

BAB IV

IMPLEMENTASI DAN EVALUASI SISTEM

4.1. Implementasi Sistem

4.1.1. Kebutuhan Sistem

Proses kebutuhan sistem adalah proses yang menjelaskan kebutuhan yang harus dipenuhi untuk menguji coba *user interface* dan menjalankan aplikasi pada penelitian ini. Adapun kebutuhan sistem ini harus terpenuhi agar aplikasi ini dapat berjalan dengan baik. Kebutuhan sistem tersebut yaitu:

1. Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras adalah komponen fisik yang membentuk suatu sistem komputer, serta peralatan-peralatan lain yang mendukung komputer dalam menjalankan tugasnya. Perangkat keras yang digunakan pada penelitian ini harus memiliki kinerja yang baik, sehingga aplikasi yang tersedia dapat diakses dengan baik. Spesifikasi perangkat keras yang digunakan pada penelitian ini adalah:

- a. Laptop dengan prosessor *Intel(R) Core(TM) i5-5200U CPU @ 2.20GHz*.
- b. Memori 4 *gigabytes* (GB) RAM.
- c. *VGA NVIDIA GeForce 820M 2 gigabytes* (GB) resolusi 1366 x 768
- d. *Printer*
- e. *Mouse*

2. Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak adalah suatu program komputer yang diperlukan pengguna untuk mengoperasikan fungsi dari perangkat keras. Perangkat lunak berperan sangat penting dalam penelitian ini karena tujuan dari penelitian ini adalah membuat suatu aplikasi penjualan. Adapun perangkat lunak yang digunakan dalam pembuatan aplikasi pada penelitian ini adalah:

- a. Sistem Operasi *Microsoft Windows 10*.
- b. *Microsoft SQL Server 2008* untuk membuat *database*.
- c. *Microsoft Visual Studio 2010* untuk melakukan pemrograman aplikasi.
- d. *PowerDesigner 15* untuk membuat *Entity Relationship Diagram*
- e. *PowerDesigner 6* untuk membuat *Context Diagram* dan *Data Flow*

Diagram

4.1.2. Pembuatan Aplikasi dan Instalasi Aplikasi

Proses ini dilakukan setelah melakukan pembuatan desain dan menjelaskan kebutuhan sistem. Proses ini menjelaskan tentang pembuatan aplikasi dan tahapan instalasi aplikasi. Aplikasi Penjualan Pada Mitra 10 Wiyung Surabaya ini dibuat dengan menggunakan Visual Studio 2010. Aplikasi ini hanya digunakan untuk satu komputer atau laptop. Untuk tahapan membuat aplikasi sebagai berikut:

1. Pembuatan *Database*

Database yang digunakan untuk pembuatan program pada penelitian ini menggunakan *database SQL Server 2008*.

2. Coding Aplikasi

Sebelum melakukan *coding*, aplikasi ini terdiri dari 12 *form* dan terdapat 3 modul/proses inti utama, yaitu:

a. Modul/proses *input* data ke dalam *database*.

Diproses ini terdapat sub proses *input* data *member* yang dilakukan di *form master member*, sub proses *input* data jenis barang yang dilakukan di *form master jenis barang*, sub proses *input* data barang yang dilakukan di *form master barang*, dan sub proses *input* data diskon yang dilakukan di *form master diskon*.

b. Modul/proses menghitung transaksi penjualan

Diproses ini terdapat proses menginputkan data transaksi dan menghitung total harga yang harus dibayar serta menghitung total penjualan dalam satu kali transaksi. Terdapat juga proses mencetak bukti transaksi dan menyimpan transaksi ke dalam *database* yang dilakukan di *form transaksi*.

c. Modul/proses pelaporan

Proses ini adalah proses untuk menampilkan, menyimpan, dan mencetak laporan. Untuk menampilkan, menyimpan dan mencetak laporan stok barang dibutuhkan *form* laporan stok barang yang akan menghasilkan laporan stok barang. Untuk menampilkan, menyimpan dan mencetak rekapitulasi penjualan dibutuhkan *form* rekapitulasi penjualan yang akan menghasilkan rekapitulasi penjualan dan total penjualan dalam satu hari. Untuk menampilkan, menyimpan dan mencetak laporan pendapatan dibutuhkan *form* laporan pendapatan yang akan menghasilkan laporan pendapatan, dan total pendapatan toko.

Setelah menjelaskan tentang alur dari modul/proses aplikasi dan *form* maka selanjutnya dilakukan tahapan coding aplikasi. Coding aplikasi adalah memberikan nilai/fungsi di *form* aplikasi agar aplikasi bisa berjalan atau berfungsi sesuai dengan alur yang sudah dirancang.

3. Instalasi Aplikasi

Untuk menjalankan aplikasi tersebut agar berjalan dengan baik diperlukan tahapan-tahapan install, tahapan instalasi sebagai berikut:

- a. Install sistem operasi *Microsoft Windows 7/8/10* pada laptop atau komputer.
- b. Install aplikasi penjualan pada laptop atau komputer.
- c. Install aplikasi *database* microsoft SQL Server 2008 pada laptop atau komputer.
- d. *Attach database* aplikasi penjualan pada laptop atau komputer.
- e. Kemudian jalankan aplikasi penjualan.

4.1.3. Uji Coba *User Interface*

Setelah kebutuhan sistem terpenuhi, maka selanjutnya adalah mengimplementasikan ke dalam Aplikasi Penjualan Pada Mitra 10 Wiyung. Pada saat aplikasi dijalankan yang pertama kali tampil adalah tampilan *login*. Dari tampilan ini, pengguna (*user*) harus *login* terlebih dulu dengan memasukkan *username*, *password*, dan jabatan agar dapat melanjutkan ke halaman menu selanjutnya dengan menekan tombol *login* atau menekan enter pada setelah mengisi semua kolom. Seperti terlihat pada gambar 4.1.

The screenshot shows a window titled "FormLOGIN" with a "LOGIN" heading. Below the heading are two input fields: "Username" containing "kasir01" and "Password" containing "kasir001". At the bottom right, there are two buttons labeled "LOGIN" and "BACK".

Gambar 4.1. Tampilan *form Login*

4.1.4. Tampilan Form Menu

Tampilan menu utama pada gambar 4.2 ini berisi tampilan menu-menu yaitu menu *form master*, menu transaksi, dan menu laporan. Untuk menu *Master* terdiri dari *form jenis barang*, *form member*, *form barang*, dan *form promo*, *form pelanggan*, dan *form COA*. Untuk menu transaksi terdiri dari *form penjualan*. Dan untuk menu laporan terdiri dari *form stok barang*, *form rekap penjualan*, *form pendapatan* dan *form jurnal*, seperti terlihat pada gambar 4.2.

The screenshot shows a window titled "FormMENU" for "MITRA 10 WIYUNG SURABAYA". The menu is organized into three columns: "MASTER", "TRANSAKSI", and "LAPORAN".

MASTER		TRANSAKSI	LAPORAN	
JENIS BARANG	MEMBER	PENJUALAN	STOK BARANG	REKAP PENJUALAN
BARANG	PROMO	PEMESANAN	PENDAPATAN	JURNAL
PELANGGAN	COA	PEMBAYARAN		

Gambar 4.2. Tampilan *form Menu*

4.1.5. Tampilan Form Master

Tampilan *form master* muncul ketika pengguna memilih *icon form master* pada menu utama dan sebelumnya sudah melakukan *login*. Tampilan *form master* ini terbagi menjadi 4 *form* yaitu *master member*, *master jenis barang*, *master diskon* serta *master barang*. Dalam *form master member* terdapat 5 *text box* (kode, nama, alamat, no telpon, diskon member), 1 *combobox* (jenis kelamin), 1 *datagridview* untuk melihat data yang sudah tersimpan, dan 3 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, dan *button* ubah untuk mengubah salah satu data yang sudah tersimpan. Seperti terlihat pada gambar 4.3.

MITRA 10 - MEMBER

Mitra 10
MILIKI MITRA 10 & BERSAMA SAMA

MEMBER

ID MEMBER : M0004

DISKON MEMBER : 30 %

NAMA :

ALAMAT :

JENIS KELAMIN :

NO. TELP :

TAMPILAN

ID_MEMBER	NAMA_MEMBER	ALAMAT_MEMBER
M0001	Agus Prima	Wiyung Pratama 10
M0002	Susiani Wira	Citraland
M0003	Ruth Handayani	Kedurus

Gambar 4.3. Tampilan *form master member*

Kemudian untuk *form master* jenis barang, terdapat 2 *textbox* untuk kode dan nama jenis barang, 1 *datagridview* untuk melihat data yang sudah tersimpan, dan 3 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, dan *button* ubah untuk mengubah salah satu data yang sudah tersimpan. Seperti terlihat pada gambar 4.4.



Gambar 4.4. Tampilan *form master* jenis barang

Untuk *form master* promo barang, terdapat 4 *textbox* (kode promo, jumlah promo, keterangan promo, dan kode barang), 2, *datetimepicker* tanggal awal dan tanggal akhir promo, 3 *datagridview* untuk melihat data yang sudah tersimpan, tampilan kode barang yang termasuk promo, dan tampilan detail promo, dan 4 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, *button* ubah untuk mengubah salah satu data yang sudah tersimpan, dan *button* untuk tambah barang promo. Seperti terlihat pada gambar 4.5

MITRA 10 - PROMO



PROMO

M. PROMO

KODE PROMO :

NAMA PROMO :

JUMLAH DISKON : %

TANGGAL AWAL :

TANGGAL AKHIR :

KETERANGAN :

D. PROMO

KODE PROMO :

KODE BARANG :

TAMPILAN MASTER PROMO

KODE_PROMO	NAMA_PROMO	JUMLAH_PP
PB0001	Promo Septem...	10
PB0002	Promo 17 Agus...	30

TAMPILAN DETAIL PROMO

KODE_PROMO	KODE_BARANG
PB0001	B0003
PB0001	B0005
PB0001	B0010

Gambar 4.5. Tampilan *form master* promo

Untuk *form master* barang, terdapat 6 *textbox* (kode barang, nama barang, kode jenis, jumlah barang, harga barang per satuan), 1 *combobox* (jenis barang), 1 *datagridview* untuk melihat data yang sudah tersimpan, dan 3 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, dan *button* ubah untuk mengubah salah satu data yang sudah tersimpan. Seperti terlihat pada gambar 4.6.

MITRA 10 - BARANG



BARANG

KODE BARANG :

KODE JENIS :

NAMA BARANG :

JUMLAH BARANG :

HARGA BARANG :

TAMPILAN

KODE_	KODE_	NAMA_BARANG	JUMI	HARGA
B0001	JB0001	Keramik Asia Tale 30x30	2000	65000
B0002	JB0001	Keramik Granit 30x30	2000	80000
B0003	JB0002	Kompor Rinna	200	250000
B0004	JB0003	Bathtub	100	25000...
B0008	JB0007	Cat AquaProfi White	300	60000
B0005	JB0004	Tenda Kemah	250	900000
B0006	JB0006	Lampu Philips LED 5W	500	100000
B0009	JB0009	Jam Dinding	200	150000
B0007	JB0007	Cat Catilac Biru	200	45000
B0010	JB0010	Pompa Hidrolik	250	250000

Gambar 4.6. Tampilan *form master* barang.

Untuk *form master* pelanggan, terdapat 4 *textbox* (kode pelanggan, nama pelanggan, alamat, dan no telpon), 1 *combobox* (jenis kelamin), 1 *datagridview* untuk melihat data yang sudah tersimpan, dan 3 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, dan *button* ubah untuk mengubah salah satu data yang sudah tersimpan. Seperti terlihat pada gambar 4.7.

The screenshot shows a software window titled "MITRA 10 - PELANGGAN". On the left, there is a form with the following fields: "ID PELANGGAN : PL0003", "NAMA :", "ALAMAT :", "JENIS KELAMIN : [dropdown]", and "NO. TELP :". Below these fields are three icons: a floppy disk (save), a pencil (edit), and a trash can (delete). On the right, a "TAMPILAN" table displays customer data:

	ID_PELANGGAN	NAMA_PELANGGAN	ALAMAT_PELANG
▶	PL0001	Harianto	Bangkingan
	PL0002	Suls	Karangpilang
*			

Gambar 4.7. Tampilan *form master* pelanggan.

Untuk *form master* COA, terdapat 2 *textbox* (kode akun, nama akun), 1 *datagridview* untuk melihat data yang sudah tersimpan, dan 3 *button*, yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang sudah tersimpan, dan *button* ubah untuk mengubah salah satu data yang sudah tersimpan. Seperti terlihat pada gambar 4.8.

MITRA 10 - CHART OF ACCOUNT

Mitra10 **CHART OF ACCOUNT**
BUKTI NO. INTERNASIONAL & KARESI INDEPENDENT

NO. AKUN :

NAMA AKUN :

TAMPILAN

	NO_AKUN	NAMA_AKUN
▶	111	Kas
	211	Pembayaran di muka
	411	Penjualan
*		

Gambar 4.8. Tampilan *form master* COA.

