Penjualan

13. *Form* Laporan Barang Datang

Form Penjualan Barang Datang berisikan tampilan barang datang untuk update stok barang, diatas form terdapat button print, refresh, dan report, Adapun form tersebut dapat dilihat pada gambar di bawah ini:



Gambar 4.12 *Form* Laporan Barang Datang

14. *Form* Laporan Stok

Form Kartu Stok berisikan tampilan barang-barang yang sudah terupdate pada barang datang, diatas form terdapat button refresh, print, dan report untuk memudahkan analisa dalam kartu stok, Adapun form tersebut dapat dilihat pada gambar di bawah ini:

MainMenu
Data Master Data Transaksi Data Laporan Ubah Password

kartustok
Tanggal: 03/09/2014 S/D 03/09/2014
Kode Barang: 6996001355756
Nama Barang: BENTOEL BIRU
Semua Barang per Hari ini

Main Report

Kartu Stock Surabaya, 03-09-2014

Nama Barang : BENTOEL BIRU Merk : ROKOK
Satuan : PACK

No	Tanggal	Awal	Terima	Keluar	Akhir
1	03-09-2014	0	10	0	10
	03/09/2014				10

Gambar 4.13 Form Laporan Kartu Stok

15. Form Minimum Stok

Form warning stok barang digunakan untuk memberikan informasi kepada karyawan maupun pengurus (user) aplikasi mengenai barang-barang dagangan apa saja yang telah kurang dari sama dengan stok minimum yang ditentukan. Adapun Form warning stok barang dapat dilihat pada gambar dibawah ini:

MainMenu
Data Master Data Transaksi Data Laporan Ubah Password

kartustok
Tanggal: 03/09/2014 S/D 03/09/2014
Kode Barang:
Nama Barang:
Semua Barang per Hari ini

Main Report

Kartu Stock

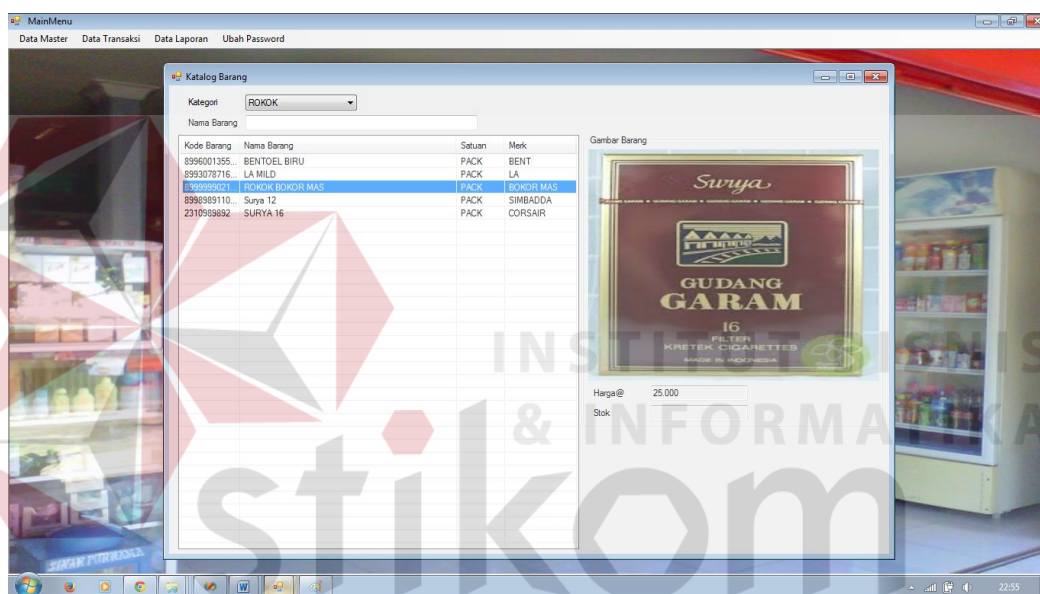
Stok Per Tanggal 03-09-2014

No	Nama Barang	Stok	Satuan	kategori
1	BENTOEL BIRU	10	PACK	ROKOK
2	Surya 12	(11)	PACK	ROKOK
3	SURYA 16	12	PACK	ROKOK

Gambar 4.14 Form Minimum Stok

16. Form Katalog Barang

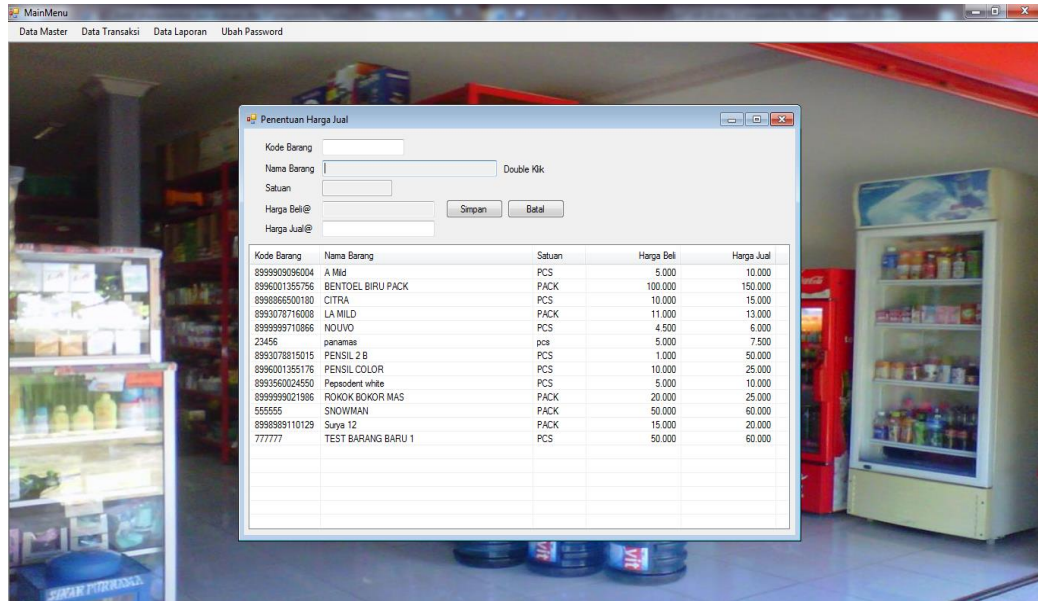
Form Katalog Barang adalah form yang berisikan barang-barang apa saja yang dijual pada Toko Agung Jaya, di dalam form terdapat kategori-kategori barang seperti sabun, alat tulis, dan lain-lain sehingga memudahkan pembeli untuk memilih barang yang akan dibeli, Adapun form tersebut dapat dilihat pada gambar di bawah ini:



