ANALISIS DAN PENGEMBANGAN SISTEM INFORMASI PEMBELIAN DAN PENJUALAN MANOKO GROSIR

Laporan Kerja Praktek

Diajukan untuk memenuhi syarat matakuliah kerja praktek
Program strata satu Jurusan Manajemen Informatika

Oleh :
Tan Ory Jaka Perdana NIM. 10504464

JURUSAN MANAJEMEN INFORMATIKA
FAKULTAS TEKNIK DAN ILMU KOMPUTER
UNIVERSITAS KOMPUTER INDONESIA
BANDUNG
2009

LEMBAR PENGESAHAN

ANALISIS DAN PENGEMBANGAN SISTEM INFORMASI PEMBELIAN DAN PENJUALAN MANOKO GROSIR

Laporan Kerja Praktek

Diajukan untuk memenuhi syarat matakuliah kerja praktek
Program strata satu Jurusan Manajemen Informatika

Oleh :
Tan Ory Jaka Perdana NIM. 10504464

Bandung, ... 2009

Pembimbing Jurusan,	Pembimbing Lapangan,
Wartika, S.Kom., MT.	Andri Jamaluddin
NIP. 4127. 70. 26. 002	

Ketua Jurusan Manajemen Informatika

Dadang Munandar, S.E., M.Si.
NIP. 4127. 70. 26. 019

KATA PENGANTAR

Segala puji dan syukur Penyusun panjatkan kepada Tuhan Yang Maha Esa, karena atas Rahmat dan Hidayah-Nya sehingga Penyusun dapat menyelesaikan Laporan Kerja Praktek. Laporan ini ditujukan untuk memenuhi dan melengkapi persyaratan Akademik Jurusan Manajemen Informatika pada Fakultas Teknik dan Ilmu Komputer di Universitas Komputer Indonesia dengan judul

"ANALISIS DAN PENGEMBANGAN SISTEM INFORMASI PEMBELIAN DAN PENJUALAN MANOKO GROSIR"

Penyusun menyadari banyak kesulitan dan hambatan yang dihadapi dalam proses penyusunan laporan Kerja Praktek ini, namun berkat kerja keras, do'a, dukungan dan bantuan dari berbagai pihak akhirnya laporan ini dapat diselesaikan. Oleh karena itu dengan segala kerendahan hati penyusun menyadari bahwa penulisan dan penyusunan laporan ini masih jauh dari kesempurnaan baik ditinjau dari kalimat maupun materinya. Hal ini mengingat keterbatasan kemampuan dan pengetahuan penyusun. Walaupun demikian usaha yang maksimal telah dilakukan dalam penyelesaian laporan ini agar dapat memenuhi harapan. Semoga laporan ini dapat bermanfaat, khususnya bagi penyusun, umumnya bagi pembaca sehingga kritik dan saran demi perbaikan sangat diharapkan.

Pada kesempatan ini, Penyusun ingin mengucapkan terima kasih yang sebesar-besarnya kepada yang terhormat :

- ✓ Bapak *Prof. Dr. Ir. H. Ukun Sastraprawira, M.Sc*, selaku Dekan Fakultas Teknik.
- ✓ Bapak *Dadang Munandar, S.E., M.Si.*, selaku Ketua Jurusan Manajemen Informatika.
- ✓ Ibu *Mayalisa Bahar, S.Kom, MT*, selaku dosen pembimbing yang telah meluangkan banyak waktu, tenaga dan pikirannya untuk membimbing dan memberikan saran dalam menyelesaikan Kerja Praktek.
- ✓ Kang *Andri Jamaludin* selaku pembimbing dari Manoko Grosir yang telah memberikan kesempatan kepada Penyusun untuk melakukan pengamatan sampai diselesaikannya Laporan Kerja Praktek.
- ✓ Ayah dan Ibu yang telah memberikan do'a, dukungan baik moril maupun materil yang tiada henti.
- ✓ Buat semua keluargaku yang selalu mendukungku.
- ✓ Semua sahabatku dari Karang Taruna RW 03 Kp. Manoko Ds. Cikahuripan Kec. Lembang yang selalu ada di saat kubutuhkan.
- ✓ Dan semua pihak yang tidak dapat Penyusun sebutkan satu persatu yang telah membantu Penyusun dalam pelaksanaan dan pembuatan Laporan Kerja Praktek.

Akhirnya penyusun hanya dapat menyampaikan do’a semoga kebaikan yang telah diberikan dari semua pihak mendapat balasan yang lebih besar dari Allah SWT, Amin.

Bandung, September 2009

Penyusun

DAFTAR ISI

LEMBAR PENGESAHAN i

KATA PENGANTAR ii

DAFTAR ISI.....v

DAFTAR TABEL.....viii

DAFTAR GAMBAR ix

DAFTAR LAMPIRAN.....xi

BAB I PENDAHULUAN.....1

1.1. Latar Belakang.....1

1.2. Identifikasi Masalah dan Rumusan Masalah 2

1.3. Maksud dan Tujuan.....3

1.4. Metode Pengembangan Sistem.....3

1.5. Batasan Masalah 5

1.6. Lokasi dan Jadwal Kerja Praktek.....6

BAB II LANDASAN TEORI.....7

2.1. Pengertian Sistem.....7

2.1.1. Elemen Sistem.....7

2.1.2. Karakteristik Sistem 9

2.1.3. Klasifikasi Sistem.....11

2.2. Pengertian Informasi.....12

2.3. Pengertian Sistem Informasi 12
2.4. Metode Analisis dan Perancangan Terstruktur 13
2.4.1. Flow Map.....13
2.4.2. Diagram Kontek 13
2.4.3. Data Flow Diagram 14
2.5. Tinjauan Perangkat Lunak Borland Delphi 7.0 15
2.5.1. Tampilan Dasar Borland Delphi 7.0.....15
2.5.2. Komponen EkszBoxVCL 18
2.5.3. Komponen PicShow 19

BAB III PROFIL PERUSAHAAN.....21
3.1. Tinjauan Umum Perusahaan 21
3.2. Struktur Organisasi 22
3.3. Deskripsi Kerja 22
3.4. Analisis Sistem yang Berjalan 26

BAB IV ANALISIS KERJA PRAKTEK 27
4.1 Analisis Sistem.....27
4.1.1. Analisis Dokumen 27
4.1.2. Analisis Prosedur yang Sedang Berjalan.....29
4.1.2.1. Flow Map yang Sedang Berjalan.....29
4.1.2.2. Diagram Kontek yang Sedang Berjalan 30
4.1.2.3. DFD yang Sedang Berjalan 30

4.1.3. Evaluasi Sistem yang Sedang Berjalan 31
4.2. Usulan Perancangan Sistem 32
4.2.1. Tujuan Perancangan Sistem 32
4.2.2. Perancangan Prosedur yang Diusulkan 32
4.2.2.1. Flow Map yang Diusulkan.....32
4.2.2.2. Diagram Kontek yang Diusulkan 33
4.2.2.3. DFD yang Diusulkan 34
4.2.2.4. Kamus Data 34
4.2.2.5. Tabel Relasi 39
4.2.2.6. Perancangan Kode 40
4.2.2.7. Struktur Menu Program 42
4.2.2.8. Kebutuhan Sistem.....43
4.2.3. Program Aplikasi yang Diusulkan.....43
4.2.4. Evaluasi Sistem yang Diusulkan 58

BAB V KESIMPULAN DAN SARAN.....59
5.1. Kesimpulan 59
5.2. Saran 59

DAFTAR PUSTAKA xii
LAMPIRAN - LAMPIRAN.....xiii

DAFTAR TABEL

Tabel 1.1. Jadwal Kerja Praktek 6

Tabel 2.1. Komponen – Komponen EkszBoxVCL 18

Tabel 2.2. Komponen – Komponen PicShow 20

Tabel 4.1. Tabel penjualan 35

Tabel 4.2 Tabel dtil_jual 36

Tabel 4.3. Tabel hak_akses 36

Tabel 4.4. Tabel barang 37

Tabel 4.5. Tabel pembelian 37

Tabel 4.6. Tabel temp 38

DAFTAR GAMBAR

Gambar 1.1. Metode Pengembangan Waterfall 4

Gambar 2.1. Tampilan Borland Delphi 7.0 16

Gambar 3.1. Struktur Organisasi Manoko Grosir 22

Gambar 4.1. Rangka Laporan Penjualan 28

Gambar 4.2. Rangka Laporan Pembelian 28

Gambar 4.3. Rangka Struk Penjualan 29

Gambar 4.4. Flow Map Sistem yang Sedang Berjalan 29

Gambar 4.5. Diagram Kontek Sistem yang Sedang Berjalan 30

Gambar 4.6. DFD Sistem yang Sedang Berjalan 31

Gambar 4.7. Flow Map yang Diusulkan 33

Gambar 4.8. Diagram Kontek yang Diusulkan 33

Gambar 4.9. DFD yang Diusulkan 34

Gambar 4.10. Tabel Relasi 40

Gambar 4.11. Struktur Menu Program 42

Gambar 4.12. Form Koneksi 44

Gambar 4.13. Form Splash 45

Gambar 4.14. Form Utama (Tab Login) 45

Gambar 4.15. Form Login 46

Gambar 4.16. Form Utama (Tab Kasir) 47

Gambar 4.17. Form Penjualan 48

Gambar 4.18. Struk Jual 48

Gambar 4.19. Form Daftar Penjualan 49
Gambar 4.20. Form Detail Penjualan 50
Gambar 4.21. Form Utama (Tab Gudang) 50
Gambar 4.22. Form Daftar Barang 51
Gambar 4.23. Form Tambah Barang 52
Gambar 4.24. Form Detail Stok 53
Gambar 4.25. Form Pembelian 53
Gambar 4.26. Form Utama (Tab Admin) 54
Gambar 4.27. Form Laporan 55
Gambar 4.28. Laporan Pembelian 55
Gambar 4.29. Laporan Penjualan 56
Gambar 4.30. Laporan Stok Barang 56
Gambar 4.31. Form Hak Akses 57
Gambar 4.32. Form Tambah Hak Akses 57
Gambar 4.33. Form Ganti Password 58

DAFTAR LAMPIRAN

Surat Pernyataan (bukti sudah melaksanakan kerja praktek).....xiii

DOKUMEN – DOKUMEN PERUSAHAAN.....xiv

- ❖ Laporan Pembelian Agustus 2009.....xiv
- ❖ Laporan Penjualan Agustus 2009.....xv
- ❖ Struk Penjualan Manoko Grosir.....xvi

Listing Program.....xvii

FORMULIR DAFTAR HADIR PRAKTEK KERJA LAPANGAN (1 lembar)

FORM PENILAIAN PRAKTEK KERJA LAPANGAN (1 lembar)

BERITA ACARA BIMBINGAN KERJA PRAKTEK (1 lembar)

CD PRAKTEK KERJA LAPANGAN (2 keping)

BAB I

PENDAHULUAN

1.1. Latar Belakang Kerja Praktek

Dalam perkembangan ilmu pengetahuan dan teknologi dewasa ini, kebutuhan akan informasi menjadi hal yang sangat penting untuk meningkatkan peradaban yang lebih maju bagi seluruh umat manusia di setiap bidang pekerjaannya masing – masing. Komputer merupakan salah satu cindera mata penting hasil dari perkembangan teknologi tersebut dan diciptakan khusus untuk mempermudah umat manusia dalam menyelesaikan setiap pekerjaannya.

Belakangan ini area penggunaan komputer telah meluas yang sebelumnya komputer termasuk barang mewah yang hanya dikenal pada daerah perkotaan, kini telah sudah menjadi barang yang merakyat yang bisa kita temukan pada daerah pedesaan. Hal ini membutikan bahwa hampir semua orang telah mampu mengoperasikan mesin ini.

Manoko adalah kampung yang terletak dalam wilayah Desa Cikahuripan Kecamatan Lembang Kabupaten Bandung Barat. Dengan meluasnya area penggunaan komputer ke dalam wilayah Desa Cikahuripan, berbagai bidang usaha pun mulai menggunakan mesin ini sebagai media atau alat pencari, penyimpan atau penyedia informasi. Terbukti dengan mulai munculnya beberapa warnet dan percetakan digital serta toko atau grosir yang menggunakan komputer sebagai alat pendataan barangnya.

Oleh karena itu penyusun mencoba untuk melakukan analisis salah satu bidang usaha tersebut, melalui pengembangan sistem informasi pada “Manoko

Grosir” yang terdapat di Kampung Manoko. Sehingga penyusun mengambil judul “Analisis dan Pengembangan Sistem Informasi Pembelian dan Penjualan Manoko Grosir”.

1.2. Identifikasi dan Rumusan Masalah

a. Identifikasi Masalah

Selama melakukan Kerja Praktek pada Manoko Grosir, penyusun melihat pendataan barang yang masuk (dibeli) dan barang yang keluar (dijual) secara langsung. Komputer pun berperan dalam pendataan tersebut dengan salah satu program aplikasinya Microsoft Excell. Namun selama pendataan tersebut, penyusun melihat terjadinya masalah yang teridentifikasi, yaitu :

- Masih terdapat kesalahan pendataan antara pihak gudang dan kasir
- Timbulnya data yang bersifat ganda (Redudansi) bahkan data yang hilang serta keakuratan data yang masih kurang

b. Rumusan Masalah

Dari identifikasi masalah di atas, penyusun dapat merumuskan masalah tersebut menjadi :

- Bagaimana cara merelasikan data penjualan dan pembelian tersebut
- Bagaimana agar setiap bagian (gudang,kasir) tidak repot dalam melakukan pendataan

1.3. Maksud dan Tujuan

Maksud dilaksanakan Kerja Praktek adalah untuk mengimplementasikan pengetahuan yang didapat di perkuliahan dengan kenyataan yang sesungguhnya di lapangan, sedangkan tujuan dilaksanakannya kerja praktek adalah untuk :

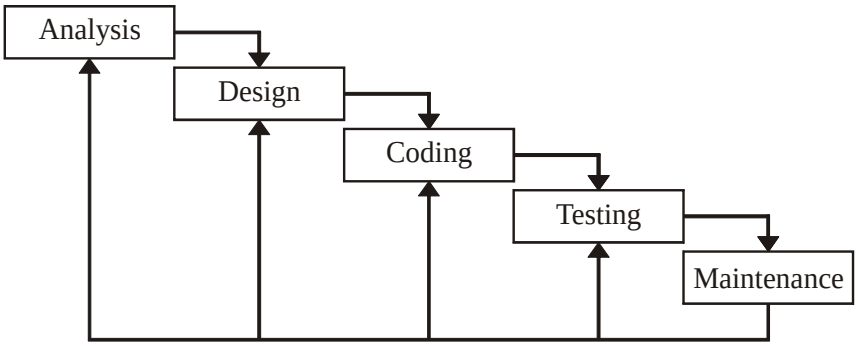
- Mengetahui Sistem Pembelian dan Penjualan yang sedang berjalan pada Manoko Grosir.
- Membuat usulan Sistem Pembelian dan Penjualan pada Manoko Grosir.

1.4. Metode Pengembangan Sistem

Metode pengembangan sistem adalah membuat suatu sistem yang baru untuk menggantikan atau memperbaharui sistem yang lama. Dalam kerja praktek ini, penyusun menggunakan model waterfall.

Nama model waterfall ini sebenarnya adalah “Linear Sequential Model”. Model ini sering disebut dengan “classic life cycle”. Model ini adalah model yang muncul pertama kali yaitu sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai didalam Software Engineering (SE). Model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, design, coding, testing dan maintenance. Disebut dengan waterfall karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan. Sebagai contoh tahap desain harus menunggu selesainya tahap sebelumnya yaitu tahap analysis.

Secara umum tahapan pada model waterfall dapat dilihat pada gambar berikut :



Sumber : Roger Presman, 2008

Gambar 1.1
Metode Pengembangan Waterfall

Gambar di atas adalah tahapan umum dari model waterfall. Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam model ini, yaitu :

- **Analysis**. Proses pencarian kebutuhan diintensifkan dan difokuskan pada software. Untuk mengetahui sifat dari program yang akan dibuat, maka para software engineer harus mengerti tentang domain informasi dari software, misalnya fungsi yang dibutuhkan, user interface, dsb.
- **Design**. Proses ini digunakan untuk mengubah kebutuhan-kebutuhan diatas menjadi representasi ke dalam bentuk “blueprint” software sebelum coding dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya.
- **Coding**. Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin, yaitu ke dalam bahasa pemrograman melalui proses coding. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh programmer.

- **Testing**. Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar software bebas dari error, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.
- **Maintenance**. Pemeliharaan suatu software diperlukan, termasuk di dalamnya adalah pengembangan, karena software yang dibuat tidak selamanya hanya seperti itu. Ketika dijalankan mungkin saja masih ada error kecil yang tidak ditemukan sebelumnya, atau ada penambahan fitur-fitur yang belum ada pada software tersebut. Pengembangan diperlukan ketika adanya perubahan dari eksternal perusahaan seperti ketika ada pergantian sistem operasi, atau perangkat lainnya.

1.5. Batasan Masalah

Agar memudahkan pembahasan selanjutnya, sehingga tidak menyimpang dari permasalahan yang akan dibahas dalam laporan ini dilakukan pembatasan masalah yaitu :

- Sistem Informasi yang dibuat hanya untuk melayani penyimpanan data penjualan, pembelian dan stock barang.
- Untuk pembelian, pada Manoko Grosir tidak menggunakan sistem pemesanan begitu juga dengan penjualannya.
- Informasi yang dihasilkan dari sistem ini berupa laporan transaksi penjualan dan laporan pembelian dengan periode tertentu. Dan untuk laporan daftar stok barang hanya sampai stok terakhir pada tanggal yang dipilih.

1.6. Lokasi dan Jadwal Kerja Praktek

Penelitian mengenai Sistem Informasi ini dilakukan pada Manoko Grosir yang beralamat di Kampung Manoko RT 01/03 Desa Cikahuripan Kecamatan Lembang Kabupaten Bandung Barat.

Waktu yang diperlukan untuk melakukan penelitian ini sekitar 6 minggu dengan rincian sebagai berikut :

Tabel 1.1
Jadwal Kerja Praktek

No.	Aktivitas	Minggu ke					
		I	II	III	IV	V	VI
1.	Observasi dan analisa di lokasi	X					
2.	Pengumpulan data	X	X				
3.	Pembuatan Program			X	X	X	X

BAB II

LANDASAN TEORI

2.1. Pengertian Sistem

Menurut Jerry FithGerald**,** sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu. Dalam artian umum sistem bisa diartikan sebagai kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.1.1. Elemen Sistem

Ada beberapa elemen yang membentuk sebuah sistem, yaitu : tujuan, masukan, proses, keluaran, batas, mekanisme pengendalian dan umpan balik serta lingkungan. Berikut penjelasan mengenai elemen-elemen yang membentuk sebuah sistem :

- **Tujuan**. Setiap sistem memiliki tujuan (Goal), entah hanya satu atau mungkin banyak. Tujuan inilah yang menjadi pemotivasi yang mengarahkan sistem. Tanpa tujuan, sistem menjadi tak terarah dan tak terkendali. Tentu saja, tujuan antara satu sistem dengan sistem yang lain berbeda.
- **Masukan**. Masukan (*input*) sistem adalah segala sesuatu yang masuk ke dalam sistem dan selanjutnya menjadi bahan yang diproses. Masukan dapat berupa hal-hal yang berwujud (tampak secara fisik) maupun yang tidak tampak. Contoh masukan yang

berwujud adalah bahan mentah, sedangkan contoh yang tidak berwujud adalah informasi (misalnya permintaan jasa pelanggan).

- **Proses**. Proses merupakan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna dan lbih bernilai, misalnya berupa informasi dan produk, tetapi juga bisa berupa hal-hal yang tidak berguna, misalnya saja sisa pembuangan atau limbah. Pada pabrik kimia, proses dapat berupa bahan mentah. Pada rumah sakit, proses dapat berupa aktivitas pembedahan pasien.
- **Keluaran**. Keluaran (*output*) merupakan hasil dari pemrosesan. Pada sistem informasi, keluaran bisa berupa suatu informasi, saran, cetakan laporan, dan sebagainya.
- **Batas**. Yang disebut batas (*boundary*) sistem adalah pemisah antara sistem dan daerah di luar sistem (lingkungan). Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem. Sebagai contoh, tim sepakbola mempunyai aturan permainan dan keterbatasan kemampuan pemain. Pertumbuhan sebuah toko kelontong dipengaruhi oleh pembelian pelanggan, gerakan pesaing dan keterbatasan dana dari bank. Tentu saja batas sebuah sistem dapat dikurangi atau dimodifikasi sehingga akan mengubah perilaku sistem. Sebagai contoh, dengan

menjual saham ke publik, sebuah perusahaan dapat mengurangi keterbasatan dana.

- **Mekanisme Pengendalian dan Umpan Balik**. Mekanisme pengendalian (*control mechanism*) diwujudkan dengan menggunakan umpan balik (*feedback*), yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan maupun proses. Tujuannya adalah untuk mengatur agar sistem berjalan sesuai dengan tujuan.
- **Lingkungan**. Lingkungan adalah segala sesuatu yang berada diluar sistem. Lingkungan bisa berpengaruh terhadap operasi sistem dalam arti bisa merugikan atau menguntungkan sistem itu sendiri. Lingkungan yang merugikan tentu saja harus ditahan dan dikendalikan supaya tidak mengganggu kelangsungan operasi sistem, sedangkan yang menguntungkan tetap harus terus dijaga, karena akan memacu terhadap kelangsungan hidup sistem.

2.1.2. Karakteristik Sistem

- **Memiliki Komponen**. Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, bekerja sama membentuk satu kesatuan. Komponen-komponen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem tidak perduli betapapun kecilnya, selalu mengandung komponen-

komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

- **Batas Sistem (Boundary)**. Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.
- **Lingkungan Luar Sistem (Environment)**. Adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem.
- **Penghubung Sistem (Interface)**. Merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya.
- **Masukan Sistem (Input)**. Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Signal input adalah energi yang diproses untuk didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah maintenance input yang

digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

- **Keluaran Sistem (Output)**. Merupakan hasil dari energi yang diolah oleh sistem.
- **Pengolah Sistem (Process)**. Merupakan bagian yang memproses masukan untuk menjadi keluaran yang diinginkan.
- **Sasaran Sistem**. Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.

2.1.3. Klasifikasi Sistem

- **Sistem Abstrak**. Sistem yang berupa pemikiran atau ide-ide yang tidak tampak secara fisik (sistem teologia)
- **Sistem Fisik**. Merupakan sistem yang ada secara fisik (sistem komputer, sistem akuntansi, sistem produksi dll.)
- **Sistem Alamiah**. Sistem yang terjadi melalui proses alam. (sistem matahari, sistem luar angkasa, sistem reproduksi dll.)
- **Sistem Buatan Manusia**. Sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut humanmachine system (contoh ; sistem informasi)
- **Sistem Tertentu (Deterministic System)**. Beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-

bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh ; sistem komputer)

- **Sistem Tak Tentu (Probabilistic System)**. Sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas.
- **Sistem Tertutup (Close System)**. Sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah relatively closed system (secara relatif tertutup, tidak benar-benar tertutup).

  Sistem Terbuka (Open System). Sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya.

2.2. Pengertian Informasi

Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi : input - proses – output.

2.3. Pengertian Sistem Informasi

Sistem Informasi adalah suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya. Atau sebuah sistem

terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi.