4.1.6. Tampilan Transaksi Penjualan

Tampilan *form* transaksi muncul ketika pengguna memilih *icon* transaksi pada menu utama dan sebelumnya sudah melakukan *login*. Tampilan *form* transaksi ini terdiri dari 14 *textbox* (no. transaksi (otomatis terisi sendiri), kode barang, nama barang, harga barang, diskon barang, harga promo, jumlah barang, total harga, sub total, pajak, total penjualan, tunai, kembali, id member), 1 *datetimepicker* untuk menampilkan tanggal, 1 *datagridview* untuk menampilkan barang yang terjual ke pelanggan, dan 4 *button* yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang ada di *datagridview*, *button* tambah untuk menambah daftar barang baru ke dalam *datagridview*, dan *button* cari untuk mencari id member. Adapun tampilan *form* transaksi adalah seperti gambar 4.9.

MITRA 10 - TRANSAKSI PENJUALAN



PENJUALAN

TUNAI :

KEMBALI :

ID MEMBER : 

TANGGAL : 

NO. TRANSAKSI :

KODE BARANG :

NAMA BARANG :

HARGA :

DISKON :

HARGA PROMO :

JUMLAH :

TOTAL HARGA :

SUB TOTAL :

PAJAK :

TOTAL JUAL :




TAMPILAN

	Kode Barang	No. Transaksi	Nama Barang
▶	TP0001	B0009	Jam Dinding
*			

Gambar 4.9. Tampilan *form* transaksi penjualan

4.1.7. Tampilan Transaksi Pemesanan

Tampilan *form* pemesanan muncul ketika pengguna memilih *icon* pemesanan pada menu utama dan sebelumnya sudah melakukan *login*. Tampilan *form* pemesanan ini terdiri dari 14 *textbox* (no. transaksi (otomatis terisi sendiri), kode barang, nama barang, harga barang, diskon barang, harga promo, jumlah barang, total harga, sub total, pajak, total penjualan, keterangan, uang muka, sisa pembayaran, id pelanggan), 1 *datetimepicker* untuk menampilkan tanggal, 1 *datagridview* untuk menampilkan barang yang terjual ke pelanggan, dan 5 *button* yaitu *button* simpan untuk menyimpan data, *button* hapus untuk menghapus salah satu data yang ada di *datagridview*, *button* pelanggan untuk membuat data pelanggan baru, *button* cari untuk mencari pelanggan yang sudah tersimpan dan

button tambah untuk menambah daftar barang baru ke dalam *datagridview*. Adapun tampilan *form* transaksi adalah seperti gambar 4.10.

MITRA 10 - TRANSAKSI PEMESANAN

Mitra10
SOLUSI PERUSAHAAN & KEBERUSAHAAN

PEMESANAN

TANGGAL : 30/08/2016

NO. TRANSAKSI : TS0011

KODE BARANG :

NAMA BARANG :

HARGA :

DISKON :

HARGA PROMO : 0

JUMLAH :

TOTAL HARGA :

SUB TOTAL : 5000000

PAJAK : 10

TOTAL JUAL : 5500000

KETERANGAN : Pembayaran DP

UANG MUKA : 1100000

SISA PEMBAYARAN : 4400000

ID PELANGGAN : PL0001

TAMPILAN

	Kode Barang	No. Transaksi	Nama Barang
▶	TS0011	B0010	Pompa Hidrolik
*			

Gambar 4.10. Tampilan *form* transaksi pemesanan

4.1.8. Tampilan Transaksi Pembayaran

Tampilan *form* pembayaran muncul ketika pengguna memilih *icon* pembayaran pada menu utama dan sebelumnya sudah melakukan *login*. Tampilan *form* pembayaran ini terdiri dari 8 *textbox* (no. transaksi (otomatis terisi sendiri), no pemesanan, pembayaran dp, pembayaran sisa, total, tunai, kembali, id member/id pelanggan), 2 *datetimepicker* untuk menampilkan tanggal sekarang dan tanggal pemesanan, dan *button* simpan untuk menyimpan data. Adapun tampilan *form* transaksi adalah seperti gambar 4.11.

FormTRANSAKSIPEMBAYARAN

Mitraio
BUILDING MATERIALS & HOME IMPROVEMENT

PEMBAYARAN

TANGGAL : 30/08/2016

NO. TRANSAKSI : TB0002

NO. PEMESANAN : TS0001

TANGGAL PESAN : 29/08/2016

PEMBAYARAN DP : 5280000

PEMBAYARAN SISA : 21120000

TOTAL : 24000000

TUNAI : 30000000

KEMBALI : 8880000

ID MEMBER / ID PELANGGAN : PL0001

KETERANGAN : Lunas

INSTITUT BISNIS & INFORMATIKA
stikom
SURABAYA

Gambar 4.11. Tampilan *form* transaksi pembayaran

4.1.9. Tampilan *Form* Laporan

Tampilan *form* laporan muncul ketika pengguna memilih *icon* laporan pada menu utama dan sebelumnya sudah melakukan *login*. Tampilan *form* laporan ini terbagi menjadi 3 yaitu *form* laporan stok barang, *form* rekapitulasi pendapatan, dan *form* laporan pendapatan. Untuk *form* laporan stok barang terdiri dari sebuah *datagridview*, dan sebuah *button* yaitu *button* cetak untuk mencetak data laporan. Adapun tampilan *form* laporan stok seperti gambar 4.12.

MITRA 10 - STOK BARANG

	KODE_BARANG	KODE_JENIS	NAMA_BARANG	JUMLAH_BARANG	HARGA_BARANG
▶	B0001	JB0001	Keramik Asia Tal...	2000	65000
	B0002	JB0001	Keramik Granit 3...	2000	80000
	B0003	JB0002	Kompom Rinna	200	250000
	B0004	JB0003	Bathtub	100	2500000
	B0008	JB0007	Cat AquaProff W...	300	60000
	B0005	JB0004	Tenda Kemah	250	900000
	B0006	JB0006	Lampu Philips LE...	500	100000
	B0009	JB0009	Jam Dinding	200	150000
	B0007	JB0007	Cat Catilac Biru	200	45000
	B0010	JB0010	Pompa Hidrolik	250	250000
*					

Gambar 4.12. Tampilan *Form* Laporan Stok

Untuk *form* laporan rekapitulasi penjualan terdiri dari sebuah *datetimepicker*, sebuah *datagridview*, dan sebuah *button* yaitu *button* cetak untuk mencetak laporan. Adapun tampilan *form* laporan rekapitulasi penjualan adalah seperti gambar 4.13.

MITRA 10 - REKAPITULASI PENJUALAN

Tanggal Transaksi 30/08/2016

	NO_PENJUALAN	KODE_BARANG	NAMA_BARANG	HARGA_BARANG	JUMLAH_JUAL	TOTAL_JU
▶	TP0002	B0001	Keramik Asia Tal...	65000	200	13000000
	TP0001	B0003	Kompom Rinna	250000	1	250000
	TP0004	B0003	Kompom Rinna	250000	1	250000
	TP0003	B0006	Lampu Philips LE...	100000	2	200000
*						

Gambar 4.13. Tampilan *form* rekapitulasi penjualan

Untuk *form* laporan rekapitulasi pembayaran terdiri dari sebuah *datetimepicker*, sebuah *datagridview*, dan sebuah *button* yaitu *button* cetak untuk mencetak laporan. Adapun tampilan *form* laporan rekapitulasi pembayaran adalah seperti gambar 4.14.

MITRA 10 - REKAPITULASI PEMBAYARAN JOB ORDER

Mitra10
BUILDING MATERIALS & HOME IMPROVEMENT

REKAP PEMBAYARAN

Tanggal Transaksi: 30/08/2016

NO_PEMBARAN	NO_PEMESANAN	KODE_BARANG	NAMA_BARANG	HARGA_BARANG	JUMLAH
TBD001	TS0001	B0002	Keramik Granit 3...	80000	300

INSTITUT BISNIS & INFORMATIKA

stikom

Gambar 4.14. Tampilan *form* rekapitulasi pembayaran

Untuk *form* jurnal penjualan terdiri dari sebuah *datetimepicker*, sebuah *button* yaitu *button* cetak untuk mencetak jurnal penjualan, dan sebuah *datagridview* untuk menampilkan rekapitulasi penjualan. Adapun tampilan *form* laporan pendapatan adalah seperti gambar 4.15.

FormJURNALPENJUALAN

Mitraio
BUILDING MATERIALS & HOME IMPROVEMENT

JURNAL PENJUALAN

Tanggal Jurnal: 30/08/2016

	NO	TANGGAL_JL	NO_	NO_PENJU	NAMA_AKU	DEBET_P	KREDIT_P	KET_JURNAL_PENJUALAN
▶	1	29/08/2016	411	TP0001	Penjualan	0	275000	
	1	29/08/2016	111	TP0001	Kas	275000	0	
	3	29/08/2016	411	TP0002	Penjualan	0	10010000	
	3	29/08/2016	111	TP0002	Kas	10010000	0	
	5	29/08/2016	411	TP0003	Penjualan	0	154000	
	5	29/08/2016	111	TP0003	Kas	154000	0	
	7	29/08/2016	411	TP0004	Penjualan	0	192500	
	7	29/08/2016	111	TP0004	Kas	192500	0	

Gambar 4.15. Tampilan *form* Jurnal Penjualan

Untuk *form* jurnal pemesanan terdiri dari sebuah *datetimepicker*, sebuah *button* yaitu *button* cetak untuk mencetak jurnal pemesanan, dan sebuah *datagridview* untuk menampilkan jurnal pemesanan. Adapun tampilan *form* laporan pendapatan adalah seperti gambar 4.16.

FormJURNALPEMESANAN

Mitraio
BUILDING MATERIALS & HOME IMPROVEMENT

JURNAL PEMESANAN

Tanggal Jurnal: 29/08/2016

	NO	TANGGAL_JURNAL	NO_PEMI	NO_	NAM.	DEBET_PEM	KREDIT_PEMESA	KET_JURNAL_PEMES.
▶	1	29/08/2016	TS0001	111	Kas	26400000	0	Pembayaran DP
	3	29/08/2016	TS0002	111	Kas	14300000	0	Pembayaran DP
	5	29/08/2016	TS0003	111	Kas	1760000	0	Pembayaran DP
	7	29/08/2016	TS0004	111	Kas	19800000	0	Pembayaran DP
	9	29/08/2016	TS0005	111	Kas	1980000	0	Pembayaran DP
	11	29/08/2016	TS0006	111	Kas	8800000	0	Pembayaran DP
	13	29/08/2016	TS0007	111	Kas	990000	0	Pembayaran DP
	15	29/08/2016	TS0008	111	Kas	990000	0	Pembayaran DP

Gambar 4.16. Tampilan *form* Jurnal Pemesanan

Untuk *form* jurnal penyesuaian terdiri dari sebuah *datetimepicker*, sebuah *button* yaitu *button* cetak untuk mencetak jurnal penyesuaian, dan sebuah *datagridview* untuk menampilkan jurnal penyesuaian. Adapun tampilan *form* laporan pendapatan adalah seperti gambar 4.17.

	NO_JURNAL_PEN	TANGGAL_JURNAL	NO_PEMBAYARAN	NO_AKUN	NAMA_AKUN	DEBET_P
▶	19	29/08/2016	TB0001	211	Pembayaran di m...	0
*	19	29/08/2016	TB0001	111	Kas	24000000

Gambar 4.17. Tampilan *form* Jurnal Penyesuaian

4.1.10. Tampilan Laporan Transaksi Penjualan

Laporan (*invoice*) penjualan ini digunakan untuk mengetahui proses transaksi penjualan yang dilakukan oleh perusahaan kepada pelanggan

MITRA 10 WIYUNG	
No. Transaksi : TP0001	Tanggal : 29/08/2016
1 Kompor Rinna	250,000.00
	250,000.00
	10
	250,000.00

Gambar 4.18. Tampilan *Invoice* Transaksi

4.1.11. Tampilan Laporan Stok Barang

Laporan ini digunakan oleh pihak perusahaan untuk mengetahui stok barang yang tersedia di perusahaan.