Gambar 4.15 Form Katalog Barang

17. Form Harga Jual

Form Harga Jual berguna sebagai penetapan harga barang yang akan dijual melalui toko, sehingga selisihnya dari harga jual dan beli adalah hasil keuntungan yang didapat oleh toko



Gambar 4.16 *Form Harga Jual*

18. *Form Ubah Password*

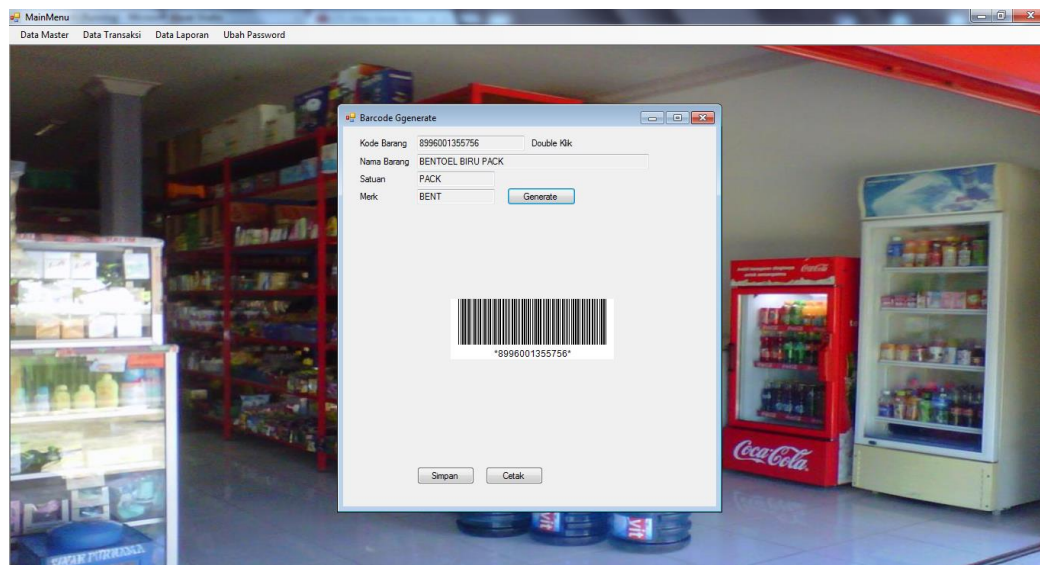
Form Ubah Password digunakan untuk pengguna aplikasi apabila ingin merubah password untuk login. Adapun form tersebut dapat dilihat pada gambar di bawah ini:

The screenshot shows a software window titled "Ubah Password" (Change Password). It contains three input fields: "User Name" (with the value "administrator"), "Password Lama" (Old Password), and "Password Baru" (New Password). Below the fields are two buttons: "Simpan" (Save) and "Cancel".

Gambar 4.17 *Form Ubah Password*

19. *Generate Barcode*

Form ini berguna untuk membuat barcode dari barang yang tidak memiliki barcode, di dalam form terdapat button simpan dan cetak setelah memilih id barang yang belum mempunyai *barcode*.

Gambar 4.18 *Form Barcode*

4.2 Evaluasi

Setelah melakukan implementasi sistem, tahap selanjutnya adalah melakukan uji coba dan evaluasi terhadap sistem. Tujuan evaluasi ini adalah untuk mengetahui apakah aplikasi yang telah dibuat sudah berjalan dengan baik dan sesuai tujuan atau *output* yang diharapkan. Evaluasi ini akan dibagi menjadi dua bagian, yaitu evaluasi uji coba fungsi aplikasi, analisa hasil coba sistem, dan evaluasi uji coba kemudahan penggunaan aplikasi. Uji coba ini akan dilakukan dengan menggunakan *black box testing*. Adapun evaluasi yang dilakukan adalah sebagai berikut:

4.2.1 Uji Coba Fungsi Aplikasi

Uji coba fungsi aplikasi ini dilakukan dengan cara mencoba melakukan *action* tertentu kepada tombol dan *input* data di dalam *textbox* ataupun data *gridview*. Adapun detil evaluasi uji coba fungsi aplikasi sebagai berikut:

a. Fungsi *Shortcut* tombol

Adapun daftar *shortcut* yang ditentukan dan direncanakan dalam pembuatan aplikasi ini antara lain sebagai berikut:

Tabel 4.2 Daftar *shortcut* tombol di dalam setiap *form*

No	Shortcut	Tombol
1	P	Print, simpan dan print
2	E	Email
3	<i>Enter</i>	OK, cari laporan, simpan, tambah ke daftar
4	<i>Esc</i>	Refresh, batal
5	<i>Home</i>	Keluar, kembali
6	<i>Delete</i>	Hapus
7	F12	Edit
8	Alt + A	Ambil foto, cari barang
9	<i>Space</i>	Pembayaran
10	<i>End</i>	Hitung

Fungsi *shortcut* telah diujikan kepada setiap *form* yang ada di dalam aplikasi dan hasilnya adalah *shortcut* tombol berjalan baik dan benar sesuai dengan perencanaan. Pada laporan ini, akan diberikan contoh cara uji coba *shortcut* tombol masing-masing 1 *form* di setiap menu aplikasi. 1 (satu) *form* untuk master, 1 (satu) *form* untuk transaksi, dan 1 (satu) *form* untuk laporan.

b. Form Master

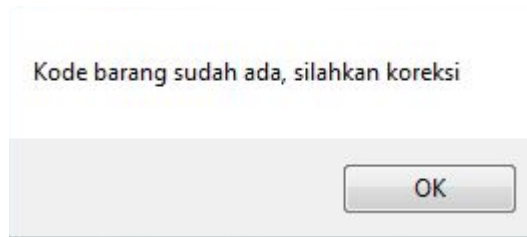
Kriteria uji coba *shortcut* pada *form* master pelanggan dapat dilihat pada tabel 4.3 di bawah ini:

Tabel 4.3 Hasil uji coba *shortcut* master

<i>Test Case ID</i>	Tujuan	<i>Input</i>	<i>Output yang diharapkan</i>	Status
1	Melakukan Fungsi <i>SAVE</i>	<i>Shortcut "Enter"</i>	Tanpa menekan tombol <i>form</i> , data pelanggan secara otomatis tersimpan. Kemudian keluar pesan "Data Telah Tersimpan"	1.Sukses 2.Data tersimpan di tabel pelanggan 3.Muncul

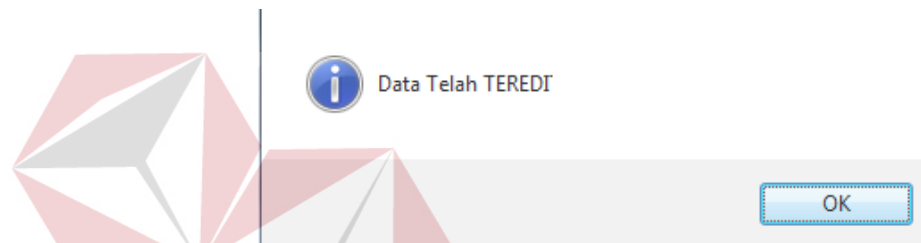
Test Case ID	Tujuan	Input	Output yang diharapkan	Status
				pesan yang diharapkan
2	Melakukan Fungsi <i>EDIT</i>	<i>Shortcut</i> "F12"	Tanpa menekan tombol <i>form</i> , data pelanggan secara otomatis teredit. Kemudian keluar pesan "Data Telah Teredit"	1.Sukses 2.Tabel pelanggan ter <i>update</i> 3.Muncul pesan yang diharapkan
3	Melakukan Fungsi HAPUS	<i>Shortcut</i> "Delete"	Tanpa menekan tombol <i>form</i> , data pelanggan secara otomatis terhapus. Kemudian keluar pesan pertanyaan "Yakin Hapus", jika dijawab "ya" maka akan keluar pesan "Data Telah Terhapus". Jika dijawab "Tidak" maka akan kembali ke <i>form</i> pelanggan	1.Sukses 2.Tabel pelanggan ter hapus 3.Muncul pesan yang diharapkan
4	Melakukan Fungsi BATAL	<i>Shortcut</i> "Esc"	Tanpa menekan tombol <i>form</i> , setiap <i>field</i> yang ada di <i>form</i> pelanggan secara otomatis akan dijadikan <i>default</i> .	1.Sukses 2. <i>Field</i> pada <i>form</i> pelanggan diganti dengan <i>default</i>
5	Melakukan Fungsi KELUAR	<i>Shortcut</i> "Home"	Tanpa menekan tombol <i>form</i> , secara otomatis <i>form</i> pelanggan akan keluar kemudian masuk ke <i>form</i> MASTER.	Sukses

Uji coba tabel 4.3 nomor 1 menghasilkan pesan konfirmasi dari data pelanggan yang dimasukkan pada tabel pelanggan yang ditandai dengan tampilnya pesan pada gambar di bawah ini:



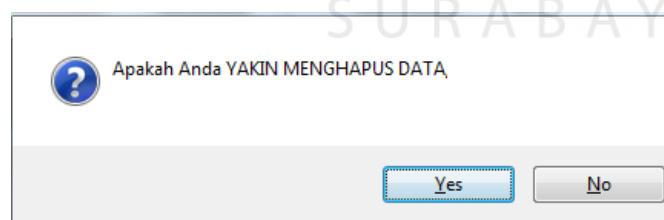
Gambar 4.19 Uji Coba Konfirmasi Data Telah Tersimpan

Uji coba tabel 4.3 nomor 2 menghasilkan pesan konfirmasi dari data pelanggan yang *diupdate* pada tabel pelanggan yang ditandai dengan tampilnya pesan pada gambar di bawah ini:



