

S K R I P S I

PENGEMBANGAN APLIKASI AKUNTANSI UNTUK HOME INDUSTRY PADA STUDI KASUS DI KABUPATEN SUMENEP

Disusun Oleh:

MOHAMMAD NURUL SHIDDEQIIN

NIM. 04.12.664

**KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
JURUSAN TEKNIK ELEKTRO S-1
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010**

204

ДАВИД
СЕРГЕЕВИЧ АРХАНГЕЛЬСКИЙ
ПОДПОЛЧЕСКИЙ ЛЕНИНСКОГО ИМПЕРИАЛЬНОГО
СЕВЕРНОГО ЛЕНИНСКОГО ГОСУДАРСТВЕННОГО УНИВЕРСИТЕТА
СЕВЕРНОГО ЛЕНИНСКОГО КОММУНИСТИЧЕСКОГО ПАРТИЗАНСКОГО
УНИВЕРСИТЕТА

ДАВИД СЕРГЕЕВИЧ
СЕВЕРНОГО ЛЕНИНСКОГО УНИВЕРСИТЕТА
СЕВЕРНОГО КОММУНИСТИЧЕСКОГО ПАРТИЗАНСКОГО УНИВЕРСИТЕТА

ДИ КРЫЛЬЯМ СВИНЕЙ
СКАЗКА МОСКОВСКОГО УНИВЕРСИТЕТА
БАЛАШИХИНСКОГО КОММУНИСТИЧЕСКОГО УНИВЕРСИТЕТА

204

LEMBAR PERSETUJUAN

PENGEMBANGAN APLIKASI AKUNTANSI UNTUK HOME INDUSTRY PADA STUDY KASUS DI KABUPATEN SUMENEP

SKRIPSI

*Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh
Gelar Sarjana Teknik Elektro Strata Satu (S-I)*

Disusun Oleh :

MOHAMMAD NURUL SHIDDIEQIIN

NIM : 04.12.664

Diperiksa dan Disetujui

Dosen Pembimbing I

M. Ibrahim Ashari, ST., MT.
NIP.Y. 1030100358

Dosen Pembimbing II

I Komang Somawirata, ST., MT.
NIP.Y. 1030100361

Mengetahui

Ketua Jurusan Teknik Elektro S-1



Ir. Yusuf Ismail Nakhoda, MT.
NIP.Y. 1018800189

JURUSAN TEKNIK ELEKTRO S-1
KONSENTRASI TEKNIK KOMPUTER DAN INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL MALANG
2010

ABSTRAK

Banyak sekali badan usaha yang bergerak di bidang jasa, dagang maupun manufaktur atau lainnya perlu melakukan pencatatan transaksi bisnis untuk mengontrol kondisi keuangan usahanya yang akan menunjukkan kondisi keuangan sesungguhnya, laba atau rugi. Banyak permasalahan timbul dari sistem akuntansi pada badan usaha terutama yang bergerak dalam bidang usaha kecil, seperti pencatatan secara *paperbase* adalah banyaknya waktu terbuang, kehilangan data dan tingkat keakurasiannya yang kurang. Padahal laporan keuangan ini akan dijadikan sebagai bahan evaluasi untuk mengambil strategi keuangan pada tahap berikutnya sehingga perusahaan tetap bertahan bahkan bisa maju.

Berdasarkan hasil survei pada sebagian *Home Industry* di Kabupaten Sumenep, dilakukanlah perancangan sistem informasi akuntansi dengan fitur yang dapat memproses transaksi penjualan, pembelian, data pegawai, data supplier, aset, data barang, dan pembuatan laporan keuangan secara komputerisasi agar dapat mempermudah usaha dagang dalam melakukan pencatatan transaksi bisnisnya.

Dengan aplikasi ini badan usaha dagang dapat lebih cepat melakukan pencatatan transaksi bisnisnya dengan tidak mengurangi kinerja bisnisnya atau kehilangan data seperti pada permasalahan yang sudah dihadapi sebelumnya.

Kata kunci: pencatatan *paperbase*, usaha dagang, komputerisasi

ABSTRACT

Many commercial entities dedicated to the manufacture, trade, or services or other business transaction records should be to monitor the financial situation of their business, which will show the true financial condition, gains or losses. Many of the problems of accounting systems in enterprises engaged primarily in the field of small enterprises, as a paperbase recording is the amount of lost time, loss of data, and less accuracy level. Whereas that financial statements will be used as materials for evaluation to take a financing for the next stage strategy so that enterprises can survive even advanced.

Based on the results of a survey on some of home industry in Sumenep, perform accounting information system with features that can process of sales, purchases, personal data, data of suppliers, goods, data, and the manufacture of computerized financial information to facilitate trade in the exercise of the business transaction log activity design.

With the implementation of commercial enterprises can more quickly make commercial relationship with the list of not reduce the performance of business or loss of data as the problems experienced by the previous.

Keywords: record-keeping of *paperbase*, trading business, computerization

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa yang telah memberikan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul :

**“PENGEMBANGAN APLIKASI AKUNTANSI
UNTUK HOME INDUSTRY PADA STUDI KASUS
DI KABUPATEN SUMENEPE”**

Skripsi ini merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatika ITN Malang dan untuk mencapai gelar Sarjana Teknik.

Keberhasilan peyelesaian laporan skripsi ini tidak lepas dari dukungan dan bantuan berbagai pihak. Untuk itu penyusun menyampaikan terima kasih kepada :

1. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE. selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Sidik Noertjahjono, MT. Selaku Dekan Fakultas Teknik Industri serta Ketua Panitia Ujian Skripsi Fakultas Teknik Industri.
3. Bapak Ir. F. Yudi Limpraptono, MT. selaku Ketua Jurusan Teknik Elektro S-1.
4. Bapak M. Ibrahim Ashari, ST., MT. selaku Dosen Pembimbing I.
5. Bapak I Komang Somawirata, ST., MT. selaku Dosen Pembimbing II.
6. Ayah dan Ibu serta saudara-saudara saya yang telah memberikan do'a restu dan dorongan semangat.

7. Teman-teman serta semua pihak yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penyusun telah berusaha semaksimal mungkin serta menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Oleh karena itu penyusun mengharapkan saran dan kritik yang bersifat membangun dari para pembaca sekalian demi kesempurnaan laporan skripsi ini.

Harapan penyusun semoga laporan skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan bagi semua kalangan dan pembaca.

Malang, Maret 2010

Penyusun

DAFTAR ISI

LEMBAR PERSETUJUAN	i
ABSTRAKSI	ii
KATA PENGANTAR	iv
DAFTAR ISI	vi
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
 BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan.....	3
1.4. Batasan Masalah.....	3
1.5. Metodologi	4
1.5.1. Metodologi Pengumpulan Data.....	4
1.5.2. Metode Perancangan Sistem	5
1.6. Sistematika Penulisan.....	6
 BAB II TINJAUAN PUSTAKA	
2.1. Pengertian Informasi	8
2.1.1. Sistem Informasi Akuntansi	8
2.1.2. Persamaan Akuntansi	9
2.1.3. Tujuan Sistem Akuntansi	10

2.2. Pengertian Database dan RDBMS	10
2.2.1. Ms. Office Access 2003	13
2.3. Pengertian <i>Interface</i>	18
2.3.1. Borland Delphi 7	19
2.4. Perangkat Analisis dan Perancangan.....	24
2.4.1. Diagram Aliran Data	24
2.4.2. Pemodelan Data.....	25

BAB III DESAIN DAN ANALISA SISTEM

3.1. Deskripsi Sistem.....	27
3.2. Spesifikasi Sistem	28
3.3. Perancangan Sistem.....	29
3.3.1. Data Flow Diagram (DFD).....	29
3.3.2. Pemodelan Data Relasional.....	35
3.4. Desain Tabel Pembentukan Sistem.....	37

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1. Hasil Perancangan Sistem	40
4.2. Pengujian Aplikasi Akuntansi.....	42
4.3. Pengujian Entri Data	44
4.3.1. Form Login.....	44
4.3.2. Form Pegawai.....	45
4.3.3. Form Supplier.....	48
4.3.4. Form Data Barang	49