Sistem ini memanfaatkan perangkat keras dan perangkat lunak komputer, prosedur manual, model manajemen dan basis data.

2.4. Metode Analisis dan Perangcangan Terstruktur

2.4.1. Flow Map

Flow Map adalah bagan alir sistem merupakan bagan yang menunjukan arus pekerjaan scara keseluruhan dari simbol-simbol untuk menggambarkan secara urut dari arus data dan dokumen baik yang diperlukan maupun yang dihasilkan.

2.4.2. Diagram Kontek

Diagram konteks merupakan alat pemodelan atau suatu diagram yang menggambarkan sistem berbasis komputer yang dirancang secara global dan merupakan suatu diagram alir data tingkat atas, dimana didalam diagram konteks ini menggambarkan seluruh jaringan, baik masukan maupun sebuah keluaran sebuah sistem.

Diagram konteks terdiri dari sebuah simbol proses tunggal yang menggambarkan sebuah sistem dan menunjukan data aliran utama untuk dan dari terminator. Diagram ini merupakan dasar yang digunakan untuk menentukan aliran data yang mengalir menuju

sistem (*input system*) dan keluar dari sistem (*output system*), yang meliputi objek berupa kesatuan luar (*ekseternal entity*). Diagram konteks dapat mendefinisikan jangkauan proses penurunan sistem informasi yaitu menetukan apa yang menjadi bagian dari sistem informasi dan apa yang tidak menjadi bagian sistem informasi.

2.4.3. Data Flow Diagram

Diagram aliran data atau DFD merupakan suatu bagan alir data yang digunakan untuk menjelaskan data yang ditransformasikan oleh suatu proses pada suatu sistem dengan menekankan pada fungsi – fungsi yang ada dalam sistem, cara menggunakan informasi yang tersimpan dan pemindahan informasi antar fungsi di dalam sistem.

Beberapa simbol yang digunakan dalam *Data Flow Diagram* (DFD) antara lain:

- **Proses (Process).** Proses adalah simbol pertama data flow diagram. Proses dilambangkan dengan lingkaran, dimana proses ini menunjukan bagian dari sistem yang mengubah satu atau lebih input dan output. Nama proses dituliskan dengan satu kata, singkatan atau kalimat sederhana.
- **Aliran Data (Flow).** Aliran Data digambarkan dengan tanda panah dari proses. Aliran data juga digunakan untuk menunjukan bagian – bagian informasi dari satu bagian ke bagian lain. Pembagian nama untuk aliran ini menunjukan

sebuah arti untuk sebuah aliran. Untuk kebanyakan sistem yang dibuat, aliran data sebenarnya mengambarkan data yakni angka, huruf, pesan, dan macam - macam informasi lainnya.

- **Simpanan Data (Storage).** Simpanan data digunakan sebagai penyimpanan bagi paket – paket data. Notasi penyimpanan data digambarkan dengan garis horizontal yang pararel. Simpanan data merupakan simpanan data dari data yang berupa suatu file atau database di sistem komputer ataupun berupa arsip atau catatan manual. Nama dari simpanan data menunjukan nama filenya.
- **Kesatuan Luar (*External Entity*).** Setiap sistem pasti mempunyai batas sistem (*boundary*) yang memisahkan suatu sistem dengan lingkungan luarnya. Kesatuan luar merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, Organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau output dari sistem.

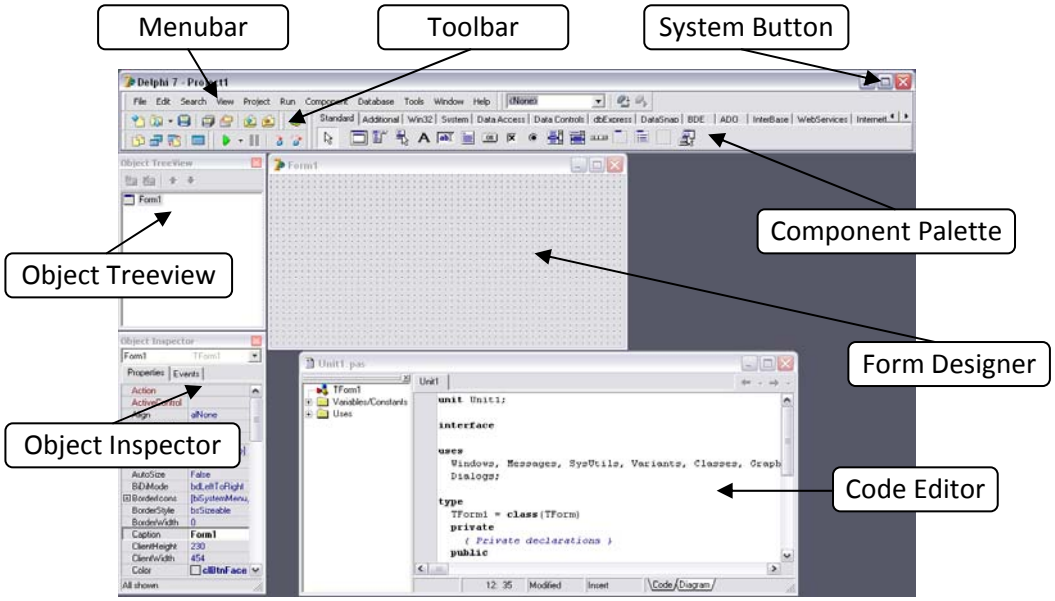
2.5. Tinjauan Perangkat Lunak Borland Delphi 7.0

Borland Delphi 7.0 adalah salah satu piranti lunak produk dari *Borland Software Corporation* versi 7 dalam bahasa pemrograman delphi. Delphi 7.0 ini merupakan bahasa pemrograman yang menggunakan metode *Graphical User Interface (GUI)* yang dimana si user dapat berinteraksi langsung dengan

komponen-komponen yang disediakan oleh Delphi dalam melakukan pemrograman.

Kekayaan bahasa di dalam Delphi 7.0 tergolong adalah bahasa yang paling mudah diantara bahasa pemrograman lainnya. Dan hampir semua bahasa di dalam Delphi 7.0 bersifat open source dengan cara menahan tombol ctrl dan mengklik kiri pada salah satu bahasa di dalam sourcecodenya.

2.5.1. Tampilan Dasar Borland Delphi 7.0



Gambar 2.1
Tampilan Borland Delphi 7.0

Gambar diatas adalah tampilan dasar Delphi 7.0. berikut adalah penjelasan dari gambar di atas :

- **Menubar**. Seperti kebanyakan program aplikasi lainnya, Delphi 7.0 juga memiliki menubar yang berisi File, Edit, Search dan lainnya.
- **Toolbar**. Merupakan kumpulan icon kecil (16x16) sebagai linker ke pada suatu perintah yang ada di dalam aplikasi.
- **System Button**. Adalah tombol bawaan dari sistem operasi yang digunakan. Di sini terdapat 3 tombol, yaitu : Minimize, Maximize dan close.
- **Component Palette**. Kumpulan – kumpulan komponen atau object yang disediakan oleh Delphi untuk mempermudahkan user dalam mendesain program. Delphi juga memungkinkan kita untuk membuat komponen – komponen tersebut atau memasang komponen eksternal lainnya.
- **Form Designer**. Tempat menyimpan komponen – komponen yang juga termasuk dalam desain secara visual.
- **Code Editor**. Area penulisan kode dari form designer atau pas unit.
- **Object Treeview**. Tampilan dari semua komponen – komponen yang diletakkan di dalam form designer dengan object form sebagai induknya.

- **Object Inspector**. Fasilitas yang berisi properti dan event yang dimiliki oleh komponen dalam form.

2.5.2. Komponen EkszBoxVCL

EkszBoxVCL merupakan komponen external untuk Delphi dan Visual C yang diciptakan oleh Kernel Master tahun 2007-2008. Komponen ini memberikan kita kesempatan untuk lebih kreatif dalam mendesain form. Komponen ini dibuat untuk penggunaan dalam Delphi 5.0, Delphi 6.0, Delphi 7.0, Delphi 2005, Delphi 2006 dan Delphi 2007 serta beberapa versi untuk pemrograman Visual C.

Berikut adalah komponen – komponen yang terdapat pada Component Package EkszBoxVCL, yaitu :

Tabel 2.1.
Komponen – Komponen EkszBoxVCL

Icon	Nama	Keterangan
	TEkDirBrowse	Sebagai explorer dalam tampilan treeview.
	TEkGradientProgressBar	Terlihat sederhana, tapi merupakan progressbar yang sangat atraktif dengan gradien warna.
	TEkImgBtn	Kontrol button yang dapat kita rubah skinnya dengan dukungan efek fading.
	TEkImgCheckBox	Kontrol checkbox yang dapat kita

		rubah skinnya dengan dukungan efek fading.
	TEkImgFade	Memberikan kita untuk melakukan efek fading antara dua image dengan beragam efek.
	TEkImgForm	Kontrol yang sangat mudah dalam memberikan skin pada form kita buat.
	TEkImgLED	Kontrol LCD dengan skin yang bisa kita rubah dan didesain dengan 7 segmen tampilan LED.
	TEkImgProgressBar	Adalah sebuah fitur progressbar kontrol yang dapat kita rubah skinnya dengan berbagai efek animasi.
	TEkImgTrackBar	Adalah sebuah fitur trackbar kontrol yang dapat kita rubah skinnya dengan berbagai efek animasi.
	TEkLabel	Kontrol label dengan dua warna.

2.5.3. Komponen PicShow

PicShow adalah sebuah komponen yang diciptakan oleh Kambiz R. Khojasteh. Dengan komponen ini, kita bisa membuat efek transisi pada dua gambar atau lebih. Pada komponen ini, kita telah disediakan berbagai macam efek transisi dan rentang waktu untuk

proses transisi tersebut. Berikut adalah beberapa komponen yang terdapat dalam unit PicShow, yaitu :

Tabel 2.2.
Komponen – Komponen PicShow

Icon	Nama	Keterangan
	TpicShow	Sebuah komponen yang memiliki dua image utama, yaitu background picture dan picture. Komponen ini bisa melakukan transisi antara kedua gambar tersebut dengan beragam efek transisi. Selain itu kedua image itu dapat kita rubah saat runtime program dengan memberikan sedikit kode.
	TDBPicShow	Adalah komponen yang dapat memuat gambar – gambar yang ada pada suatu tabel dalam database dengan berbagai macam efek transisi.

BAB III

PROFIL PERUSAHAAN

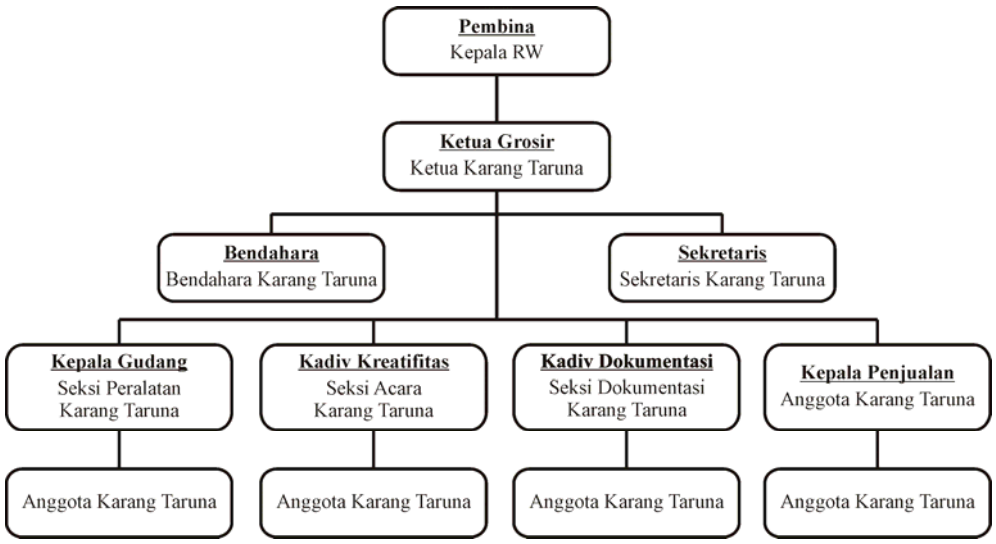
3.1. Tinjauan Umum Perusahaan

Manoko Grosir adalah buah karya anak – anak Karang Taruna RW 03 Kampung Manoko Desa Cikahuripan Kecamatan Lembang Kabupaten Bandung Barat yang terbentuk pada tanggal 29 Agustus 2008. Dengan persetujuan dari Kepala RW 03 Deden Gumilar S.Pd. serta sumbangan dana warga RW 03 yang baik berupa materi, tenaga serta bangunan sebagai tempatnya, terbentuk usaha dagang Karang Taruna RW 03 yang dinamakan Manoko Grosir. Dengan slogan *Sasareungan urang tiasa* yang dalam bahasa Indonesia berati bersama kita bisa, Manoko Grosir bertujuan untuk membangun Karang Taruna dan warga di daerahnya.

Manoko Grosir sangat membantu perekonomian warga sekitar. Bagaimana tidak, karena sebelum adanya grosir ini, warga harus mengeluarkan ongkos untuk berbelanja ke pasar Lembang yang sekarang ongkos tersebut dapat ditekan dengan adanya grosir ini. Selain itu dengan adanya grosir ini, permintaan sumbangan terhadap warga pun akan berkurang, seperti dana untuk hari kemerdekaan atau pembangunan tempat beribadah dan event lainnya karena dibantu oleh laba dari grosir ini. Dan para anak muda terutama anggota Karang Taruna RW 03 bisa menambah uang saku mereka ketimbang menganggur dengan membantu pekerjaan pada grosir ini. Manoko Grosir menawarkan beberapa kebutuhan pada warga sekitar, seperti sembako dan kebutuhan lainnya.

3.2. Struktur Organisasi

Seperti pada organisasi lainnya, Manoko Grosir adalah salah satu organisasi yang terstruktur. Berikut adalah bagan struktur organisasi Manoko Grosir :



Gambar 3.1.
Struktur Organisai Manoko Grosir

3.3. Deskripsi Kerja

Berikut ini adalah bidang kerja dari bagian – bagian peran pengelola pada Manoko Grosir, yaitu :

- Pembina

  Merupakan orang yang mempunyai jabatan tertinggi pada daerah tersebut (Kepala RW). Sebagai seorang pembina, dia harus berperan dalam memberikan pengarahan dari problematika atau hal – hal lainnya yang terjadi di dalam organisasi. Selain itu pembina juga menjadi tempat bertanya ketua grosir dalam hal seputar grosir.

- Ketua Grosir

  Ketua grosir dipegang oleh Ketua Karang Taruna. Ada beberapa tugas dari seorang ketua, yaitu :

  - Mengkoordinasi bagian – bagian yang berada pada grosir.
  - Mengawasi aktivitas sehari – hari di grosir.
  - Memberikan bimbingan serta petunjuk tentang pekerjaan dari masing – masing bagian di bawahnya.
  - Bertanggung jawab terhadap perkembangan atau penurunan kinerja grosir secara keseluruhan.
  - Menerima semua laporan dari Kadiv Dokumentasi.
  - Berhak memberikan izin atau tidak atas segala uang yang akan keluar dari grosir.

- Bendahara

  Peran ini dipegang oleh Bendahara Karang Taruna. Berikut adalah tugas dari bendahara, yaitu :

  - Mengelola semua keuangan grosir.
  - Menyimpan uang dari pemasukan grosir.
  - Memberikan uang yang akan keluar atas izin dari ketua.
  - Membuat laporan keuangan untuk diserahkan kepada Kadiv dokumentasi.
  - Bertanggung jawab atas semua pekerjaannya kepada ketua grosir.

- Sekretaris

  Posisi ini dipegang oleh Sekretaris Karang Taruna. Pekerjaan utama sekretaris adalah membantu pekerjaan ketua grosir, terutama dalam hal pencatatan.

- Kepala Gudang

  Ketua divisi ini dipegang oleh Seksi Peralatan Karang Taruna. Berikut adalah tugas dari Kepala Gudang, yaitu :

  - Mengawasi segala aktivitas yang berhubungan dengan barang dagangan dan inventory grosir.
  - Memberikan pekerjaan kepada anggota di bawahnya baik dalam pembelian barang dagangan maupun perawatan inventory grosir.
  - Mendata semua transaksi pembelian.
  - Bertanggung jawab atas barang dagangan dan inventory yang ada di grosir terhadap ketua grosir.

- Kadiv Kreatifitas

  Kepala divisi ini dipegang oleh Seksi Acara Karang Taruna dengan tugas sebagai berikut :

  - Menjaga kenyaman pembeli ataupun pekerja di grosir dengan memberikanan tatanan ruang grosir secara kreatif.
  - Memberikan bimbingan serta apresiasi terhadap produk kreatif karya anggota di bawahnya.

- Menyeleksi karya – karya anggota di bawahnya untuk dijadikan bahan penghias ruangan grosir.
- Bertanggung jawab atas semua pekerjaannya terhadap ketua grosir.

➢ Kadiv Dokumentasi

Kepala divisi ini dipegang oleh Seksi Dokumentasi Karang Taruna. Adapun tugasnya adalah sebagai berikut :

- Mengawasi, memberi bimbingan dan memberikan pekerjaan kepada anggota di bawahnya dalam pembuatan laporan.
- Membuat laporan berdasarkan data – data transaksi dari bagian gudang dan kasir.
- Memberikan laporan secara berkala kepada ketua grosir atau sesuai permintaan dari ketua grosir.
- Bertanggung jawab atas laporan – laporan tersebut terhadap ketua grosir.

➢ Kepala Penjualan

Kepala penjualan dipegang oleh salah satu anggota karang taruna atas pilihan bersama. Tugas kepala penjualan adalah :

- Mengkoordinir segala pekerjaan anggota di bawahnya.
- Mendata semua transaksi penjualan.
- Bertanggung jawab atas semua transaksi penjualan yang ada di grosir kepada ketua grosir.

- Anggota Karang Taruna level terbawah

  Adalah orang – orang yang dikoordinir langsung oleh ketua secara bersama sesuai dengan kemampuannya masing – masing dalam penempatan bagiannya. Tugas utama mereka adalah melakukan segala pekerjaan yang diperintahkan dan bertanggung jawab atas pekerjaannya itu kepada bagian yang berada di atasnya masing – masing.

3.4. Analisis Sistem yang Berjalan

Pada Manoko Grosir, Sistem Pembelian dan Penjualan di dalamnya telah memakai komputer dalam melakukan pencatatan. Namun sayang, program yang dipakai adalah excell. Walaupun excell memiliki pencarian data yang lumayan cepat, namun hasil dari pencarian tersebut tidak variatif seperti yang bisa dilakukan oleh query.

Selain itu, pencatatan tidak dilakukan langsung oleh bagian penjualan dan pembelian. Namun mereka hanya menyimpan struk penjualan dan pembelian yang nantinya akan di rekap oleh pihak dokumentasi dalam membuat laporan, dengan kata lain itu bisa disebut dengan kerja secara dua kali dan terlalu memberatkan pihak dokumentasi yang akan kita sebut sebagai admin.

Bentuk laporan di sini pun tidak begitu detail, seperti untuk satu barang akan di akumulasikan dalam sehari. Jadi kita tidak bisa tahu detail transaksi tersebut.

BAB IV

ANALISIS KERJA PRAKTEK

4.1 Analisis Sistem Manoko Grosir

Sebagai kesatuan dari elemen – elemen di dalamnya, Manoko Grosir merupakan sebuah sistem. Di dalam sistem Manoko Grosir terdapat beberapa sub sistem seperti contoh dalam pembahasan ini adalah sistem pembelian dan penjualan. Berikut adalah hasil analisis sistem pembelian dan penjualan yang sedang berjalan di Manoko Grosir.

4.1.1. Analisis Dokumen

Ada beberapa dokumen yang penyusun dapatkan selama melakukan kerja praktek di Manoko Grosir, yaitu :

- Laporan Penjualan bulan Agustus 2009

  Dalam laporan yang dibuat dari Microsoft Excell ini berisi daftar semua transaksi penjualan Manoko Grosir selama bulan Agustus tahun 2009 dalam bentuk tabel dengan 5 kolom, yaitu *tanggal*, *nama barang*, *harga*, *banyaknya dan jumlah*. Berikut adalah rangka laporan penjualan yang dikutip dari lampiran :

MANOKO GROSIR
Laporan Penjualan
Agustus 2009

Tanggal	Nama Barang	Harga	Banyaknya	Jumlah
			Total	

Ketua Grosir Dokumentasi

Gambar 4.1
Rangka Laporan Penjualan

❖ Laporan Pembelian bulan Agustus 2009

Bentuk dari laporan ini tidak jauh beda dengan laporan penjualan yaitu terdiri dari 5 kolom. Berikut adalah adalah rangka laporan pembelian yang dikutip dari lampiran :

MANOKO GROSIR
Laporan Pembelian
Agustus 2009

Tanggal	Nama Barang	Harga	Banyaknya	Jumlah
			Total	

Ketua Grosir Dokumentasi

Gambar 4.2
Rangka Laporan Pembelian

❖ Struk Penjualan Manoko Grosir

Struk Penjualan ini adalah Nota Penjualan pada Manoko Grosir. Disini terdapat beberapa field, yaitu *tanggal, kepada, no. Struk, nama barang, harga, banyaknya, jumlah dan total*. Berikut adalah rangka dari struk penjualan Manoko Grosir yang dikutip dari lampiran :

MANOKO GROSIR Tanggal :
Kp. Manoko RT 01/03 Kepada :
Ds. Cikahuripan – Kec. Lembang

No. Struk :

Nama Barang	Harga	Banyaknya	Jumlah

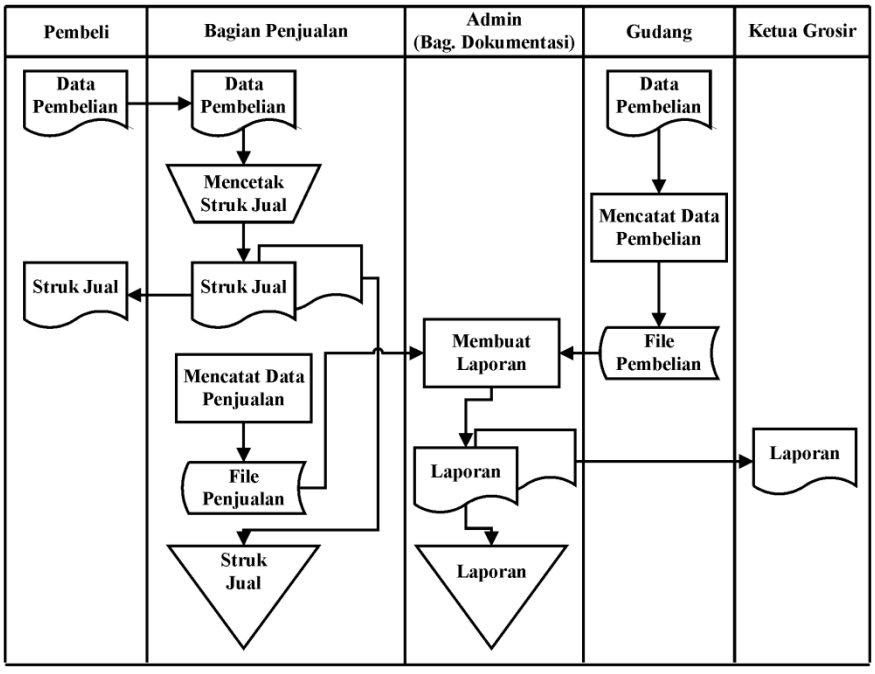
Gambar 4.3
Rangka Struk Penjualan

4.1.2. Analisis Prosedur yang Sedang Berjalan

Berikut adalah hasil analisis dari Sistem Informasi Pembelian dan Penjualan yang sedang berjalan di Manoko Grosir :