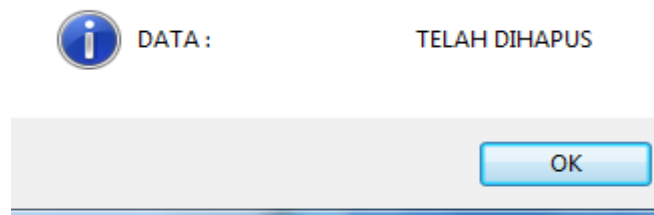
Gambar 4.20 Uji Coba Konfirmasi Data Telah Teredit

Uji coba tabel 4.3 nomor 3 menghasilkan pesan konfirmasi dari data pelanggan yang akan dihapus pada tabel pelanggan yang ditandai dengan tampilnya pesan pada gambar di bawah ini:



Gambar 4.21 Uji Coba Konfirmasi Data Akan Dihapus

Jika *user* menekan tombol “No” maka akan kembali ke *form* pelanggan, jika *user* menekan tombol “Yes” maka data pelanggan yang akan terhapus pada tabel pelanggan yang ditandai dengan tampilnya pesan pada gambar di bawah ini:



Gambar 4.22 Uji Coba Konfirmasi Data Terhapus

c. *Form* Transaksi Penjualan

Kriteria uji coba *shortcut* pada *form* transaksi penjualan dapat dilihat pada tabel 4.4 di bawah ini:

Tabel 4.4 Hasil uji coba *shortcut* transaksi penjualan

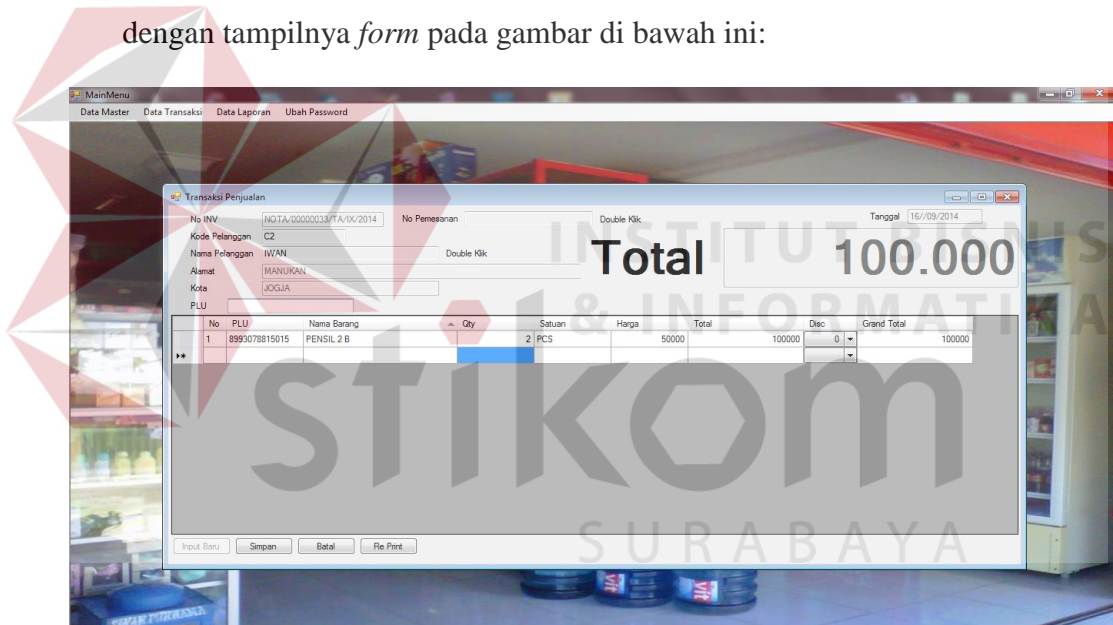
<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
1	Melakukan Fungsi Tambah Ke Daftar	<i>Shortcut</i> "Enter"	Tanpa menekan tombol <i>form</i> , data penjualan secara otomatis masuk ke dalam data <i>gridview</i> .	1.Sukses 2.Data penjualan masuk ke dalam data <i>gridview</i>
2	Melakukan Fungsi Cari Barang	<i>Shortcut</i> "Alt+A"	Tanpa menekan tombol <i>form</i> , daftar barang dagangan secara otomatis keluar di dalam <i>form</i> daftar barang.	1.Sukses 2.Muncul <i>form</i> barang dagangan disertai data barang
3	Melakukan Fungsi Pembayaran	<i>Shortcut</i> "Space"	Tanpa menekan tombol <i>form</i> , secara otomatis akan keluar <i>form</i> pembayaran	1.Sukses 2.Muncul <i>form</i> pembayaran 3. <i>Grand total</i> sesuai

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
			penjualan sesuai <i>grand total</i> .	
4	Melakukan Fungsi Batal	<i>Shortcut "Esc"</i>	Tanpa menekan tombol <i>form</i> , setiap <i>field</i> yang ada di <i>form</i> penjualan secara otomatis akan dijadikan <i>default</i> . Kemudian muncul pesan "Transaksi Dibatalkan"	1.Sukses 2. <i>Field</i> pada <i>form</i> penjualan diganti dengan <i>default</i> 3.Muncul pesan yang direncanakan
5	Melakukan Fungsi Keluar	<i>Shortcut "Home"</i>	Tanpa menekan tombol <i>form</i> , secara otomatis <i>form</i> penjualan akan keluar kemudian masuk ke <i>form</i> MASTER.	Sukses
6	Melakukan Fungsi Hapus Daftar	Klik Data <i>Gridview</i>	Akan keluar pesan konfirmasi "Yakin Hapus?" Jika dijawab "Yes" maka data akan terhapus dari data <i>gridview</i> . Jika dijawab "No" maka akan kembali ke <i>textbox</i> id barang.	1.Sukses 2.Muncul pesan konfirmasi hapus
7	Melakukan	<i>Shortcut</i>	Tanpa	1.Sukses