4.3.5. Form Pembelian	51
4.3.6. Form Penjualan.....	52
4.3.7. Aset Usaha Dagang	54
4.3.8. Laporan Laba Rugi.....	55
4.3.9. Laporan Perubahan Modal	56
4.3.10. Laporan Neraca	57
4.3.11. Printer Setup.....	58
4.4. Perbandingan Pencatatan Sistem <i>Paperbase</i> dengan Sistem Komputerisasi	59
BAB V PENUTUP	
5.1. Kesimpulan	60
5.2. Saran	61
DAFTAR PUSTAKA	62
LAMPIRAN-LAMPIRAN	

DAFTAR TABEL

3.1. Tabel Aset.....	37
3.2. Tabel Barang	37
3.4. Tabel Pegawai.....	38
3.5. Tabel Supplier.....	38
3.6. Tabel Pembelian.....	39
3.7. Tabel Penjualan	39
5.1. Tabel Perbandingan Pencatatan Sistem <i>Paperbase</i> dengan Sistem Komputerisasi	59

DAFTAR GAMBAR

2.1. <i>Ms. Office Access 2003</i>	15
2.2. Tables	16
2.3. Query	17
2.4. Relationships	18
2.5. Borland Delphi 7	21
2.6. ADO pada Component Palette	22
2.7. Jet 4.0 OLE Provider	23
2.8. Pemilihan <i>Database</i>	23
2.9. Simbol Utama DFD	24
3.1. <i>Conteks Diagram</i> Sistem Informasi Akuntansi.....	31
3.2. DFD level 0 Sistem Informasi Akuntansi	32
3.3. DFD level 1 Validasi Login	33
3.4. DFD level1 Pembuatan Laporan	33
3.5. DFD level1 Pemvalidasian Data Barang.....	34
3.6. DFD level 1 Proses Pembelian.....	34
3.7. DFD level 1 Proses Penjualan	35
3.8. <i>Entity Relationship Diagram (ERD)</i>	36
4.1. Struktur Menu Aplikasi.....	41
4.2. Pengisian User dan Password	42
4.3. Struktur Menu Utama	43
4.4. Form Login.....	44

4.5.	Pesan Error pada kesalahan.....	44
4.6.	Penginputan Data Pegawai	45
4.7.	Menyimpan Data Pegawai.....	45
4.8.	Mengedit data dan peringatan penggunaan <i>username</i>	46
4.9.	Pesan Penyimpanan, penghapusan dan pengeditan data.....	47
4.10.	Form Supplier.....	48
4.11.	Peringatan Nama Sama	48
4.12.	Form Data Barang.....	49
4.13.	Peringatan penggunaan nama yang sama	50
4.14.	Form Pembelian	51
4.15.	Form Penjualan	52
4.16.	Peringatan Stok.....	53
4.17.	<i>Form</i> Aset Usaha Dagang	54
4.18.	Laporan Laba Rugi.....	55
4.19.	Laporan Perubahan Modal	56
4.20.	Laporan Neraca	57
4.21.	Printer Setup	58

BAB I

PENDAHULUAN

1.1. Latar Belakang

Banyak sekali perusahaan yang bergerak di bidang jasa, dagang maupun manufaktur atau lainnya perlu melakukan pencatatan transaksi bisnis untuk mengontrol kondisi keuangan usahanya. Karena dari laporan keuangan akan dapat menunjukkan keadaan keuangan dari perusahaan yang sesungguhnya, laba atau rugi. Banyak permasalahan timbul dari sistem akuntansi pada perusahaan terutama yang bergerak dalam bidang usaha kecil, seperti pencatatan secara *paperbase* adalah banyaknya waktu terbuang, kehilangan data dan tingkat keakurasiannya yang kurang. Padahal laporan keuangan ini akan dijadikan sebagai bahan evaluasi untuk mengambil strategi keuangan pada tahap berikutnya sehingga perusahaan tetap bertahan bahkan bisa maju.

Di indonesia perusahaan dagang banyak ditemui mulai dari level kecil, menengah sampai yang besar, namun proses bisnis dan sistem akuntansi yang dipakai hampirlah sama antara satu perusahaan dengan lainnya. Terutama kota-kota kecil yang sedang tumbuh, sangat butuh adanya pengelolaan keuangan dan jalannya transaksi barang pada usaha dagang. Seperti halnya *Home Industry* di Kota Sumenep dimana bidang usahanya masih banyak yang bergerak dalam usaha dagang menengah kebawah dan sumber daya manusianya yang rata-rata belum mengenal arti

pengelolaan keuangan serta transaksi bisnis juga sistem komputerisasi, sehingga nantinya bisa terus berkembang seperti kota-kota lainnya.

Usaha dagang yang dijalankan pada umumnya masih banyak menggunakan sistem pencatatan secara *paperbase* dalam pembuatan serta pencatatan jalannya transaksi dan laporan, yang dapat menghambat bahkan memperlambat kinerja bisnis. Karena kurang memadainya sarana informasi dan sering terjadi kehilangan data informasi tersebut, maka dibutuhkan sistem informasi akuntansi yang memadai, untuk proses pencatatan transaksi-transaksi yang dilakukan secara komputerisasi sehingga menghasilkan informasi yang dapat menaikkan kinerja bisnis dan menghasilkan laporan keuangan yang tepat.

Aplikasi akuntansi yang akan dibuat, berdasarkan hasil survei pada sebagian *Home Industry* di Kabupaten Sumenep. Kemudian dilakukan perancangan sistem informasi akuntansi dengan fitur yang dapat memproses transaksi penjualan, pembelian, data pegawai, data supplier, aset, data barang, dan pembuatan laporan keuangan secara komputerisasi agar dapat mempermudah usaha dagang dalam melaksanakan sistem akuntansinya.

1.2. Rumusan Masalah

Berdasarkan hal di atas maka timbul suatu perumusan masalah, bagaimana mendesain aplikasi akuntansi tentang pengelolaan data yang nantinya dapat mengontrol jalannya transaksi barang sehingga menunjang kinerja *Home Industry*.

1.3. Tujuan

Tujuan yang ingin dicapai dalam pelaksanaan skripsi ini membuat Aplikasi Akuntansi yang dapat menghasilkan informasi yang tepat, akurat dan dapat diandalkan sebagai bahan pengambilan keputusan sehingga membantu proses bisnis usaha dagang yang dijalankan.

1.4. Batasan Masalah

Agar permasalahan mengarah sesuai dengan tujuan maka pembahasan dibatasi pada hal-hal sebagai berikut:

1. Penerapan sistem informasi ini hanya dirancang pada usaha dagang (*Home Industry*) di Kabupaten Sumenep.
2. Tidak membahas Teknologi Pengamanan.
3. Secara garis besar sistem informasi ini memiliki beberapa modul aplikasi, dalam hal ini dibatasi hanya pada modul sistem informasi usaha dagang (*Home Industry*) di Kabupaten Sumenep.
4. Sistem dikembangkan dengan menggunakan aplikasi *Borland Delphi 7* dan sistem database digunakan *Microsoft Office access 2003*.

1.5. Metodologi

1.5.1. Metode Pengumpulan Data

Sebelum melakukan desain aplikasi, dilakukan survei ke beberapa usaha dagang yang masih menggunakan sistem pencatatan manual. Hasil survei digunakan sebagai pedoman dalam mendesain aplikasi yang dibutuhkan *Home Industry* di Kabupaten Sumenep.

Metode yang digunakan dalam pengumpulan data adalah :

1. Studi Lapangan

Dengan metode ini data-data diperoleh langsung dari sumber yang bersangkutan, dimana peneliti berhadapan langsung dengan obyek yang diteliti, yang dilakukan dengan cara :

a. Survei

Teknik pengumpulan data dengan cara terjun secara langsung dan mencatat secara sistematis terhadap objek masalah.

b. Wawancara / Interview

Teknik pengumpulan data dengan jalan mengadakan komunikasi atau tanya jawab secara langsung dengan pimpinan atau pegawai *Home Industry* di Kabupaten Sumenep tentang sistem yang diterapkan.

2. Studi Pustaka / Literatur

Pengumpulan data ini dilakukan dengan cara mencari bahan-bahan kepustakaan sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan obyek penelitian.