4.1.2.1. Flow Map Sistem yang Sedang Berjalan

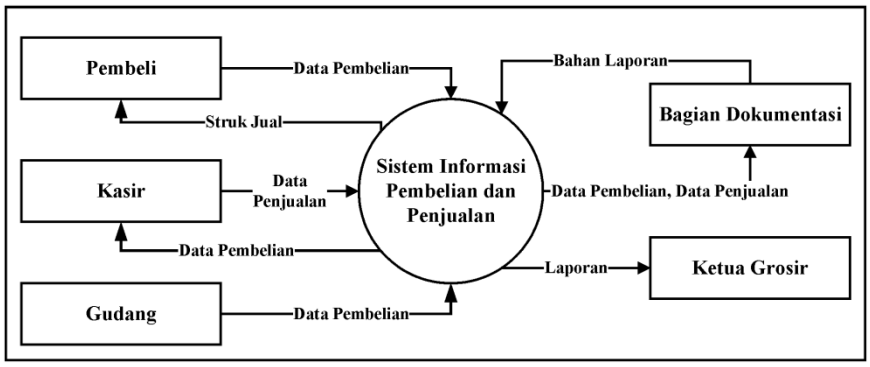
Berikut ini adalah flow map sistem yang sedang berjalan :



Gambar 4.4
Flow Map Sistem yang Sedang Berjalan

4.1.2.2. Diagram Kontek Sistem yang Sedang Berjalan

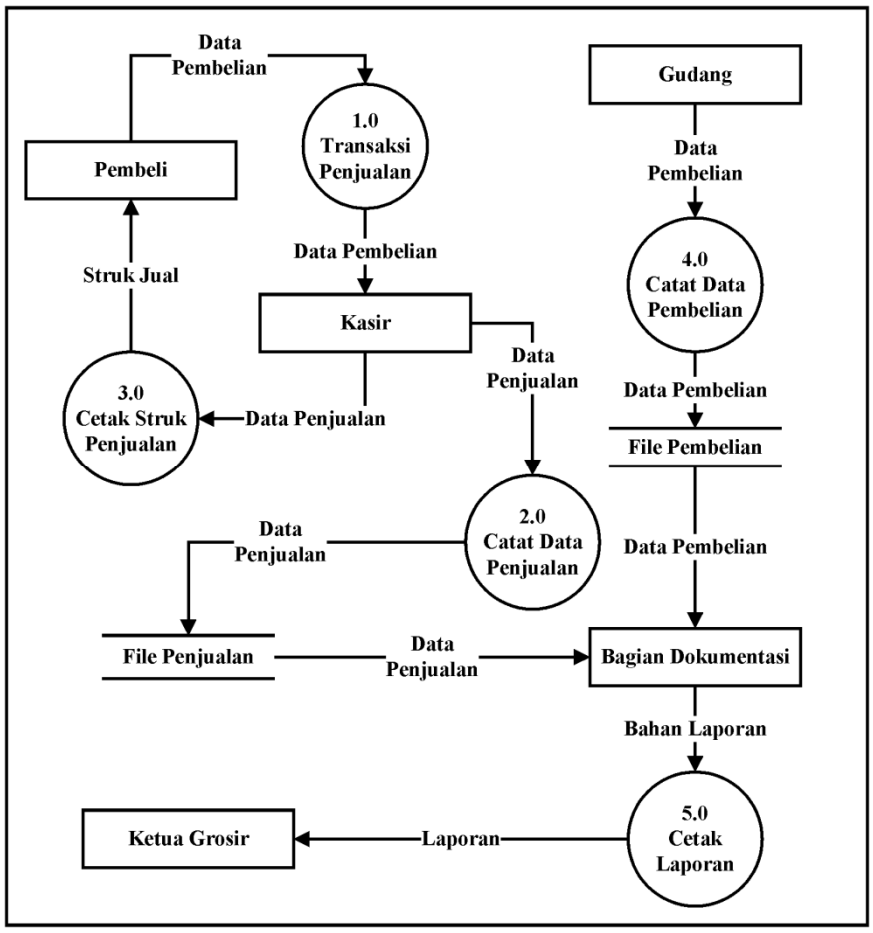
Berikut ini adalah diagram kontek sistem yang sedang berjalan :



Gambar 4.5
Diagram Kontek Sistem yang Sedang Berjalan

4.1.2.3. DFD Sistem yang Sedang Berjalan

Berikut ini adalah DFD (*Data Flow Diagram*) sistem yang sedang berjalan :



Gambar 4.6
DFD Sistem yang Sedang Berjalan

4.1.3. Evaluasi Sistem yang Sedang Berjalan

Jika melihat dari hasil analisis di atas, pada flow map terdapat dua buah file karena sistem ini tidak memiliki database melainkan dua file excell (data pembelian, data penjualan). Selain itu, kasir tidak diberi fasilitas untuk melihat harga barang. Untuk melihat harga barang, kasir harus mengingat harga barang tersebut atau membuka file daftar harga.

4.2 Usulan Perancangan Sistem

Setelah melakukan analisis dan melihat kelemahan dari sistem yang telah ada pada Manoko Grosir, maka penyusun memberikan sebuah usulan perancangan sistem.

4.2.1. Tujuan Perancangan Sistem

Tujuan perancangan sistem ini adalah untuk mempercepat pengolahan data dan informasi terutama dalam pembuatan laporan secara efektif, sehingga data yang ada akan tersimpan dengan baik dan jika ada data yang tidak diperlukan bisa dihilangkan, sehingga terdapat pengumpulan data yang tersusun rapi dan seragam.

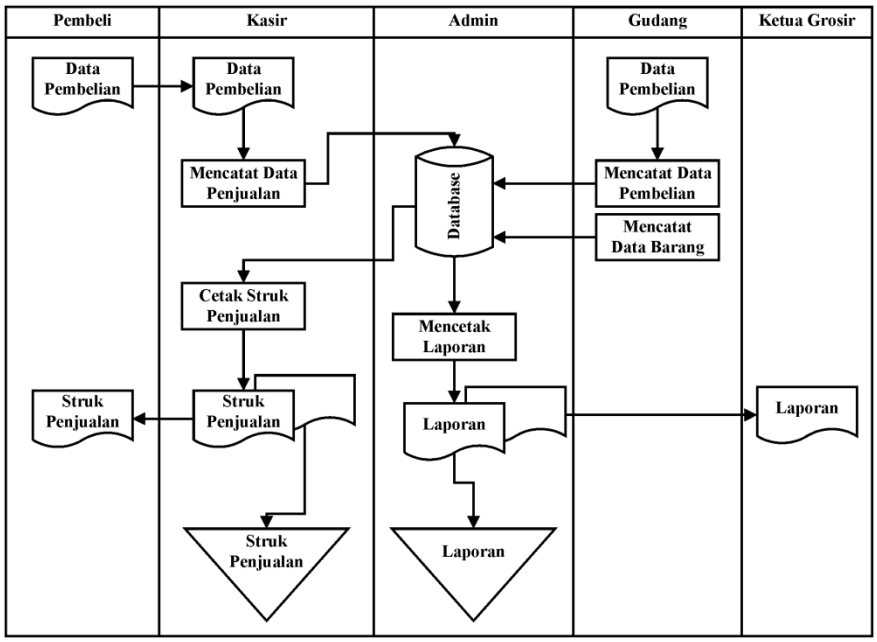
4.2.2. Perancangan Prosedur yang Diusulkan

Setelah melakukan analisis terhadap sistem yang sedang berjalan, penyusun mencoba merancang sebuah prusedur usulan dengan tidak merubah penuh sistem yang ada.

Berikut adalah rancangan sistem yang akan diusulkan :

4.2.2.1. Flow Map yang Diusulkan

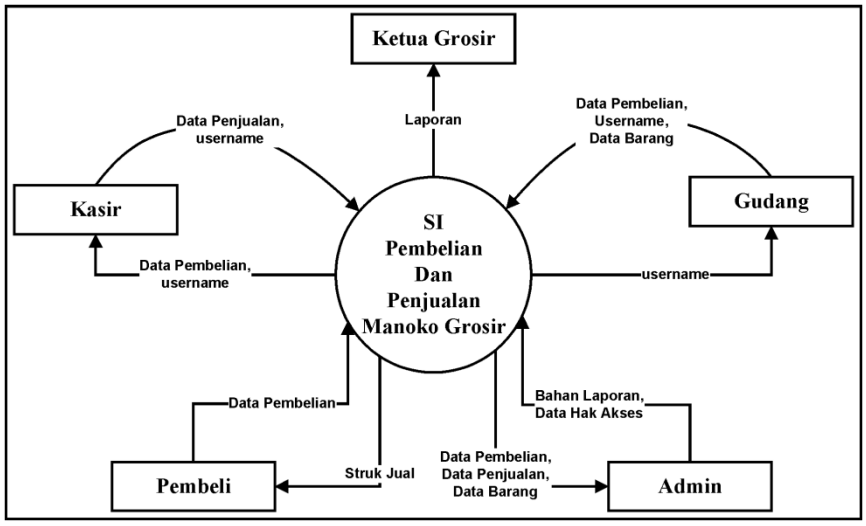
Tidak seperti dengan flow map yang sedang berjalan. Dalam flow map usulan ini, penyusun mengganti dua buah tempat penyimpanan data yang berupa *excell* dengan sebuah database yang menggunakan *SQL Server*. Berikut ini adalah flow map yang diusulkan :



Gambar 4.7
Flow Map yang Diusulkan

4.2.2.2. Diagram Kontek yang Diusulkan

Untuk diagram kontek, tidak mengalami perubahan. Karena diagram kontek usulan hasil pertimbangan penyusun persis sama dengan sebelumnya, yaitu :

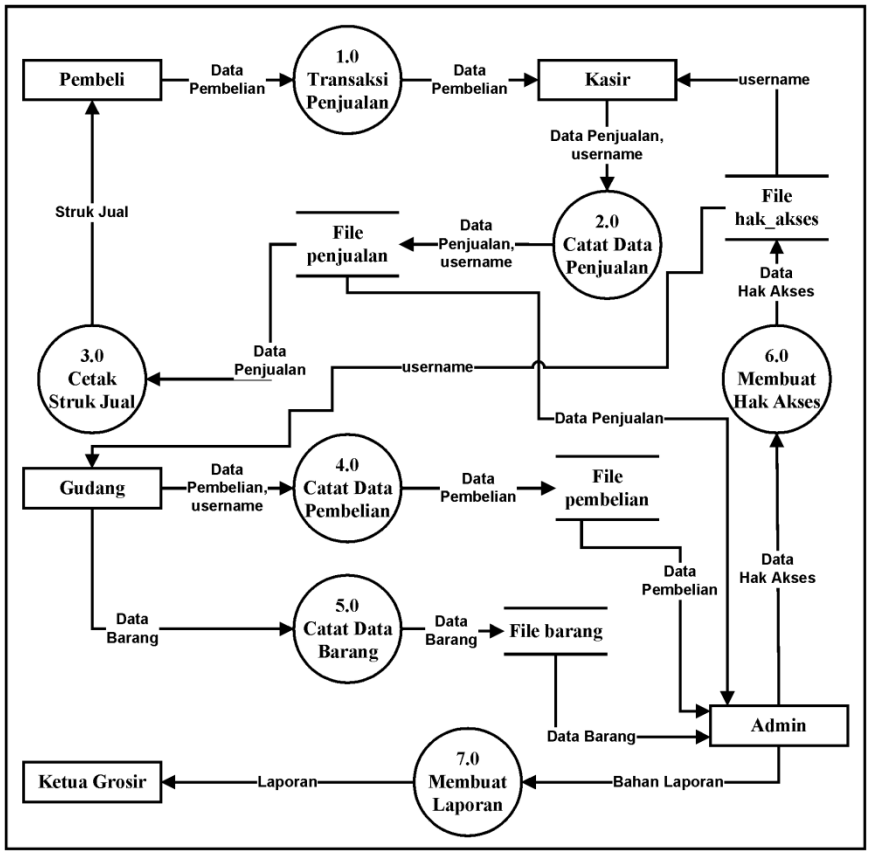


Gambar 4.8
Diagram Kontek yang Diusulkan

4.2.2.3. DFD yang Diusulkan

Walaupun pada diagram kontek tidak mengalami perubahan, namun pada *data flow diagram (DFD)* terdapat beberapa perubahan sesuai dengan flow map usulan.

Berikut adalah DFD yang diusulkan :



Gambar 4.9
DFD yang Diusulkan

4.2.2.4. Kamus Data

Kamus data merupakan tempat penyimpanan informasi dari aliran-aliran data atau file-file yang digunakan dalam sebuah sistem informasi. Dalam perancangan program aplikasi *Sistem*

Pembelian dan Penjualan Manoko Grosir terdiri dari beberapa tabel, yaitu :

❖ Tabel penjualan

Tabel Penjualan adalah file tempat menyimpan semua transaksi penjualan.

Attribut Kunci : kode_jual

Tabel 4.1
Tabel penjualan

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	kode_jual	varchar	11	Kode jual
2.	nama_pembeli	varchar	30	Nama pembeli
3.	tgl_jual	varchar	10	Tanggal penjualan
4.	total_harga	float	8	Total harga
5.	bag_kasir	varchar	20	Id kasir yang mencatat transaksi penjualan

❖ Tabel dtil_jual

Tabel ini berisi tentang detail dari penjualan.

Tabel 4.2
Tabel dtil_jual

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	kode_jual	varchar	11	Kode jual
2.	kode_brg	varchar	7	Kode barang
3.	qty	float	8	Quantity
4.	harga_satuan	float	8	Harga satuan
5.	jml_harga	float	8	Jumlah harga

❖ Tabel hak_akses

Tabel ini merupakan penyimpanan data hak akses untuk login program dan identitas suatu transaksi.

Attribut kunci : username

Tabel 4.3
Tabel hak_akses

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	username	varchar	20	Username
2.	pass	varchar	20	Password
3.	jenis	char	1	Jenis hak akses

❖ Tabel barang

Tabel ini berisi data barang.

Attribut kunci : kode_brg

Tabel 4.4
Tabel barang

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	kode_brg	varchar	7	Username
2.	nama_brg	varchar	30	Nama barang
3.	satuan	varchar	20	Satuan barang
4.	harga_jual	float	8	Harga jual satuan
5.	stok_awal	float	8	Stok awal barang
6.	terhapus	char	1	Status terhapus
7.	stok_minim	float	8	Stok minimal barang

❖ Tabel pembelian

Tempat penyimpanan transaksi pembelian

Attribut kunci : kode_beli

Tabel 4.5
Tabel pembelian

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	kode_beli	varchar	7	Username
2.	tgl_beli	varchar	10	Tanggal pembelian
3.	bag_gudang	varchar	20	Id gudang

				yang mencatat transaksi
4.	kode_brg	varchar	7	Kode barang
5.	qty	float	8	Quantity
6.	harga_satuan	float	8	Harga satuan barang
7.	jumlah_harga	float	8	Jumlah harga
8.	dari	varchar	30	Tempat membeli barang

❖ Tabel temp

Tabel temp digunakan untuk mendata stok barang yang digunakan dalam pembuatan laporan.

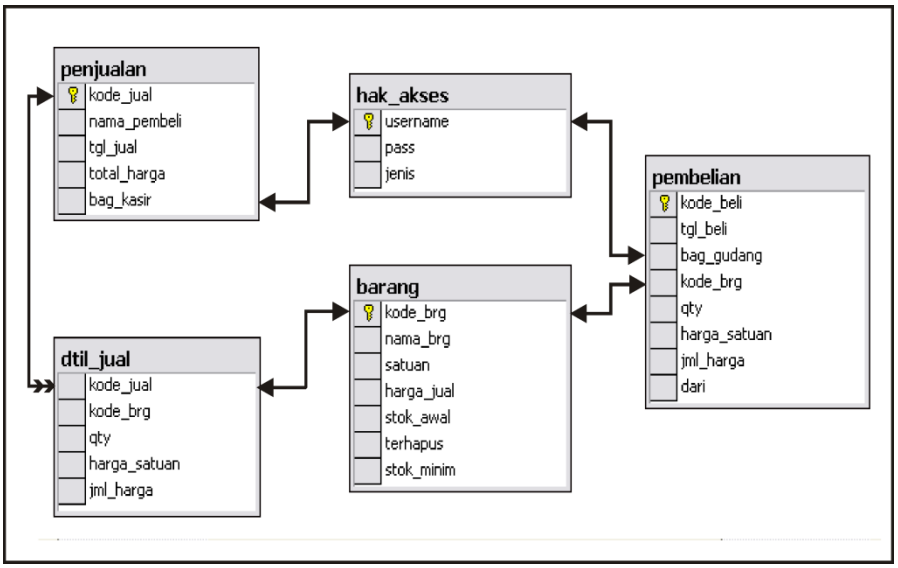
Tabel 4.6
Tabel temp

No.	Nama Field	Tipe Data	Ukuran	Keterangan
1.	k_brg	varchar	7	Kode barang
2.	n_brg	varchar	30	Nama barang
3.	satuan	varchar	20	Satuan barang
4.	awal	float	8	Stok awal barang
5.	jml_beli	float	8	Jumlah

				quantity pembelian
6.	jml_jual	float	8	Jumlah quantity penjualan
7.	akhir	float	8	Stok akhir barang

4.2.2.5. Tabel Relasi

Relasi antar tabel merupakan suatu proses mengorganisasikan *file* untuk menghilangkan group elemen yang berulang-ulang. Proses relasi antar tabel adalah mngelompokan data menjadi tabel-tabel yang menunjukan entity dan relasinya yang berfungsi untuk mengakses data item sedemikian rupa sehingga *database* tersebut mudah dimodifikasi. Tabel relasi pada program aplikasi *Sistem Pembelian dan Penjualan Manoko Grosir* dapat dilihat pada gambar dibawah ini :



Gambar 4.10
Tabel Relasi

4.2.2.6. Perancangan Kode

Pengkodean digunakan untuk mempermudah dalam membedakan data-data yang terlibat dalam satu tabel.

Berikut ini adalah kode – kode dan struktur kode yang ada pada database *Sistem Pembelian dan Penjualan Manoko Grosir* :

1. kode_jual

   kode_jual dibuat untuk mengidentifikasi suatu transaksi penjualan. Kode ini dibentuk dari kumpulan huruf dan angka.

   Format kode jual :

   J 0 1 2 3 4 5 0 1 2 3

   I II III

   I : J - untuk identifikasi Jual

II : 6 digit untuk tanggal dengan format *ddmmyy*

III : 4 digit untuk urutan transaksi penjualan pada tanggal tersebut

2. username

   username dibuat untuk mengidentifikasi hak akses untuk login dan id pencatat pada transaksi. Kode ini tidak mempunyai format tertentu. Namun seperti kode lainnya, kode ini bersifat *unik*. Kode ini bersifat *lowercase* dan *tidak memiliki spasi*.

3. kode_brg

   kode_brg dibuat untuk mengidentifikasi barang. Kode ini dibentuk dari kumpulan huruf dan angka.

   Format kode_brg :

   B R G 0 1 2 3
   I II

   I : BRG – untuk identifikasi Barang

   II : 4 digit untuk urutan barang.

4. kode_beli

   kode_beli dibuat untuk mengidentifikasi suatu transaksi pembelian. Kode ini dibentuk dari kumpulan huruf dan angka.

   Format kode beli :

B 0 1 2 3 4 5 0 1 2 3

I II III

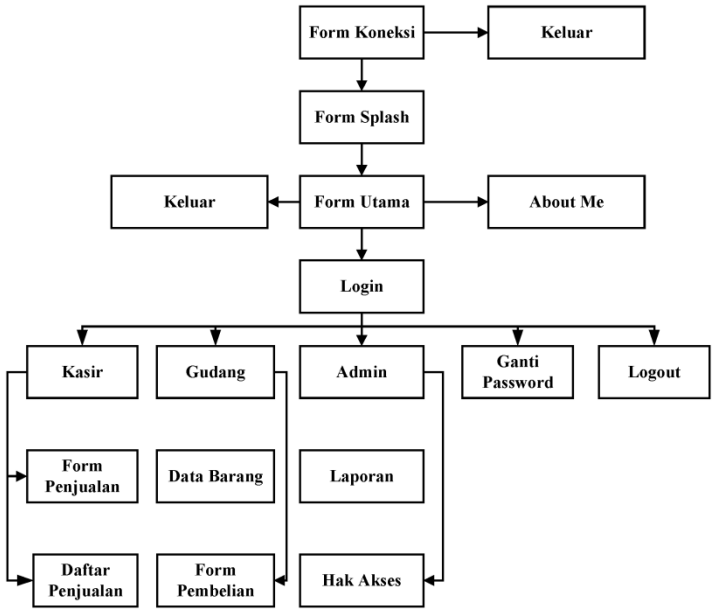
I : B - untuk identifikasi Beli

II : 6 digit untuk tanggal dengan format *ddmmyy*

III : 4 digit untuk urutan transaksi pembelian pada tanggal tersebut

4.2.2.7. Struktur Menu Program

Struktur menu merupakan bentuk utama dari suatu rancangan program untuk memudahkan dalam melihat dan menggunakan sesuai dengan kebutuhan struktur menu porgram aplikasi *Sistem Pembelian dan Penjualan Manoko Grosir*. Adapun struktur menu program tersebut adalah sebagai berikut :



Gambar 4.11
Struktur Menu Program

4.2.2.8. Kebutuhan Sistem

Program aplikasi ini dapat berjalan pada spesifikasi komputer minimum yaitu :

- Sistem Operasi semua versi Windows di atas Win Me.
- Processor yang mendukung SO Windows Me ke atas.
- RAM 256 MB (Dianjurkan 512 MB).
- VGA dan Monitor dengan resolusi 1024 x 768 pixels dan kualitas warna 32 Bit.
- Printer (dianjurkan berwarna).

4.2.3. Program Aplikasi yang Diusulkan

Setelah melalui perancangan di atas, maka penyusun membuat sebuah program aplikasi *Sistem Pembelian dan Penjualan Manoko Grosir*. Berikut ini tampilan adalah program yang diusulkan :

- **Form Koneksi**

  Form ini digunakan untuk menentukan nama server dan database yang dipakai.



Gambar 4.12
Form Koneksi

Dari gambar di atas, form tersebut terdiri dari 2 buah inputan (nama / IP server dan nama database). Pada saat program pertama kali dijalankan dalam sistem, nama / ip server akan otomatis terisi dengan nama komputer dari sistem operasi komputer tersebut serta nama database *db_mg*. Jika server terkoneksi maka program akan menyimpan dua buah inputan itu ke dalam registry sistem operasi, sehingga kita tidak perlu menginputkan nama server dan databasenya nanti.