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
	Fungsi HITUNG	“End”	menekan tombol <i>form</i> , perhitungan akan dilakukan atas (<i>textbox</i> bayar) - (<i>textbox</i> pembulatan) = (<i>textbox</i> kembali).	2.Perhitungan benar
8	Melakukan Fungsi SIMPAN	<i>Shortcut</i> “Enter”	Tanpa menekan tombol <i>form</i> , data transaksi penjualan tersimpan di tabel transaksi dan detail penjualan. Kemudian Keluar pesan “Transaksi Tersimpan”	1.Sukses 2.Data tersimpan 3.Keluar Pesan yang diharapkan
9	Melakukan Fungsi Simpan & <i>Print</i>	<i>Shortcut</i> “P”	Tanpa menekan tombol <i>form</i> , akan keluar pesan “ <i>Print</i> atau kirim email” beserta nota penjualan	1.Sukses 2. <i>Form</i> konfirmasi <i>print</i> atau email keluar beserta nota penjualan
10	Melakukan Fungsi <i>PRINT</i>	<i>Shortcut</i> “P”	Tanpa menekan tombol <i>form</i> , nota akan tercetak. Kemudian kembali ke transaksi penjualan	1.Sukses 2.Nota tercetak 3.Kembali ke <i>form</i> transaksi penjualan
11	Melakukan Fungsi EMAIL	<i>Shortcut</i> “E”	Tanpa menekan tombol <i>form</i> ,	1.Sukses 2. <i>Form</i> konfirmasi email keluar

<i>Test Case ID</i>	<i>Tujuan</i>	<i>Input</i>	<i>Output yang diharapkan</i>	<i>Status</i>
			akan keluar pesan konfirmasi email	
12	Melakukan Fungsi Kirim Email	<i>Shortcut “Enter”</i> pada tombol <i>OK</i>	Tanpa menekan tombol pada <i>form</i> , nota akan terkirim dan keluar pesan “Terkirim”	1.Sukses 2.Nota terkirim ke email tujuan 3.Keluar pesan “Terkirim”

Uji coba tabel 4.4 menghasilkan data penjualan data *gridview* yang ditandai dengan tampilnya *form* pada gambar di bawah ini:



Gambar 4.23 Uji Penjualan

d. Fungsi Toolbox Di Dalam *Form*

Pada pengujian *toolbox* ini seluruh *form* telah diuji dan hasilnya sesuai dengan rencana. *Toolbox* berjalan lancar, benar, dan baik sesuai dengan fungsi yang telah ditetapkan.

e. Toolbox Error Provider

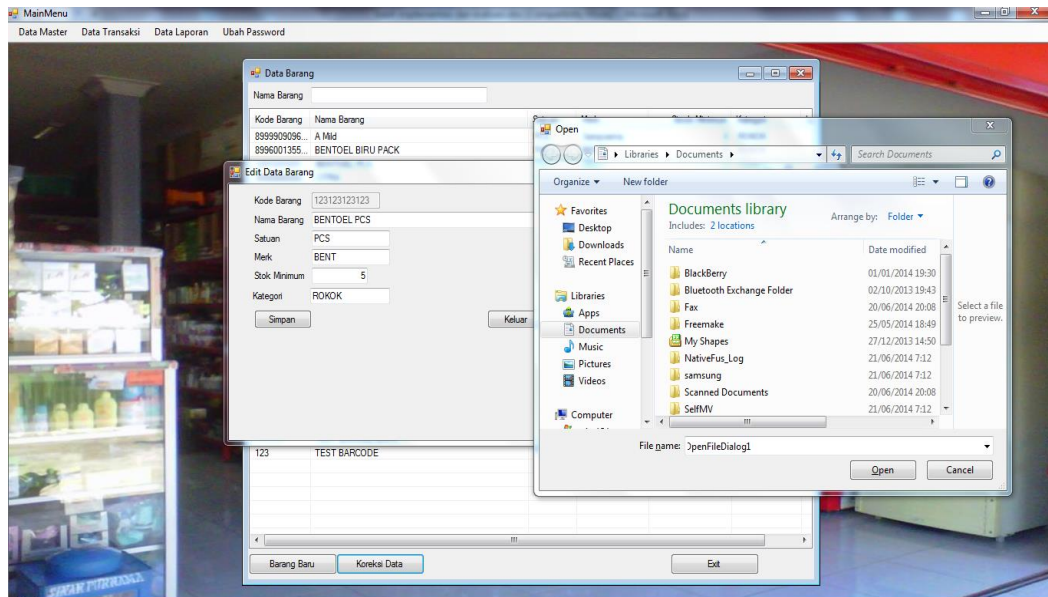
Uji coba *toolbox error provider* telah dilakukan di semua *form* pada aplikasi ini dan hasilnya *error provider* berjalan dengan baik dan benar. Pada laporan tugas akhir ini, akan ditampilkan salah satu uji coba *error provider* pada *form* master pelanggan. Jika *textbox* membutuhkan *input* angka, *user* harus memberikan *input* angka jika tidak *error provider* akan berjalan. Begitupun juga jika *textbox* membutuhkan *input* huruf, *user* harus memberikan *input* huruf jika tidak *error provider* akan berjalan. Adapun uji coba pada *form* master pelanggan