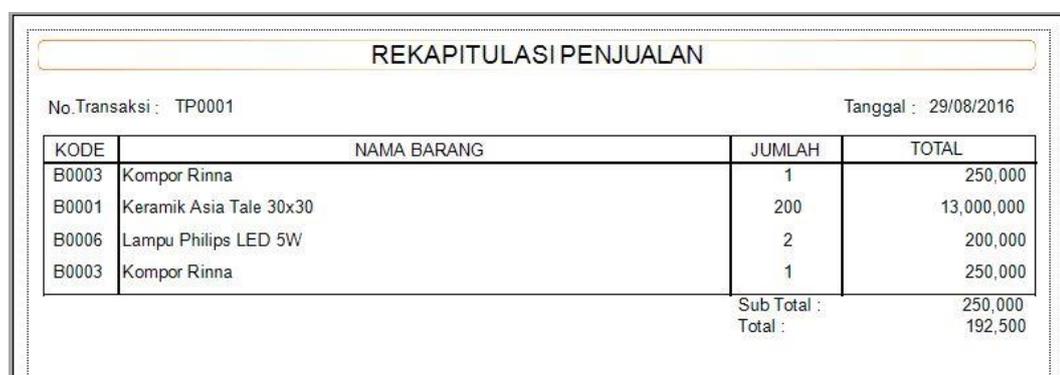


KODE BARANG	NAMA BARANG	HARGA	TOTAL STOK
B0001	Keramik Asia Tale 30x30	65,000	2,000
B0002	Keramik Granit 30x30	80,000	2,000
B0003	Kompot Rinna	250,000	200
B0004	Bathtub	2,500,000	100
B0008	Cat AquaProff White	60,000	300
B0005	Tenda Kemah	900,000	250
B0006	Lampu Philips LED 5W	100,000	500
B0009	Jam Dinding	150,000	200
B0007	Cat Catilac Biru	45,000	200
B0010	Pompa Hidrolik	250,000	250

Gambar 4.19. Tampilan Laporan Stok Barang

4.1.12. Tampilan Laporan Rekapitulasi Penjualan

Laporan ini digunakan oleh pihak perusahaan untuk mengetahui hasil rekapitulasi penjualan yang terjadi dalam satu hari



REKAPITULASI PENJUALAN			
No.Transaksi : TP0001		Tanggal : 29/08/2016	
KODE	NAMA BARANG	JUMLAH	TOTAL
B0003	Kompot Rinna	1	250,000
B0001	Keramik Asia Tale 30x30	200	13,000,000
B0006	Lampu Philips LED 5W	2	200,000
B0003	Kompot Rinna	1	250,000
		Sub Total :	250,000
		Total :	192,500

Gambar 4.20. Tampilan Laporan Rekapitulasi Penjualan

4.1.13. Tampilan Laporan Rekapitulasi Pembayaran

Laporan ini digunakan oleh pihak perusahaan untuk mengetahui hasil rekapitulasi pembayaran yang terjadi

REKAPITULASI PEMBAYARAN					
No. Transaksi Pembayaran : TB0001			Tanggal : 30/08/2016		
No Pesanan	Nama Barang	Jumlah	Total Bayar	Sudah Dibayar	Sisa Pembayaran
TS0001	Keramik Granit 30x30	300	24,000,000	5,280,000	21,120,000

Gambar 4.21. Tampilan Laporan Rekapitulasi pembayaran

4.1.14. Tampilan Jurnal Penjualan

Laporan ini dapat digunakan oleh pihak perusahaan untuk mengetahui jurnal penjualan dari hasil penjualan

 JURNAL PENJUALAN PER PERIODE 29/08/2016						
NO	NO PENJUALAN	NO AKUN	NAMA AKUN	DEBET	KREDIT	KETERANGAN
1	TP0001	411	Penjualan	0	275,000	
1	TP0001	111	Kas	275,000	0	
3	TP0002	411	Penjualan	0	10,010,000	
3	TP0002	111	Kas	10,010,000	0	
5	TP0003	411	Penjualan	0	154,000	
5	TP0003	111	Kas	154,000	0	
7	TP0004	411	Penjualan	0	192,500	
7	TP0004	111	Kas	192,500	0	

Gambar 4.22. Tampilan Jurnal Penjualan

4.1.15. Tampilan Jurnal Pemesanan

Laporan ini dapat digunakan oleh pihak perusahaan untuk mengetahui jurnal pemesanan dari hasil pemesanan barang

 JURNAL PEMESANAN PER PERIODE 30/08/2016						
NO	NO PEMESANAN	NO AKUN	NAMA AKUN	DEBET	KREDIT	KETERANGAN
19	TS0010	211	Pembayaran di muka	0	5,500,000	Pembayaran DP
19	TS0010	111	Kas	5,500,000	0	Pembayaran DP
21	TS0011	211	Pembayaran di muka	0	5,500,000	Pembayaran DP
21	TS0011	111	Kas	5,500,000	0	Pembayaran DP

Gambar 4.23. Tampilan Jurnal pemesanan

4.1.16. Tampilan Jurnal Penyesuaian

Laporan ini dapat digunakan oleh pihak perusahaan untuk mengetahui jurnal penjualan dari hasil penjualan

 JURNAL PEMESANAN PER PERIODE 29/08/2016						
NO	NO PEMBAYARAN	NO AKUN	NAMA AKUN	DEBET	KREDIT	KETERANGAN
19	TB0001	211	Pembayaran di muka	0	24,000,000	Lunas
19	TB0001	111	Kas	24,000,000	0	Lunas

Gambar 4.24. Tampilan Jurnal Penyesuaian

4.2. Evaluasi Sistem

Setelah melakukan kebutuhan sistem, pembuatan program, instalasi program dan uji coba *user interface*, selanjutnya adalah melakukan evaluasi terhadap sistem. Evaluasi dilakukan dengan dua cara yaitu evaluasi terhadap *form* pendukung dengan evaluasi dasar serta *form-form* yang sesuai tujuan aplikasi dibuat dengan hasil analisis. Tujuan evaluasi ini adalah untuk mengetahui apakah fungsi pada aplikasi yang telah dibuat ini sudah berjalan dengan baik dan sesuai

dengan tujuan atau *output* yang diharapkan. Adapun evaluasi uji coba yang dilakukan adalah sebagai berikut:

4.2.1. Uji coba *form master member*

Form master member digunakan untuk menyimpan data pelanggan yang ingin mendaftar menjadi anggota (*member*). Di *form master member* terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data pelanggan dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input data member*:



The image shows a screenshot of a web form for member registration. The form is overlaid on a background featuring a large, stylized red and white geometric logo on the left and a watermark for 'INSTITUT BISNIS & INFORMATIKA Stikom SURABAYA' on the right. The form fields are as follows:

ID MEMBER :	<input type="text" value="M0002"/>
DISKON MEMBER :	<input type="text" value="20"/> %
NAMA :	<input type="text" value="Susan"/>
ALAMAT :	<input type="text" value="Dukuh Kupang"/>
JENIS KELAMIN :	<input type="text" value="Perempuan"/>
NO. TELP :	<input type="text" value="089762200981"/>

Gambar 4.25. Textbox untuk *input* data

Pada gambar 4.25., ID member dan diskon akan otomatis terisi sesuai banyaknya member yang ada. Admin akan mengisi kolom nama, alamat, no telpon. Kemudian akan memilih jenis kelamin di *combobox*, dan memberikan diskon kepada member. Tabel 4.1. menunjukkan pengujian terhadap *textbox*.

Tabel 4.1 Pengujian *Textbox form member*

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Datagridview*

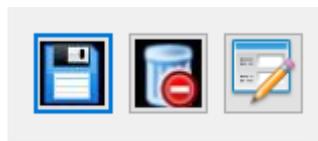
TAMPILAN

	ID_MEMBER	NAMA_MEMBER	ALAMAT_MEMBER
▶	M0001	Agus Prima	Wiyung Pratama 10
	M0002	Susiani Wira	Citraland
	M0003	Ruth Handayani	Kedurus
*			

Gambar 4.26. *Datagridview form master member*

Datagridview menunjukkan kepada admin data *member* yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data *member* seperti ID *member*, nama, alamat, no telpon, jenis kelamin, dan diskon untuk *member*.

3. *Button*

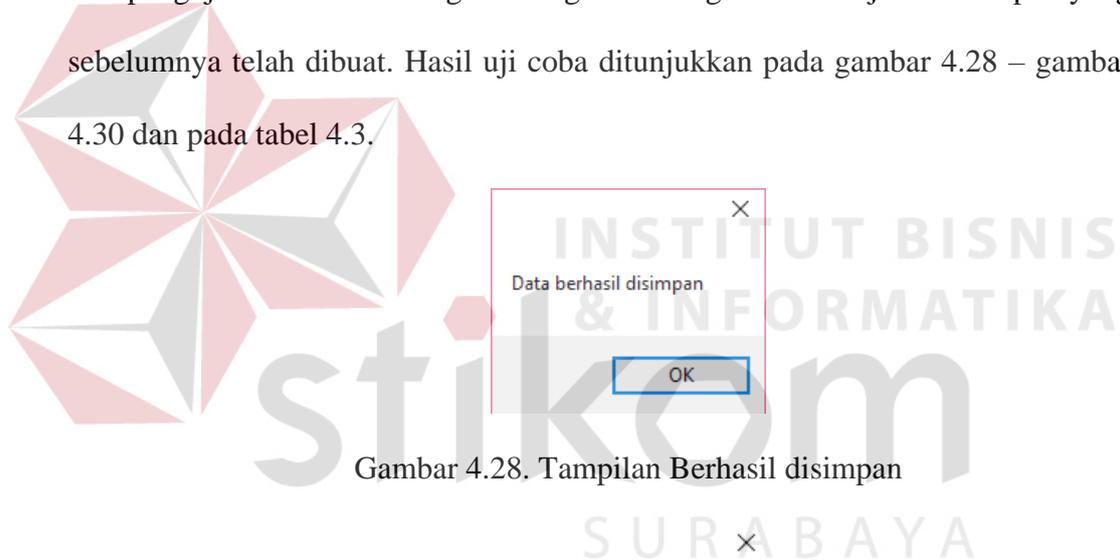
Gambar 4.27. *Button* simpan, hapus dan ubah data *member*.

Button simpan berfungsi untuk menyimpan data *member* ke dalam database, *button* hapus berfungsi untuk menghapus salah satu data *member* yang ada di database, dan *button* ubah berfungsi untuk mengubah salah satu data *member* yang ada di database.

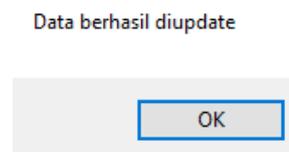
Tabel 4.2. Pengujian *button* pada *form member*

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>database</i> di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah <i>database</i> di <i>datagridview</i>	Dapat dilakukan

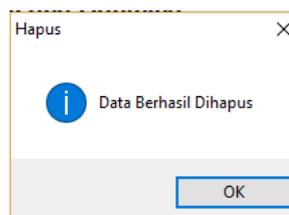
Tabel 4.2. menunjukkan hasil pengujian pada setiap *button* di *form member*. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.28 – gambar 4.30 dan pada tabel 4.3.



Gambar 4.28. Tampilan Berhasil disimpan

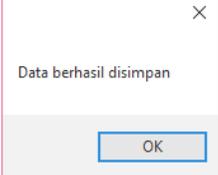
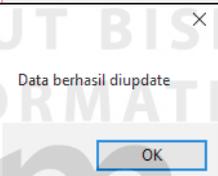
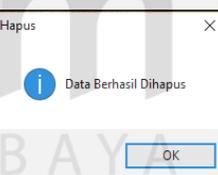


Gambar 4.29. Tampilan Berhasil diupdate



Gambar 4.30. Tampilan Berhasil dihapus

Tabel 4.3. Hasil uji coba *form master member*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master member</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database member</i> dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.28.
2	Mengubah data <i>master member</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database member</i> dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.29.
3	Menghapus data <i>master member</i>	Menekan <i>button</i> hapus data	Menghapus dari <i>database member</i> dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.30.

4.2.2. Uji coba *form master jenis barang*

Form master jenis barang digunakan untuk menyimpan data jenis barang yang ada di toko. Di *form master jenis barang* terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data jenis barang dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data jenis barang:



KODE JENIS : JB0006
NAMA JENIS : Houseware

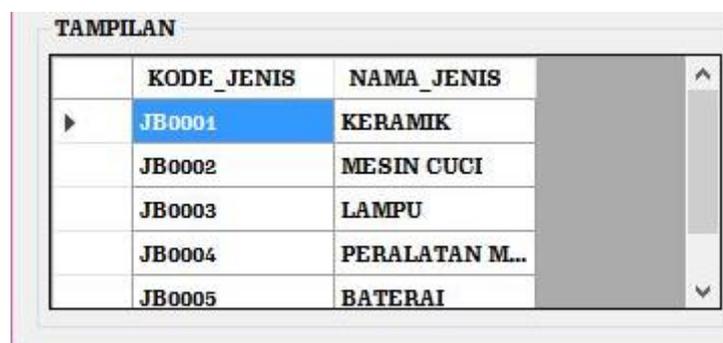
Gambar 4.31. *Textbox* untuk *input* data jenis barang

Pada gambar 4.31., kode jenis barang akan otomatis terisi sesuai banyaknya jenis barang yang ada. Admin akan mengisi kolom nama, jenis barang di *textbox*. Tabel 4.4. menunjukkan pengujian terhadap *textbox*

Tabel 4.4 Pengujian *Textbox form* jenis barang

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Datagridview*



TAMPILAN

	KODE_JENIS	NAMA_JENIS
▶	JB0001	KERAMIK
	JB0002	MESIN CUCI
	JB0003	LAMPU
	JB0004	PERALATAN M...
	JB0005	BATERAI

Gambar 4.32. *Datagridview form master jenis barang*

Datagridview menunjukkan kepada admin data jenis barang yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data jenis barang seperti kode jenis barang, dan nama jenis barang.