1.5.2. Metode Perancangan Sistem

Metode yang digunakan dalam perancangan Sistem Informasi usaha dagang (*Home Industry*) di Kabupaten Sumenep adalah :

1. Rekayasa Sistem

Tahap ini ditekankan pada pengumpulan kebutuhan pengguna tingkatan sistem dengan mendefinisikan konsep sistem beserta *interface* yang menghubungkannya dengan lingkungan. Hasil dari tahap ini adalah spesifikasi sistem.

2. Analisis

Tahap ini melakukan pengumpulan kebutuhan elemen-elemen ditingkat perangkat lunak. Dengan analisa harus dapat ditentukan kebutuhan data, fungsi, proses atau *procedure* yang diperlukan beserta unjuk kerjanya, dan *interface*-nya. Hasil akhir tahap ini adalah spesifikasi kebutuhan perangkat lunak.

3. Design

Spesifikasi perangkat lunak yang dihasilkan dari tahap analisa ditransformasikan kedalam bentuk arsitektur perangkat lunak yang memiliki karakteristik mudah dimengerti dan tidak sulit untuk diimplementasikan. Tahap ini terdiri dari dua tahap, yaitu *preliminary design* dan *detailed design*. Subtahap pertama menghasilkan rancangan yang bersifat global, sedangkan subtahap yang kedua menghasilkan rancangan detil sehingga semua kelas, tipe data, fungsi dan prosedurennya terdefinisi.

4. Coding

Tahap ini dilakukan implementasi hasil rancangan kedalam baris-baris kode program yang dapat dimengerti oleh mesin.

5. Testing

Pengujian dilakukan untuk setiap modul. Jika hasil pengujian tidak menemukan adanya masalah, modul-modul yang terpisah tersebut diintegrasikan untuk mendapatkan perangkat lunak yang utuh. Kemudian, dilakukan pengujian ditingkat perangkat lunak yang memfokuskan pada masalah-masalah logika internal, fungsi eksternal, potensi masalah yang mungkin terjadi dan pemeriksaan hasil.

1.6. Sistematika Penulisan

Untuk mempermudah dan memahami pembahasan skripsi ini, maka peneliti menyajikan secara sistematis sebagai berikut:

Bab I : PENDAHULUAN

Berisi Latar Belakang, Rumusan Masalah, Tujuan, Batasan Masalah, Metodologi dan Sistematika Penulisan.

Bab II : DASAR TEORI

Berisi tentang landasan teori mengenai sistem informasi Akuntansi, teori *Borland Delphi 7* dan teori *Ms. Office Access 2003*.

Bab III : PERANCANGAN SISTEM

Bab ini berisi perencanaan objek uji, prosedur percobaan dan menjelaskan metode penelitian yang dilakukan.

Bab IV : IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang Implementasi dari Hasil Desain Aplikasi Akuntansi dan pengujian pada aplikasi.

Bab V : PENUTUP

Berisi tentang kesimpulan dan saran yang diambil berdasarkan hasil pembahasan skripsi ini.

BAB II

TINJAUAN PUSTAKA

2.1. Pengertian Informasi

Informasi adalah merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian (*event*) yang nyata (*fact*) yang digunakan untuk pengambilan keputusan. Data yang telah diolah ke dalam suatu bentuk yang berguna bagi penerimanya secara nyata atau berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa depan. Secara garis besar informasi (*Information*) adalah data yang telah diproses atau diolah menjadi bentuk (*substansi*) lain yang lebih berguna dan lebih berarti bagi yang menerimanya.

Informasi harus berguna dan mempunyai manfaat bagi pemakainya sehingga diharapkan informasi tersebut dapat berfungsi maksimal bagi pemakainya serta tepat sasaran dari hak penerima informasi sesuai dengan wewenang dan tingkat kebutuhan akan informasi, sehingga tidak semua orang bisa mendapatkan informasi.

2.1.1. Sistem Infomasi Akuntansi

Akuntansi adalah proses yang tediri dari 3 aktivitas, yaitu identifikasi, merekam, dan mengkomunikasikan kejadian ekonomi pada sebuah organisasi. Definisi sistem informasi akuntansi adalah memproses data dan transaksi untuk menyediakan informasi yang dibutuhkan oleh pemakai untuk perencanaan, pengendalian, dan pengoperasian bisnis. Untuk menghasilkan informasi yang

dibutuhkan untuk pengambilan keputusan, maka sistem informasi akuntansi harus dapat mengumpulkan transaksi dan data lain dan memasukkannya ke dalam sistem tersebut, dapat memproses dan menyimpan data tersebut untuk masa yang akan datang, dan menyediakan informasi yang dibutuhkan oleh pemakai berupa laporan yang menghasilkan informasi yang dapat diandalkan untuk pengambilan keputusan nantinya.

2.1.2. Persamaan Akuntansi

Menurut persamaan akuntansi, harta perusahaan pada dasarnya merupakan hak dari pihak lain yang dalam hal ini termasuk pemilik perusahaan. Selain menggunakan modal dari pemilik, juga menggunakan pinjaman untuk membiayai operasinya. Karena konsep kesatuan usaha menghendaki pemisahan modal pemilik dari kekayaan atau harta perusahaan, maka ketiga unsur persamaan akuntansi dinyatakan sebagai berikut :

$$H = U + M$$

di mana:

H = Harta atau aktiva

U = Utang atau kewajiban

M = Modal pemilik

Pada persamaan di atas, penggunaan utang akan menambah harta perusahaan. Tetapi harta perusahaan tersebut tidak seluruhnya menjadi hak pemilik. Karena itu, sisipan kanan persamaan memisahkan utang dan modal pemilik. Dengan kata lain,

bertambahnya harta perusahaan tidak selalu mencerminkan bertambahnya modal pemilik.

2.1.3. Tujuan Sistem Akuntansi

Akuntansi adalah kegiatan yang berhubungan dengan pengelolaan keuangan suatu organisasi, dalam kegiatan akuntansi orang mencatat, mendokumentasikan dan membuat laporan berbagai hal yang berkaitan dengan kegiatan keuangan suatu organisasi. Sesuai dengan pengertian akuntansi, maka secara umum tujuan pengembangan sistem akuntansi adalah sebagai berikut :

- 1) Untuk menyediakan informasi bagi pengelolaan usaha baru.
- 2) Untuk memperbaiki informasi yang dihasilkan oleh sistem yang sudah ada, baik mengenai mutu, ketetapan penyajian, maupun struktur informasinya.
- 3) Untuk menghargai pengendalian akuntansi dan pengecekan intern, yaitu untuk memperbaiki tingkat keandalan (*reliability*) informasi akuntansi dan untuk menyediakan catatan lengkap, mengenai pertanggungjawaban dan perlindungan kekayaan perusahaan.
- 4) Untuk mengurangi biaya klerikal dalam menyelenggarakan catatan akutansi.

2.2. Pengertian Database dan RDBMS

Pengertian secara umum Database adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah

skema atau struktur tertentu, tersimpan di *hardware* komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu.

Basis data (*database*) akan berguna jika ada pengelola/penggeraknya. Yang menjadi pengelola/penggeraknya secara langsung adalah program/aplikasi (*software*). Prinsip utama database adalah pengaturan data/arsip yang nantinya dijadikan sebagai informasi. Tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan data/arsip.

Alasan diperlukannya Database sebagai tempat penyimpanan data adalah :

1. Merupakan salah satu komponen penting dalam sistem informasi, karena merupakan dasar dalam menyediakan informasi.
2. Menentukan kualitas informasi : akurat, tepat pada waktunya dan relevan.
3. Informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya.
4. Mengurangi duplikasi data (*data redundancy*).
5. Hubungan data dapat ditingkatkan (*data relatability*).
6. Mengurangi pemborosan tempat simpanan luar (pencatatan manual).