Gambar 4.24 Uji Coba *Error Provider*

f. Toolbox Open File Dialog

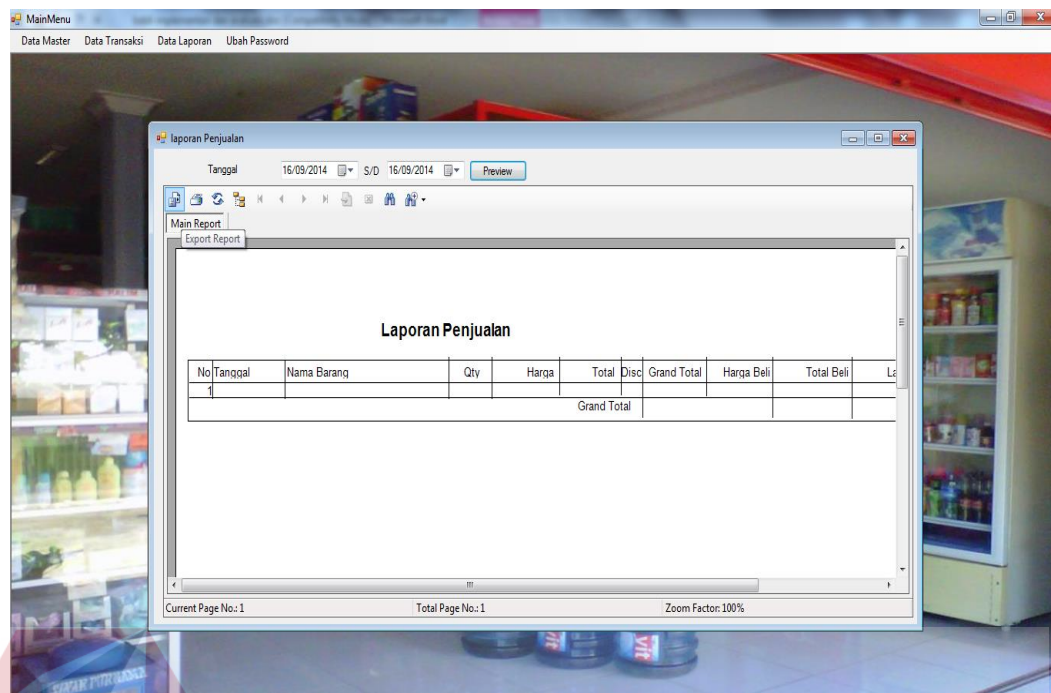
Uji coba *toolbox open file dialog* telah dilakukan di *form* pelanggan dan *form* pengguna, hasilnya kedua *form* tersebut berjalan dengan sukses, baik, dan benar. Pada laporan tugas akhir ini, akan ditampilkan salah satu uji coba *open file dialog* pada *form* master pelanggan. Jika tombol diberikan *event click* dan diisi dengan *open file dialog*, maka jika tombol tersebut di klik oleh *user* akan keluar *open file dialog*.



Gambar 4.25 Uji Coba *Open File Dialog*

g. *Toolbox Tool Tip*

Uji coba *toolbox tool tip* telah dilakukan di *form* transaksi penjualan dan *form* master barang, hasilnya kedua *form* tersebut berjalan dengan sukses, baik, dan benar. Pada laporan tugas akhir ini, akan ditampilkan salah satu uji coba *tool tip* pada *form* transaksi penjualan. Jika tombol diberikan *event mousehover* dan diisi dengan *tool tip*, maka jika tombol tersebut diberikan arah oleh *pointer* maka akan keluar kalimat sesuai dengan *setting* di dalam *coding*. Adapun uji coba pada *form* transaksi penjualan



Gambar 4.26 Uji Coba *Tool Tip*

- h. Fungsi *Query Maintenance (Select, Insert, Update, Delete)* dan *Cek Database*
- Fungsi *query maintenance (select, insert, update, delete)* dan *cek database* adalah suatu fungsi yang mutlak digunakan di setiap *form*, baik secara keseluruhan maupun sebagian. Misalkan pada setiap *form* master fungsi *query maintenance (select, insert, update, delete)* dan *cek database* digunakan, tetapi pada *form* laporan biasanya hanya fungsi *query cek database*. Berikut penjelasan mengenai fungsi *query*:

- i. *Form Login*

Form login digunakan untuk melakukan penyeleksian terhadap *user* yang masuk ke dalam sistem. Apabila *user* ingin masuk ke dalam sistem, *user* harus memasukkan *username* dan *password* kedalam *textbox username* dan *textbox password* yang tersedia di *form login*. Selanjutnya sistem melakukan seleksi terhadap *username* dan *password* yang telah dimasukkan dan

kemudian sistem menampilkan menu sesuai dengan hak akses yang diberikan untuk setiap bagian pengurus maupun setiap bagian pengguna.

Tabel 4.6 Daftar uji coba *username* dan *password*

NO	Nama Field	Data 1	Data 2	Jenis User
1	<i>Textbox Username</i>	31101	kasir	Pengguna
2	<i>Textbox Password</i>	10001	12345	Pemilik