3. *Button*



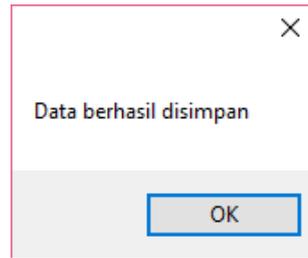
Gambar 4.33. *Button* simpan, hapus dan ubah data jenis barang.

Button simpan berfungsi untuk menyimpan data jenis barang ke dalam database, *button* hapus berfungsi untuk menghapus salah satu data jenis barang yang ada di database, dan *button* ubah berfungsi untuk mengubah salah satu data jenis barang yang ada di database.

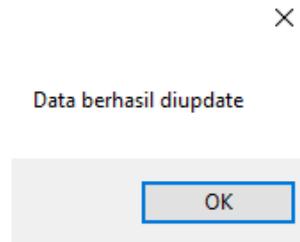
Tabel 4.5. Pengujian *button* pada *form* jenis barang

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus database di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah database di <i>datagridview</i>	Dapat dilakukan

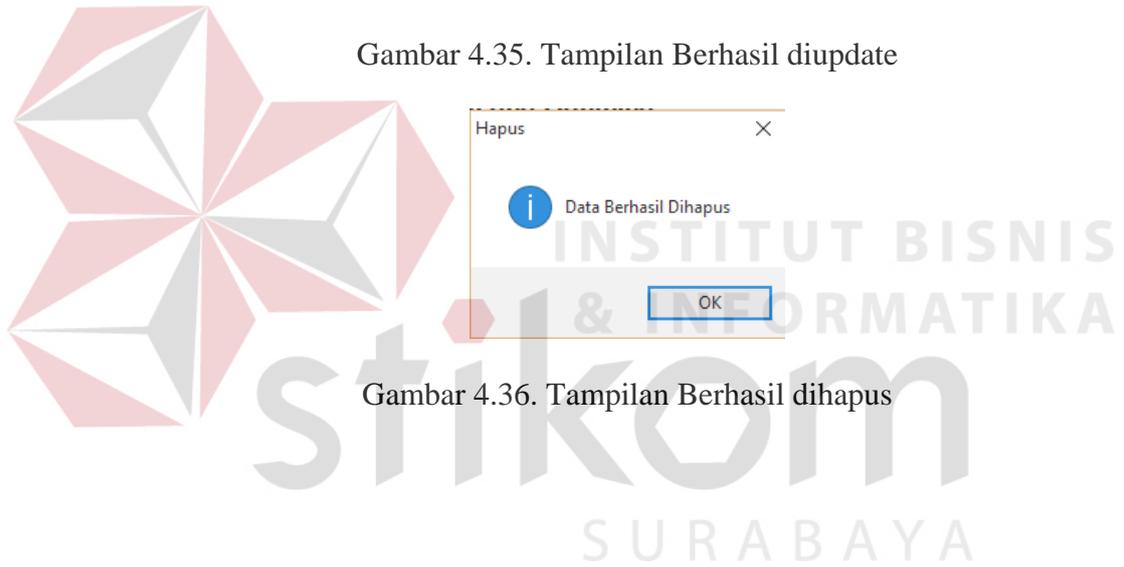
Tabel 4.5. menunjukkan hasil pengujian pada setiap *button* di *form* jenis baang. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.34. – gambar 4.36., dan pada tabel 4.6.



Gambar 4.34. Tampilan Berhasil disimpan

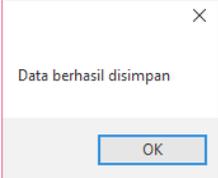
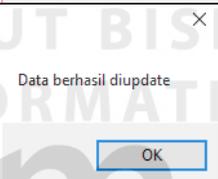
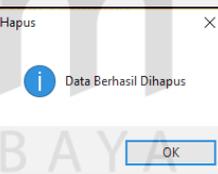


Gambar 4.35. Tampilan Berhasil diupdate



Gambar 4.36. Tampilan Berhasil dihapus

Tabel 4.6. Hasil uji coba *form master jenis barang*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master jenis barang</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> mesin dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.34.
2	Mengubah data <i>master jenis barang</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database</i> jenis barang dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.35.
3	Menghapus data <i>master jenis barang</i>	Menekan <i>button</i> hapus data	Menghapus dari <i>database</i> jenis barang dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.36.

4.2.3. Uji coba *form master promo*

Form master promo digunakan untuk menyimpan data promo yang ada di toko. Di *form master promo* terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data promo dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data promo:

The image shows a screenshot of a web form titled "M. PROMO". The form contains several input fields:

- KODE PROMO :** A text box containing "PB0005".
- NAMA PROMO :** A text box containing "Cuci Gudang".
- JUMLAH DISKON :** A text box containing "70" followed by a percentage sign "%".
- TANGGAL AWAL :** A date picker showing "18/08/2016".
- TANGGAL AKHIR :** A date picker showing "18/08/2016".
- KETERANGAN :** A large text area for notes.

 A watermark for "INSTITUT BISNIS & INFORMATIKA STIKOM SURABAYA" is visible in the background.

Gambar 4.37. Textbox untuk *input* data promo

Pada gambar 4.37., kode promo akan otomatis terisi sesuai banyaknya promo yang ada. Admin akan mengisi kolom jumlah promo, tanggal awal dan akhir promo, dan keterangan promo di *textbox*. Tabel 4.7. menunjukkan pengujian terhadap *textbox*

Tabel 4.7 Pengujian *Textbox form* promo

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Datagridview*

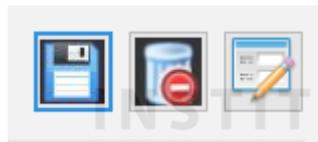


	KODE_DISKON	NAMA_DISKON	JUMLAH_DI ^
▶	PB0001	Promo 17 Agus...	70
	PB0002	Promo Ulang T...	50

Gambar 4.38. *Datagridview form master promo*

Datagridview menunjukkan kepada admin data promo yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data promo seperti kode promo, nama promo, dan jumlah promo.

3. *Button*



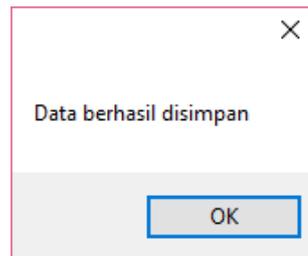
Gambar 4.39. *Button* simpan, hapus dan ubah data promo.

Button simpan berfungsi untuk menyimpan data promo ke dalam *database*, *button* hapus berfungsi untuk menghapus salah satu data promo yang ada di *database*, dan *button* ubah berfungsi untuk mengubah salah satu data promo yang ada di *database*.

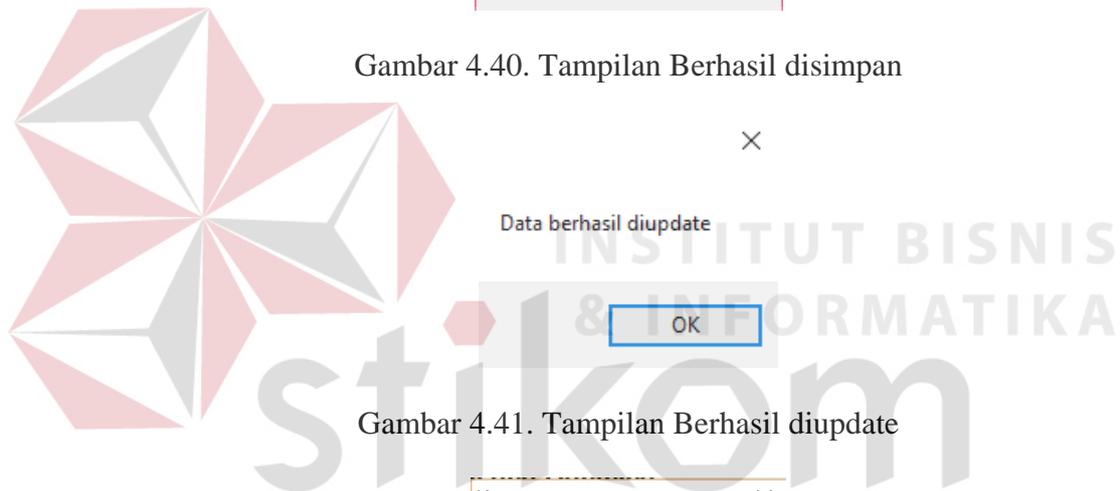
Tabel 4.8. Pengujian *button* pada *form* promo

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>database</i> di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah <i>database</i> di <i>datagridview</i>	Dapat dilakukan

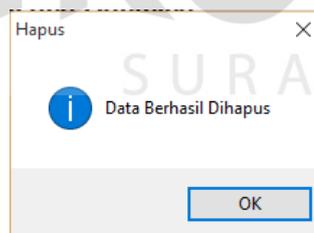
Tabel 4.8. menunjukkan hasil pengujian pada setiap *button* di *form* promo. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.40. – gambar 4.42., dan pada tabel 4.9.



Gambar 4.40. Tampilan Berhasil disimpan

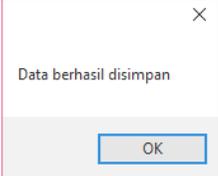
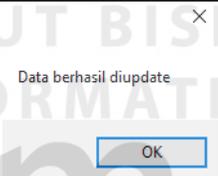
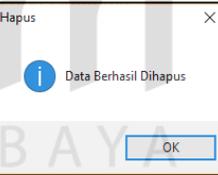


Gambar 4.41. Tampilan Berhasil diupdate



Gambar 4.42. Tampilan Berhasil dihapus

Tabel 4.9. Hasil uji coba *form master promo*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master promo</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> mesin dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.40.
2	Mengubah data <i>master promo</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database</i> diskon dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.41.
3	Menghapus data <i>master promo</i>	Menekan <i>button</i> hapus data	Menghapus dari <i>database</i> promo dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.42.

4.2.4. Uji coba *form master barang*

Form master barang digunakan untuk menyimpan data barang yang ada di toko. Di *form master* barang terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data barang dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data barang:

The image shows a web form for entering item data. It contains the following fields:

- KODE BARANG :** B0006
- KODE JENIS :** JB0003 (dropdown menu showing LAMPU)
- NAMA BARANG :** Philips
- JUMLAH BARANG :** 200
- HARGA BARANG :** 60000

Gambar 4.43. *Textbox* untuk *input* data barang

Pada gambar 4.43., kode barang akan otomatis terisi sesuai banyaknya barang yang ada. Admin akan mengisi kolom nama barang, jumlah barang, dan harga barang di *textbox*. Untuk jenis barang admin akan memilih di *combobox*, dan kode jenis barang akan terisi secara otomatis sesuai dengan jenis barang yang dipilih. Begitu juga dengan diskon barang. Tabel 4.10. menunjukkan pengujian terhadap *textbox*

Tabel 4.10 Pengujian *Textbox form* barang

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan
3.	Mengeluarkan kode barang dan kode diskon secara otomatis sesuai pilihan admin	Dapat dilakukan

2. *Datagridview*

TAMPILAN

	KODE_BARANG	KODE_JENIS	NAMA_BARANG	JU
▶	B0001	JB0001	KERAMIK ASL...	997
	B0002	JB0002	MESIN CUCI P...	299
	B0003	JB0003	LAMPU PHILIP...	1500
	B0004	JB0004	DONGKRAK HL...	500
	B0005	JB0005	BATERAI ABC ...	3000
*				

Gambar 4.44. *Datagridview form master barang*

Datagridview menunjukkan kepada admin data barang yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data barang seperti kode barang, kode jenis, kode diskon, nama jenis barang, nama barang, jumlah barang, harga barang, dan diskon barang.