Basis data (*database*) juga berarti suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi. Basis data dimaksudkan untuk mengatasi problem pada sistem yang memakai pendekatan berbasis berkas. Sebuah konsep database memiliki beberapa hal sebagai berikut :

- Entitas : merupakan tempat informasi direkam, dapat berupa orang, tempat, kejadian dan lain-lain. Sebagai contoh dalam kasus administrasi siswa maka terdapat entity siswa, mata kuliah, guru, pembayaran.
- Atribut : disebut juga data elemen, data field, atau data item yang digunakan untuk menerangkan suatu entitas dan mempunyai harga tertentu, misalnya atribut dari entitas siswa diterangkan oleh, nama, tanggal lahir, alamat.
- Data Value : informasi atau data aktual yang disimpan pada tiap data, elemen, atau atribut.
- File/Tabel : kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.
- Record/Tuple : kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

Untuk mengelola basis data diperlukan perangkat lunak yang disebut DBMS.

DBMS adalah perangkat lunak sistem yang memungkinkan para pemakai membuat, memelihara, mengontrol, dan mengakses basis data dengan cara yang praktis dan efisien.

Relational Database adalah kumpulan data yang saling berelasi yang dipakai/ada dalam suatu limgkup tertentu, misalkan instansi, perusahaan dan lain-lain atau kasus tertentu. RDBMS (Relational Database Management System) merupakan koleksi atau kumpulan data yang di dalamnya memiliki suatu sistem yang mengatur

relasi di dalamnya bersama dengan satu set program yang berfungsi untuk melakukan manajemen sistem terhadap data tersebut.

Selanjutnya dalam RDBMS semua data disimpan dalam tabel-tabel, di mana sebuah tabel menyimpan informasi mengenai sebuah subjek tertentu. Dengan RDBMS, sebuah database akan dengan mudah dikelola walaupun jumlah datanya banyak dan kompleks, seperti pendefenisian data, mana data yang akan dimuat ke dalam sebuah database, bagaimana mengelolanya, serta bagaimana membagi data. Ide RDBMS ini yaitu menggunakan konsep matematika aljabar relasional untuk membagi data dalam beberapa himpunan (set) yang saling berhubungan dalam subset. Dalam model relasional, data dipisahkan dalam beberapa set yang pararel dengan struktur tabel. Struktur tabel ini mengandung elemen data individual yang disebut kolom atau *field*. Satu set kumpulan kolom disebut *record*.

2.2.1. Ms. Office Access 2003

Salah satu produk dari *Microsoft Corporation* ini mempunyai kegunaan antara lain sebagai *database* yang didalamnya terdapat tabel, *query*, *form*, *report*, keamanan database dan *VBA* (*Visual Basic for Application*). Yang menjadikannya lebih populer adalah kemudahannya dalam mengelola data, dengan menggunakan antarmuka yang mudah dipahami oleh orang awam sekali pun. Poin-poinnya antara lain : - Perancangan dan pembuatan *database*.

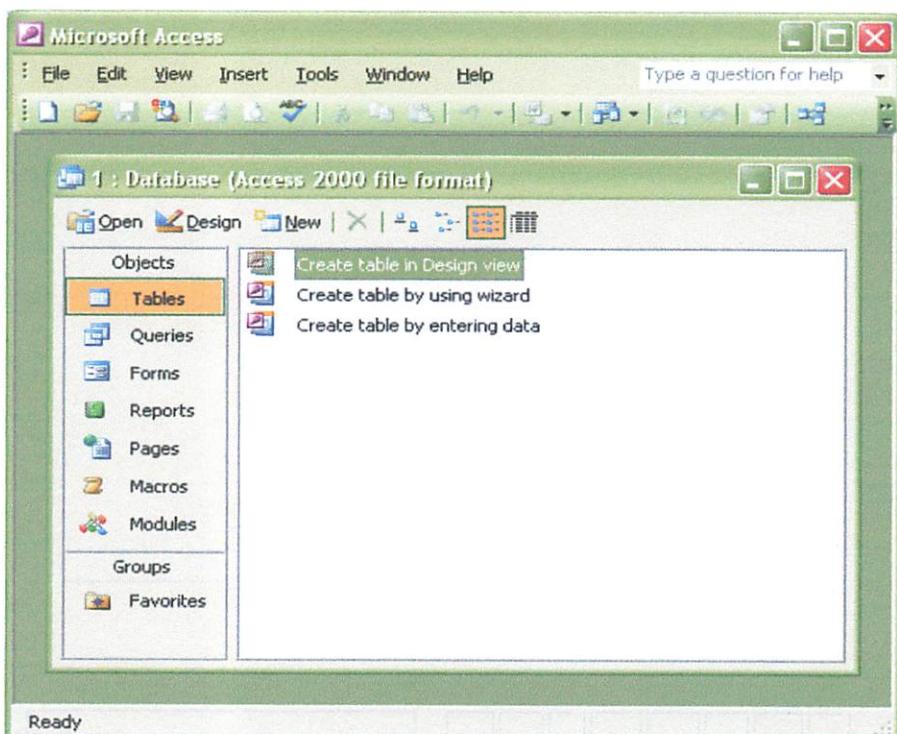
- Pengelolaan dan relasi tabel.
- Pembuatan dan penggunaan form.

- Proses memasukkan data ke *datasheet* dan ekspor data ke program atau ke *database* lain.
- Pengurutan dan *filter record*.
- Pembuatan dan pengelolaan *query*.
- Pembuatan laporan dan label surat.

Database pada Ms Access 2003 terdiri atas tables, query, form, report, macros, modules yang saling terkait atau saling berhubungan. Penjelasan sebagai berikut :

- a. **Tables**, berupa tabel kumpulan data yang merupakan komponen utama dari sebuah database.
- b. **Queries**, digunakan untuk mencari dan menampilkan data yang memenuhi syarat tertentu dari satu tabel atau lebih. Juga digunakan untuk meng-*update* atau menghapus beberapa *record* data pada suatu saat yang sama. Selain itu digunakan untuk menjalankan perhitungan terhadap sekelompok data.
- c. **Forms**, digunakan untuk menampilkan data, mengisi data, mengubah data yang ada di dalam tabel. Ketika *form* dibuka, Access 2003 mengambil data dari satu tabel atau lebih dan menampilkannya ke layar monitor dengan menggunakan *layout* yang dibuat melalui *Form Wizard* atau dari *layout* yang dirancang sendiri oleh user.
- d. **Reports**, digunakan untuk menampilkan laporan hasil analisa data. *Report* yang telah dikelompokkan dapat dicetak, dihitung subtotal dan total datanya berdasarkan kriteria tertentu. Juga dapat dibuat dengan berisikan grafik atau label data.

- e. **Pages**, digunakan untuk membuat halaman Web (page) berupa *data access page* yang dapat ditempatkan di *server* sistem jaringan intranet atau internet.
- f. **Macros**, untuk mengotomatisasi perintah-perintah yang sering digunakan dalam mengolah data.
- g. **Modules**, digunakan untuk perancangan berbagai modul aplikasi pengolahan database tingkat lanjut sesuai dengan kebutuhan yang berisi kode *Visual Basic for Application* yang ditulis untuk menangani even dalam Ms. Access 2003.

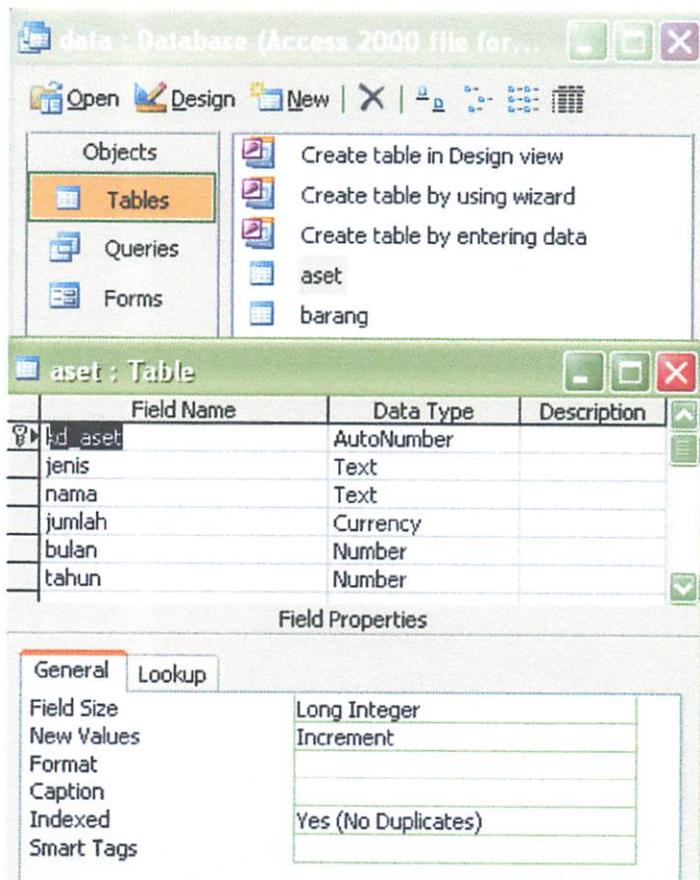


Gambar 2.1. Ms. Office Access 2003

Penggunaan Ms.Access pada aplikasi yang digunakan sebagai database, menggunakan :