- **Form Splash**

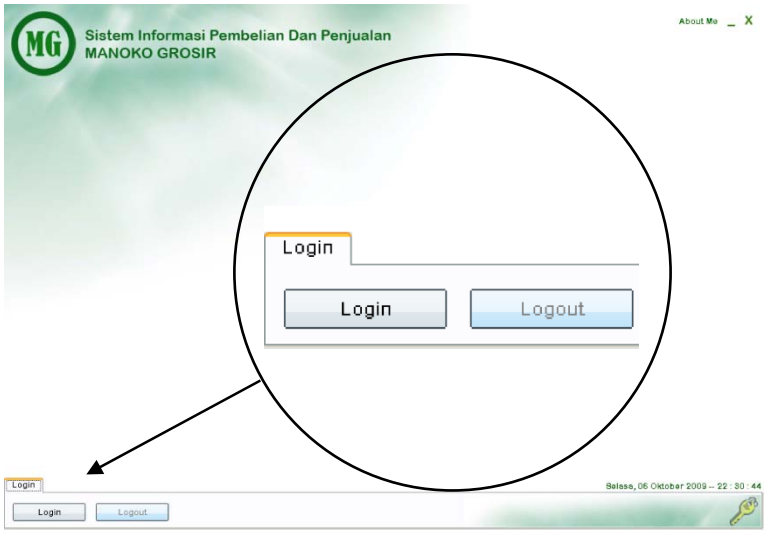
  Form ini merupakan form sambutan dengan efek fade in saat tampil dan fade out saat tertutup. Form ini hanya sekedar tampilan pembuka program.



Gambar 4.13
Form Splash

➢ **Form Utama (Tab Login)**

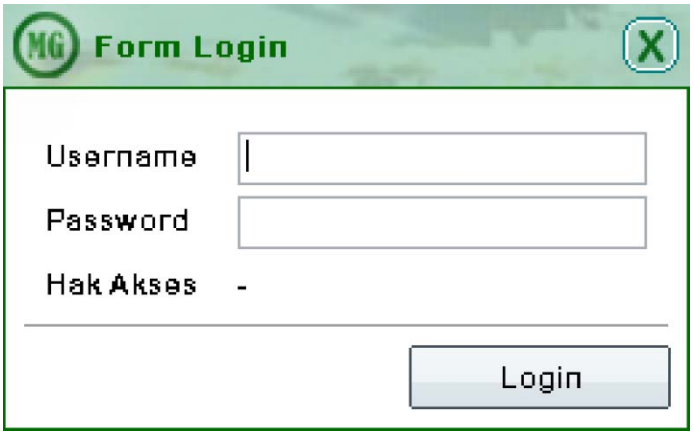
Adalah form induk dari program ini. Form ini terdapat 1 buah page tab. Pada page tab ini terdapat tab login, kasir, gudang dan admin. Namun untuk tab kasir, gudang dan admin tidak akan tampil sebelum user melakukan login sesuai jenis hakaksesnya masing – masing.



Gambar 4.14
Form Utama (Tab Login)

➢ **Form Login**

Digunakan untuk login user agar bisa mengakses program.



Gambar 4.15
Form Login

Pada form ini terdapat inputan username dan password. Bila username yang diinputkan terdaftar, maka hak akses akan muncul sesuai dengan hak akses yang dia miliki. Terdapat tiga jenis hak akses, yaitu :

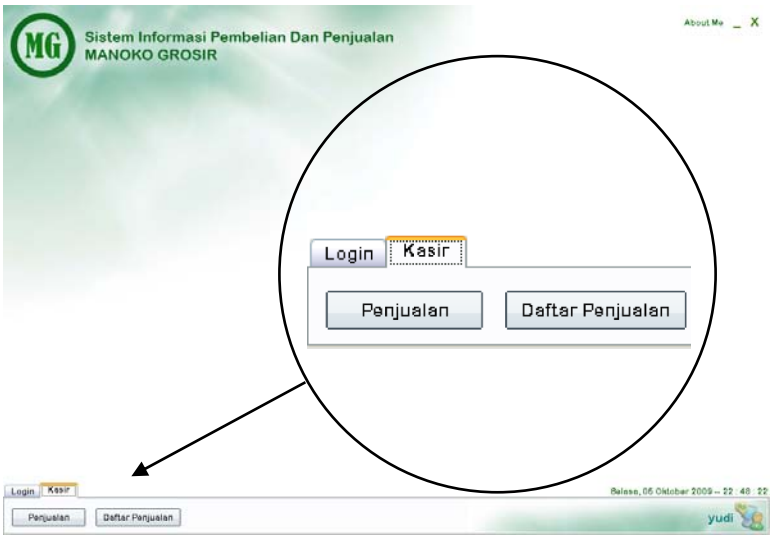
- Kasir : Untuk pengolahan data penjualan (tab kasir)
- Gudang : Untuk pengolahan data pembelian (tab gudang)
- Admin : Untuk pengolahan data hak akses dan laporan (tab admin)

Jika user menginputkan dengan benar, maka tombol login pada form utama akan berubah menjadi tombol ganti password dan sebuah tab akan muncul sesuai dengan hak aksesnya.

➢ **Form Utama (Tab Kasir)**

Jika pada form login user memiliki hak sebagai kasir dan passwordnya benar, maka tab kasir akan muncul pada

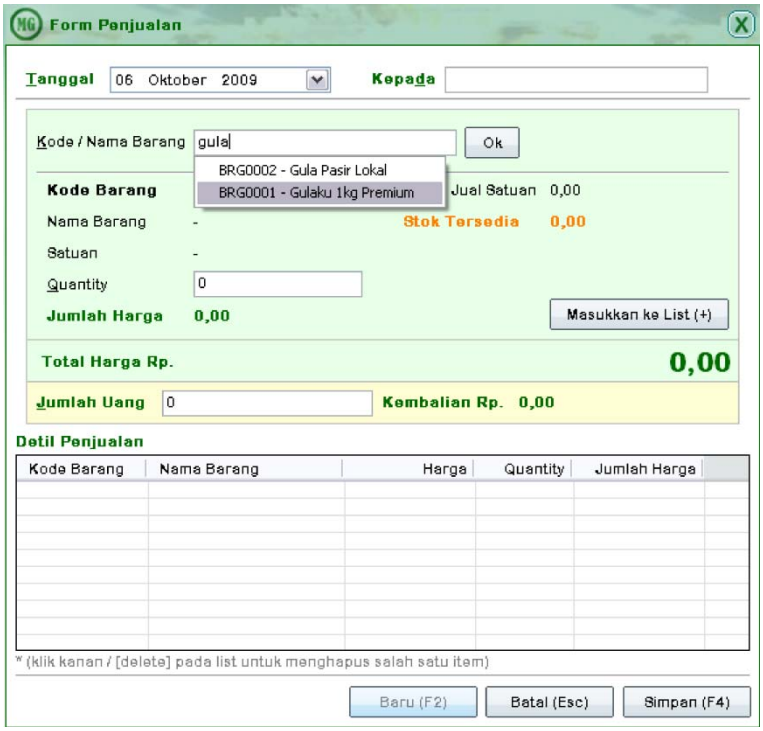
form utama. Pada tab ini terdapat dua buah tombol menu, yaitu *Penjualan* dan *Daftar Penjualan*.



Gambar 4.16
Form Utama (Tab Kasir)

- **Form Penjualan**

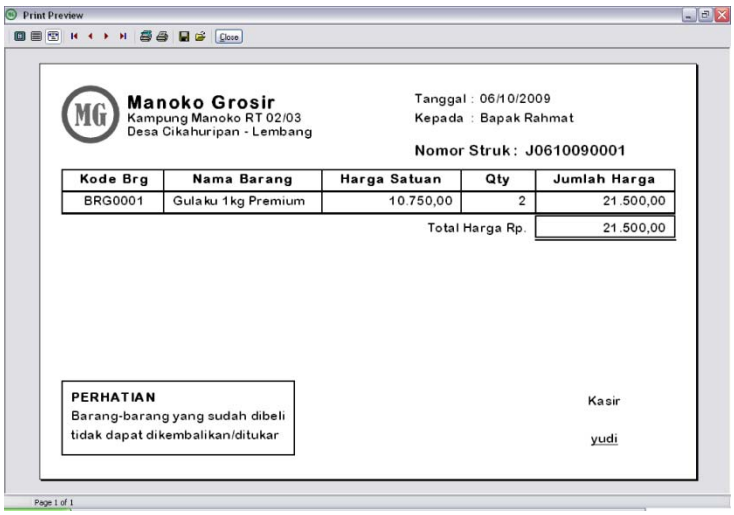
Form ini digunakan oleh kasir dalam setiap transaksi penjualan. Di sini penyusun memberikan teknik mencari data barang dengan menggunakan edit box. Kasir hanya menginputkan beberapa karaktek dari nama atau kode barang tersebut, maka sebuah popupmenu yang berisi hasil pencarian barang akan muncul di bawah edit box tersebut.



Gambar 4.17
Form Penjualan

➢ **Struk Jual**

Setelah kasir menyimpan data penjualan pada form penjualan, maka struk jual akan tampil.



Gambar 4.18
Struk Jual

- **Form Daftar Penjualan**

Form ini membantu kasir untuk melihat transaksi penjualan yang ia catat. Kasir hanya tinggal memilih tanggal transaksi, maka semua transaksi yang ia catat pada tanggal tersebut akan tampil pada list.



Gambar 4.19
Form Daftar Penjualan

- **Form Detail Penjualan**

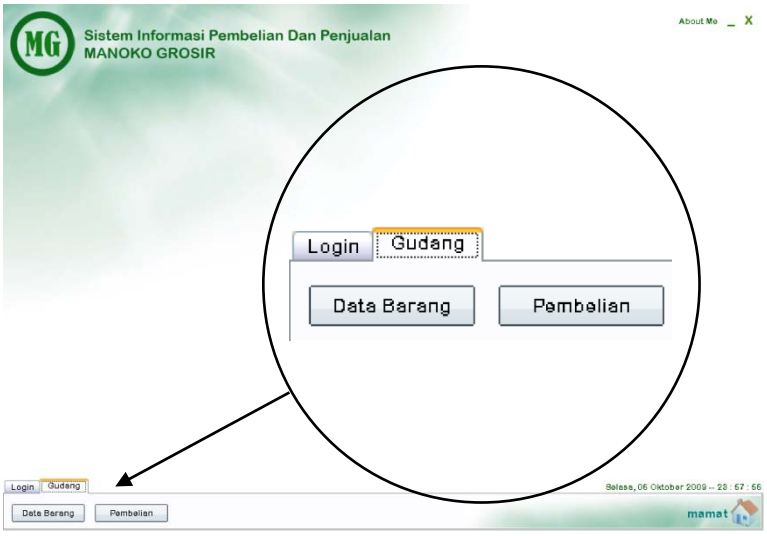
Form ini akan tampil jika kasir meilih salah satu transaksi pada list yang terdapat pada form daftar penjualan. Form ini mempermudah kasir untuk menghapus data transaksi penjualan yang salah atau mencetak ulang struk jual pada transaksi yang ia catat. Untuk mencetak ulang struk, program ini akan kembali memanggil struk jual dengan kode penjualan yang dipilih pada list.



Gambar 4.20
Form Detail Penjualan

➢ **Form Utama (Tab Gudang)**

Jika pada form login user memiliki hak sebagai gudang dan passwordnya benar, maka tab gudang akan muncul pada form utama. Pada tab ini terdapat dua buah tombol menu, yaitu *Data Barang* dan *Pembelian*.



Gambar 4.21
Form Utama (Tab Gudang)

- **Form Daftar Barang**

Form ini merupakan form master barang. Disini penyusun menggunakan metode pencarian secara query. Untuk mencari barang, user hanya tinggal menginputkan beberapa karakter dari nama atau kode barang yang ia cari dan hasil pencarian pun akan masuk kedalam list.



Gambar 4.22
Form Daftar Barang

Form ini juga memiliki tombol tambah, edit, hapus dan stok detail barang.

- **Form Tambah Barang / Edit Barang**

Form ini digunakan untuk menginputkan data barang baru. Disini terdapat field *nama barang, satuan, harga jual, stok awal* dan *stok minimal*. Untuk kode barang, program akan melakukannya otomatis setelah data inputan disimpan. Hal ini dibentuk agar tidak terjadi duplikat data pada tabel

barang ketika terjadi penyimpanan data secara bersamaan pada client / server.



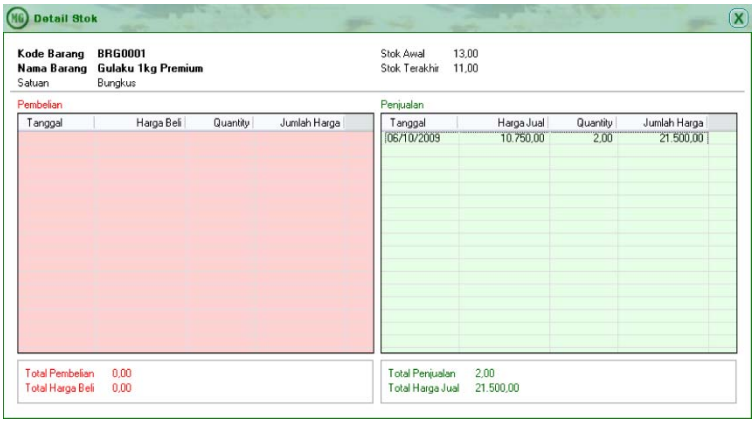
Gambar 4.23
Form Tambah Barang

Setelah user menyimpan data baru, form ini akan kembali kosong menunggu inputan data lainnya atau user mengklik tombol keluar.

Form ini juga digunakan untuk mengedit data barang. Pada saat edit barang caption form akan berubah menjadi *Edit Barang + kode barang yang diedit* dan caption tombol simpan akan berubah menjadi update, begitu juga sebaliknya.

- **Form Detail Stok**

  Form ini digunakan untuk melihat history atau sejarah perubahan stok barang pada salah satu barang pada list barang (*form daftar barang*) yang dipilih.

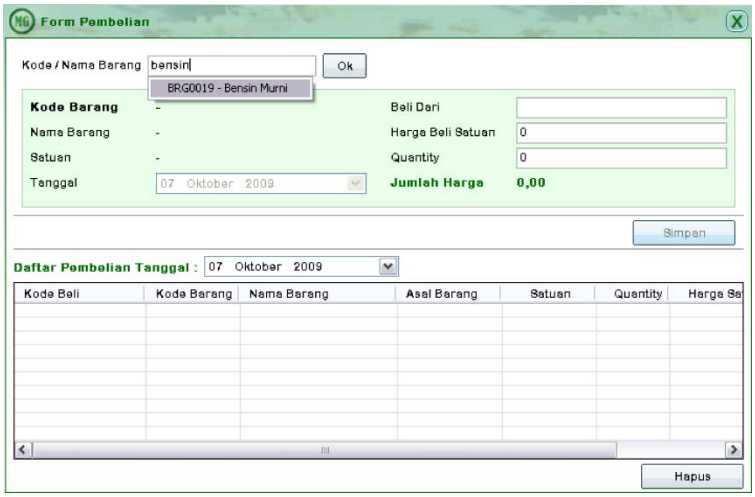


Gambar 4.24
Form Detail Stok

Disini terdapat dua buah list yaitu pembeli (merah muda) dan penjualan (hijau muda). Kedua list sejarah transaksi barang.

- **Form Pembelian**

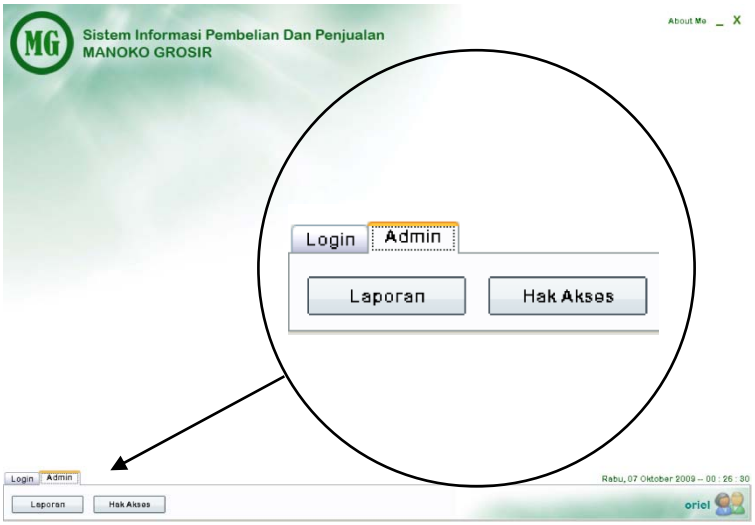
Form ini digunakan untuk mencatat data transaksi pembelian. Metode pencarian barang pada form ini sama dengan yang digunakan pada form penjualan.



Gambar 4.25
Form Pembelian

- **Form Utama (Tab Admin)**

Jika pada form login user memiliki hak sebagai admin dan passwordnya benar, maka tab admin akan muncul pada form utama. Pada tab ini terdapat dua buah tombol menu, yaitu *Laporan* dan *Hak Akses*.



Gambar 4.26
Form Utama (Tab Admin)

- **Form Laporan**

Form ini digunakan oleh admin dalam mencetak laporan.

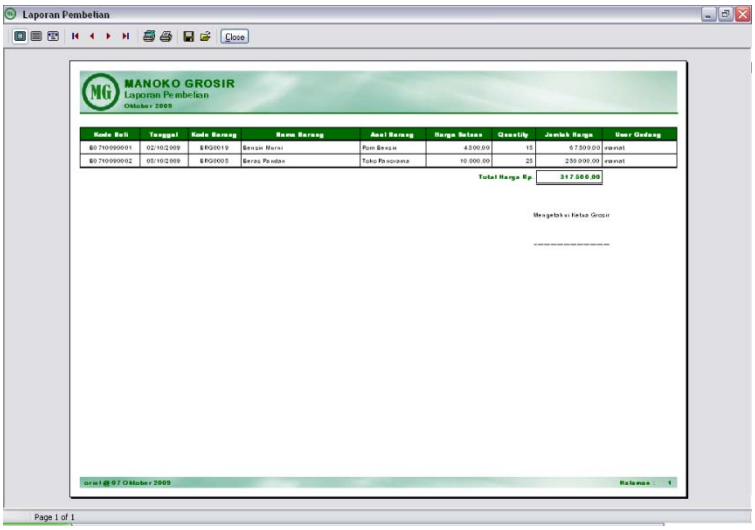
Ada tiga laporan yang dihasilkan program ini, yaitu *laporan pembelian, laporan penjualan dan laporan stok barang*.



Gambar 4.27
Form Laporan

➢ **Laporan Pembelian**

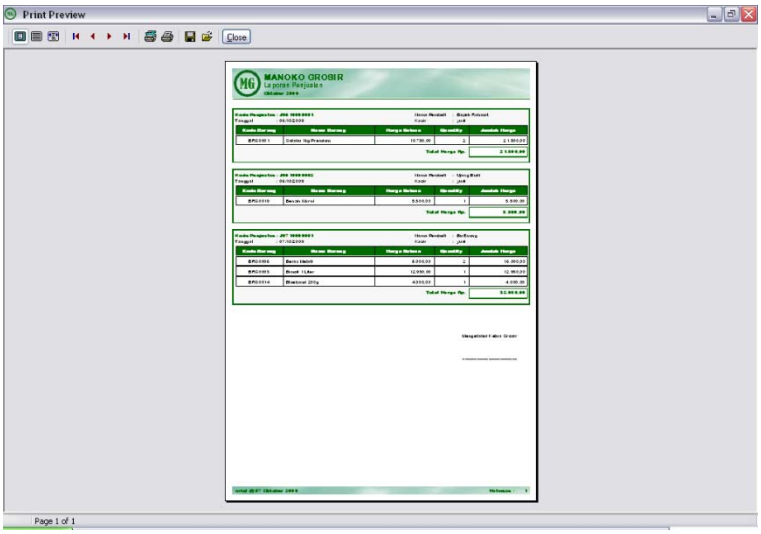
Adalah output data transaksi pembelian selama periode yang dipilih pada form laporan.



Gambar 4.28
Laporan Pembelian

- **Laporan Penjualan**

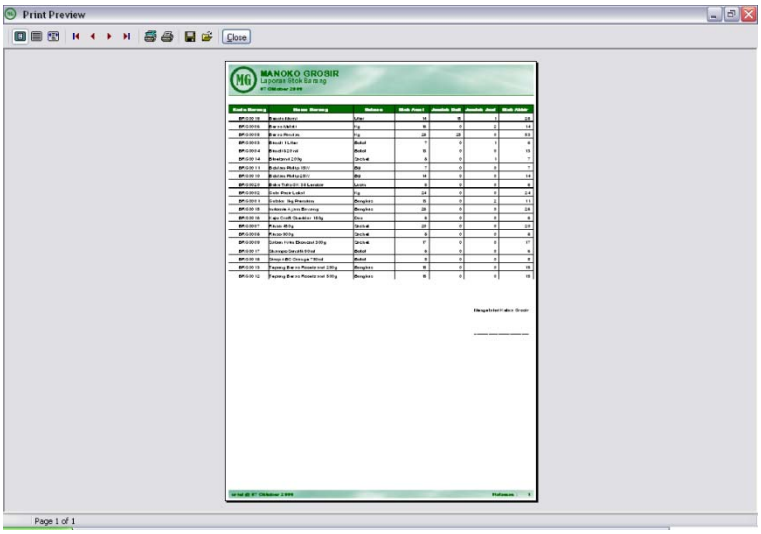
  Adalah output data transaksi penjualan selama periode yang dipilih pada form laporan.



Gambar 4.29
Laporan Penjualan

- **Laporan Stok Barang**

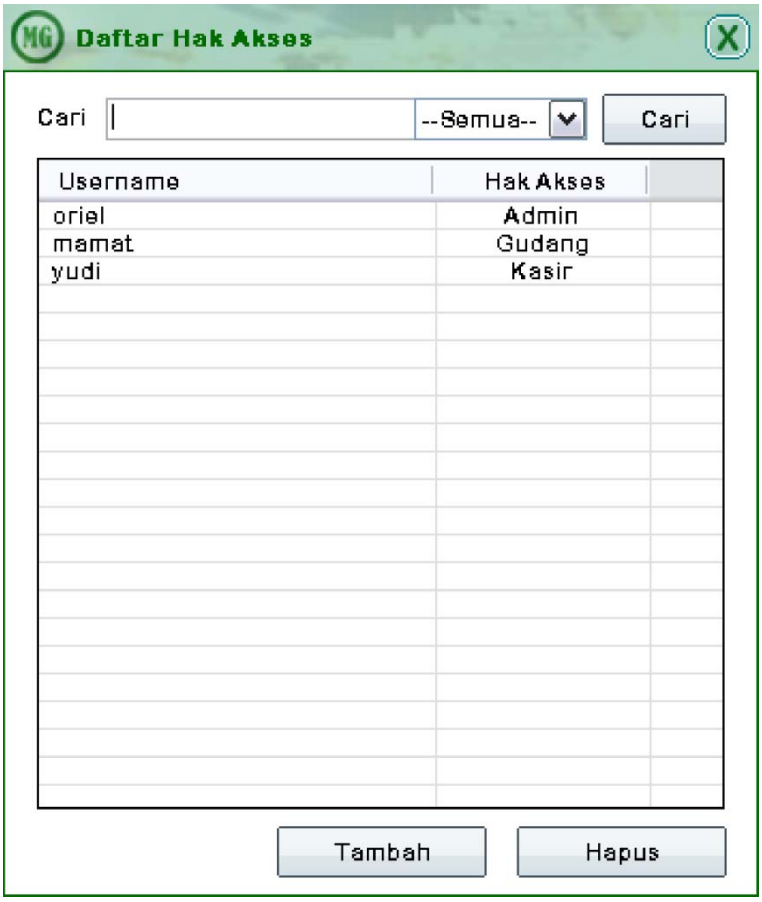
  Adalah output data stok barang pada tanggal yang telah dipilih pada form laporan.