3. *Button*



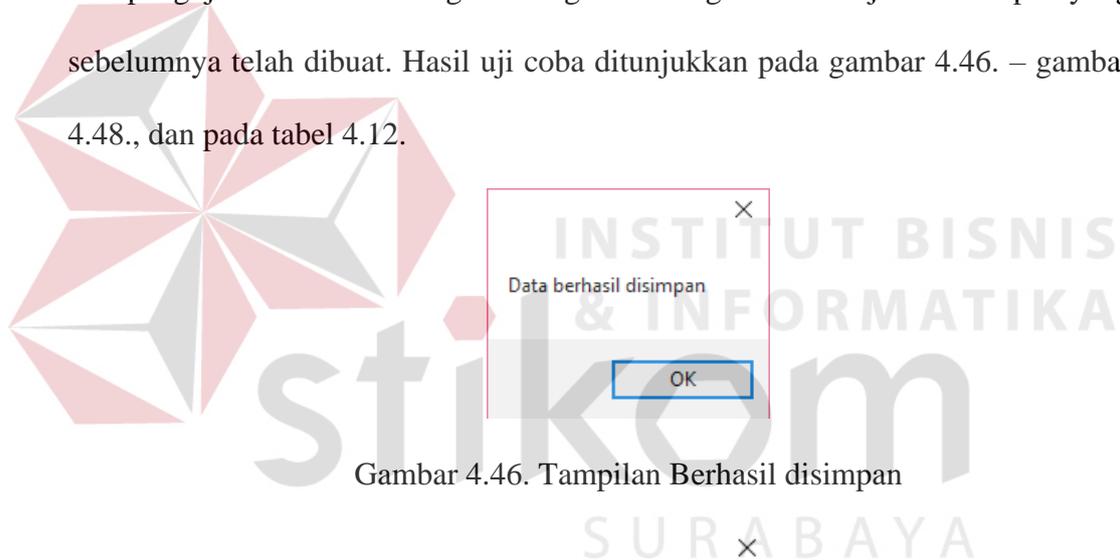
Gambar 4.45. *Button* simpan, hapus dan ubah data barang.

Button simpan berfungsi untuk menyimpan data barang ke dalam *database*, *button* hapus berfungsi untuk menghapus salah satu data barang yang ada di *database*, dan *button* ubah berfungsi untuk mengubah salah satu data barang yang ada di *database*.

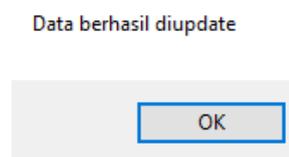
Tabel 4.11. Pengujian *button* pada *form* barang

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>database</i> di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah <i>database</i> di <i>datagridview</i>	Dapat dilakukan

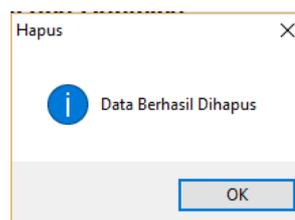
Tabel 4.11. menunjukkan hasil pengujian pada setiap *button* di *form* barang. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.46. – gambar 4.48., dan pada tabel 4.12.



Gambar 4.46. Tampilan Berhasil disimpan

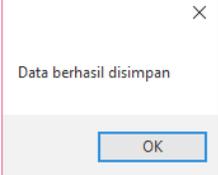
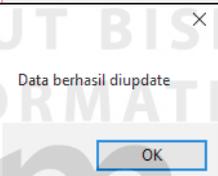
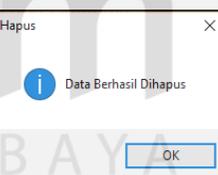


Gambar 4.47. Tampilan Berhasil diupdate



Gambar 4.48. Tampilan Berhasil dihapus

Tabel 4.12. Hasil uji coba *form master barang*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master barang</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> barang dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.46.
2	Mengubah data <i>master barang</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database</i> barang dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.47.
3	Menghapus data <i>master barang</i>	Menekan <i>button</i> hapus data	Manghapus dari <i>database</i> barang dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.48.

4.2.5. Uji coba *form master* pelanggan

Form master barang digunakan untuk menyimpan data barang yang ada di toko. Di *form master* pelanggan terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data pelanggan dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data barang:



The image shows a screenshot of a web form for customer registration. The form contains the following fields:

- ID PELANGGAN :** A text box containing the value "PL0002".
- NAMA :** A text box containing the value "Budl".
- ALAMAT :** A text box containing the value "Wiyung".
- JENIS KELAMIN :** A dropdown menu (combobox) with "Laki-Laki" selected.
- NO. TELP :** A text box containing the value "08172823382".

Gambar 4.49. Textbox untuk *input* data pelanggan

Pada gambar 4.49., id pelanggan akan otomatis terisi sesuai banyaknya pelanggan yang ada. Admin akan mengisi kolom nama pelanggan, alamat pelanggan, dan no telpon barang di *textbox*. Untuk jenis kelamin admin akan memilih di *combobox*. Begitu juga dengan diskon barang. Tabel 4.13. menunjukkan pengujian terhadap *textbox*

Tabel 4.13. Pengujian *Textbox form* pelanggan

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Datagridview*

TAMPILAN

	ID_PELANGGAN	NAMA_PELANGG	ALAMAT	JENIS
▶	PL0001	Susi	Kebraon	Peremp
*				

Gambar 4.50. *Datagridview form master* pelanggan

Datagridview menunjukkan kepada admin data pelanggan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data barang seperti id pelanggan, nama pelanggan, alamat, jenis kelamin, dan no telpon.

3. *Button*



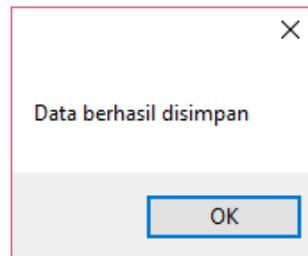
Gambar 4.51. *Button* simpan, hapus dan ubah data pelanggan.

Button simpan berfungsi untuk menyimpan data pelanggan ke dalam database, *button* hapus berfungsi untuk menghapus salah satu data pelanggan yang ada di database, dan *button* ubah berfungsi untuk mengubah salah satu data pelanggan yang ada di database.

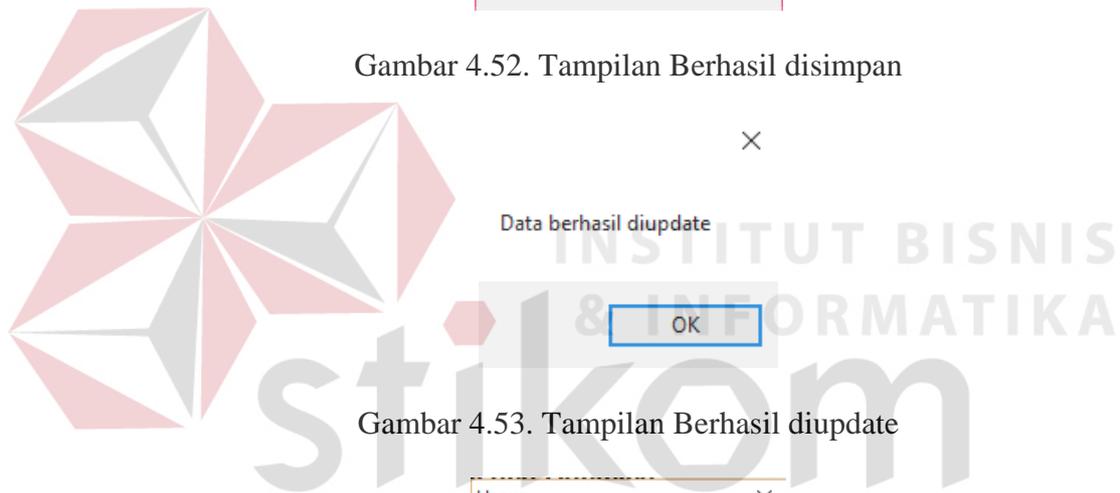
Tabel 4.14. Pengujian *button* pada *form* pelanggan

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus database di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah database di <i>datagridview</i>	Dapat dilakukan

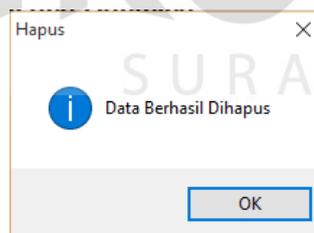
Tabel 4.14. menunjukkan hasil pengujian pada setiap *button* di *form* pelanggan. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.52. – gambar 4.54., dan pada tabel 4.15.



Gambar 4.52. Tampilan Berhasil disimpan

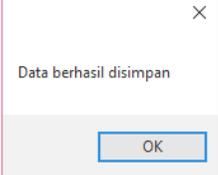
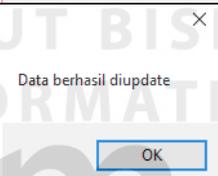
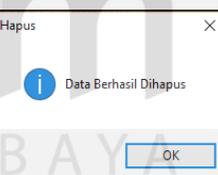


Gambar 4.53. Tampilan Berhasil diupdate



Gambar 4.54. Tampilan Berhasil dihapus

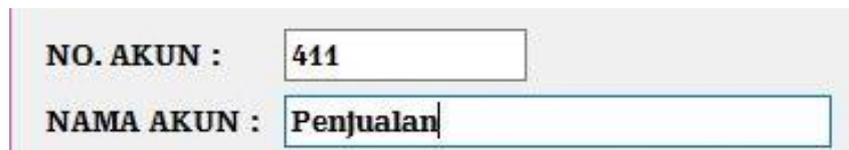
Tabel 4.15. Hasil uji coba *form master pelanggan*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master pelanggan</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> pelanggan dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.52.
2	Mengubah data <i>master pelanggan</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database</i> pelanggan dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.53.
3	Menghapus data <i>master pelanggan</i>	Menekan <i>button</i> hapus data	Manghapus dari <i>database</i> pelanggan dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.54.

4.2.6. Uji coba *form master COA*

Form master COA digunakan untuk menyimpan data COA yang ada di toko. Di *form master COA* terdapat *input* data dan simpan data ke dalam *database*. Pihak admin akan memasukkan data COA dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data barang:



The screenshot shows a form with two input fields. The first field is labeled 'NO. AKUN :' and contains the value '411'. The second field is labeled 'NAMA AKUN :' and contains the value 'Penjualan'.

Gambar 4.55. *Textbox* untuk *input* data COA

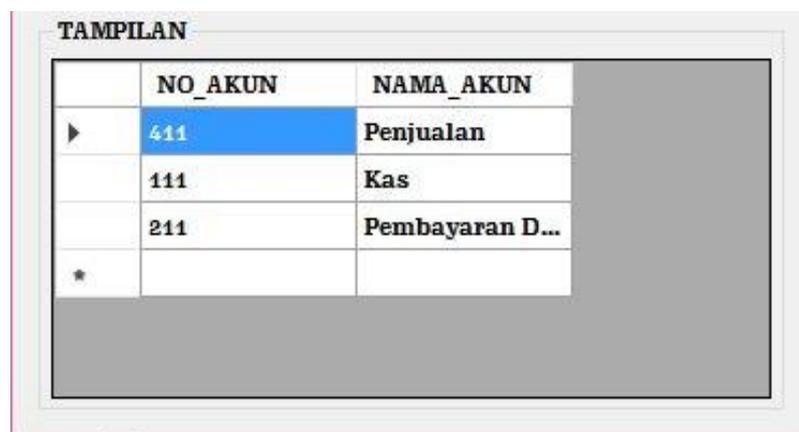
Pada gambar 4.55., Admin akan mengisi kolom no akun, dan nama akun.

Tabel 4.16. menunjukkan pengujian terhadap *textbox*

Tabel 4.16. Pengujian *Textbox form COA*

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Datagridview*



The screenshot shows a DataGridView window titled 'TAMPILAN'. It contains a table with two columns: 'NO_AKUN' and 'NAMA_AKUN'. The first row is highlighted in blue and contains the values '411' and 'Penjualan'. The second row contains '111' and 'Kas'. The third row contains '211' and 'Pembayaran D...'. There is a small asterisk icon in the first column of the fourth row.

	NO_AKUN	NAMA_AKUN
▶	411	Penjualan
	111	Kas
	211	Pembayaran D...
*		

Gambar 4.56. *Datagridview form master COA*

Datagridview menunjukkan kepada admin data COA yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data COA seperti no akun dan nama akun.

3. *Button*



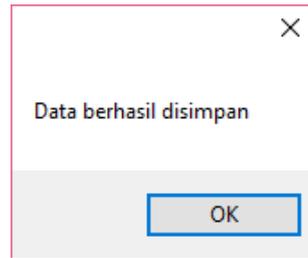
Gambar 4.57. *Button* simpan, hapus dan ubah data COA.

Button simpan berfungsi untuk menyimpan data COA ke dalam *database*, *button* hapus berfungsi untuk menghapus salah satu data COA yang ada di *database*, dan *button* ubah berfungsi untuk mengubah salah satu data COA yang ada di *database*.