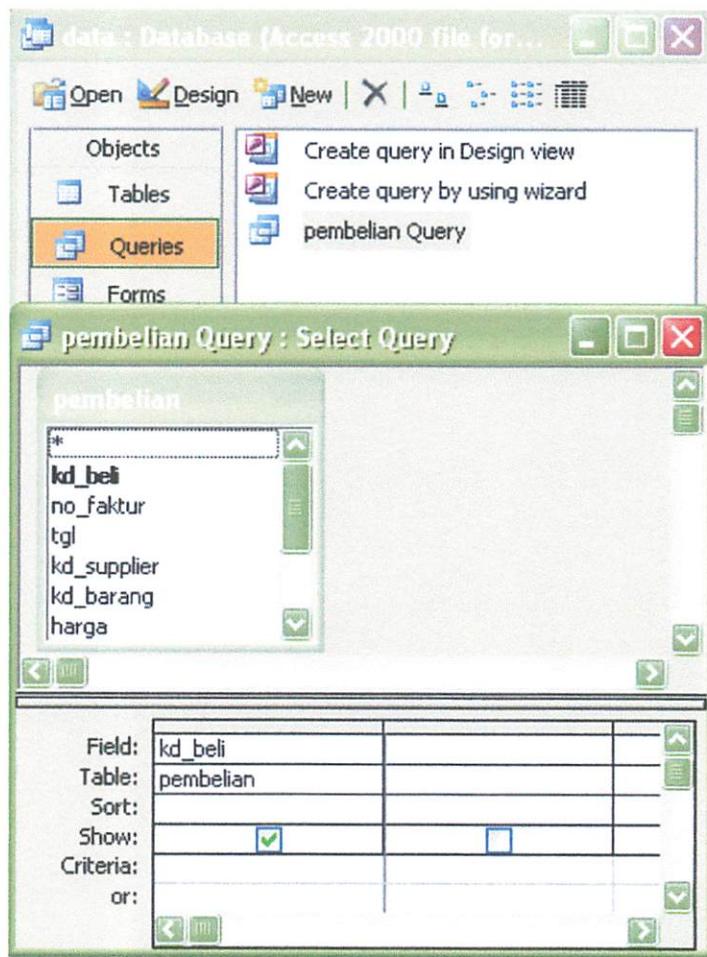
- ❖ Tables untuk mengumpulkan data yang merupakan komponen utama dari sebuah database. Data tersebut berupa tabel dimana banyaknya disesuaikan

dengan kebutuhan program yang akan dibuat, seperti pada gambar di bawah ini :



Gambar 2.2. Tables

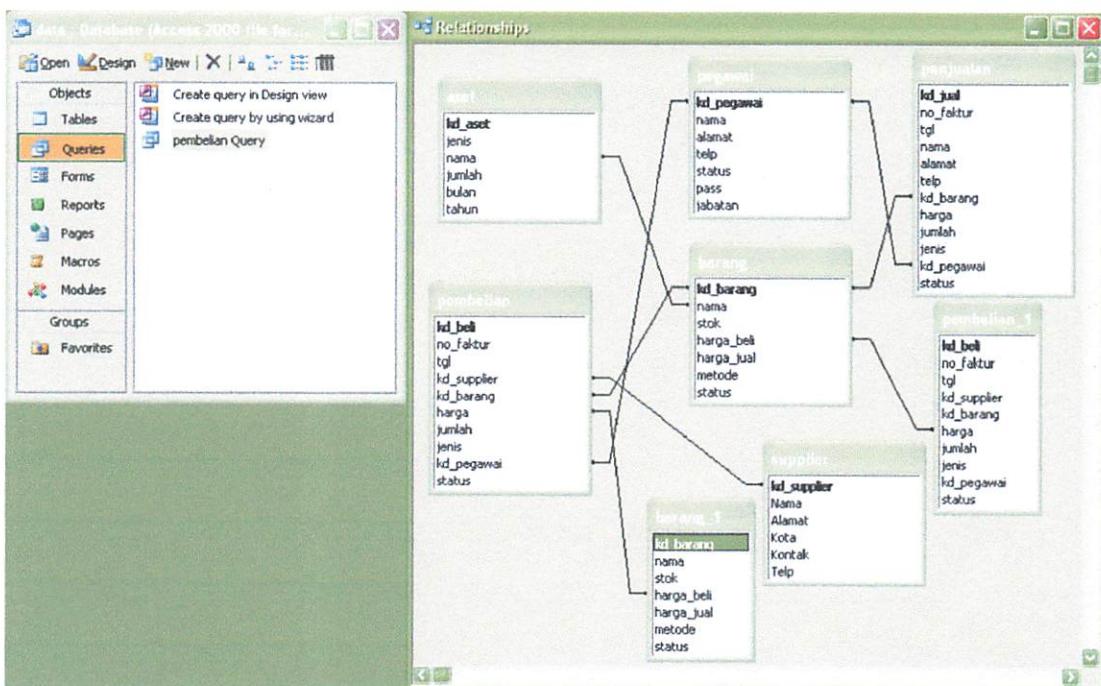
- ❖ Query, untuk mencari dan menampilkan data yang memenuhi syarat tertentu dari satu tabel atau lebih. Juga digunakan untuk meng-update atau menghapus beberapa *record* data pada suatu saat yang sama. Selain itu digunakan untuk menjalankan perhitungan terhadap sekelompok data.



Gambar 2.3. Query

- ❖ Relationships, untuk menghubungkan antar tabel pada *database* sehingga dapat digambarkan dalam bentuk fisik yang paling umum dengan sebutan Diagram Relasi Entitas (*Entity Relationship Diagram/ERD*) dimana pada dasarnya diagram ini dibuat sebagai gambaran tentang hubungan antar atribut/field dalam data fisik/tabel/entitas yang terlibat pada setiap modul pemrograman yang sangat berguna sebagai referensi dalam pembuatan

program. dibawah ini digambarkan relasi antar tabel yang satu dengan yang lain.



Gambar 2.4. Relationships

2.3. Pengertian *Interface*

Interface adalah penghubung antara dua sistem atau alat. Media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Keluaran (*output*) dari suatu subsistem akan menjadi masukan (*input*) untuk subsistem lainnya dengan melalui penghubung. Dengan penghubung satu subsistem dapat terintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

Interface ini, meliputi:

1. Perangkat yang dipakai untuk mengerjakan sesuatu, dan perangkat yang secara tidak langsung mengontrol perangkat lunak.
2. Piranti *input* atau *output*.
3. Prosedur pemakaian perangkat.

Dalam terminologi perangkat lunak, *interface* bisa diartikan sebagai tampilan atau cara perangkat lunak bersangkutan berinteraksi dengan penggunanya. Sedangkan dalam terminologi perangkat keras, *interface* mengacu kepada standar yang digunakan oleh suatu *peripheral* tertentu untuk berhubungan dengan *peripheral* lainnya dalam satu sistem.