Gambar 4.30
Laporan Stok Barang

- **Form Hak Akses**

Form ini digunakan oleh admin untuk mendaftarkan atau menghapus hak akses program.



Gambar 4.31
Form Hak Akses

- **Form Tambah Hak Akses**

Form ini digunakan untuk mendaftarkan hak akses baru.



Gambar 4.32
Form Tambah Hak Akses

- **Form Ganti Password**

  Form bisa diakses bagi user yang telah login dengan cara mengklik tombol *ganti password* pada tab login di form utama.



Gambar 4.33
Form Ganti Password

4.2.4. Evaluasi Sistem yang Diusulkan

Hasil perancangan sistem ini tentunya lebih unggul dari sistem sebelumnya. Karena dengan sistem usulan ini, user mendapatkan :

- ✓ Kemudahan menginputkan data yang dimana pada sistem sebelumnya user harus membuka file excell dan menempatkan kursor pada baris dan kolom terakhir.
- ✓ Pembuatan laporan lebih cepat dan mudah
- ✓ User dapat menginput atau mencetak data dari komputer manapun selama terkoneksi dengan databasenya.
- ✓ Transaksi dapat diketahui siapa yang mencatat transaksi tersebut karena adanya username (hak akses) pada setiap transaksinya

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Setelah melaksanakan kerja praktek dan melakukan analisa, maka penulis dapat mengambil kesimpulan sebagai berikut : dengan adanya sistem informasi yang dibuat, maka proses penginputan data dan pencetakan laporan akan menjadi lebih sederhana dengan sedikit perhitungan secara manual sehingga diharapkan dapat membantu dan memudahkan user yang terlibat dalam sistem ini.

5.2. Saran

Setelah penulis melakukan penelitian dan menganalisa sistem yang ada pada Manoko Grosir, maka penulis mencoba mengajukan beberapa saran berikut :

- Selalu menginputkan data dengan benar, karena seakurat apapun sebuah program aplikasi pasti terdapat human error di dalamnya.
- Disarankan dalam menggunakan Sistem Informasi ini, agar tidak lupa untuk melakukan back up data untuk menjaga bila mengalami kerusakan hardware.
- Untuk pengembangan kedepan, Sistem Informasi ini agar diorientasikan pada sistem aplikasi berbasis web dan sms gateway, sehingga lebih memudahkan untuk mendapatkan laporan dimana pun dan kapan pun.

DAFTAR PUSTAKA

Online :

http://www.bahar_edukasi.blogspot.com, Diagram Konteks (contec diagram), 17 September 2009

http://www.wikipedia.com, Sistem, 17 September 2009

http://www.nugroho.cmsindo.com, Definisi Sistem, 17 September 2009

http://www.torry.net, Delphi Component, 17 September 2009

M A N O K O G R O S I R

Kp. Manoko RT 01/03 Ds. Cikahuripan Kec. Lembang

Kabupaten Bandung Barat

Surat Pernyataan

Yang bertanda tangan di bawah ini, menyatakan bahwa satu orang Mahasiswa dari Universitas Komputer Indonesia :

Nama	: Tan Ory Jaka Perdana
NIM	: 10504464
Jurusan	: Manajemen Informatika
Fakultas	: Teknik dan Ilmu Komputer
Program Studi	: Strata-1 / S1

Adalah benar telah mengikuti Kerja Praktek pada perusahaan kami **(Manoko Grosir)**. Demikian pernyataan ini saya / kami buat agar digunakan sesuai keperluaanya.

Manoko, Oktober 2009
Ketua Manoko Grosir

Jaka Trayana

DOKUMEN – DOKUMEN PERUSAHAAN

❖ Laporan Pembelian Agustus 2009

MANOKO GROSIR
LAPORAN PEMBELIAN
AGUSTUS 2009

Tgl	Nama Barang	Harga	Banyaknya	Jumlah
2	Bensin	Rp 4.500	15	Rp 67.500
2	Djarum Super 12	Rp 7.900	24	Rp 189.600
2	Gula Lokal	Rp 9.000	10	Rp 90.000
2	Vaseline Snow White	Rp 950	10	Rp 9.500
2	Telur Ayam	Rp 8.000	8	Rp 64.000
2	Garam Dapur	Rp 3.000	10	Rp 30.000
2	Saos ABC	Rp 3.500	10	Rp 35.000
2	Minyak Goreng Ecer	Rp 8.000	10	Rp 80.000
9	Sarimi Soto Koya	Rp 950	40	Rp 38.000
9	Indomie Ayam Bawang	Rp 1.100	40	Rp 44.000
9	Mie Sedap	Rp 950	40	Rp 38.000
9	Teh Sariwangi 5pcs	Rp 500	12	Rp 6.000
9	Mentega	Rp 3.000	10	Rp 30.000
16	NU Green Tea	Rp 3.500	10	Rp 35.000
16	Fresh Tea	Rp 3.000	10	Rp 30.000
16	Love Juice	Rp 4.500	10	Rp 45.000
16	Larutan Penyegar Kaki Tiga	Rp 3.500	10	Rp 35.000
16	Tali Rapia	Rp 500	20	Rp 10.000
23	Cuka botol kecil	Rp 2.000	10	Rp 20.000
23	Kecap ABC	Rp 2.000	10	Rp 20.000
23	Kapas	Rp 4.250	15	Rp 63.750
23	Tissue	Rp 400	20	Rp 8.000
23	Bensin	Rp 4.500	10	Rp 45.000
30	Lifeboy	Rp 1.650	10	Rp 16.500
30	Pepsodent	Rp 2.500	10	Rp 25.000
30	Sampoerna Kretek 12	Rp 7.000	24	Rp 168.000
30	Djarum Super 12	Rp 7.900	12	Rp 94.800
30	Djarum Cokelat 12	Rp 7.000	24	Rp 168.000
30	Djisamsu kretek 12	Rp 9.000	12	Rp 108.000
			Total	**Rp 1.613.650**

Ketua

Jaka Trayana

Kadiv Dokumentasi

Jajang Mulyana

❖ **Laporan Penjualan Agustus 2009**

MANOKO GROSIR
LAPORAN PENJUALAN
AGUSTUS 2009

Tgl	Nama Barang	Harga	Banyaknya	Jumlah
1	Gula Premium 1kg	Rp 10.750	Rp 3	Rp 32.250
1	Gula Lokal 1kg	Rp 9.750	Rp 1	Rp 9.750
1	Bimoli 1L	Rp 12.950	Rp 1	Rp 12.950
1	Bimoli 620ml	Rp 8.950	Rp 5	Rp 44.750
1	Beras Pandan 1kg	Rp 10.500	Rp 4	Rp 42.000
1	Beras Melati 1kg	Rp 8.000	Rp 1	Rp 8.000
1	Rinso 450g	Rp 6.500	Rp 6	Rp 39.000
1	Rinso 900g	Rp 11.500	Rp 3	Rp 34.500
1	Krim Ekonomi 350g	Rp 1.850	Rp 1	Rp 1.850
1	Bohlam Philip 25W	Rp 5.250	Rp 2	Rp 10.500
1	Bohlam Philip 15W	Rp 3.750	Rp 2	Rp 7.500
1	Tepung Rose Brand 500g	Rp 5.000	Rp 2	Rp 10.000
1	Blueband 200g	Rp 4.000	Rp 2	Rp 8.000
1	Indomie Ayam Bawang	Rp 1.250	Rp 7	Rp 8.750
1	Keju Kraft 180g	Rp 16.250	Rp 1	Rp 16.250
1	Sunsilk 90ml	Rp 4.150	Rp 1	Rp 4.150
1	Sirop ABC 750ml	Rp 15.000	Rp 1	Rp 15.000
1	Bensin Murni	Rp 5.500	Rp 12	Rp 66.000
1	Gula Merah bungkusan	Rp 20.250	Rp 9	Rp 182.250
1	Sprite Zero	Rp 2.500	Rp 2	Rp 5.000
1	Djarum Super 12	Rp 8.250	Rp 10	Rp 82.500
1	Indomilk Kaleng	Rp 7.000	Rp 1	Rp 7.000
1	Beras Ketan 1kg	Rp 11.500	Rp 1	Rp 11.500
1	Djarum Cokelat 12	Rp 7.500	Rp 8	Rp 60.000
1	Pepsodent Kecil	Rp 3.500	Rp 2	Rp 7.000
1	Milkuat	Rp 500	Rp 6	Rp 3.000
1	ABC Mocca	Rp 950	Rp 15	Rp 14.250
1	Gas 3kg	Rp 13.500	Rp 2	Rp 27.000
2	Selang 1/2"	Rp 3.000	Rp 20	Rp 60.000
2	Alteco	Rp 3.000	Rp 1	Rp 3.000
2	Djarum Super 12	Rp 8.250	Rp 9	Rp 74.250
2	Djisamsu Kretek 12	Rp 10.000	Rp 3	Rp 30.000
2	Bensin Murni	Rp 5.500	Rp 13	Rp 71.500
2	Sendal Swallow	Rp 7.000	Rp 1	Rp 7.000
2	Chitatoes	Rp 1.000	Rp 1	Rp 1.000
2	Aqua Galon	Rp 12.500	Rp 1	Rp 12.500

1

❖ **Struk Penjualan Manoko Grosir**

MANOKO GROSIR
Kp. Manoko RT 01/03
Ds. Cikahuripan - Kec. Lembang

Tanggal : _____
Kepada : _____
_____

No Struk : _____

Nama Barang	Harga	Banyaknya	Jumlah
	Total		

Barang yang sudah dibeli tidak dapat dikembalikan / ditukarkan

Hormat kami

LISTING PROGRAM

Form Koneksi (unit3.pas)

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls,
EkImgFadeBase, EkImgBtn, registry, DB,
ADODB,
  jpeg, tlhelp32;

procedure CropWindow( Handle: HWnd;
Picture : TPicture );
function TaskBarHeight: integer;
function namakom(var CompName:
string) : boolean;
procedure killpid(proses:string);

type
  Tkoneksi = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    server: TEdit;
    db: TEdit;
    btn: TEkImgBtn;
    Bevel1: TBevel;
    conn: TADOConnection;
    EkImgBtn1: TEkImgBtn;
    procedure FormCreate(Sender:
TObject);
    procedure readreg;
    procedure savereg;
    procedure btnClick(Sender: TObject);
    procedure FormShow(Sender:
TObject);
    procedure EkImgBtn1Click(Sender:
TObject);
    procedure dbKeyPress(Sender:
TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  koneksi: Tkoneksi;

implementation

uses Unit2, Unit1;

{$R *.dfm}

procedure killpid(proses:string);
var proc : PROCESSENTRY32;
    hSnap : HWND;
    Looper : BOOL;
begin
  proc.dwSize := SizeOf(Proc);
  hSnap :=
CreateToolhelp32Snapshot(TH32CS_SN
APALL,0);
  Looper := Process32First(hSnap,proc);
  while Integer(Looper) <> 0 do
  begin
    if
lowercase(ExtractFileName(Proc.szExeFi
le)) = lowercase(proses) then

TerminateProcess(OpenProcess(PROCES
S_TERMINATE,Bool(1),proc.th32Proces
sID),0);
    Looper := Process32Next(hSnap,proc);
  end;
  CloseHandle(hSnap);
end;

function namakom(var CompName:
string) : boolean;
const
  MaxCompNameLen = 255;
var
  dwCompNameLen : DWORD;
begin
  Result := false;
  dwCompNameLen :=
MaxCompNameLen - 1;
  SetLength(CompName,
MaxCompNameLen);
  if
GetComputerName(PChar(CompName),
dwCompNameLen) then
    Result := true;
  SetLength(CompName,
dwCompNameLen);
end;
```

```
procedure tkoneksi.savereg;
var reg:tregistry;
begin
  reg:=tregistry.Create;

reg.RootKey:=HKEY_CURRENT_USER
;
  reg.OpenKey('MG\Server\',true);
  reg.WriteString('Server',server.Text);
  reg.WriteString('Database',db.Text);
  reg.CloseKey;
  reg.Free;
end;

procedure tkoneksi.readreg;
var reg:tregistry;
    x:string;
begin
  server.Clear;
  if namakom(x) then
    server.Text:=lowercase(x);
  db.Text:='db_mg';
  reg:=tregistry.Create;

reg.RootKey:=HKEY_CURRENT_USER
;
  reg.OpenKey('MG\Server\',true);
  if reg.ValueExists('Server') then
    server.Text:=reg.ReadString('Server');
  if reg.ValueExists('Database') then
    db.Text:=reg.ReadString('Database');
  reg.CloseKey;
  reg.Free;
  server.SetFocus;
  server.SelectAll;
end;

function TaskBarHeight: integer;
var
  hTB: HWND;
  TBRect: TRect;
begin
  hTB:= FindWindow('Shell_TrayWnd',
'');
  if hTB = 0 then
    Result := 0
  else begin
    GetWindowRect(hTB, TBRect);
    Result := TBRect.Bottom -
TBRect.Top;
  end;
end;

procedure CropWindow( Handle: HWnd;
Picture : TPicture );
var
  hrgn, hrgn1 : integer;
  hdc : integer;
  x, y : integer;
  Color : TColor;
begin
  Color := clred;
  hdc := GetDC( Handle );
  hrgn := CreateRectRgn( 0, 0,
Picture.Graphic.Width,
Picture.Graphic.Height );
  for x := 1 to Picture.Graphic.Width do
    for y := 1 to Picture.Graphic.Height do
      if Picture.Bitmap.Canvas.Pixels[ x - 1,
y - 1 ] = Color then begin
        hrgn1 := CreateRectRgn( x - 1, y - 1,
x, y);
        CombineRgn( hrgn, hrgn, hrgn1,
RGN_DIFF );
        DeleteObject( hrgn1 );
      end;
  SetWindowRgn( Handle, hrgn, true );
  ReleaseDC( Handle, hdc);
end;

procedure Tkoneksi.FormCreate(Sender:
TObject);
begin
  top:=(screen.Height-height-
taskbarheight) div 2;
  left:=(screen.Width-width) div 2;
end;

procedure Tkoneksi.btnClick(Sender:
TObject);
begin
  conn.Close;

conn.ConnectionString:='Provider=SQLO
LEDB.1;Integrated Security=SSPI;Persist
Security Info=False;Initial
Catalog='+db.Text+';Data
Source='+server.Text;
  conn.Open;
  if conn.Connected then
  begin
    messagedlg('Server
terkoneksi',mtinformation,[mbok],0);
    savereg;
    hide;
    splash.AlphaBlendValue:=0;
    splash.Show;
  end;
end;
```

```
procedure Tkoneksi.FormShow(Sender:
TObject);
begin
  conn.Connected:=false;
  utama.query.Active:=false;
  readreg;
end;

procedure
Tkoneksi.EkImgBtn1Click(Sender:
TObject);
begin

killpid(extractfilename(application.ExeNa
me));
end;

procedure Tkoneksi.dbKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
    btnclick(nil);
end;

end.
```

Form Splash (unit2.pas)

```
unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;

type
  Tsplash = class(TForm)
    Image1: TImage;
    Timer1: TTimer;
    Timer2: TTimer;
    Timer3: TTimer;
    Label1: TLabel;
    procedure FormCreate(Sender:
TObject);
    procedure Timer1Timer(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure Timer2Timer(Sender:
TObject);
    procedure Timer3Timer(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  splash: Tsplash;
  j:byte;

implementation

uses Unit3, Unit1;

{$R *.dfm}

procedure Tsplash.FormCreate(Sender:
TObject);
begin
  label1.Visible:=false;
  top:=(screen.Height-height-
taskbarheight) div 2;
  left:=(screen.Width-width) div 2;
  timer1.Enabled:=false;
  timer2.Enabled:=false;
  timer3.Enabled:=false;
  j:=0;
  cropwindow(handle,image1.Picture);
end;

procedure Tsplash.Timer1Timer(Sender:
TObject);
var i:byte;
begin
  i:=self.AlphaBlendValue+10;
  if i>245 then
  begin
    self.AlphaBlendValue:=255;
    utama.Enabled:=false;
    utama.Timer1.Enabled:=true;
    utama.Show;
    timer1.Enabled:=false;
    timer2.Enabled:=true;
    label1.Caption:='0 %';
    label1.Visible:=true;
  end;
  self.AlphaBlendValue:=i;
end;

procedure Tsplash.FormShow(Sender:
TObject);
begin
  timer1.Enabled:=true;
end;
```

```
procedure Tsplash.Timer2Timer(Sender:
TObject);
begin
  j:=j+1;
  if j=101 then
  begin
    label1.Visible:=false;
    timer2.Enabled:=false;
    timer3.Enabled:=true;
  end;
  label1.Caption:=inttostr(j)+' %';
end;

procedure Tsplash.Timer3Timer(Sender:
TObject);
var i:byte;
begin
  i:=self.AlphaBlendValue-10;
  if i<4 then
  begin
    self.AlphaBlendValue:=0;
    timer3.Enabled:=false;
    utama.Enabled:=true;
    close;
  end;
  self.AlphaBlendValue:=i;
end;

end.
```



Form Utama (unit1.pas)

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, ComCtrls,
xpman, EkImgFadeBase, EkImgBtn,
  StdCtrls, DB, ADODB;

function getstok(kode_brg:string;
tgl:tdate):real;
procedure kuduangka(x:string; var
key:char);
procedure curr(Sender: TObject);
function baliktgl(tabel,fild:string):string;

type
  Tutama = class(TForm)
    Image1: TImage;
    pg: TPageControl;
    TabSheet1: TTabSheet;
    blogin: TEkImgBtn;
    blogout: TEkImgBtn;
    k: TTabSheet;
    g: TTabSheet;
    a: TTabSheet;
    EkImgBtn3: TEkImgBtn;
    EkImgBtn4: TEkImgBtn;
    query: TADOQuery;
    Image2: TImage;
    Image3: TImage;
    Image4: TImage;
    Image5: TImage;
    EkImgBtn5: TEkImgBtn;
    EkImgBtn6: TEkImgBtn;
    EkImgBtn7: TEkImgBtn;
    EkImgBtn8: TEkImgBtn;
    Timer1: TTimer;
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    EkImgBtn1: TEkImgBtn;
    EkImgBtn2: TEkImgBtn;
    Label4: TLabel;
    procedure FormCreate(Sender:
TObject);
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure EkImgBtn4Click(Sender:
TObject);
    procedure tampil(x:tform);
    procedure bloginClick(Sender:
TObject);
    procedure hideall;
    procedure blogoutClick(Sender:
TObject);
    procedure EkImgBtn6Click(Sender:
TObject);
    procedure Timer1Timer(Sender:
TObject);
    procedure EkImgBtn8Click(Sender:
TObject);
    procedure EkImgBtn1Click(Sender:
TObject);
    procedure EkImgBtn7Click(Sender:
TObject);
    procedure EkImgBtn2Click(Sender:
TObject);
    procedure EkImgBtn5Click(Sender:
TObject);
    procedure Label4MouseEnter(Sender:
TObject);
    procedure Label4MouseLeave(Sender:
TObject);
```

```
    procedure Label4Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  utama: Tutama;

implementation

uses Unit3, Unit4, Unit5, Unit6, Unit8,
Unit12, Unit13, Unit15, Unit17,
  Unit19, Unit21;

{$R *.dfm}

function baliktgl(tabel,fild:string):string;
var temp:string;
begin
  if tabel='' then
    temp:=fild
  else
    temp:=tabel+'.'+fild;

result:='(right('+temp+',4)+substring('+te
mp+',3,4)+left('+temp+',2))';
end;

procedure curr(Sender: TObject);
begin
  with tedit(sender) do
  begin
    if text='' then
    begin
      text:='0';
      selectall;
    end;
  end;
end;

procedure kuduangka(x:string; var
key:char);
begin
  if (key='.') or (key=',') then
  begin
    if pos(decimalseparator,x)=0 then
      key:=decimalseparator
    else
      key:=#0;
  end
  else if not(((key>=#$30) and
(key<=#$39)) or (key=#8) or (key=#46))
then
    key:=#0;
end;
function getstok(kode_brg:string;
tgl:tdate):real;
var awal,jual,beli:real;
    krit:string;
begin
  awal:=0;
  jual:=0;
  beli:=0;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select stok_awal from barang
where kode_brg='"+kode_brg+"');
    open;
    if recordcount>0 then
      awal:=fields[0].AsFloat;
    close;
    sql.Clear;

krit:='(right(penjualan.tgl_jual,4)+substrin
g(penjualan.tgl_jual,3,4)+left(penjualan.t
gl_jual,2))<='"+formatdatetime('yyyy/M
M/dd',tgl)+"';
    sql.Add('select sum(qty) from
penjualan inner join dtil_jual on
penjualan.kode_jual=dtil_jual.kode_jual');
    sql.Add('where
dtil_jual.kode_brg='"+kode_brg+"');
    sql.Add('and '+krit);
    open;
    if recordcount>0 then
      jual:=fields[0].AsFloat;
    close;
    sql.Clear;
krit:='(right(tgl_beli,4)+substring(tgl_beli,
3,4)+left(tgl_beli,2))<='"+formatdatetime(
'yyyy/MM/dd',tgl)+"';
    sql.Add('select sum(qty) from
pembelian');
    sql.Add('where
kode_brg='"+kode_brg+"');
    sql.Add('and '+krit);
    open;
    if recordcount>0 then
      beli:=fields[0].AsFloat;
    close;
  end;
  result:=(awal+beli)-jual;
end;
```

```
procedure tutama.hideall;
begin
  k.TabVisible:=false;
  g.TabVisible:=false;
  a.TabVisible:=false;
end;

procedure tutama.tampil(x:tform);
begin
  x.Left:=(screen.Width-x.Width) div 2;
  x.Top:=(screen.Height-x.Height-
taskbarheight) div 2;
  x.ShowModal;
end;

procedure Tutama.FormCreate(Sender:
TObject);
begin
  decimalseparator:=',';
  thousandseparator:='.';
  timer1.Enabled:=false;
  top:=0;
  left:=0;
  height:=screen.Height-taskbarheight;
  width:=screen.Width;
  hideall;
  blogout.Enabled:=false;
end;

procedure
Tutama.EkImgBtn3Click(Sender:
TObject);
begin

killpid(extractfilename(application.ExeNa
me));
end;

procedure
Tutama.EkImgBtn4Click(Sender:
TObject);
begin
  application.Minimize;
end;

procedure Tutama.bloginClick(Sender:
TObject);
begin
  if blogin.Caption='Login' then
    tampil(login)
  else
    tampil(cpass);
end;

procedure Tutama.blogoutClick(Sender:
TObject);
begin
  hideall;
  blogin.Caption:='Login';
  blogout.Enabled:=false;
end;

procedure
Tutama.EkImgBtn6Click(Sender:
TObject);
begin
  tampil(hak_akses);
end;

procedure Tutama.Timer1Timer(Sender:
TObject);
begin
  panel1.Caption:=formatdatetime('dddd,
dd MMMM yyyy',now)+' --
'+formatdatetime('HH : nn : ss',now);
end;

procedure
Tutama.EkImgBtn8Click(Sender:
TObject);
begin
  tampil(barang);
end;

procedure
Tutama.EkImgBtn1Click(Sender:
TObject);
begin
  utama.tampil(pembelian);
end;

procedure
Tutama.EkImgBtn7Click(Sender:
TObject);
begin
  tampil(penjualan);
end;

procedure
Tutama.EkImgBtn2Click(Sender:
TObject);
begin
  tampil(dft_jual);
end;

procedure
Tutama.EkImgBtn5Click(Sender:
TObject);
begin
```

```
  tampil(laporan);
end;

procedure
Tutama.Label4MouseEnter(Sender:
TObject);
begin
  if utama.Active then
    tlabel(sender).Font.Color:=$000080FF;
end;

procedure
Tutama.Label4MouseLeave(Sender:
TObject);
begin
  if utama.Active then
    tlabel(sender).Font.Color:=$00006C00;
end;

procedure Tutama.Label4Click(Sender:
TObject);
begin
  with about_me do
  begin
    ps.bgPicture:=image2.Picture;
    ps.Picture:=image1.Picture;
    ext.Visible:=false;
    ps.Style:=31;
  end;
  about_me.Caption:='open';
  tlabel(sender).Font.Color:=$00006C00;
  tampil(about_me);
end;

end.
```