Tabel 4.7 Hasil uji coba *username* dan *password*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1	Deskripsi <i>username</i> , <i>password</i> login yang valid	Memasukkan data 1	<i>Form Login</i> Tertutup dan menu pada <i>form</i> utama aktif sesuai hak akses pengguna	1.Sukses 2. <i>Login</i> Berhasil 3.Tampil <i>Form</i> utama sesuai hak akses
2	Deskripsi <i>username</i> , <i>password</i> login yang valid	Memasukkan data 2	<i>Form Login</i> Tertutup dan menu pada <i>form</i> utama aktif sesuai hak akses pemilik	1.Sukses 2. <i>Login</i> Berhasil 3.Tampil <i>Form</i> utama sesuai hak akses
3	Deskripsi <i>username</i> login valid, <i>password</i> login yang tidak valid	Memasukkan <i>username</i> = 31101, <i>password</i> = 99999	Muncul pesan " <i>Password Salah</i> "	1.Sukses 2. <i>Login</i> Gagal 3.Muncul pesan yang diharapkan
4	Deskripsi <i>username</i> login, <i>password</i> login yang kosong	Memasukkan <i>username</i> = , <i>password</i> = .	Muncul pesan " <i>Isi Textbox</i> "	1.Sukses 2. <i>Login</i> Gagal 3.Muncul pesan yang diharapkan
5	Deskripsi <i>logout</i>	Menekan tombol <i>logout</i>	Muncul pesan konfirmasi <i>logout</i> dan kembali ke <i>form</i> login	1.Sukses 2. <i>Form</i> utama keluar 3.Kembali ke menu <i>login</i>

Uji coba tabel 4.7 melakukan uji coba untuk *login* dengan *username* dan *password* salah satu pengguna.



Gambar 4.27 Uji Login User

j. *Form Master*

Setiap *form* master telah diuji coba fungsi *query*, hasil uji terhadap semua *form* master sukses dan berhasil melakukan perintah *select*, *insert*, *update*, maupun *delete*. Pada laporan tugas akhir ini, akan ditampilkan salah satu dari sejumlah *form* master tentang tahap uji coba. Pada kesempatan ini, *form* yang ditampilkan di laporan untuk contoh tahap uji coba adalah *form* master pelanggan. Adapun proses uji coba dan hasil uji coba pada *form* master pelanggan sebagai berikut:

Tabel 4.8 Daftar Uji Coba *Form* Master Pelanggan

No	Nama Field	Data Input SIMPAN	Data Input EDIT
1	ID pelanggan	123456	123456
2	Nama pelanggan	di Agung	Perdana Agung
5	Tanggal lahir		
6	Kota lahir		
7	Selular pelanggan		
8	Alamat rumah pelanggan		
9	Kota rumah pelanggan		

No	Nama Field	Data Input SIMPAN	Data Input EDIT
10	Alamat tinggal pelanggan	Manukan III / 29	Perum Candi
11	Kota tinggal pelanggan	Surabaya	Sidoarjo
12	Email pelanggan	diagung@gmail.com	perdanaagung@gmail.com
13	Jenis kelamin pelanggan	Laki-Laki	Laki-Laki
14	Foto pelanggan	E:\cobaFotoSimpan.jpg	E:\cobaFotoEdit.jpg

Tabel 4.9 Hasil uji coba Form Master Pelanggan

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1	Memastikan data masuk ke dalam <i>database</i> setelah ditekan tombol <i>simpan</i> pada <i>form</i> pelanggan	Memasukkan data <i>input</i> SIMPAN pada table 4.8	1. <i>Query Coding</i> benar 2. Keluar Pesan “Data telah tersimpan” 3. Data masuk ke dalam <i>database</i> 4. Keluar pada data <i>gridview form</i> pelanggan 5. Foto pelanggan disimpan dengan <i>binary</i> data	Sukses Sesuai <i>output</i> yang diharapkan
2	Memastikan data di dalam <i>database</i> telah <i>teredit</i> setelah ditekan tombol <i>edit</i> pada <i>form</i> pelanggan	Memasukkan data <i>input</i> EDIT pada table 4.8	1. <i>Query Coding</i> Benar 2. Keluar pesan “Data telah teredit” 3. Data di dalam <i>database</i> telah diganti dengan data baru 4. Data <i>Edit</i> keluar pada data <i>gridview form</i> pelanggan	Sukses Sesuai <i>output</i> yang diharapkan
3	Memastikan data di dalam <i>database</i> telah tehapus	Klik pada data <i>gridview</i> data atas ID = “123456”	1. <i>Query Coding</i> Benar 2. Keluar pesan konfirmasi hapus, jika dijawab “YES”, maka 3. Keluar Pesan “Data telah tehapus”	Sukses Sesuai <i>output</i> yang diharapkan

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
	setelah ditekan tombol hapus pada <i>form</i> pelanggan	, atas nama = "Perdana Agung"	4.Data di dalam <i>database</i> telah dihapus atau dihilangkan	
4	Memastikan data <i>gridview</i> pada <i>form</i> pelanggan telah terisi sesuai <i>query</i> pada <i>coding</i> dan tabel pelanggan	Lihat pada data <i>gridview</i> pada <i>form</i> pelanggan	1. <i>Query Coding</i> Benar 2.Data pada tabel pelanggan keluar seluruhnya pada data <i>gridview</i>	Sukses Sesuai <i>output</i> yang diharapkan

k. Transaksi Penjualan

Setiap *form* transaksi telah diuji coba fungsi *query*, hasil uji terhadap semua *form* transaksi sukses dan berhasil melakukan perintah *insert* (Simpan Transaksi). Pada kesempatan ini, *form* yang ditampilkan di laporan untuk contoh tahap uji coba simpan data transaksi adalah *form* transaksi penjualan. Adapun proses uji coba dan hasil uji coba pada *form* transaksi penjualan sebagai berikut:

Tabel 4.10 Daftar Uji Coba *Form* Transaksi Penjualan Dalam 1 x Transaksi

No	Nama Field	Data Barang 1	Data Barang 2
1	No transaksi	<i>Default</i> sistem	<i>Default</i> sistem
2	Tgl / Jam	<i>Default</i> Sistem	<i>Default</i> sistem
3	Nama pelanggan	<i>Select</i> sistem : "di Agung Perdana"	<i>Select</i> sistem : "di Agung Perdana"
4	Kode barang	B001	B002
5	Nama barang	<i>Select</i> sistem : "SNOWMAN BIRU"	<i>Select</i> sistem : "PENGHAPUS 2B"
6	Qty	10	5
7	Diskon	<i>Select</i> sistem : "0"	<i>Select</i> sistem : "0"
8	Jenis penjualan	Item	Box