2.3.1. Borland Delphi 7

Delphi adalah salah satu bahasa pemrograman berbasis visual yang digunakan untuk membuat program aplikasi pada komputer. Bahasa pemrograman yang digunakan sebenarnya merupakan turunan dari bahasa pemrograman pascal, yang dahulu pada Delphi dikenal sebagai objek pascal. Delphi telah terbukti mampu menghasilkan *software-software* yang berkualitas, baik yang berskala besar maupun kecil, teknologi yang digunakanpun cukup *up-to-date*. Ada jenis pemrograman non-prosedural seperti pemrograman untuk kecerdasan buatan seperti bahasa Prolog. Delphi termasuk Keluarga Visual sekelas *Visual Basic*, *Visual C*, artinya perintah-perintah untuk membuat objek dapat dilakukan secara visual. Pemrograman ini memungkinkan penggunaannya berkreasi lebih baik dalam menghasilkan suatu program aplikasi, dimana pemrogram tinggal memilih objek apa yang ingin dimasukkan

kedalam *Form/Window*, lalu tingkah laku objek tersebut saat menerima *event/aksi* tinggal dibuat programnya. Delphi merupakan bahasa berorientasi objek, artinya nama objek, properti dan *methode/procedure* dikemas menjadi satu kemasan (*encapsulate*).

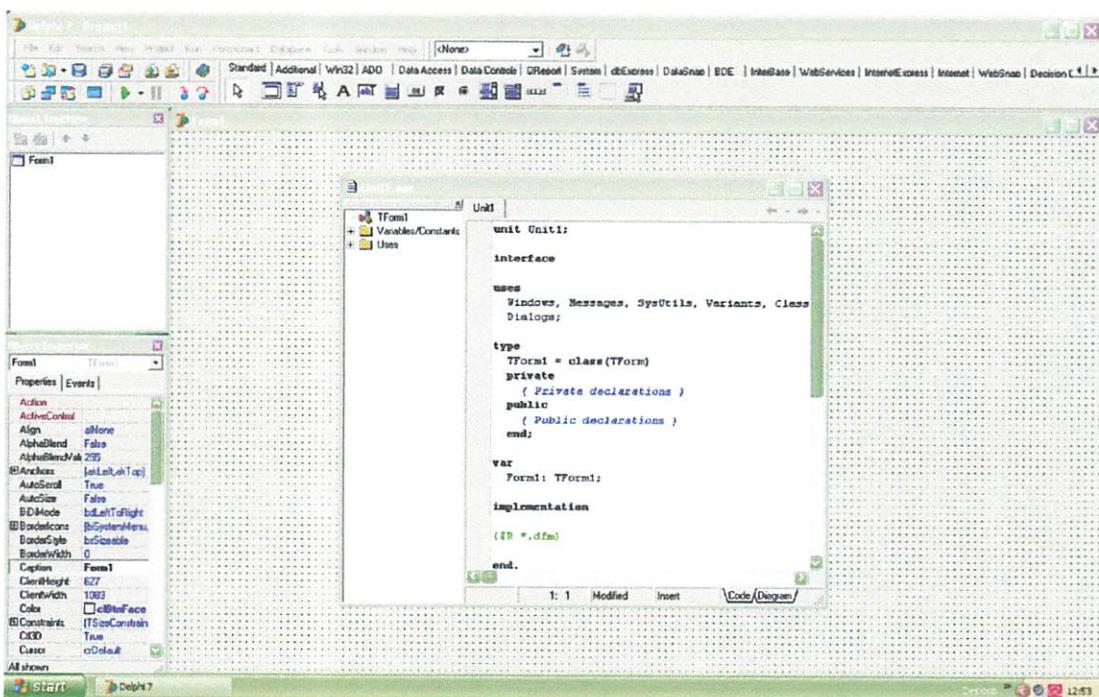
Kelebihannya terletak pada ketelitian dan aturan penulisan programnya, karena harus pakai prosedur dan deklarasi tiap-tiap *tool*. Kekurangannya terletak pada tingkat kepraktisannya, karena banyak orang yang memilih visual basic untuk kepraktisannya. Delphi disebut juga *visual programming* artinya komponen-komponen yang ada tidak hanya berupa teks (yang sebenarnya program kecil) tetapi muncul berupa gambar-gambar.

Form Delphi pada umumnya hanya ditampilkan 4 buah, tetapi sebenarnya Delphi masih menyediakan banyak *form* lainnya untuk mendukung pekerjaan pemrograman. Pada aplikasi digunakan antara lain :

- a. ***Form Menu***, *form* ini adalah *form* utama IDE Delphi, dengan melalui menu-menunya berfungsi untuk mengakses seluruh utilitas yang disediakan Delphi, baik yang berhubungan dengan pembuatan Aplikasi, pengaturan *file*, konfigurasi IDE, dan banyak *setting* lainnya.
- b. ***Form Designer***, *form* ini merupakan salah satu *form* utama Delphi, khususnya untuk pemrograman aplikasi berbasis *Window* (Delphi juga bisa untuk membuat aplikasi berbasis DOS). Fungsi utamanya menjadi *object parent* bagi komponen-komponen Delphi lainnya (media kanvas untuk melukis objek-objek lainnya).
- c. ***Object Inspector***, terdiri dari dua bagian. Pertama adalah *Properties* untuk mengatur *property* komponen yang sedang aktif. Kedua adalah *Event*

sebagai *trigger/pemicu* komponen yang bersangkutan untuk menjalankan perintah tertentu.

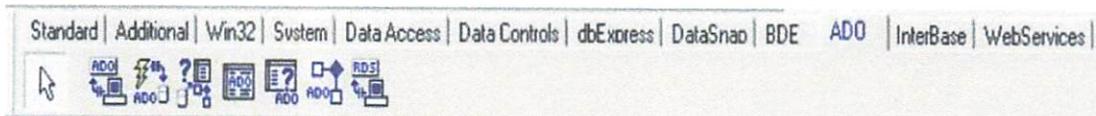
- d. *Unit Editor*, form ini berfungsi sebagai media *programmer* untuk mencurahkan ide-idenya, tentunya dalam sintaks-sintaks *programming* yang dimengerti oleh Delphi.
- e. *Object Treeview*, form ini adalah media navigasi bagi *programmer* untuk berpindah-pindah dari satu komponen ke komponen lainnya.



Gambar 2.5. Borland Delphi 7

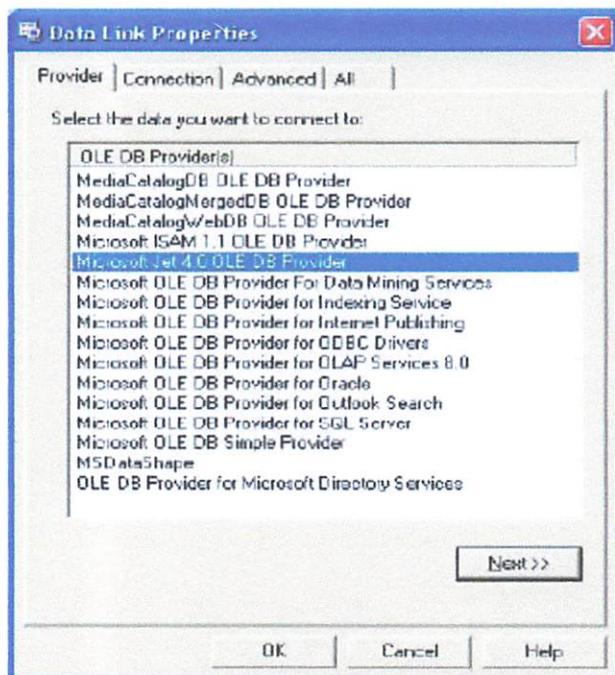
- f. *Component Palette*, komponen ini adalah peralatan yang telah disediakan oleh Delphi. Berfungsi sebagai kotak-kotak yang siap digunakan /ditempelkan pada media *form Designer*.

- g. **Data Access** dan **BDE** banyak berperan pada penentuan sumber data seperti tabel, *SQL*, *query* maupun koneksi sivitasnya. Contoh komponen ini yaitu *Table* dan *Datasource*.
- h. **Data Control** lebih menekankan pada pengaturan data yang terpasang pada aplikasi seperti DBLabel, DBNavigator, DBGrid, DBImage, DBMemo, DBEdit, DBCombobox dan lain sebagainya.
- i. **ADO** digunakan untuk komunikasi Program dengan Database *MS Access* sehingga data yang dibuat dengan *MS Access* akan dapat diintegrasikan dengan Delphi.