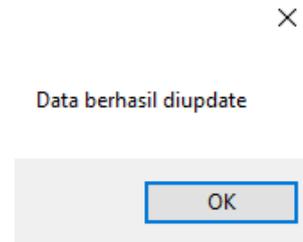
Tabel 4.17. Pengujian *button* pada *form* COA

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>database</i> di <i>datagridview</i>	Dapat dilakukan
3	Melakukan ubah <i>database</i> di <i>datagridview</i>	Dapat dilakukan

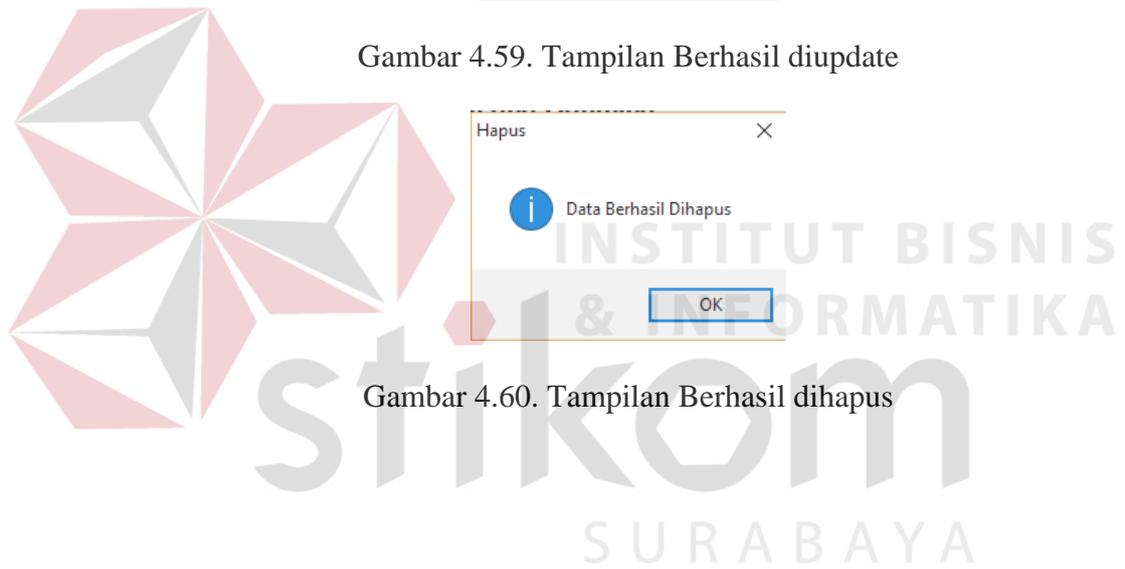
Tabel 4.17. menunjukkan hasil pengujian pada setiap *button* di *form* COA. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.58. – gambar 4.60., dan pada tabel 4.18.



Gambar 4.58. Tampilan Berhasil disimpan

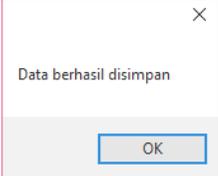
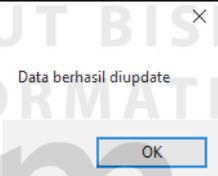
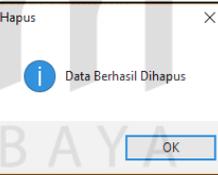


Gambar 4.59. Tampilan Berhasil diupdate



Gambar 4.60. Tampilan Berhasil dihapus

Tabel 4.18. Hasil uji coba *form master COA*

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data <i>master COA</i>	Menekan <i>button</i> simpan data	Tersimpan ke <i>database COA</i> dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.58.
2	Mengubah data <i>master COA</i>	Menekan <i>button</i> ubah data	Mengubah dari <i>database COA</i> dan tampil kotak dialog “data berhasil diupdate”		Sesuai gambar 4.59.
3	Menghapus data <i>master COA</i>	Menekan <i>button</i> hapus data	Manghapus dari <i>database COA</i> dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.60.

4.2.7. Uji coba *form* transaksi penjualan

Form penjualan digunakan untuk menyimpan data penjualan yang ada di toko. Di *form* penjualan terdapat *input* data dan simpan data ke dalam *database*. Pihak kasir akan memasukkan data penjualan dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data penjualan:

The screenshot shows a software window titled "MITRA 10 - TRANSAKSI PENJUALAN". The window contains a form for recording a sale. On the left, there is a logo for "Mitra10" and the word "PENJUALAN". The form fields are as follows:

- TANGGAL: 18/08/2016
- NO. TRANSAKSI: TP0001
- KODE BARANG: B0001
- NAMA BARANG: KERAMIK ASIA TILE 30X30
- HARGA: 62000
- DISKON: 0
- JUMLAH: 30
- TOTAL HARGA: 186000
- SUB TOTAL: (empty)
- PAJAK: 10
- TOTAL JUAL: 0

On the right side of the form, there are input fields for:

- TUNAI: (empty)
- KEMBALI: (empty)
- ID MEMBER: 0

Below the ID MEMBER field is a magnifying glass icon. At the bottom right, there is a section labeled "TAMPILAN" which is currently empty. A large watermark "STIKOM SURABAYA" is overlaid on the form.

Gambar 4.61. Textbox untuk *input* data penjualan

Pada gambar 4.61., no transaksi akan otomatis terisi sesuai banyaknya transaksi yang ada. Kasir akan mengisi kolom kode barang di *textbox*, kemudian nama barang, harga barang, dan diskon barang akan muncul secara otomatis setelah kasir mengisi kode barang. Kemudian kasir akan mengisi jumlah barang yang akan dibeli dan secara otomatis total harga barang akan muncul. Adapun rumus untuk menghitung total harga adalah:

$$total = (jumlah \times harga) - (jumlah \times harga \times diskon) \dots\dots\dots(4.1)$$

Kemudian sub total adalah total harga semua barang yang ingin dibeli oleh pelanggan. Sub total didapat setelah kasir menekan tombol tambah.

Kemudian total penjualan didapat setelah sub total ditambahkan dengan pajak. Rumusnya adalah:

$$total\ penjualan = subtotal + \left(\frac{subtotal \times pajak}{100} \right) \dots\dots\dots(4.2)$$

Untuk kolom tunai, kasir akan mengisi sesuai dengan uang yang dibayarkan pelanggan. Setelah mengisi, maka kolom uang kembali akan otomatis terisi untuk menunjukkan uang kembali. Adapun rumusnya adalah:

$$kembali = tunai - total\ penjualan \dots\dots\dots(4.3)$$

Kolom ID *member* harus diisi terlebih dahulu untuk dapat menampilkan total jual. Jika pelanggan merupakan *member*, maka akan mendapat diskon member sebesar 30% di kolom total jual. Total penjualan jika sudah terdaftar menjadi member didapat dengan rumus:

$$total\ penjualan1 = subtotal + \left(\frac{subtotal \times pajak}{100} \right) \dots\dots\dots(4.4)$$

$$total\ penjualan = total\ penjualan1 - (total\ penjualan1 \times 0.3) \dots\dots\dots(4.5)$$

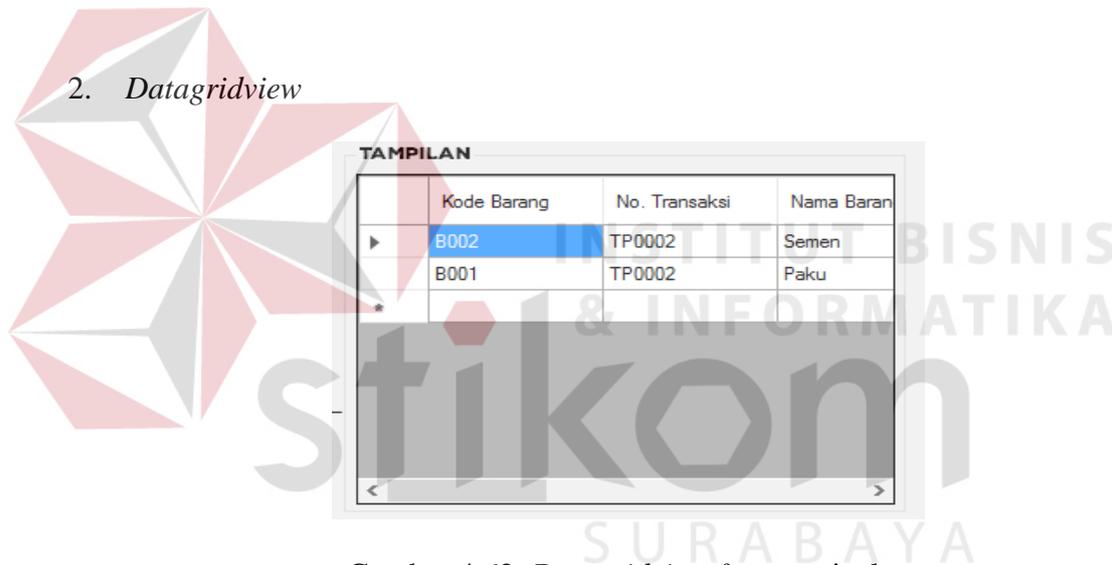
Jika pelanggan bukan *member*, namun berbelanja hingga melebihi atau sama dengan Rp 1.000.000, -, maka akan muncul suatu pertanyaan apakah pelanggan ingin mendaftar menjadi member. Jika memilih ya, maka akan tampil *form master member*.

Tabel 4.19. menunjukkan pengujian terhadap *textbox*

Tabel 4.19. Pengujian *Textbox form* penjualan

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan
3.	Mengeluarkan kode jenis transaksi dan kode diskon secara otomatis sesuai pilihan kasir	Dapat dilakukan
4.	Menghitung secara otomatis total, total penjualan, dan kembali	Dapat dilakukan
5.	Menghitung secara otomatis total penjualan ketika sudah terdaftar menjadi member	Dapat dilakukan
6.	Menampilkan <i>messagebox</i> konfirmasi ingin menjadi member	Dapat dilakukan

2. *Datagridview*

Gambar 4.62. *Datagridview form* penjualan

Datagridview menunjukkan kepada kasir data penjualan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data transaksi seperti kode barang, no transaksi, nama barang, harga per satuan, jumlah barang, diskon barang, harga seluruhnya, dan tanggal transaksi.

3. *Button*

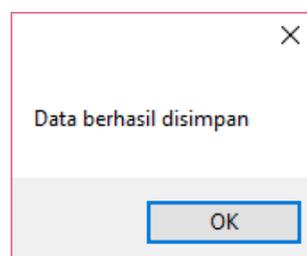
Gambar 4.63. *Button* simpan, hapus, tambah, dan cari pada data penjualan.

Button simpan berfungsi untuk menyimpan data penjualan ke dalam *database*, *button* hapus berfungsi untuk menghapus salah satu data penjualan yang ada di *database*. Kemudian *button* tambah adalah untuk menambah data barang pada penjualan. *Button* cari untuk mencari member yang sudah tersimpan di *database*.

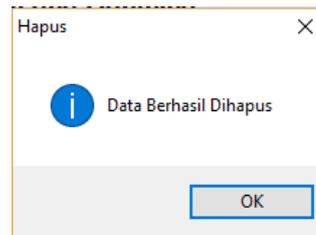
Tabel 4.20. Pengujian *button* pada *form* penjualan

No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>datagridview</i>	Dapat dilakukan
3	Melakukan cetak nota penjualan	Dapat dilakukan
4.	Melakukan penambahan data barang di transaksi penjualan	Dapat dilakukan
5.	Melakukan pencarian <i>member</i> di <i>database</i>	Dapat dilakukan

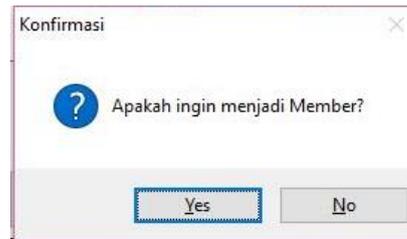
Tabel 4.20. menunjukkan hasil pengujian pada setiap *button* di *form* penjualan. Proses simpan dan tambah tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.64. – gambar 4.66., dan pada tabel 4.21.



Gambar 4.64. Tampilan Berhasil disimpan



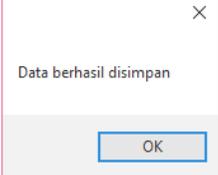
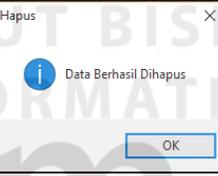
Gambar 4.65. Tampilan Berhasil dihapus



Gambar 4.66. Tampilan Konfirmasi



Tabel 4.21. Hasil uji coba *form* transaksi penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	Kesimpulan
1	Menyimpan data transaksi	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> mesin dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.64.
2	Menghapus data transaksi di <i>datagridview</i>	Menekan <i>button</i> hapus data	Menghapus dari <i>datagridview</i> transaksi dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.65.
3	Menampilkan <i>messagebox</i> konfirmasi ingin menjadi member	Melakukan transaksi penjualan diatas Rp 1.000.000,-	Menampilkan <i>messagebox</i> “apakah anda ingin menjadi <i>member</i> ?”		Sesuai gambar 4.66.