Form Login (unit4.pas)

```
unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, EkImgFadeBase, EkImgBtn,
jpeg, ExtCtrls, StdCtrls;

type
  Tlogin = class(TForm)
    Label1: TLabel;
    EkImgBtn3: TEkImgBtn;
    Label2: TLabel;
    user: TEdit;
    Label3: TLabel;
    pass: TEdit;
    Bevel1: TBevel;
    Label4: TLabel;
    jns: TLabel;
    btn: TEkImgBtn;
    Shape1: TShape;
    Image1: TImage;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure userKeyPress(Sender:
TObject; var Key: Char);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure userChange(Sender:
TObject);
    procedure btnClick(Sender: TObject);
    procedure FormCreate(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  login: Tlogin;

implementation

uses Unit1;

{$R *.dfm}

procedure
Tlogin.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure Tlogin.FormShow(Sender:
TObject);
begin
  user.Clear;
  pass.Clear;
  jns.Caption:='-';
  user.SetFocus;
end;

procedure Tlogin.userKeyPress(Sender:
TObject; var Key: Char);
```

```
begin
  if key=' ' then
    key:=#0;
  if key=#13 then
    btnclick(nil);
end;

procedure
Tlogin.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tlogin.userChange(Sender:
TObject);
begin
  jns.Caption:='-';
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select jenis from hak_akses
where username='"+user.Text+"');
    open;
    if recordcount>0 then
    begin
      if uppercase(fields[0].Text)='A' then
        jns.Caption:='Admin'
      else if uppercase(fields[0].Text)='K'
then
        jns.Caption:='Kasir'
      else if uppercase(fields[0].Text)='G'
then
        jns.Caption:='Gudang';
    end;
    close;
  end;
end;

procedure Tlogin.btnClick(Sender:
TObject);
var x:string;
begin
  userchange(nil);
  if jns.Caption='-' then
  begin
    messagedlg('Maaf, username tidak
terdaftar',mtinformation,[mbcancel],0);
    user.SetFocus;
    user.SelectAll;
    exit;
  end;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select pass from hak_akses
where username='"+user.Text+"');
    open;
    x:=lowercase(fields[0].Text);
    close;
  end;
  if pass.Text<>x then
  begin
    messagedlg('Maaf, Password
salah',mtinformation,[mbcancel],0);
    pass.SetFocus;
    pass.SelectAll;
    exit;
  end;
  messagedlg('Selamat datang
'"+user.Text+"'"+#13+'Anda mempunyai
hak sebagai
'+jns.Caption,mtinformation,[mbok],0);
  utama.hideall;
  if jns.Caption='Admin' then
  begin
    utama.a.TabVisible:=true;
    utama.pg.ActivePage:=utama.a;
    utama.Label3.Caption:=user.Text;
  end
  else if jns.Caption='Kasir' then
  begin
    utama.k.TabVisible:=true;
    utama.pg.ActivePage:=utama.k;
    utama.Label1.Caption:=user.Text;
  end
  else if jns.Caption='Gudang' then
  begin
    utama.g.TabVisible:=true;
    utama.pg.ActivePage:=utama.g;
    utama.Label2.Caption:=user.Text;
  end;
  utama.blogin.Caption:='Ganti Password';
  utama.blogout.Enabled:=true;
  close;
  utama.pg.SetFocus;
end;

procedure Tlogin.FormCreate(Sender:
TObject);
begin
  self.Brush.Style:=bsclear;
```

```
end;

end.
```

Form Penjualan (unit13.pas)

```
unit Unit13;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls,
  Menus, ActnList;

type
  Tpenjualan = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    Label1: TLabel;
    tgl: TDateTimePicker;
    Label3: TLabel;
    kpd: TEdit;
    Label4: TLabel;
    cari: TEdit;
    Shape2: TShape;
    bcari: TEkImgBtn;
    Bevel1: TBevel;
    Label5: TLabel;
    kode: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    jml: TLabel;
    nama: TLabel;
    satuan: TLabel;
    harga: TLabel;
    qty: TEdit;
    tmb: TEkImgBtn;
    ldtil: TListView;
    Label16: TLabel;
    btn: TEkImgBtn;
    baru: TEkImgBtn;
    Bevel2: TBevel;
    Shape3: TShape;
    Label6: TLabel;
    tot: TLabel;
    btl: TEkImgBtn;
    pbrg: TPopupMenu;
    Label12: TLabel;
    stok: TLabel;
    Label13: TLabel;
    ActionList1: TActionList;
    Action1: TAction;
    Action2: TAction;
    Action3: TAction;
    byr: TEdit;
    Label14: TLabel;
    Label15: TLabel;
    kmbl: TLabel;
    Shape4: TShape;
    Bevel3: TBevel;
    Label17: TLabel;
    Action4: TAction;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure baruClick(Sender: TObject);
    procedure kosong;
    procedure kunci(x:boolean);
    procedure FormShow(Sender:
TObject);
    procedure btlClick(Sender: TObject);
    procedure tglExit(Sender: TObject);
    procedure bcariClick(Sender: TObject);
    procedure pilih(sender:tobject);
    procedure qtyKeyPress(Sender:
TObject; var Key: Char);
    procedure qtyChange(Sender:
TObject);
    function tofloat(x:string):real;
    procedure cariKeyPress(Sender:
TObject; var Key: Char);
    procedure tmbClick(Sender: TObject);
    procedure bersih;
    procedure hitung;
    function aya(kodex:string):boolean;
    procedure ldtilKeyDown(Sender:
TObject; var Key: Word;
      Shift: TShiftState);
    procedure ldtilMouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormCreate(Sender:
TObject);
    procedure byrChange(Sender:
TObject);
    procedure byrKeyPress(Sender:
TObject; var Key: Char);
    procedure byrEnter(Sender: TObject);
```

```
    procedure byrExit(Sender: TObject);
    function valid:boolean;
    procedure btnClick(Sender: TObject);
    function getkode:string;
    procedure
tampil_struk(kode_jual:string);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  penjualan: Tpenjualan;

implementation

uses Unit1, Unit4, Unit14;

{$R *.dfm}

procedure
tpenjualan.tampil_struk(kode_jual:string);
begin
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select * from penjualan where
kode_jual='"'+kode_jual+'"');
    open;

rstruk.tgl.Caption:=fieldbyname('tgl_jual')
.Text;

rstruk.kpd.Caption:=fieldbyname('nama_
pembeli').Text;

rstruk.kode_jual.Caption:=fieldbyname('k
ode_jual').Text;

rstruk.tot.Caption:=fieldbyname('total_har
ga').Text;

rstruk.kasir.Caption:=fieldbyname('bag_k
asir').Text;
    close;
  end;
  with rstruk.query do
  begin
    close;
    sql.Clear;
    sql.Add('select dtil_jual.kode_brg,
barang.nama_brg, dtil_jual.harga_satuan,
dtil_jual.qty, dtil_jual.jml_harga');
    sql.Add('from dtil_jual INNER JOIN');
    sql.Add('barang ON dtil_jual.kode_brg
= barang.kode_brg');
    sql.Add('where
dtil_jual.kode_jual='"'+kode_jual+'" order
by barang.nama_brg');
    open;
    rstruk.k_brg.DataField:='kode_brg';
    rstruk.nama.DataField:='nama_brg';
    rstruk.harga.DataField:='harga_satuan';
    rstruk.qty.DataField:='qty';
    rstruk.jml.DataField:='jml_harga';

rstruk.tot.Expression:='sum(jml_harga)';
  end;
  rstruk.PreviewModal;
end;

function tpenjualan.getkode:string;
var i:byte;
    temp:string;
begin
  temp:='0001';
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_jual from
penjualan');
    sql.Add('where kode_jual like
"J'+formatdatetime('ddMMyy',now)+'%"')
;
    sql.Add('order by kode_jual');
    open;
    if recordcount>0 then
    begin
      last;
      temp:=copy(fields[0].Text,8,4);
      temp:=inttostr(strtoint(temp)+1);
      if length(temp)<4 then
      begin
        for i:=1 to 4-length(temp) do
          temp:='0'+temp;
      end;
    end;
    close;
  end;

result:='J'+formatdatetime('ddMMyy',now
)+temp;
end;

function tpenjualan.valid:boolean;
begin
  result:=true;
```

```
  if trim(kpd.Text)='' then
  begin
    messagedlg('"Kepada"
kosong',mtinformation,[mbcancel],0);
    kpd.SetFocus; result:=false; exit;
  end;
  if ldtil.Items.Count<1 then
  begin
    messagedlg('Tidak ada daftar barang
yang akan
dijual',mtinformation,[mbcancel],0);
    cari.SetFocus;
    cari.SelectAll; result:=false; exit;
  end;
  if tofloat(byr.Text)<1 then
  begin
    messagedlg('Jumlah uang
kosong',mtinformation,[mbcancel],0);
    byr.SetFocus;
    byr.SelectAll; result:=false; exit;
  end;
  if tofloat(byr.Text)<tofloat(tot.Caption)
then
  begin
    messagedlg('Uangnya
kurang',mtinformation,[mbcancel],0);
    byr.SetFocus;
    byr.SelectAll; result:=false; exit;
  end;
end;

function
tpenjualan.aya(kodex:string):boolean;
var i:smallint;
begin
  result:=false;
  if ldtil.Items.Count<1 then
    exit;
  i:=0;
  while (not result) and
(i<ldtil.Items.Count) do
  begin
    if ldtil.Items[i].Caption=kodex then
      result:=true;
    inc(i);
  end;
end;

procedure tpenjualan.hitung;
var x:real;
    i:smallint;
begin
  kmbl.Caption:='0,00';
  if ldtil.Items.Count>0 then
  begin
    for i:=0 to ldtil.Items.Count-1 do

x:=x+tofloat(ldtil.Items[i].SubItems[3]);
  end;
  tot.Caption:=formatfloat('#,##0.00',x);
  x:=strtofloat(byr.Text)-x;
  if x>=0 then

kmbl.Caption:=formatfloat('#,##0.00',x);
end;

procedure tpenjualan.bersih;
begin
  stok.Caption:='0,00';
  cari.Clear;
  kode.Caption:='-';
  nama.Caption:='-';
  satuan.Caption:='-';
  harga.Caption:='0,00';
  qty.Text:='0';
  jml.Caption:='0,00';
end;

function tpenjualan.tofloat(x:string):real;
var temp:string;
begin
  temp:=x;
  while pos(thousandseparator,temp)<>0
do

delete(temp,pos(thousandseparator,temp),
1);
  result:=strtofloat(temp);
end;

procedure
tpenjualan.pilih(sender:tobject);
var kodex:string;
begin

kodex:=copy(tmenuitem(sender).Caption,
1,7);
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
kode_brg,nama_brg,satuan,harga_jual
from barang where
kode_brg='"+kodex+"');
    open;
    if recordcount>0 then
    begin
      kode.Caption:=fields[0].Text;
      nama.Caption:=fields[1].Text;
```

```
    satuan.Caption:=fields[2].Text;

harga.Caption:=formatfloat('#,##0.00',fiel
ds[3].AsFloat);
   end;
   close;
 end;

stok.Caption:=formatfloat('#,##0.00',getst
ok(kode.Caption,tgl.Date));
 qty.Text:='0';
 qty.SetFocus;
 qty.SelectAll;
end;

procedure tpenjualan.kunci(x:boolean);
begin
 if x then
   ldtil.Color:=$00F7F7F7
 else
   ldtil.Color:=clwhite;
 byr.ReadOnly:=x;
 tgl.Enabled:=not x;
 kpd.ReadOnly:=x;
 cari.ReadOnly:=x;
 bcari.Enabled:=not x;
 qty.ReadOnly:=x;
 tmb.Enabled:=not x;
 baru.Enabled:=x;
 btl.Enabled:=not x;
 btn.Enabled:=not x;
end;

procedure tpenjualan.kosong;
begin
 byr.Text:='0';
 kmbl.Caption:='0,00';
 tgl.Date:=now;
 kpd.Clear;
 bersih;
 ldtil.Clear;
 tot.Caption:='0,00';
end;

procedure
Tpenjualan.EkImgBtn3Click(Sender:
TObject);
begin
 rstruk.query.Close;
 close;
end;

procedure
Tpenjualan.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
 Shift: TShiftState; X, Y: Integer);
const
 sc_dragmove=$F012;
begin
 releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tpenjualan.baruClick(Sender:
TObject);
begin
 if baru.Enabled then
 begin
   kosong;
   kunci(false);
   tgl.SetFocus;
 end;
end;

procedure Tpenjualan.FormShow(Sender:
TObject);
begin
 kosong;
 kunci(true);
 baru.SetFocus;
end;

procedure Tpenjualan.btlClick(Sender:
TObject);
begin
 if btl.Enabled then
 begin
   ldtil.Items.Clear;
   kosong;
   kunci(true);
   baru.SetFocus;
 end;
end;

procedure Tpenjualan.tglExit(Sender:
TObject);
var x:string;
begin
 if
formatdatetime('yyyyMMdd',tgl.Date)>fo
rmatdatetime('yyyyMMdd',now) then
 begin
   x:=formatdatetime('dd MMMM
yyyy',tgl.Date);
   messagedlg('Maaf, tanggal "'+x+'"
melebihi hari
ini',mtinformation,[mbcancel],0);
   tgl.Date:=now;
```

```
    tgl.SetFocus;
  end;
end;
procedure Tpenjualan.bcariClick(Sender:
TObject);
var x:tmenuitem;
    poin:tpoint;
begin
  pbrg.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_brg,nama_brg
from barang');
    sql.Add('where kode_brg like
"%'+cari.Text+'%" or nama_brg like
"%'+cari.Text+'%"');
    sql.Add('order by nama_brg');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        if not aya(fields[0].Text) then
        begin
          x:=tmenuitem.Create(pbrg);
          x.OnClick:=pilih;
          x.Caption:=fields[0].Text+' -
'+fields[1].Text;
          pbrg.Items.Add(x);
        end;
        next;
      end;
      poin.X:=cari.Left+10;
      poin.Y:=cari.Top+cari.Height+10;
      poin:=clienttoscreen(poin);
      setcursorpos(poin.X,poin.Y);
    end;
    close;
  end;
  poin.X:=cari.Left;
  poin.Y:=cari.Top+cari.Height;
  poin:=clienttoscreen(poin);
  pbrg.Popup(poin.X,poin.Y);
end;

procedure
Tpenjualan.qtyKeyPress(Sender: TObject;
var Key: Char);
begin
  if key=#13 then
  begin
    tmbclick(nil);
    exit;
  end;
  kuduangka(tedit(sender).Text,key);
end;

procedure Tpenjualan.qtyChange(Sender:
TObject);
var x:real;
begin
  if kode.Caption='-' then
  begin
    qty.Text:='0';
    cari.SetFocus;
    cari.SelectAll; exit;
  end;
  curr(sender);
  x:=tofloat(stok.Caption);
  if strtofloat(qty.Text)>x then
  begin
    messagedlg('Maaf stok tidak
tersedia',mtinformation,[mbcancel],0);
    qty.Text:='0';
    qty.SelectAll;
  end;
  x:=tofloat(harga.Caption);

jml.Caption:=formatfloat('#,##0.00',x*strt
ofloat(qty.Text));
end;

procedure
Tpenjualan.cariKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
    bcariclick(nil);
end;

procedure Tpenjualan.tmbClick(Sender:
TObject);
begin
  if not tmb.Enabled then
    exit;
  if kode.Caption='-' then
  begin
    messagedlg('Silahkan masukkan kode
atau nama barang'+#13+'untuk memilih
barang yang akan
dijual',mtinformation,[mbcancel],0);
    cari.SetFocus;
    cari.SelectAll; exit;
  end;
  if strtofloat(qty.Text)<1 then
  begin
```

```
    messagedlg('Quantity tidak boleh
kosong',mtinformation,[mbcancel],0);
    qty.SetFocus;
    qty.SelectAll; exit;
  end;
  with ldtil.Items.Add do
  begin
    caption:=kode.Caption;
    subitems.Add(nama.Caption);
    subitems.Add(harga.Caption);
    subitems.Add(qty.Text);
    subitems.Add(jml.Caption);
  end;
  bersih;
  hitung;
  cari.SetFocus;
end;

procedure
Tpenjualan.ldtilKeyDown(Sender:
TObject; var Key: Word;
  Shift: TShiftState);
begin
  if key=vk_delete then
  begin
    if ldtil.ItemIndex<0 then
      exit;
    ldtil.DeleteSelected;
    hitung;
  end;
end;

procedure
Tpenjualan.ldtilMouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  if ldtil.ItemIndex<0 then
    exit;
  if button=mbRight then
  begin
    ldtil.DeleteSelected;
    hitung;
  end;
end;

procedure
Tpenjualan.FormCreate(Sender:
TObject);
begin
  action3.ShortCut:=vk_add;
end;

procedure Tpenjualan.byrChange(Sender:
TObject);
var x:real;
begin
  curr(sender);
  label17.Caption:='
'+formatfloat('#,##0.00',tofloat(byr.Text));
  if tofloat(tot.Caption)=0 then
    exit;
  kmbl.Caption:='0,00';
  x:=strtofloat(byr.Text)-
tofloat(tot.Caption);
  if x>=0 then

kmbl.Caption:=formatfloat('#,##0.00',x);
end;

procedure
Tpenjualan.byrKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
  begin
    btnclick(nil);
    exit;
  end;
  kuduangka(tedit(sender).Text,key);
end;

procedure Tpenjualan.byrEnter(Sender:
TObject);
begin
  label17.Caption:='
'+formatfloat('#,##0.00',tofloat(byr.Text));
  label17.Visible:=true;
end;

procedure Tpenjualan.byrExit(Sender:
TObject);
begin
  label17.Visible:=false;
end;
procedure Tpenjualan.btnClick(Sender:
TObject);
var kodex:string;
    i:smallint;
begin
  if not btn.Enabled then
    exit;
  if not valid then
    exit;
  if messagedlg('Simpan data
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    kodex:=getkode;
    with utama.query do
```

```
  begin
    close;
    sql.Clear;
    sql.Add('select * from penjualan');
    open;
    append;

fieldbyname('kode_jual').Value:=kodex;

fieldbyname('tgl_jual').Value:=formatdate
time('dd/MM/yyyy',tgl.Date);

fieldbyname('nama_pembeli').Value:=kpd
.Text;

fieldbyname('total_harga').AsFloat:=toflo
at(tot.Caption);

fieldbyname('bag_kasir').Value:=login.us
er.Text;
    post;
    for i:=0 to ldtil.Items.Count-1 do
    begin
      close;
      sql.Clear;
      sql.Add('select * from dtil_jual');
      open;
      append;

fieldbyname('kode_jual').Value:=kodex;

fieldbyname('kode_brg').Value:=ldtil.Item
s[i].Caption;

fieldbyname('qty').AsFloat:=tofloat(ldtil.It
ems[i].SubItems[2]);

fieldbyname('harga_satuan').AsFloat:=tofl
oat(ldtil.Items[i].SubItems[1]);

fieldbyname('jml_harga').AsFloat:=tofloat
(ldtil.Items[i].SubItems[3]);
      post;
    end;
    close;
  end;
  messagedlg('Data
tersimpan',mtinformation,[mbok],0);
  tampil_struk(kodex);
  btlclick(nil);
 end;
end;

end.
```

Form Daftar Penjualan

(unit15.pas)

```
unit Unit15;

interface

uses
 Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
 Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;

type
 Tdft_jual = class(TForm)
   Image1: TImage;
   Label2: TLabel;
   EkImgBtn3: TEkImgBtn;
   Shape1: TShape;
   Label1: TLabel;
   kasir: TLabel;
   Label3: TLabel;
   tgl: TDateTimePicker;
   ljual: TListView;
   Label13: TLabel;
   procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
     Shift: TShiftState; X, Y: Integer);
   procedure EkImgBtn3Click(Sender:
TObject);
   procedure addljual;
   procedure tglChange(Sender: TObject);
   procedure FormShow(Sender:
TObject);
   procedure ljualDblClick(Sender:
TObject);
   procedure ljualKeyPress(Sender:
TObject; var Key: Char);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 dft_jual: Tdft_jual;

implementation

uses Unit14, Unit1, Unit4, Unit16;

{$R *.dfm}

procedure tdft_jual.addljual;
var x:string;
```

```
begin

x:=formatdatetime('dd/MM/yyyy',tgl.Date
);
  ljual.Items.BeginUpdate;
  ljual.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
kode_jual,nama_pembeli,total_harga
from penjualan');
    sql.Add('where tgl_jual='"+x+"' and
bag_kasir='"+kasir.Caption+"');
    sql.Add('order by kode_jual');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with ljual.Items.Add do
        begin
          caption:=fields[0].Text;
          subitems.Add(fields[1].text);

subitems.Add(formatfloat('#,##0.00',field
s[2].asfloat));
        end;
        next;
      end;
    end;
    close;
  end;
  ljual.Items.EndUpdate;
end;

procedure
Tdft_jual.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure
Tdft_jual.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure Tdft_jual.tglChange(Sender:
TObject);
begin
  addljual;
end;
procedure Tdft_jual.FormShow(Sender:
TObject);
begin
  kasir.Caption:=login.user.Text;
  tgl.Date:=now;
  tglchange(nil);
end;

procedure
Tdft_jual.ljualDblClick(Sender: TObject);
begin
  if ljual.ItemIndex<0 then
    exit;
  with ljual.Items[ljual.ItemIndex] do
  begin
    dtil_jual.kode.Caption:=caption;

dtil_jual.tgl.Caption:=formatdatetime('dd
MMMM yyyy',tgl.Date);
    dtil_jual.nama.Caption:=subitems[0];
    dtil_jual.kasir.Caption:=kasir.Caption;
    dtil_jual.tot.Caption:=subitems[1];
  end;
  utama.tampil(dtil_jual);
end;

procedure
Tdft_jual.ljualKeyPress(Sender: TObject;
var Key: Char);
begin
  if key=#13 then
    ljualDblClick(nil);
end;

end.
```