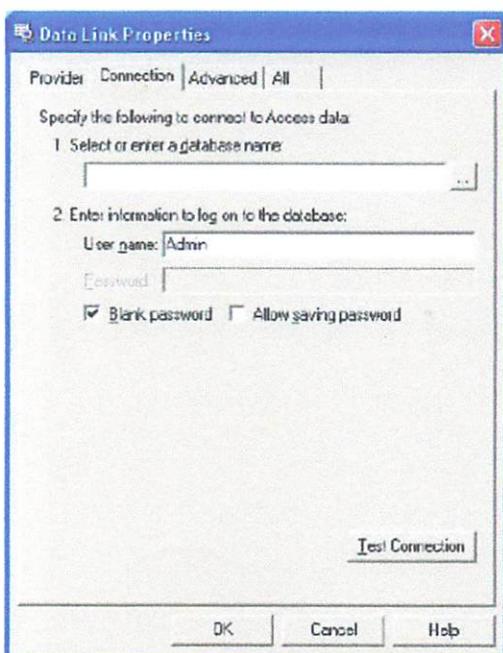
Gambar 2.6. ADO pada Component Palette

- j. Pada pengkoneksian *database* digunakan komponen ADO Connection dengan jalur konesinya memakai Jet 4.0 OLE Provider.



Gambar 2.7. Jet 4.0 OLE Provider

Setelah penentuan jalur koneksi telah dilakukan, maka dilakukan pengkonfigurasian koneksi dan pemilihan database yang akan digunakan.



Gambar 2.8. Pemilihan Database

2.4. Perangkat Analisis dan Perancangan

2.4.1. Diagram Aliran Data

Diagram Aliran Data atau yang biasa disebut dengan *DFD (Data Flow Diagram)* merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program.

Ada beberapa simbol DFD yang banyak dipakai, yaitu :

No.	Simbol	Penjelasan
01		Kesatuan luar adalah lingkungan luar sistem yang dapat berupa orang, departemen atau sistem lain yang memberikan input ataupun menerima output dari sistem.
02		Arus data adalah aliran data yang mengalir diantara proses, simpanan data dan kesatuan luar.
03		Proses adalah kerja atau kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk kedalam proses untuk dihasilkan arus data yang akan keluar dari proses.
04		Simpanan Data adalah merupakan simpanan data yang berupa file.

Gambar 2.9. Simbol Utama DFD

2.4.2. Pemodelan Data

Model data adalah sekumpulan cara / peralatan / *tool* untuk mendeskripsikan data-data, hubungannya satu sama lain, semantiknya, serta batasan konsistensi.

Ada dua model data, yaitu : *Entity Relationship Diagram* (ERD) dan model relasional. Keduanya menyediakan cara untuk mendeskripsikan perancangan basis data pada peringkat logika.

Model ERD atau *Conceptual Data Model* (CDM) adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (*entity*) serta hubungan (*relationship*) antara entitas-entitas itu.

Model Relasional atau *Physical Data Model* (PDM) adalah model yang menggunakan sejumlah tabel untuk menggambarkan data serta hubungan antara data-data tersebut. Setiap tabel mempunyai sejumlah kolom di mana setiap kolom memiliki nama yang unik.

Di dalam ERD atau CDM maupun PDM, relasi (hubungan) setiap entitas mempunyai derajat hubungan (kardinalitas) yang menunjukkan jumlah maksimum entitas yang dapat berelasi dengan entitas yang lain. Relasi kardinalitas yang terjadi di antara dua himpunan entitas dapat berupa :

- a. 1 ke 1 (*one to one*), setiap entitas pada suatu himpunan entitas berhubungan paling banyak satu entitas pada himpunan entitas yang lain, begitu juga sebaliknya.
- b. 1 ke N (*one to many*), setiap entitas berhubungan dengan banyak entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.

- c. N ke 1 (*many to one*), setiap entitas berhubungan dengan paling banyak satu entitas pada himpunan entitas yang lain, tetapi tidak sebaliknya.
- d. N ke N (*many to many*), setiap entitas pada suatu himpunan dapat berhubungan dengan entitas pada himpunan entitas yang lain, demikian sebaliknya.

BAB III

DESAIN DAN ANALISA SISTEM

3.1. Deskripsi Sistem

Sistem Informasi Akuntansi berguna untuk menghasilkan informasi yang dibutuhkan untuk pengambilan keputusan, maka sistem informasi akuntansi harus dapat mengumpulkan data transaksi dan data-data lain dan memasukkannya ke dalam sistem tersebut, dapat memproses dan menyimpan data tersebut untuk masa yang akan datang, dan menyediakan informasi yang dibutuhkan oleh pemakai berupa laporan yang menghasilkan informasi yang dapat diandalkan sebagai pengambilan keputusan terhadap perusahaan.

Tujuan dari sistem ini adalah untuk mengubah sistem akuntansi yang dulunya manual ke sistem yang telah ter-komputerisasi dalam pemrosesan data dengan pengoperasian yang lebih mudah dan relatif cepat dalam pekerjaan dan terstruktur, penyimpanan data yang lebih aman dan lebih lengkap, penyediaan proses pelaporan informasi akuntansi yang lebih tepat dan akurat. Data yang diolah dalam sistem ini adalah data transaksi yang terjadi pada setiap harinya sehingga menghasilkan pelaporan yang jelas tepat dan akurat sehingga nantinya bisa dipakai sebagai pengambil keputusan atas usaha dagang yang dijalankan.

3.2. Spesifikasi Sistem

Sebelum melakukan desain aplikasi, dilakukan survei ke beberapa usaha dagang (*Home Industry*) di Kabupaten Sumenep yang masih menggunakan sistem pencatatan akuntansi secara manual. Hasil survei digunakan sebagai pedoman dalam mendesain aplikasi yang dibutuhkan usaha dagang secara umum.

Dari hasil analisa dan survei maka dirancang desain aplikasi yang telah disesuaikan dengan kebutuhan usaha dagang dalam pelaksanaan bisnisnya, berupa *form*:

1. Pembelian, sebagai pengolah data pada transaksi pembelian barang.
2. Penjualan, sebagai pengolah data pada transaksi penjualan barang.
3. Aset, sebagai pencatat aset usaha dagang.
4. Supplier, sebagai pencatat data supplier.
5. Pegawai, sebagai pencatat data pegawai.
6. Barang, sebagai pencatat semua data barang.
7. Laporan, sebagai kegiatan pelaporan untuk pengambilan keputusan nantinya.

Form yang telah didesain nantinya akan dijadikan sebagai program yang telah sesuai dengan tingkat kebutuhan *home industry* (usaha dagang) berdasarkan hasil survey.

3.3. Perancangan Sistem

3.3.1. Data Flow Diagram (DFD)

Untuk menggambarkan aliran data pada sistem digunakan *Data Flow Diagram* (DFD). DFD merupakan alur data dengan konsep penggambaran analisa maupun rancangan sistem, diagram tersebut memperlihatkan dari mana data dimasukkan dan data apa yang dihasilkan dari setiap proses. Pembuatan DFD dimulai dari *context diagram* yang menggambarkan garis besar proses yang terdapat pada Gambar 3.1.

Kesatuan terluar atau *entity* yang terlibat dalam sistem perusahaan dagang pada Gambar 3.1 secara umum ada lima yaitu: Supplier sebagai pihak yang memasok barang, Pelanggan sebagai pihak yang membeli barang, Administrasi sebagai pihak yang mencatat transaksi mengenai barang dagang pada transaksi pembelian dan transaksi penjualan, Kepala Gudang sebagai pihak yang mengatur barang masuk dan keluar, dan Pemilik sebagai pihak yang berkepentingan untuk mendapat laporan dari seluruh proses yang terjadi dalam usaha tersebut.

Setelah pembuatan *context diagram*, dilanjutkan untuk memecah penggambaran aliran data menjadi bagian yang lebih detail dengan pembuatan DFD level 0 dan dapat dilihat pada Gambar 3.2.

Pada Gambar 3.2, sistem dapat dibagi dalam lima proses besar, yaitu: login, pembelian, penjualan, laporan, dan barang. Proses login adalah proses pemilihan hak akses.