4.2.8. Uji coba *form* transaksi pemesanan

Form pemesanan digunakan untuk menyimpan data pemesanan yang ada di toko. Di *form* pemesanan terdapat *input* data dan simpan data ke dalam *database*. Pihak kasir akan memasukkan data pemesanan dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data pemesanan:

Gambar 4.67. Textbox untuk *input* data pemesanan

Pada gambar 4.67., no transaksi akan otomatis terisi sesuai banyaknya transaksi yang ada. Kasir akan mengisi kolom kode barang di *textbox*, kemudian nama barang, harga barang, dan diskon barang akan muncul secara otomatis setelah kasir mengisi kode barang. Kemudian kasir akan mengisi jumlah barang yang akan dibeli dan secara otomatis total harga barang akan muncul. Adapun rumus untuk menghitung total harga adalah:

$$total = (jumlah \times harga) - (jumlah \times harga \times diskon) \dots\dots\dots(4.1)$$

Kemudian sub total adalah total harga semua barang yang ingin dibeli oleh pelanggan. Sub total didapat setelah kasir menekan tombol tambah.

Kemudian total penjualan didapat setelah sub total ditambahkan dengan pajak. Rumusnya adalah:

$$\text{total penjualan} = \text{subtotal} + \left(\frac{\text{subtotal} \times \text{pajak}}{100} \right) \dots\dots\dots(4.2)$$

Kemudian uang muka adalah jumlah uang yang dibayarkan di muka oleh pelanggan kepada kasir sebesar 20% dari total jual, dan sisa pembayaran adalah jumlah sisa yang harus dibayarkan oleh pelanggan. Uang muka dan sisa pembayaran didapatkan dengan rumus:

$$\text{uang muka} = \text{total penjualan} \times 0.2 \dots\dots\dots(4.3)$$

$$\text{sisa pembayaran} = \text{total penjualan} - \text{uang muka} \dots\dots\dots(4.4)$$

ID pelanggan adalah kolom untuk pelanggan yang tidak terdaftar *member*.

Tabel 4.22. menunjukkan pengujian terhadap *textbox*

Tabel 4.22. Pengujian *Textbox form* pemesanan

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan
3.	Mengeluarkan kode jenis transaksi dan kode diskon secara otomatis sesuai pilihan kasir	Dapat dilakukan
4.	Menghitung secara otomatis total, total penjualan, dan kembali	Dapat dilakukan
5.	Menghitung secara otomatis uang muka dan sisa pembayaran	Dapat dilakukan

2. *Datagridview*

TAMPILAN

	Kode Barang	No. Transaksi	Nama Barang
▶	B0003	TS0002	LAMPU PHILIP..
*			

Gambar 4.68. *Datagridview form* pemesanan

Datagridview menunjukkan kepada kasir data pemesanan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data transaksi seperti kode barang, no transaksi, nama barang, harga per satuan, jumlah barang, diskon barang, harga seluruhnya, dan tanggal pesan.

3. *Button*



Gambar 4.69. *Button* simpan, hapus, tambah data pemesanan, cari pelanggan, dan tambah pelanggan.

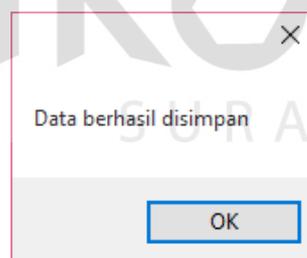
Button simpan berfungsi untuk menyimpan data pemesanan ke dalam database, *button* hapus berfungsi untuk menghapus salah satu data pemesanan yang ada di database. Kemudian *button* tambah adalah untuk menambah data barang pada pemesanan. *Button* cari untuk mencari pelanggan yang sudah tersimpan di database, dan *button* buat pelanggan untuk membuat data pelanggan baru di database.

Tabel 4.23. Pengujian *button* pada *form* pemesanan

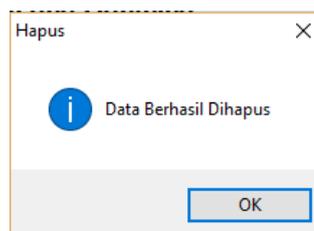
No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan
2	Melakukan hapus <i>datagridview</i>	Dapat dilakukan
3	Melakukan cetak nota pemesanan	Dapat dilakukan
4.	Melakukan penambahan data barang di transaksi penjualan	Dapat dilakukan
5.	Melakukan pencarian <i>member</i> di <i>database</i>	Dapat dilakukan
6.	Melakukan penambahan data pelanggan baru di <i>database</i>	Dapat dilakukan

Tabel 4.15. menunjukkan hasil pengujian pada setiap *button* di *form* pemesanan.

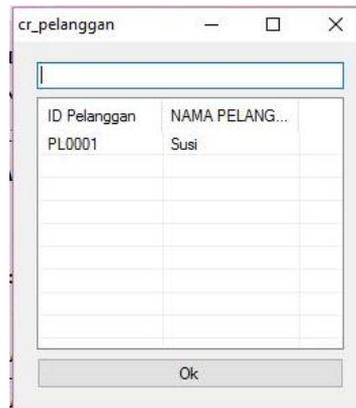
Proses simpan dan tambah tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.70. – gambar 4.73., dan pada tabel 4.24.



Gambar 4.70. Tampilan Berhasil disimpan

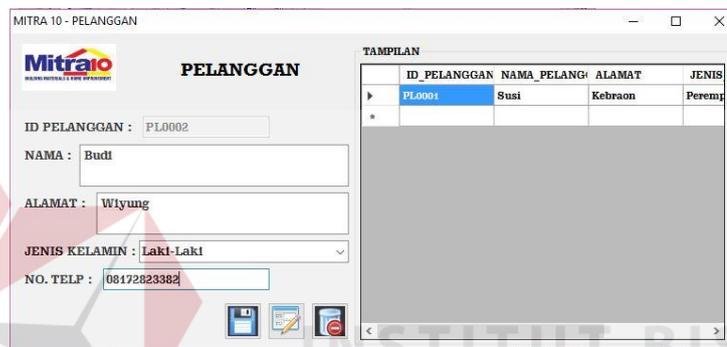


Gambar 4.71. Tampilan Berhasil dihapus



ID Pelanggan	NAMA PELANG...
PL0001	Susi

Gambar 4.72. Tampilan daftar pelanggan



Mitra 10
Mitra 10 - PELANGGAN

PELANGGAN

ID PELANGGAN : PL0002

NAMA : Budi

ALAMAT : Wiyung

JENIS KELAMIN : Laki-Laki

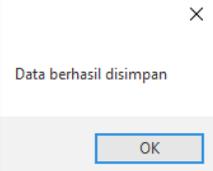
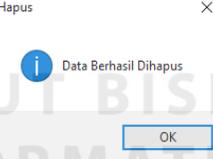
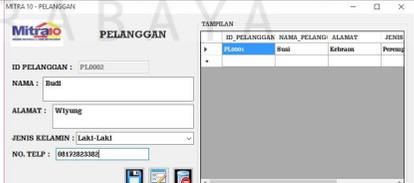
NO. TELP : 08172823382

TAMPILAN

ID_PELANGGAN	NAMA_PELANG	ALAMAT	JENIS
PL0001	Susi	Kebraon	Peremp

Gambar 4.73. Tampilan form pelanggan.

Tabel 4.24. Hasil uji coba *form* transaksi pemesanan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Menyimpan data pemesanan	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> pemesanan dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.70.
2	Menghapus data pemesanan di <i>datagridview</i>	Menekan <i>button</i> hapus data	Menghapus dari <i>datagridview</i> pemesanan dan tampil kotak dialog “data berhasil dihapus”		Sesuai gambar 4.71.
3	Mencari data pelanggan di <i>database</i>	Menekan <i>button</i> cari data	Mencari dari <i>database</i> pelanggan dan tampil kotak dialog “pelanggan”		Sesuai gambar 4.72.
4	Menambah data pelanggan di <i>database</i>	Menekan <i>button</i> buat data pelanggan	Menampilkan <i>form</i> pelanggan		Sesuai gambar 4.73.

4.2.9. Uji coba *form* transaksi pembayaran

Form pembayaran digunakan untuk menyimpan data pembayaran yang ada di toko. Di *form* pembayaran terdapat *input* data dan simpan data ke dalam *database*. Pihak kasir akan memasukkan data pembayaran dan menyimpan data tersebut ke dalam *database*. Di *form* ini terdapat:

1. *Textbox input* data barang:



The screenshot shows a payment form with the following fields and values:

TANGGAL :	18/08/2016
NO. TRANSAKSI :	TB0001
NO. PEMESANAN :	TS0001
TANGGAL PESAN :	17/08/2016
PEMBAYARAN DP :	704000
PEMBAYARAN SISA :	2816000
TOTAL :	3200000
TUNAI :	3000000
KEMBALI :	184000
ID MEMBER / ID PELANGGAN :	PL0001

The form is overlaid on a background featuring a red and white geometric logo and a watermark for 'STIKOM SURABAYA INSTITUT BISNIS & INFORMATIKA'.

Gambar 4.74. Textbox untuk *input* data pembayaran

Pada gambar 4.74., no transaksi akan otomatis terisi sesuai banyaknya transaksi yang ada. Kasir akan mengisi kolom no pemesanan di *textbox*, kemudian tanggal pesan, pembayaran dp, pembayaran sisa, total, dan id member akan muncul secara otomatis setelah kasir mengisi no pemesanan. Kemudian kasir akan mengisi tunai dan secara otomatis kembali akan muncul. Adapun rumus kembali adalah sebagai berikut:

$$\text{kembali} = \text{tunai} - \text{total penjualan} \dots\dots\dots(4.3)$$

Tabel 4.25. Pengujian *Textbox form* pembayaran

No.	Pengujian	Hasil
1.	Melakukan penginputan kosong	Tidak dapat dilakukan
2.	<i>Input</i> data yang sama	Dapat dilakukan

2. *Button*

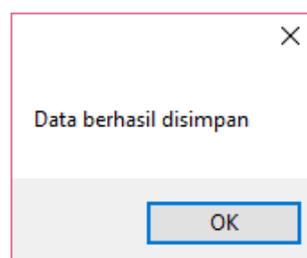
Gambar 4.75. *Button* simpan pembayaran.

Button simpan berfungsi untuk menyimpan data pembayaran ke dalam *database*.

Tabel 4.26. Pengujian *button* pada *form* pembayaran

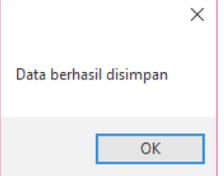
No	Pengujian	Hasil
1	Melakukan penyimpanan jika ada data yang kosong	Tidak dapat dilakukan

Tabel 4.26. menunjukkan hasil pengujian pada setiap *button* di *form* pembayaran. Proses simpan tidak dapat dilakukan jika masih ada data yang belum diisi. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba simpan yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada gambar 4.76. dan pada tabel 4.27.



Gambar 4.76. Tampilan Berhasil disimpan

Tabel 4.27. Hasil uji coba *form* transaksi pembayaran

Test Case ID	Test Case	Test Data	Output yang diharapkan	Actual Output	Kesimpulan
1	Menyimpan data pembayaran	Menekan <i>button</i> simpan data	Tersimpan ke <i>database</i> pembayaran dan tampil kotak dialog “data berhasil disimpan”		Sesuai gambar 4.75.



4.2.10. Uji coba *form* laporan stok barang

Form laporan stok barang digunakan untuk menyimpan data laporan stok barang yang ada di toko. Di *form* ini terdapat:

1. *Datagridview*



	KODE_BARANG	KODE_JENIS	NAMA_BARANG	JUMLAH
▶	B0001	JB0001	KERAMIK ASIA ...	997
	B0002	JB0002	MESIN CUCI PA...	299
	B0003	JB0003	LAMPU PHILIPS...	1500
	B0004	JB0004	DONGKRAK HID...	500
	B0005	JB0005	BATERAI ABC AA	3000
*				

Gambar 4.77. *Datagridview form* laporan stok barang

Datagridview menunjukkan kepada manajer data laporan stok barang yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang data laporan stok barang seperti kode barang, kode jenis, nama barang, jumlah stok barang, dan tanggal laporan stok barang.

2. *Button*



Gambar 4.78. *Button* cetak laporan stok barang.

Button cetak berfungsi untuk mencetak laporan stok barang

Tabel 4.28. Pengujian *button* pada *form* laporan stok barang

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

Tabel 4.28. menunjukkan hasil pengujian pada setiap *button* di *form* laporan stok. Hasil dari pengujian akan dibandingkan dengan rancangan desain uji coba yang sebelumnya telah dibuat. Hasil uji coba ditunjukkan pada tabel 4.29.