Form Detail Penjualan (unit16.pas)

```
unit Unit16;

interface

uses
```

```
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;

type
  Tdtil_jual = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    Label1: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    kode: TLabel;
    tgl: TLabel;
    nama: TLabel;
    kasir: TLabel;
    ldtil: TListView;
    Bevel1: TBevel;
    Label10: TLabel;
    tot: TLabel;
    baru: TEkImgBtn;
    EkImgBtn1: TEkImgBtn;
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure baruClick(Sender: TObject);
    procedure EkImgBtn1Click(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  dtil_jual: Tdtil_jual;

implementation

uses Unit14, Unit13, Unit1, Unit15;

{$R *.dfm}

procedure
Tdtil_jual.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure
Tdtil_jual.EkImgBtn3Click(Sender:
TObject);
begin
  rstruk.query.Close;
  close;
end;

procedure Tdtil_jual.baruClick(Sender:
TObject);
begin
  penjualan.tampil_struk(kode.Caption);
end;

procedure
Tdtil_jual.EkImgBtn1Click(Sender:
TObject);
begin
  if messagedlg('Hapus transaksi
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('delete from penjualan where
kode_jual="'+kode.Caption+'"');
      execsql;
      sql.Clear;
      sql.Add('delete from dtil_jual where
kode_jual="'+kode.Caption+'"');
      execsql;
    end;
    messagedlg('Transaksi
terhapus',mtinformation,[mbok],0);
    dft_jual.addljual;
    close;
  end;
end;
```

```
procedure Tdtil_jual.FormShow(Sender:
TObject);
begin
  ldtil.Items.BeginUpdate;
  ldtil.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select dtil_jual.kode_brg,
barang.nama_brg, dtil_jual.harga_satuan,
dtil_jual.qty, dtil_jual.jml_harga');
    sql.Add('from dtil_jual INNER JOIN');
    sql.Add('barang ON dtil_jual.kode_brg
= barang.kode_brg');
    sql.Add('where
dtil_jual.kode_jual='"+kode.Caption+"'
order by barang.nama_brg');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with ldtil.Items.Add do
        begin
          caption:=fields[0].Text;
          subitems.Add(fields[1].text);

subitems.Add(formatfloat('#,##0.00',field
s[2].AsFloat));

subitems.Add(formatfloat('#,##0.00',field
s[3].AsFloat));

subitems.Add(formatfloat('#,##0.00',field
s[4].AsFloat));
        end;
        next;
      end;
    end;
    close;
  end;
  ldtil.Items.EndUpdate;
  ldtil.SetFocus;
end;

end.
\
```

Form Barang (unit8.pas)

```
unit Unit8;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;

type
  Tbarang = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    lbrg: TListView;
    btn: TEkImgBtn;
    EkImgBtn1: TEkImgBtn;
    EkImgBtn2: TEkImgBtn;
    Label1: TLabel;
    cari: TEdit;
    bcari: TEkImgBtn;
    edt: TEkImgBtn;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure addlbrg;
    procedure addstok;
    procedure cariKeyPress(Sender:
TObject; var Key: Char);
    procedure bcariClick(Sender: TObject);
    procedure FormShow(Sender:
TObject);
    procedure
lbrgAdvancedCustomDrawSubItem(Send
er: TCustomListView;
      Item: TListItem; SubItem: Integer;
State: TCustomDrawState;
      Stage: TCustomDrawStage; var
DefaultDraw: Boolean);
    procedure btnClick(Sender: TObject);
    procedure edtClick(Sender: TObject);
    procedure EkImgBtn1Click(Sender:
TObject);
    procedure EkImgBtn2Click(Sender:
TObject);
    procedure lbrgDblClick(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  barang: Tbarang;
```

```
implementation

uses Unit1, Unit9, Unit10;

{$R *.dfm}

procedure tbarang.addstok;
var i:integer;
begin
  if lbrg.Items.Count<1 then
    exit;
  lbrg.Items.BeginUpdate;
  for i:=0 to lbrg.Items.Count-1 do

lbrg.Items[i].SubItems[3]:=formatfloat('#,
##0.00',getstok(lbrg.Items[i].Caption,now
));
  lbrg.Items.EndUpdate;
end;

procedure Tbarang.addlbrg;
begin
  lbrg.Items.BeginUpdate;
  lbrg.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
kode_brg,nama_brg,satuan,harga_jual,sto
k_minim from barang');
    sql.Add('where nama_brg like
"%'+cari.Text+'%" and terhapus<>"1"
order by kode_brg');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with lbrg.Items.Add do
        begin
          caption:=fields[0].Text;
          subitems.Add(fields[1].text);
          subitems.Add(fields[2].text);

subitems.Add(formatfloat('#,##0.00',field
s[3].AsFloat));
          subitems.Add(' ');

subitems.Add(formatfloat('#,##0.00',field
s[4].AsFloat));
        end;
        next;
      end;
    end;
    close;
  end;
  lbrg.Items.EndUpdate;
  addstok;
end;

procedure
Tbarang.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Tbarang.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tbarang.cariKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
  begin
    addlbrg;
    cari.SelectAll;
  end;
end;

procedure Tbarang.bcariClick(Sender:
TObject);
begin
  addlbrg;
  cari.SetFocus;
  cari.SelectAll;
end;

procedure Tbarang.FormShow(Sender:
TObject);
begin
  cari.Clear;
  addlbrg;
  if lbrg.Items.Count>0 then
    lbrg.ItemIndex:=0;
end;
```

```
procedure
Tbarang.lbrgAdvancedCustomDrawSubIt
em(Sender: TCustomListView;
  Item: TListItem; SubItem: Integer; State:
TCustomDrawState;
  Stage: TCustomDrawStage; var
DefaultDraw: Boolean);
begin
  if
strtofloat(item.SubItems[3])<=strtofloat(it
em.SubItems[4]) then
  begin
    if subitem=4 then
      sender.Canvas.Font.Color:=clred;
  end;
end;

procedure Tbarang.btnClick(Sender:
TObject);
begin
  tbrg.Label2.Caption:='Tambah Barang
Baru';
  tbrg.kosong;
  utama.tampil(tbrg);
end;

procedure Tbarang.edtClick(Sender:
TObject);
var kode:string;
begin
  if lbrg.ItemIndex<0 then
  begin
    messagedlg('Tidak ada barang yang
dipilih',mtinformation,[mbcancel],0);
    lbrg.SetFocus; exit;
  end;

kode:=lbrg.Items[lbrg.itemindex].Caption
;
  tbrg.Label2.Caption:='Edit "'+kode+'";
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
nama_brg,satuan,harga_jual,stok_awal,sto
k_minim from barang where
kode_brg="'+kode+'");
    open;
    with tbrg do
    begin
      nama.Text:=fields[0].Text;
      satuan.Text:=fields[1].Text;

hjual.Text:=floattostr(fields[2].AsFloat);
stok.Text:=floattostr(fields[3].AsFloat);

minim.Text:=floattostr(fields[4].AsFloat);
    end;
    close;
  end;
  utama.tampil(tbrg);
end;

procedure
Tbarang.EkImgBtn1Click(Sender:
TObject);
var kode,nama:string;
    ada:boolean;
begin
  if lbrg.ItemIndex<0 then
  begin
    messagedlg('Tidak ada yang
dipilih',mtinformation,[mbcancel],0);
    lbrg.SetFocus; exit;
  end;

nama:=lbrg.Items[lbrg.itemindex].SubIte
ms[0];
  if messagedlg('Hapus
"'+nama+'"'+#13+'Anda
Yakin?',mtconfirmation,[mbyes,mbno],0)
=mryes then
  begin

kode:=lbrg.Items[lbrg.itemindex].Caption
;
    ada:=false;
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('select kode_brg from
pembelian where kode_brg="'+kode+'");
      open;
      if recordcount>0 then
        ada:=true;
      close;
      sql.Clear;
      sql.Add('select kode_brg from
dtil_jual where kode_brg="'+kode+'");
      open;
      if recordcount>0 then
        ada:=true;
      close;
      sql.Clear;
      if not ada then
        sql.Add('delete from barang where
kode_brg="'+kode+'")
```

```
    else
      sql.Add('update barang set
terhapus="1" where
kode_brg="'+kode+'");
    execsql;
   end;
   messagedlg('"'+nama+'" telah
terhapus',mtinformation,[mbok],0);
   addlbrg;
  end;
end;

procedure
Tbarang.EkImgBtn2Click(Sender:
TObject);
begin
  if lbrg.ItemIndex<0 then
  begin
   messagedlg('Tidak ada barang yang
dipilih',mtinformation,[mbcancel],0);
   lbrg.SetFocus; exit;
  end;
  with lbrg.Items[lbrg.ItemIndex] do
  begin
   dtil_stok.kode.Caption:=caption;
   dtil_stok.nama.Caption:=subitems[0];
   dtil_stok.satuan.Caption:=subitems[1];
   with utama.query do
   begin
    close;
    sql.Clear;
    sql.Add('select stok_awal from barang
where kode_brg="'+caption+'");
    open;
    if recordcount>0 then

dtil_stok.awal.Caption:=formatfloat('#,##
0.00',fields[0].AsFloat);
    close;
   end;
   dtil_stok.stok.Caption:=subitems[3];
  end;
  utama.tampil(dtil_stok);
end;

procedure Tbarang.lbrgDblClick(Sender:
TObject);
begin
  if lbrg.ItemIndex>-1 then
   edtclick(nil);
end;

end.
```

Form Tambah / Edit Barang (unit9.pas)

```
unit Unit9;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg;

type
  Ttbrg = class(TForm)
   Image1: TImage;
   Label2: TLabel;
   EkImgBtn3: TEkImgBtn;
   Shape1: TShape;
   nama: TEdit;
   Label1: TLabel;
   Label3: TLabel;
   satuan: TEdit;
   Label4: TLabel;
   hjual: TEdit;
   Label5: TLabel;
   stok: TEdit;
   Bevel1: TBevel;
   btn: TEkImgBtn;
   Label6: TLabel;
   minim: TEdit;
   procedure EkImgBtn3Click(Sender:
TObject);
   procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
     Shift: TShiftState; X, Y: Integer);
   procedure hjualKeyPress(Sender:
TObject; var Key: Char);
   procedure hjualChange(Sender:
TObject);
   function getkode:string;
   function aya(var kode:string):boolean;
   procedure btnClick(Sender: TObject);
   procedure kosong;
   procedure namaKeyPress(Sender:
TObject; var Key: Char);
   procedure FormShow(Sender:
TObject);
  private
   { Private declarations }
  public
   { Public declarations }
  end;

var
```

```
  tbrg: Ttbrg;

implementation

uses Unit1, Unit8;

{$R *.dfm}

procedure ttbrg.kosong;
begin
  nama.Clear;
  satuan.Clear;
  hjual.Text:='0';
  stok.Text:='0';
  minim.Text:='0';
end;

function ttbrg.aya(var
kode:string):boolean;
begin
  kode:='';
  result:=false;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_brg from barang
where nama_brg like "'+nama.Text+'"');
    open;
    if recordcount>0 then
    begin
      kode:=fields[0].Text;
      result:=true;
    end;
    close;
  end;
end;

function ttbrg.getkode:string;
var i:byte;
    temp:string;
begin
  temp:='0001';
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_brg from barang
order by kode_brg');
    open;
    if recordcount>0 then
    begin
      last;
      temp:=copy(fields[0].Text,4,4);
      temp:=inttostr(strtoint(temp)+1);
      if length(temp)<4 then
      begin
        for i:=1 to 4-length(temp) do
          temp:='0'+temp;
      end;
    end;
    close;
  end;
  result:='BRG'+temp;
end;

procedure Ttbrg.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Ttbrg.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Ttbrg.hjualKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
  begin
    btnclick(nil);
    exit;
  end;
  kuduangka(tedit(sender).Text,key);
end;

procedure Ttbrg.hjualChange(Sender:
TObject);
begin
  curr(sender);
end;

procedure Ttbrg.btnClick(Sender:
TObject);
var kode:string;
begin
  if trim(nama.Text)='' then
  begin
    messagedlg('Nama barang tidak boleh
kosong',mtinformation,[mbcancel],0);
```

```
    nama.SetFocus; exit;
  end;
  if trim(satuan.Text)='' then
  begin
    messagedlg('Satuan barang tidak boleh
kosong',mtinformation,[mbcancel],0);
    satuan.SetFocus; exit;
  end;
  if strtofloat(hjual.Text)=0 then
  begin
    messagedlg('Harga jual tidak boleh
nol',mtinformation,[mbcancel],0);
    hjual.SetFocus;
    hjual.SelectAll; exit;
  end;
  if strtofloat(minim.Text)=0 then
  begin
    messagedlg('Stok minimal tidak boleh
nol',mtinformation,[mbcancel],0);
    minim.SetFocus;
    minim.SelectAll; exit;
  end;
  if pos('Tambah',label2.Caption)<>0 then
  begin
    if aya(kode) then
    begin
      messagedlg(''+nama.Text+'' telah
terdaftar dengan kode
'+kode+'',mtinformation,[mbcancel],0);
      nama.SetFocus;
      nama.SelectAll; exit;
    end;
  end;
  if messagedlg('Simpan Data
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    if pos('Tambah',label2.Caption)<>0
then
      kode:=getkode
    else
      kode:=copy(label2.Caption,7,7);
    with utama.query do
    begin
      close;
      sql.Clear;
      if pos('Tambah',label2.Caption)<>0
then
        sql.Add('select * from barang')
      else
        sql.Add('select * from barang where
kode_brg='+kode+'');
      open;
      if pos('Tambah',label2.Caption)<>0
then
        append
      else
        edit;
      if pos('Tambah',label2.Caption)<>0
then

fieldbyname('kode_brg').Value:=kode;

fieldbyname('nama_brg').Value:=nama.Te
xt;

fieldbyname('satuan').Value:=satuan.Text;

fieldbyname('harga_jual').AsFloat:=strtofl
oat(hjual.Text);

fieldbyname('stok_awal').AsFloat:=strtofl
oat(stok.Text);

fieldbyname('stok_minim').AsFloat:=strto
float(minim.Text);
      if pos('Tambah',label2.Caption)<>0
then
        fieldbyname('terhapus').Value:='0';
      post;
      close;
    end;
    messagedlg('Data
tersimpan',mtinformation,[mbok],0);
    if pos('Tambah',label2.Caption)<>0
then
    begin
      kosong;
      formshow(nil);
    end
    else
    begin
      close;
      barang.lbrg.SetFocus;
    end;
    barang.addlbrg;
  end;
end;

procedure Ttbrg.namaKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
    btnclick(nil);
end;

procedure Ttbrg.FormShow(Sender:
TObject);
begin
  nama.SetFocus;
```

```
  nama.SelectAll;
end;

end.
```

Form Detail Stok (unit10.pas)

```
unit Unit10;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;

type
  Tdtil_stok = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    Label1: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    kode: TLabel;
    nama: TLabel;
    satuan: TLabel;
    stok: TLabel;
    lbeli: TListView;
    Bevel1: TBevel;
    Label10: TLabel;
    tambah: TLabel;
    Label12: TLabel;
    kurang: TLabel;
    Label14: TLabel;
    beli: TLabel;
    Label16: TLabel;
    jual: TLabel;
    Label6: TLabel;
    awal: TLabel;
    Label8: TLabel;
    Bevel2: TBevel;
    Label9: TLabel;
    ljual: TListView;
    Bevel3: TBevel;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure addlbeli;
    procedure addljual;
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  dtil_stok: Tdtil_stok;

implementation

uses Unit1;

{$R *.dfm}

procedure tdtil_stok.addljual;
var krit:string;
begin
  krit:=baliktgl('penjualan','tgl_jual');
  ljual.Items.BeginUpdate;
  ljual.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select penjualan.tgl_jual,
dtil_jual.harga_satuan, dtil_jual.qty,
dtil_jual.jml_harga');
    sql.Add('from penjualan inner join
dtil_jual on penjualan.kode_jual =
dtil_jual.kode_jual');
    sql.Add('where
dtil_jual.kode_brg='"+kode.Caption+"");
    sql.Add('and
'+krit+'<='"+formatdatetime('yyyy/MM/d
d',now)+"");
    sql.Add('order by '+krit);
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with ljual.Items.Add do
        begin
          caption:=fields[0].Text;

subitems.Add(formatfloat('#,##0.00',field
s[1].asfloat));

subitems.Add(formatfloat('#,##0.00',field
s[2].asfloat));
```

```
subitems.Add(formatfloat('#,##0.00',field
s[3].asfloat));
        end;
        next;
      end;
    end;
    close;
    sql.Clear;
    sql.Add('select
sum(dtil_jual.qty),sum(dtil_jual.jml_harg
a)');
    sql.Add('from penjualan inner join
dtil_jual on penjualan.kode_jual =
dtil_jual.kode_jual');
    sql.Add('where
dtil_jual.kode_brg='"+kode.Caption+"');
    sql.Add('and
'+krit+'<='"+formatdatetime('yyyy/MM/d
d',now)+"');
    open;

kurang.Caption:=formatfloat('#,##0.00',fi
elds[0].AsFloat);

jual.Caption:=formatfloat('#,##0.00',fields
[1].AsFloat);
    close;
  end;
  ljual.Items.EndUpdate;
end;

procedure tdtil_stok.addlbeli;
var krit:string;
begin
  krit:=baliktgl('','tgl_beli');
  lbeli.Items.BeginUpdate;
  lbeli.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
tgl_beli,harga_satuan,qty,jml_harga');
    sql.Add('from pembelian');
    sql.Add('where
kode_brg='"+kode.Caption+"');
    sql.Add('and
'+krit+'<='"+formatdatetime('yyyy/MM/d
d',now)+"');
    sql.Add('order by '+krit);
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with lbeli.Items.Add do
        begin
          caption:=fields[0].Text;

subitems.Add(formatfloat('#,##0.00',field
s[1].asfloat));

subitems.Add(formatfloat('#,##0.00',field
s[2].asfloat));

subitems.Add(formatfloat('#,##0.00',field
s[3].asfloat));
        end;
        next;
      end;
    end;
    close;
    krit:=baliktgl('','tgl_beli');
    sql.Clear;
    sql.Add('select
sum(qty),sum(jml_harga)');
    sql.Add('from pembelian');
    sql.Add('where
kode_brg='"+kode.Caption+"');
    sql.Add('and
'+krit+'<='"+formatdatetime('yyyy/MM/d
d',now)+"');
    open;

tambah.Caption:=formatfloat('#,##0.00',fi
elds[0].AsFloat);

beli.Caption:=formatfloat('#,##0.00',fields
[1].AsFloat);
    close;
  end;
  lbeli.Items.EndUpdate;
end;

procedure
Tdtil_stok.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Tdtil_stok.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
```

```
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tdtil_stok.FormShow(Sender:
TObject);
begin
  addlbeli;
  addljual;
end;

end.
```

Form Pembelian (unit12.pas)

```
unit Unit12;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls,
EkImgFadeBase, EkImgBtn, jpeg, Menus,
  ComCtrls;

type
  Tpembelian = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    cari: TEdit;
    Label1: TLabel;
    pbrg: TPopupMenu;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    kode: TLabel;
    nama: TLabel;
    satuan: TLabel;
    jml: TLabel;
    harga: TEdit;
    qty: TEdit;
    btn: TEkImgBtn;
    Bevel2: TBevel;
    Label12: TLabel;
    dari: TEdit;
    bcari: TEkImgBtn;
    Shape2: TShape;
    Label9: TLabel;
    tgl: TDateTimePicker;
    lbeli: TListView;
    Bevel1: TBevel;
    Label10: TLabel;
    EkImgBtn1: TEkImgBtn;
    tgl2: TDateTimePicker;
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure hargaChange(Sender:
TObject);
    procedure hargaKeyPress(Sender:
TObject; var Key: Char);
    procedure cariKeyPress(Sender:
TObject; var Key: Char);
    procedure bcariClick(Sender: TObject);
    procedure pilih(sender:tobject);
    procedure kosong;
    procedure FormShow(Sender:
TObject);
    procedure kunci(x:boolean);
    procedure cariChange(Sender:
TObject);
    procedure btnClick(Sender: TObject);
    procedure FormCreate(Sender:
TObject);
    procedure tglExit(Sender: TObject);
    function getkode:string;
    procedure addlbeli;
    procedure tglChange(Sender: TObject);
    procedure EkImgBtn1Click(Sender:
TObject);
    procedure tgl2Change(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  pembelian: Tpembelian;

implementation

uses Unit1, Unit4;

{$R *.dfm}

procedure tpembelian.addlbeli;
begin
  lbeli.Items.BeginUpdate;
```

```
  lbeli.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select pembelian.kode_beli,
pembelian.kode_brg, barang.nama_brg,
pembelian.dari, barang.satuan,
pembelian.qty, pembelian.harga_satuan,');
    sql.Add('pembelian.jml_harga');
    sql.Add('from pembelian INNER
JOIN');
    sql.Add('barang ON
pembelian.kode_brg = barang.kode_brg');
    sql.Add('where
pembelian.tgl_beli='''+formatdatetime('dd
/MM/yyyy',tgl2.Date)+'''');
    sql.Add('order by pembelian.kode_beli
desc');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with lbeli.Items.Add do
        begin
          caption:=fields[0].Text;
          subitems.Add(fields[1].text);
          subitems.Add(fields[2].text);
          subitems.Add(fields[3].text);
          subitems.Add(fields[4].text);

subitems.Add(formatfloat('#,##0.00',field
s[5].asfloat));

subitems.Add(formatfloat('#,##0.00',field
s[6].asfloat));

subitems.Add(formatfloat('#,##0.00',field
s[7].asfloat));
        end;
        next;
      end;
    end;
    close;
  end;
  lbeli.Items.EndUpdate;
end;
function tpembelian.getkode:string;
var i:byte;
    temp:string;
begin
  temp:='0001';
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_beli from
pembelian');
    sql.Add('where kode_beli like
''B'+formatdatetime('ddMMyy',now)+'%''
);
    sql.Add('order by kode_beli');
    open;
    if recordcount>0 then
    begin
      last;
      temp:=copy(fields[0].Text,8,4);
      temp:=inttostr(strtoint(temp)+1);
      if length(temp)<4 then
      begin
        for i:=1 to 4-length(temp) do
          temp:='0'+temp;
      end;
    end;
    close;
  end;

result:='B'+formatdatetime('ddMMyy',no
w)+temp;
end;

procedure tpembelian.kunci(x:boolean);
begin
  tgl.Enabled:=not x;
  dari.ReadOnly:=x;
  harga.ReadOnly:=x;
  qty.ReadOnly:=x;
  btn.Enabled:=not x;
end;

procedure tpembelian.kosong;
begin
  kode.Caption:='-';
  nama.Caption:='-';
  satuan.Caption:='-';
  dari.Clear;
  harga.Text:='0';
  qty.Text:='0';
  jml.Caption:='0'+decimalseparator+'00';
end;

procedure
tpembelian.pilih(sender:tobject);
var kodex:string;
begin
  tgl.Date:=now;
  kosong;
  tglchange(nil);
```

```
kodex:=copy(tmenuitem(sender).Caption,
1,7);
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select
kode_brg,nama_brg,satuan from barang
where kode_brg='"+kodex+"');
    open;
    if recordcount>0 then
    begin
      kode.Caption:=fields[0].Text;
      nama.Caption:=fields[1].Text;
      satuan.Caption:=fields[2].Text;
    end;
    close;
  end;
  kunci(false);
  tgl.SetFocus;
end;

procedure
Tpembelian.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure
Tpembelian.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Tpembelian.hargaChange(Sender:
TObject);
begin
  curr(sender);

jml.Caption:=formatfloat('#,##0.00',strtofl
oat(harga.Text)*strtofloat(qty.Text));
end;

procedure
Tpembelian.hargaKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
  begin
    btnclick(nil);
    exit;
  end;
  kuduangka(tedit(sender).Text,key);
end;

procedure
Tpembelian.cariKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=#13 then
  begin
    bcariclick(nil);
  end;
end;

procedure Tpembelian.bcariClick(Sender:
TObject);
var x:tmenuitem;
    poin:tpoint;
begin
  pbrg.Items.Clear;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select kode_brg,nama_brg
from barang');
    sql.Add('where kode_brg like
"%'+cari.Text+'%" or nama_brg like
"%'+cari.Text+'%"');
    sql.Add('order by nama_brg');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        x:=tmenuitem.Create(pbrg);
        x.OnClick:=pilih;
        x.Caption:=fields[0].Text+' -
'+fields[1].Text;
        pbrg.Items.Add(x);
        next;
      end;
    end;
    close;
  end;
  poin.X:=cari.Left;
```

```
  poin.Y:=cari.Top+cari.Height;
  poin:=clienttoscreen(poin);
  pbrg.Popup(poin.X,poin.Y);
end;

procedure
Tpembelian.FormShow(Sender: TObject);
begin
  tgl.Date:=now;
  cari.Clear;
  kosong;
  kunci(true);
  tglchange(nil);
  cari.SetFocus;
end;

procedure
Tpembelian.cariChange(Sender:
TObject);
begin
  kosong;
  kunci(true);
end;

procedure Tpembelian.btnClick(Sender:
TObject);
var kodex:string;
begin
  if trim(dari.Text)='' then
  begin
    messagedlg('Asal barang belum
diisi',mtinformation,[mbcancel],0);
    dari.SetFocus; exit;
  end;
  if strtofloat(harga.Text)<1 then
  begin
    messagedlg('Harga beli tidak boleh
kosong',mtinformation,[mbcancel],0);
    harga.SetFocus;
    harga.SelectAll; exit;
  end;
  if strtofloat(qty.Text)<1 then
  begin
    messagedlg('Jumlah barang tidak boleh
kosong',mtinformation,[mbcancel],0);
    qty.SetFocus;
    qty.SelectAll; exit;
  end;
  if messagedlg('Simpan data
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    kodex:=getkode;
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('select * from pembelian');
      open;
      append;

fieldbyname('kode_beli').Value:=kodex;

fieldbyname('tgl_beli').Value:=formatdate
time('dd/MM/yyyy',tgl.Date);

fieldbyname('bag_gudang').Value:=login.
user.Text;

fieldbyname('kode_brg').Value:=kode.Ca
ption;

fieldbyname('qty').AsFloat:=strtofloat(qty
.Text);

fieldbyname('harga_satuan').AsFloat:=strt
ofloat(harga.Text);

fieldbyname('jml_harga').AsFloat:=strtofl
oat(qty.Text)*strtofloat(harga.Text);
      fieldbyname('dari').Value:=dari.Text;
      post;
      close;
    end;
    messagedlg('Data
tersimpan',mtinformation,[mbok],0);
    cari.Clear;
    kosong;
    kunci(true);
    tglchange(nil);
    cari.SetFocus;
  end;
end;

procedure
Tpembelian.FormCreate(Sender:
TObject);
begin
  kosong;
end;

procedure Tpembelian.tglExit(Sender:
TObject);
var x:string;
begin
  if
formatdatetime('yyyyMMdd',tgl.Date)>fo
rmatdatetime('yyyyMMdd',now) then
  begin
```

```
    x:=formatdatetime('dd MMMM
yyyy',tgl.Date);
    messagedlg('Maaf, tanggal '''+x+'''
melebihi hari
ini',mtinformation,[mbcancel],0);
    tgl.Date:=now;
    tgl.SetFocus;
    addlbeli; exit;
  end;
end;

procedure Tpembelian.tglChange(Sender:
TObject);
begin
  tgl2.Date:=tgl.Date;
  addlbeli;
end;

procedure
Tpembelian.EkImgBtn1Click(Sender:
TObject);
var x:string;
begin
  if lbeli.ItemIndex<0 then
  begin
    messagedlg('Pilih salah satu transaksi
pada list',mtinformation,[mbcancel],0);
    exit;
  end;
  x:=lbeli.Items[lbeli.itemindex].Caption;
  if messagedlg('Hapus transaksi
pembelian '''+x+''''+#13+'Anda
yakin?',mtconfirmation,[mbyes,mbno],0)
=mryes then
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('delete from pembelian where
kode_beli='''+x+'''');
      execsql;
    end;
    messagedlg('Data
terhapus',mtinformation,[mbok],0);
    addlbeli;
  end;
end;

procedure
Tpembelian.tgl2Change(Sender:
TObject);
begin
  addlbeli;
end;

end.
```