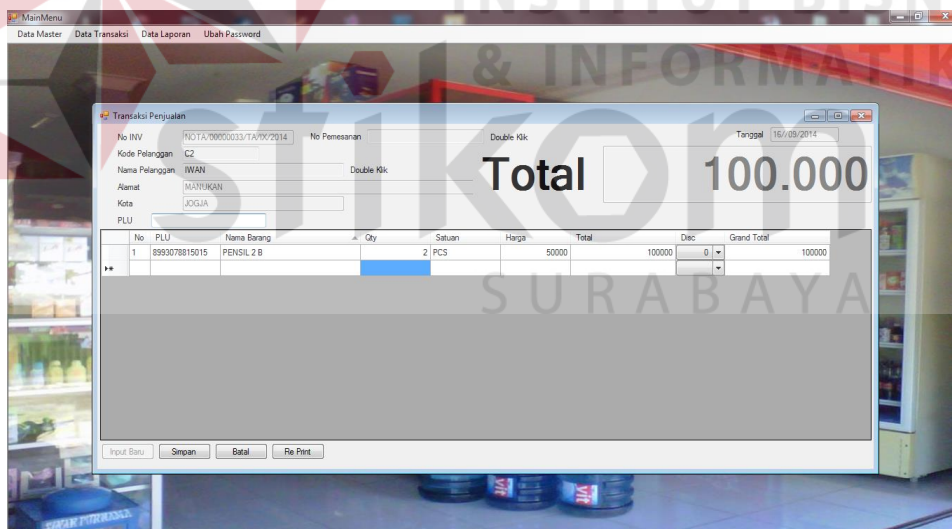
No	Nama Field	Data Barang 1	Data Barang 2
9	Data <i>gridview</i> :		
	Kode transaksi	<i>Default</i> sistem	<i>Default</i> sistem
	ID barang	B001	B002
	Nama barang	SNOWMAN BIRU	<i>PENGHAPUS 2B</i>
	Jumlah barang	10	5
	Jenis penjualan	<i>ITEM</i>	<i>BOX</i>
	Harga barang	1.100,00	25.100,00
	Diskon	0	0
	Sub total	11.000,00	125.500,00
	Isi per <i>boX</i>	15	12
	Keluar <i>item</i>	10	60

Tabel 4.11 Hasil uji coba *Form* Transaksi Penjualan Dalam 1 x Transaksi

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1	Memastikan data barang masuk didalam data <i>gridview</i> beserta perhitungan sub total dan <i>grand total</i>	Data barang kemudian tekan tombol tambah ke daftar	1.Data barang masuk ke dalam data <i>gridview</i> 2.Perhitungan sub total benar 3.Perhitungan <i>grand total</i> benar	Sukses Sesuai <i>output</i> yang diharapkan
2	Memastikan penjualan barang mempunyai stok	Data barang	1.Jika masih ada stok, maka langsung masuk ke daftar 2.Jika stok habis atau kurang, maka akan tampil pesan “Jumlah Barang Melebihi Stok <i>Ready</i> ”	Sukses Sesuai <i>output</i> yang diharapkan
3	Memastikan perhitungan nominal kembali dan data transaksi dapat disimpan di tabel transaksi penjualan dan detil transaksi penjualan	Data barang	1.Dapat menghitung nominal kembali 2. <i>Query coding</i> benar 3.Tampil Pesan “Transaksi Tersimpan” 4.Data transaksi masuk ke dalam tabel transaksi penjualan 5.Data transaksi masuk ke dalam tabel detil transaksi penjualan	Sukses Sesuai <i>output</i> yang diharapkan
4	Memastikan	Data	1.Jika ada barang yang kode	Sukses

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
	jika ada <i>input</i> data dengan kode barang yang sama, maka data sebelumnya akan ditambah data <i>input</i> yang baru.	barang (Jumlah Penjualan)	barang sama dengan <i>input</i> an sebelumnya, data yang sebelumnya tersebut akan diperbarui dengan data yang baru (ditambah). Tetapi jika jenis penjualannya beda akan membuat baru 2.Sub total akan terhitung ulang 3. <i>Grand</i> total akan terhitung ulang	Sesuai <i>output</i> yang diharapkan

Uji coba tabel 4.11 melakukan uji coba memastikan data barang dapat masuk di dalam data *gridview* beserta perhitungan sub total dan *grand* total. Uji coba tersebut ditandai dengan tampilnya *form*



Gambar 4.28 Uji Coba Penjualan Total Hitung

4.2.2 Analisa Hasil Uji Coba Sistem

Aplikasi yang telah dibuat telah diuji coba dan berjalan sesuai rencana. Setiap fungsi pada aplikasi telah diuji coba oleh penulis sendiri sebanyak 3 kali uji coba. Pengujian fungsi diuji coba setiap selesai melakukan *coding* atau

membuat fungsi tertentu. Misalnya fungsi simpan data pelanggan, penulis melakukan uji coba sebanyak 3 kali pada fungsi simpan data pelanggan dan begitu seterusnya dilakukan penulis pada setiap fungsi yang telah dibuat. Setelah dilakukan uji coba sebanyak 3 kali dan tidak mengalami kegagalan atau *error*, maka fungsi tersebut dinyatakan berhasil dibuat.