Tabel 4.29. Hasil uji coba *form* laporan stok barang

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>																																												
1	Mencetak data laporan stok barang	Menekan <i>button</i> simpan data	Tercetak laporan stok barang	 <table border="1"> <thead> <tr> <th>KODE BARANG</th> <th>NAMA BARANG</th> <th>KANTOR</th> <th>TOTAL STOK</th> </tr> </thead> <tbody> <tr> <td>8001</td> <td>Keramik ubin Dini 30x30</td> <td>42,000</td> <td>2,000</td> </tr> <tr> <td>8002</td> <td>Keramik Kuning 30x30</td> <td>30,000</td> <td>2,000</td> </tr> <tr> <td>8003</td> <td>Keramik Hitam</td> <td>250,000</td> <td>200</td> </tr> <tr> <td>8004</td> <td>Batu-batu</td> <td>2,500,000</td> <td>100</td> </tr> <tr> <td>8005</td> <td>Cat Dapur/Meja</td> <td>40,000</td> <td>500</td> </tr> <tr> <td>8006</td> <td>Tambak Keramik</td> <td>900,000</td> <td>200</td> </tr> <tr> <td>8006</td> <td>Lampas Peralas 100 2X</td> <td>100,000</td> <td>500</td> </tr> <tr> <td>8009</td> <td>Jas Dapur</td> <td>100,000</td> <td>200</td> </tr> <tr> <td>8007</td> <td>Cek Gelas 500</td> <td>40,000</td> <td>200</td> </tr> <tr> <td>8008</td> <td>Peralas 10 50x50</td> <td>200,000</td> <td>200</td> </tr> </tbody> </table>	KODE BARANG	NAMA BARANG	KANTOR	TOTAL STOK	8001	Keramik ubin Dini 30x30	42,000	2,000	8002	Keramik Kuning 30x30	30,000	2,000	8003	Keramik Hitam	250,000	200	8004	Batu-batu	2,500,000	100	8005	Cat Dapur/Meja	40,000	500	8006	Tambak Keramik	900,000	200	8006	Lampas Peralas 100 2X	100,000	500	8009	Jas Dapur	100,000	200	8007	Cek Gelas 500	40,000	200	8008	Peralas 10 50x50	200,000	200	Sesuai gambar 4.19.
KODE BARANG	NAMA BARANG	KANTOR	TOTAL STOK																																														
8001	Keramik ubin Dini 30x30	42,000	2,000																																														
8002	Keramik Kuning 30x30	30,000	2,000																																														
8003	Keramik Hitam	250,000	200																																														
8004	Batu-batu	2,500,000	100																																														
8005	Cat Dapur/Meja	40,000	500																																														
8006	Tambak Keramik	900,000	200																																														
8006	Lampas Peralas 100 2X	100,000	500																																														
8009	Jas Dapur	100,000	200																																														
8007	Cek Gelas 500	40,000	200																																														
8008	Peralas 10 50x50	200,000	200																																														



4.2.11. Uji coba *form* rekapitulasi penjualan

Form rekapitulasi penjualan digunakan untuk melihat dan mencetak data rekapitulasi penjualan yang ada di toko. Pihak manajer akan memilih tanggal penjualan dan menampilkan data tersebut. Di *form* ini terdapat:

1. *Datagridview*

	NO_PENJUALAN	KODE_BARANG	NAMA_BARANG	HARGA_BARANG	JUMLAH_JUAL	TOTAL_JU
▶	TP0002	B0001	Keramik Asia Tal...	65000	200	13000000
	TP0001	B0003	Kompom Rinna	250000	1	250000
	TP0004	B0003	Kompom Rinna	250000	1	250000
	TP0003	B0006	Lampu Philips LE...	100000	2	200000
*						

Gambar 4.79. *Datagridview form* rekapitulasi penjualan

Datagridview menunjukkan kepada manajer data transaksi penjualan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang transaksi no transaksi, kode barang, nama barang, harga barang, jumlah jual, total jual.

2. *Button*



Gambar 4.80. *Button* cetak rekapitulasi penjualan.

Button cetak berfungsi untuk mencetak bukti rekapitulasi.

Tabel 4.30. Pengujian *button* pada *form* rekapitulasi penjualan

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

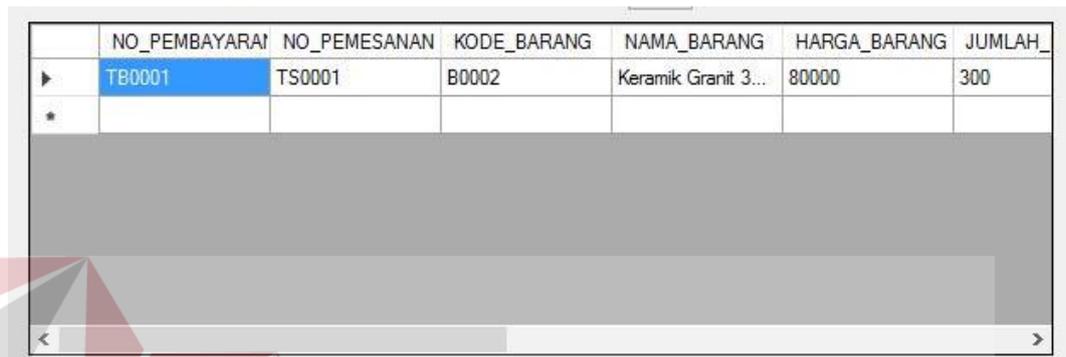
Tabel 4.31. Hasil uji coba *form* rekapitulasi penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Mencetak rekapitulasi penjualan	Menekan <i>button</i> cetak data	Tercetak rekapitulasi penjualan		Sesuai gambar 4.20.
2	Mencari data laporan transaksi di <i>datagridview</i>	Memilih tanggal rekapitulasi	Data transaksi tampil di <i>datagridview</i>		Sesuai gambar 4.79.

4.2.12. Uji coba *form* rekapitulasi pembayaran

Form rekapitulasi pembayaran digunakan untuk melihat dan mencetak data rekapitulasi pembayaran yang ada di toko. Pihak manajer akan memilih tanggal pembayaran dan menampilkan data tersebut. Di *form* ini terdapat:

1. *Datagridview*



	NO_PEMBARAAN	NO_PEMESANAN	KODE_BARANG	NAMA_BARANG	HARGA_BARANG	JUMLAH
▶	TB0001	TS0001	B0002	Keramik Granit 3...	80000	300
*						

Gambar 4.81. *Datagridview form* rekapitulasi pembayaran

Datagridview menunjukkan kepada manajer data transaksi penjualan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang transaksi seperti no pembayaran, no pemesanan, kode barang, nama barang, harga barang, jumlah barang, tanggal transaksi.

2. *Button*



Gambar 4.82. *Button* cetak rekapitulasi pembayaran.

Button cetak berfungsi untuk mencetak bukti rekapitulasi

Tabel 4.32. Pengujian *button* pada *form* rekapitulasi pembayaran

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

Tabel 4.33. Hasil uji coba *form* rekapitulasi pembayaran

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Mencetak rekapitulasi pembayaran	Menekan <i>button</i> cetak data	Tercetak rekapitulasi pembayaran		Sesuai gambar 4.21.
2	Mencari data laporan transaksi di <i>datagridview</i>	Memilih tanggal rekapitulasi	Data transaksi tampil di <i>datagridview</i>		Sesuai gambar 4.81.

4.2.13. Uji coba *form* jurnal penjualan

Form jurnal penjualan digunakan untuk melihat dan mencetak jurnal penjualan. Pihak manajer akan memilih tanggal penjualan dan menampilkan data tersebut. Di *form* ini terdapat:

1. *Datagridview*

	NO	TANGGAL_JU	NO_	NO_PENJU	NAMA_AKU	DEBET_P	KREDIT_P	KET_JURNAL_PENJUALAN
▶	1	29/08/2016	411	TP0001	Penjualan	0	275000	
	1	29/08/2016	111	TP0001	Kas	275000	0	
	3	29/08/2016	411	TP0002	Penjualan	0	10010000	
	3	29/08/2016	111	TP0002	Kas	10010000	0	
	5	29/08/2016	411	TP0003	Penjualan	0	154000	
	5	29/08/2016	111	TP0003	Kas	154000	0	
	7	29/08/2016	411	TP0004	Penjualan	0	192500	
	7	29/08/2016	111	TP0004	Kas	192500	0	

Gambar 4.83. *Datagridview form* jurnal penjualan

Datagridview menunjukkan kepada manajer data transaksi penjualan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang transaksi seperti tanggal jurnal, no akun, no penjualan, nama akun, debit, kredit, dan keterangan.

2. *Button*



Gambar 4.84. *Button* cetak jurnal penjualan.

Button cetak berfungsi untuk mencetak jurnal penjualan.

Tabel 4.34. Pengujian *button* pada *form* jurnal penjualan

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

Tabel 4.35. Hasil uji coba *form* jurnal penjualan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Mencetak jurnal penjualan	Menekan <i>button</i> cetak data	Tercetak jurnal penjualan		Sesuai gambar 4.22.
2	Mencari data jurnal penjualan di <i>datagridview</i>	Memilih tanggal jurnal	Data jurnal tampil di <i>datagridview</i>		Sesuai gambar 4.83.

4.2.14. Uji coba *form* jurnal pemesanan

Form jurnal penjualan digunakan untuk melihat dan mencetak jurnal pemesanan. Pihak manajer akan memilih tanggal pemesanan dan menampilkan data tersebut. Di *form* ini terdapat:

1. *Datagridview*

	NO	TANGGAL_JURNAL	NO_PEMI	NO_	NAM.	DEBET_PEM	KREDIT_PEMESA	KET_JURNAL_PEMES.
▶	1	29/08/2016	TS0001	111	Kas	26400000	0	Pembayaran DP
	3	29/08/2016	TS0002	111	Kas	14300000	0	Pembayaran DP
	5	29/08/2016	TS0003	111	Kas	17600000	0	Pembayaran DP
	7	29/08/2016	TS0004	111	Kas	19800000	0	Pembayaran DP
	9	29/08/2016	TS0005	111	Kas	19800000	0	Pembayaran DP
	11	29/08/2016	TS0006	111	Kas	88000000	0	Pembayaran DP
	13	29/08/2016	TS0007	111	Kas	9900000	0	Pembayaran DP
	15	29/08/2016	TS0008	111	Kas	9900000	0	Pembayaran DP

Gambar 4.85. *Datagridview form* jurnal pemesanan

Datagridview menunjukkan kepada manajer data transaksi pemesanan yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang transaksi no pemesanan, tanggal jurnal, no akun, nama akun, debit, dan kredit, dan keterangan jurnal.

2. *Button*



Gambar 4.86. *Button* cetak jurnal pemesanan.

Button cetak berfungsi untuk mencetak jurnal pemesanan.

Tabel 4.36. Pengujian *button* pada *form* jurnal pemesanan

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

Tabel 4.37. Hasil uji coba *form* jurnal pemesanan

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Mencetak jurnal penjualan	Menekan <i>button</i> cetak data	Tercetak jurnal pemesanan		Sesuai gambar 4.23.
2	Mencari data jurnal pemesanan di <i>datagridview</i>	Mencari tanggal jurnal	Data jurnal pemesanan tampil di <i>datagridview</i>		Sesuai gambar 4.85.

stikom
SURABAYA

4.2.15. Uji coba *form* jurnal penyesuaian

Form jurnal penjualan digunakan untuk melihat dan mencetak jurnal penyesuaian. Pihak manajer akan memilih tanggal penyesuaian dan menampilkan data tersebut. Di *form* ini terdapat:

1. *Datagridview*

	NO_JURNAL_PEN	TANGGAL_JURNA	NO_PEMBAYARAN	NO_AKUN	NAMA_AKUN	DEBET_P
▶	19	29/08/2016	TB0001	211	Pembayaran di m...	0
	19	29/08/2016	TB0001	111	Kas	24000000
*						

Gambar 4.87. *Datagridview* form jurnal penyesuaian

Datagridview menunjukkan kepada manajer data jurnal penyesuaian yang sudah disimpan di dalam database. *Datagridview* berisi informasi tentang jurnal seperti no jurnal, tanggal jurnal, no pembayaran, no akun, nama akun, debet, dan kredit, dan keterangan jurnal.

2. *Button*



Gambar 4.88. *Button* cetak jurnal penyesuaian.

Button cetak berfungsi untuk mencetak jurnal penyesuaian.

Tabel 4.38. Pengujian *button* pada *form* jurnal penyesuaian

No	Pengujian	Hasil
1.	Melakukan cetak laporan	Dapat dilakukan

Tabel 4.39. Hasil uji coba *form* jurnal penyesuaian

<i>Test Case ID</i>	<i>Test Case</i>	<i>Test Data</i>	<i>Output yang diharapkan</i>	<i>Actual Output</i>	<i>Kesimpulan</i>
1	Mencetak jurnal penyesuaian	Menekan <i>button</i> cetak data	Tercetak jurnal penyesuaian		Sesuai gambar 4.24.
2	Mencari data jurnal penyesuaian di <i>datagridview</i>	Mencari tanggal jurnal	Data jurnal pemesanan tampil di <i>datagridview</i>		Sesuai gambar 4.87.