Form Laporan (unit17.pas)

```
unit Unit17;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;
type
  Tlaporan = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    priode: TRadioGroup;
    tgl: TDateTimePicker;
    bulan: TComboBox;
    tahun1: TEdit;
    tahun2: TEdit;
    btn: TEkImgBtn;
    EkImgBtn2: TEkImgBtn;
    Shape2: TShape;
    jns: TComboBox;
    Label3: TLabel;
    Shape3: TShape;
    tgl2: TDateTimePicker;
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure tahun1KeyPress(Sender:
TObject; var Key: Char);
    procedure priodeClick(Sender:
TObject);
    procedure FormShow(Sender:
TObject);
    procedure connbeli;
    procedure btnClick(Sender: TObject);
    procedure connjual;
    procedure diskonek;
    procedure connstok;
    procedure EkImgBtn2Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  laporan: Tlaporan;

implementation

uses Unit18, Unit4, Unit19, Unit1,
Unit13, Unit20;

{$R *.dfm}

procedure tlaporan.connstok;
var
k_brg,n_brg,satuan,awal,jml_jual,jml_bel
i,akhir:tstrings;
    i:smallint;
    x:string;
begin

x:=formatdatetime('yyyy/MM/dd',tgl2.Dat
e);
  k_brg:=tstringlist.Create;
  jml_beli:=tstringlist.Create;
  jml_jual:=tstringlist.Create;
  akhir:=tstringlist.Create;
  n_brg:=tstringlist.Create;
  satuan:=tstringlist.Create;
  awal:=tstringlist.Create;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('delete from temp where k_brg
like "%%"');
    execsql;
    sql.Clear;
    sql.Add('select
kode_brg,nama_brg,satuan,stok_awal
from barang where terhapus<>"1"');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        k_brg.Add(fields[0].text);
        n_brg.Add(fields[1].text);
        satuan.Add(fields[2].text);
        awal.Add(fields[3].text);
        next;
      end;
    end;
    close;
  end;
  if k_brg.Count<1 then
    exit;
  for i:=0 to k_brg.Count-1 do
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('select sum(dtil_jual.qty)
from dtil_jual INNER JOIN penjualan
ON
dtil_jual.kode_jual=penjualan.kode_jual');
      sql.Add('where
'+baliktgl('penjualan','tgl_jual')+'<="'+x+'"
');
      sql.Add('and
dtil_jual.kode_brg="'+k_brg[i]+'"');
      open;

jml_jual.Add(formatfloat('#,##0.00',fields
[0].asfloat));
      close;
      sql.Clear;
      sql.Add('select sum(qty) from
pembelian where
kode_brg="'+k_brg[i]+'"');
      sql.Add('and
'+baliktgl('','tgl_beli')+'<="'+x+'"');
      open;

jml_beli.Add(formatfloat('#,##0.00',fields
[0].asfloat));
      close;
    end;

akhir.Add(formatfloat('#,##0.00',getstok(
k_brg[i],tgl2.Date)));
  end;
  for i:=0 to k_brg.Count-1 do
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('select * from temp');
      open;
      append;
      fieldbyname('k_brg').Text:=k_brg[i];
      fieldbyname('n_brg').Text:=n_brg[i];
      fieldbyname('satuan').Text:=satuan[i];

fieldbyname('awal').AsFloat:=penjualan.t
ofloat(awal[i]);

fieldbyname('jml_beli').AsFloat:=penjual
an.tofloat(jml_beli[i]);
```

```
fieldbyname('jml_jual').AsFloat:=penjual
an.tofloat(jml_jual[i]);

fieldbyname('akhir').AsFloat:=penjualan.t
ofloat(akhir[i]);
      post;
      close;
    end;
  end;
end;

procedure tlaporan.diskonek;
begin
  rstok.tbl.Close;
  rbeli.query.Close;
  rjual.qjual.Close;
  rjual.dtilx.Close;
  rjual.nama.Close;
end;

procedure tlaporan.connjual;
var krit:string;
begin
  diskonek;
  case priode.ItemIndex of
    0 : begin

rjual.priod.Caption:=formatdatetime('dd
MMMM yyyy',tgl.Date);
         krit:='where
tgl_jual='"+formatdatetime('dd/MM/yyyy'
,tgl.Date)+"";
        end;
    1 : begin
         rjual.priod.Caption:=bulan.Text+'
'+tahun1.Text;
         krit:='where tgl_jual like
"%'+inttostr(bulan.ItemIndex+1)+'/'+tahu
n1.Text+"";
        end;
    2 : begin
         rjual.priod.Caption:='Tahun
'+tahun2.Text;
         krit:='where tgl_jual like
"%/'+tahun2.Text+"";
        end;
  end;
  with rjual.qjual do
  begin
    close;
    sql.Clear;
    sql.Add('select * from penjualan');
    sql.Add(krit);
    sql.Add('order by
'+baliktgl('','tgl_jual')+',kode_jual');
    open;
  end;
  rjual.dtilx.Open;
  rjual.nama.Open;
  rjual.cetak.Caption:=login.user.Text+' @
'+formatdatetime('dd MMMM
yyyy',now);
  rjual.PreviewModal;
end;

procedure tlaporan.connbeli;
var krit:string;
begin
  diskonek;
  case priode.ItemIndex of
    0 : begin

rbeli.priod.Caption:=formatdatetime('dd
MMMM yyyy',tgl.Date);
         krit:='where
pembelian.tgl_beli='"+formatdatetime('dd
/MM/yyyy',tgl.Date)+"";
        end;
    1 : begin
         rbeli.priod.Caption:=bulan.Text+'
'+tahun1.Text;
         krit:='where pembelian.tgl_beli like
"%'+inttostr(bulan.ItemIndex+1)+'/'+tahu
n1.Text+"";
        end;
    2 : begin
         rbeli.priod.Caption:='Tahun
'+tahun2.Text;
         krit:='where pembelian.tgl_beli like
"%/'+tahun2.Text+"";
        end;
  end;
  with rbeli.query do
  begin
    close;
    sql.Clear;
    sql.Add('select pembelian.kode_beli,
pembelian.tgl_beli, pembelian.kode_brg,
barang.nama_brg, pembelian.dari,
pembelian.harga_satuan, pembelian.qty,');
    sql.Add('pembelian.jml_harga,
pembelian.bag_gudang');
    sql.Add('from pembelian INNER
JOIN');
    sql.Add('barang ON
pembelian.kode_brg = barang.kode_brg');
    sql.Add(krit);
```

```
    sql.Add('order by
'+baliktgl('pembelian','tgl_beli')+',pembeli
an.kode_beli');
    open;
    rbeli.kode.DataField:='kode_beli';
    rbeli.tgl.DataField:='tgl_beli';
    rbeli.k_brg.DataField:='kode_brg';
    rbeli.nama.DataField:='nama_brg';
    rbeli.asal.DataField:='dari';
    rbeli.harga.DataField:='harga_satuan';
    rbeli.qty.DataField:='qty';
    rbeli.jml.DataField:='jml_harga';
    rbeli.user.DataField:='bag_gudang';
    rbeli.tot.Expression:='sum(jml_harga)';
  end;
  rbeli.cetak.Caption:=login.user.Text+' @
'+formatdatetime('dd MMMM
yyyy',now);
  rbeli.PreviewModal;
end;

procedure
Tlaporan.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure
Tlaporan.EkImgBtn3Click(Sender:
TObject);
begin
  diskonek;
  close;
end;

procedure
Tlaporan.tahun1KeyPress(Sender:
TObject; var Key: Char);
begin
  if not(((key>=#$30) and (key<=#$39))
or (key=#8) or (key=#46)) then
    key:=#0;
end;

procedure Tlaporan.priodeClick(Sender:
TObject);
begin
  tgl.Enabled:=false;
  bulan.Enabled:=false;
  tahun1.Enabled:=false;
  tahun2.Enabled:=false;
  case priode.ItemIndex of
    0 : tgl.Enabled:=true;
    1 : begin
          bulan.Enabled:=true;
          tahun1.Enabled:=true;
        end;
    2 : tahun2.Enabled:=true;
  end;
end;

procedure Tlaporan.FormShow(Sender:
TObject);
begin
  jns.ItemIndex:=0;
  priode.ItemIndex:=0;
  priodeclick(nil);
  tgl.Date:=now;
  tgl2.Date:=now;

bulan.ItemIndex:=strtoint(formatdatetime
('MM',now))-1;

tahun1.Text:=formatdatetime('yyyy',now)
;

tahun2.Text:=formatdatetime('yyyy',now)
;
  priode.SetFocus;
end;

procedure Tlaporan.btnClick(Sender:
TObject);
begin
  case jns.ItemIndex of
    0 : connbeli;
    1 : connjual;
  end;
end;

procedure
Tlaporan.EkImgBtn2Click(Sender:
TObject);
begin
  diskonek;
  connstok;
  rstok.tbl.Open;
  rstok.tbl.Sort:='n_brg ASC';
  rstok.priod.Caption:=formatdatetime('dd
MMMM yyyy',tgl2.Date);
  rstok.cetak.Caption:=login.user.Text+'
@ '+rstok.priod.Caption;
  rstok.PreviewModal;
```

```
end;

end.
```

Form Hak Akses (unit6.pas)

```
unit Unit6;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg, ComCtrls;

type
  Thak_akses = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    lhak: TListView;
    tmb: TEkImgBtn;
    del: TEkImgBtn;
    Label1: TLabel;
    cari: TEdit;
    jns: TComboBox;
    go: TEkImgBtn;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure cariKeyPress(Sender:
TObject; var Key: Char);
    procedure addlhak;
    procedure jnsChange(Sender: TObject);
    procedure goClick(Sender: TObject);
    procedure FormShow(Sender:
TObject);
    procedure tmbClick(Sender: TObject);
    procedure delClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  hak_akses: Thak_akses;

implementation

uses Unit1, Unit7, Unit4;

{$R *.dfm}

procedure thak_akses.addlhak;
var krit:string;
begin
  lhak.Items.BeginUpdate;
  lhak.Items.Clear;
  krit:='where username like
"%'+cari.Text+'%"';
  case jns.ItemIndex of
    1 : krit:=krit+' and jenis like "k"';
    2 : krit:=krit+' and jenis like "g"';
    3 : krit:=krit+' and jenis like "a"';
  end;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select username,jenis from
hak_akses');
    sql.Add(krit);
    sql.Add('order by jenis,username');
    open;
    if recordcount>0 then
    begin
      first;
      while not eof do
      begin
        with lhak.Items.Add do
        begin
          caption:=fields[0].Text;
          if uppercase(fields[1].Text)='A'
then
            subitems.Add('Admin')
          else if uppercase(fields[1].Text)='K'
then
            subitems.Add('Kasir')
          else if uppercase(fields[1].Text)='G'
then
            subitems.Add('Gudang');
        end;
        next;
      end;
    end;
    close;
  end;
  lhak.Items.EndUpdate;
  if lhak.Items.Count>0 then
    lhak.ItemIndex:=0;
  cari.SetFocus;
  cari.SelectAll;
end;
```

```
procedure
Thak_akses.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Thak_akses.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure
Thak_akses.cariKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=' ' then
    key:=#0
  else if key=#13 then
    addlhak;
end;

procedure Thak_akses.jnsChange(Sender:
TObject);
begin
  addlhak;
end;

procedure Thak_akses.goClick(Sender:
TObject);
begin
  addlhak;
end;

procedure
Thak_akses.FormShow(Sender: TObject);
begin
  cari.Clear;
  jns.ItemIndex:=0;
  addlhak;
end;

procedure Thak_akses.tmbClick(Sender:
TObject);
begin
  utama.tampil(thak);
end;

procedure Thak_akses.delClick(Sender:
TObject);
var user:string;
begin
  if lhak.ItemIndex<0 then
  begin
    messagedlg('Pilih salah satu item yang
mau dihapus pada
list',mtinformation,[mbcancel],0);
    lhak.SetFocus;
    exit;
  end;

user:=lhak.Items[lhak.itemindex].Caption
;
  if user=login.user.Text then
  begin
    messagedlg('Maaf, "'+user+'" adalah
username
anda',mtinformation,[mbcancel],0);
    lhak.SetFocus;
    exit;
  end;
  if messagedlg('Hapus
"'+user+'"'+#13+'Anda
yakin?',mtconfirmation,[mbyes,mbno],0)
=mryes then
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('delete from hak_akses where
username="'+user+'"');
      execsql;
    end;
    messagedlg('"'+user+'" telah
terhapus',mtinformation,[mbok],0);
    addlhak;
  end;
end;

end.
```

Form Tambah Hak Akses (unit7.pas)

```
unit Unit7;

interface

uses
```

```
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, EkImgFadeBase,
EkImgBtn, StdCtrls, jpeg;

type
  Tthak = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    Shape1: TShape;
    user: TEdit;
    Label1: TLabel;
    Label3: TLabel;
    jns: TComboBox;
    Label4: TLabel;
    pass: TEdit;
    Bevel1: TBevel;
    btn: TEkImgBtn;
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure userKeyPress(Sender:
TObject; var Key: Char);
    procedure FormShow(Sender:
TObject);
    procedure btnClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  thak: Tthak;

implementation

uses Unit1, Unit6;

{$R *.dfm}

procedure
Tthak.EkImgBtn3Click(Sender: TObject);
begin
  close;
end;

procedure
Tthak.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tthak.userKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=' ' then
    key:=#0
  else if key=#13 then
    btnclick(nil);
end;

procedure Tthak.FormShow(Sender:
TObject);
begin
  jns.ItemIndex:=0;
  user.Clear;
  pass.Clear;
  jns.SetFocus;
end;

procedure Tthak.btnClick(Sender:
TObject);
var aya:boolean;
begin
  if length(user.Text)=0 then
  begin
    messagedlg('Maaf, username tidak
boleh
kosong',mtinformation,[mbcancel],0);
    user.SetFocus;
    exit;
  end;
  aya:=false;
  with utama.query do
  begin
    close;
    sql.Clear;
    sql.Add('select username from
hak_akses where
username='"+user.Text+"');
    open;
    if recordcount>0 then
      aya:=true;
    close;
  end;
  if aya then
  begin
    messagedlg('Maaf, username sudah
terdaftar',mtinformation,[mbcancel],0);
```

```
    user.SetFocus;
    user.SelectAll;
    exit;
  end;
  if length(pass.Text)=0 then
  begin
    messagedlg('Maaf, Password tidak
boleh
kosong',mtinformation,[mbcancel],0);
    pass.SetFocus;
    exit;
  end;
  if messagedlg('Simpan data
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('select username,pass,jenis
from hak_akses');
      open;
      append;
      fields[0].Text:=user.Text;
      fields[1].Text:=pass.Text;
      case jns.ItemIndex of
        0 : fields[2].Text:='K';
        1 : fields[2].Text:='G';
        2 : fields[2].Text:='A';
      end;
      post;
      close;
    end;
    messagedlg('Data
tersimpan',mtinformation,[mbok],0);
    close;
    hak_akses.cari.Clear;
    hak_akses.jns.ItemIndex:=0;
    hak_akses.addlhak;
  end;
end;

end.
```

Form Ganti Password (unit5.pas)

```
unit Unit5;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls,
EkImgFadeBase, EkImgBtn, jpeg;

type
  Tcpass = class(TForm)
    Image1: TImage;
    Label2: TLabel;
    EkImgBtn3: TEkImgBtn;
    lama: TEdit;
    Shape1: TShape;
    Label1: TLabel;
    Label3: TLabel;
    user: TLabel;
    Label5: TLabel;
    jns: TLabel;
    Label7: TLabel;
    baru: TEdit;
    Label8: TLabel;
    ulang: TEdit;
    Bevel1: TBevel;
    btn: TEkImgBtn;
    procedure lamaKeyPress(Sender:
TObject; var Key: Char);
    procedure EkImgBtn3Click(Sender:
TObject);
    procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton;
      Shift: TShiftState; X, Y: Integer);
    procedure FormShow(Sender:
TObject);
    procedure btnClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  cpass: Tcpass;

implementation

uses Unit4, Unit1;

{$R *.dfm}

procedure Tcpass.lamaKeyPress(Sender:
TObject; var Key: Char);
begin
  if key=' ' then
    key:=#0
  else if key=#13 then
    btnclick(nil);
```

```
end;

procedure
Tcpass.EkImgBtn3Click(Sender:
TObject);
begin
  close;
end;

procedure
Tcpass.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
const
  sc_dragmove=$F012;
begin
  releasecapture;

self.Perform(wm_syscommand,sc_dragm
ove,0);
end;

procedure Tcpass.FormShow(Sender:
TObject);
begin
  user.Caption:=login.user.Text;
  jns.Caption:=login.jns.Caption;
  lama.Clear;
  baru.Clear;
  ulang.Clear;
  lama.SetFocus;
end;

procedure Tcpass.btnClick(Sender:
TObject);
begin
  if lama.Text<>login.pass.Text then
  begin
    messagedlg('Maaf, Password lama
salah',mtinformation,[mbcancel],0);
    lama.SetFocus;
    lama.SelectAll;
    exit;
  end;
  if length(baru.Text)=0 then
  begin
    messagedlg('Maaf, password baru
belum diisi',mtinformation,[mbcancel],0);
    baru.SetFocus;
    baru.SelectAll;
    exit;
  end;
  if baru.Text<>ulang.Text then
  begin
    messagedlg('Maaf, kedua Password
tidak sama',mtinformation,[mbcancel],0);
    baru.SetFocus;
    baru.SelectAll;
    exit;
  end;
  if messagedlg('Simpan data
ini?',mtconfirmation,[mbyes,mbno],0)=mr
yes then
  begin
    with utama.query do
    begin
      close;
      sql.Clear;
      sql.Add('update hak_akses set
pass='"+baru.Text+"' where
username='"+user.Caption+"');
      execsql;
    end;
    messagedlg('Password telah
dirubah',mtinformation,[mbok],0);
    close;
    utama.pg.SetFocus;
  end;
end;

end.
```

Form About Me (unit21.pas)

```
unit Unit21;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, EkImgFadeBase, EkImgBtn,
jpeg, ExtCtrls, PicShow;

type
  Tabout_me = class(TForm)
    ext: TEkImgBtn;
    ps: TPicShow;
    Image1: TImage;
    Image2: TImage;
    procedure extClick(Sender: TObject);
    procedure psStop(Sender: TObject);
    procedure FormShow(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  about_me: Tabout_me;

implementation

{$R *.dfm}

procedure Tabout_me.extClick(Sender:
TObject);
begin
  caption:='close';
  ps.Style:=99;
  ps.Picture:=image2.Picture;
  ps.BgPicture:=image1.Picture;
  ext.Visible:=false;
  ps.Execute;
end;

procedure Tabout_me.psStop(Sender:
TObject);
begin
  if caption='open' then
    ext.Visible:=true
  else
    close;
end;

procedure Tabout_me.FormShow(Sender:
TObject);
begin
  ps.Execute;
end;

end.
```