Proses pembelian adalah proses yang berhubungan dengan aliran data pembelian barang, sistem pembayaran dan data supplier. Proses penjualan adalah

proses aliran data transaksi penjualan yang melibatkan data pelanggan. Proses barang adalah proses pengolahan data barang baik masuk maupun keluar yang nantinya akan mengolah persediaan barang. Proses laporan adalah proses yang berhubungan dengan pelaporan setiap transaksi yang berlangsung selama bisnis berjalan.

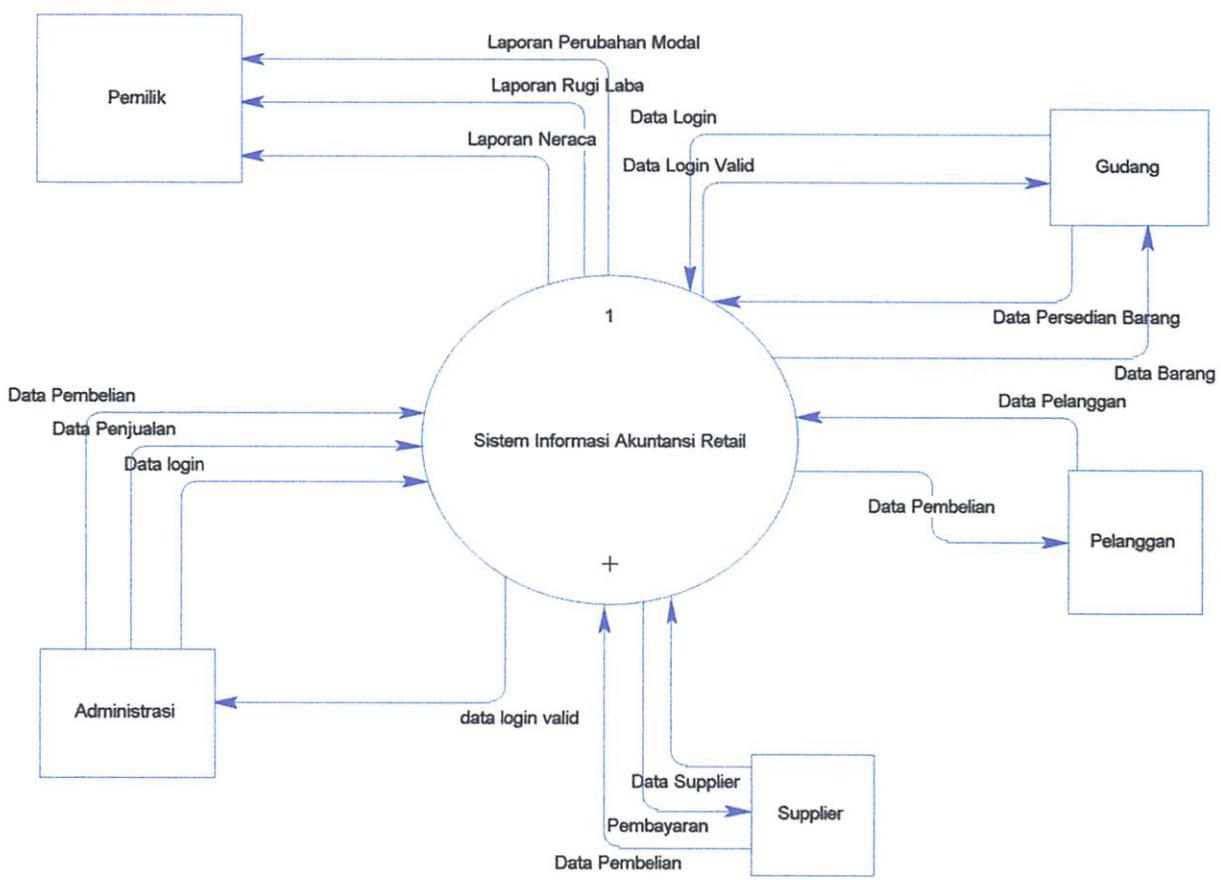
Langkah selanjutnya pengembangan proses login dengan menggambarkan proses aliran data dalam pemvalidasian data login yang berhubungan dengan data pegawai, seperti pada gambar 3.3.

Penjelasan Gambar 3.4. adalah penggambaran proses pembuatan laporan yang menghasilkan laporan laba rugi, laporan perubahan modal, dan laporan neraca. Pada proses ini melibatkan data transaksi pembelian, data transaksi penjualan, data barang, dan data aset.

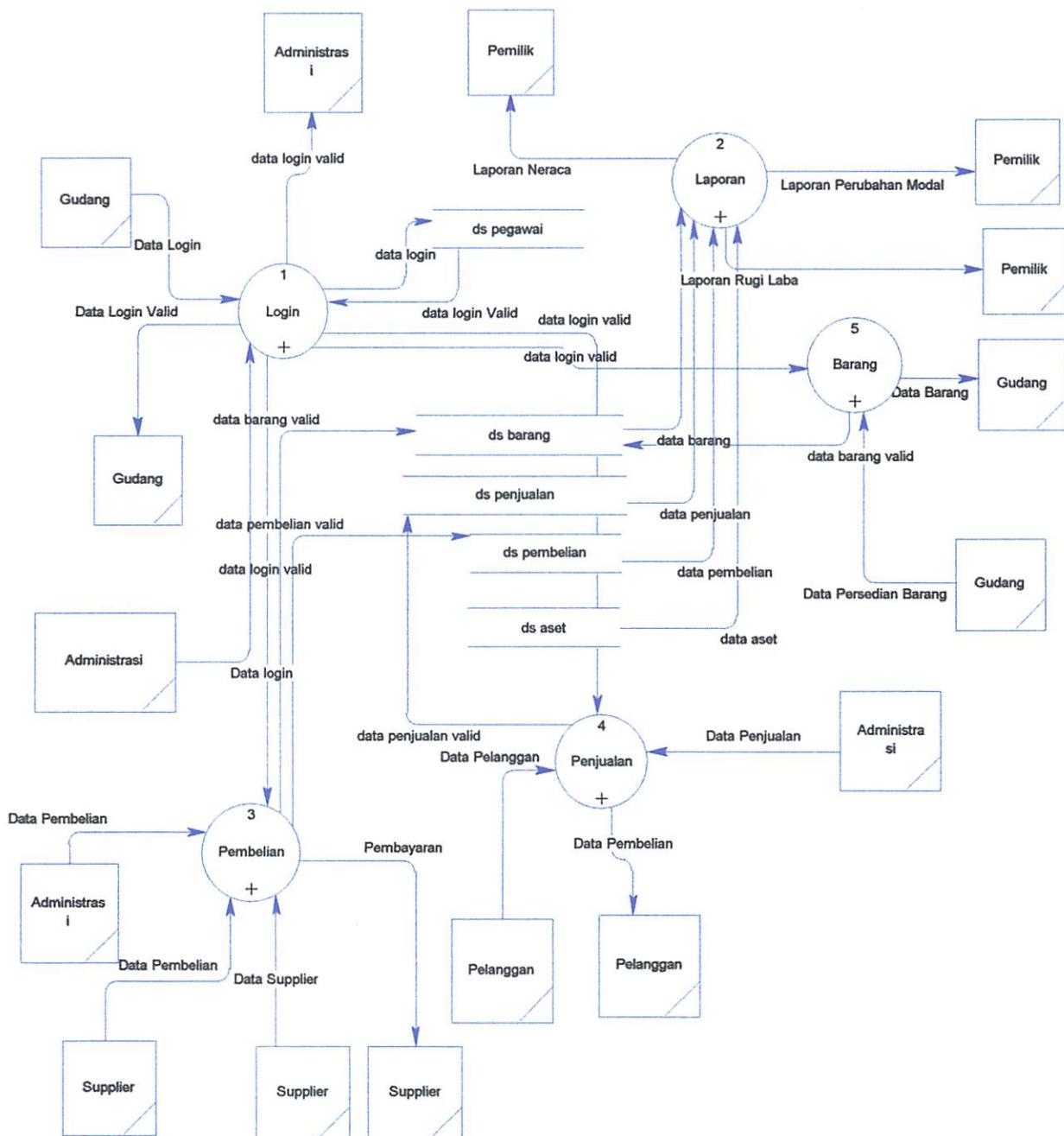
Proses barang dikembangkan lagi dengan proses pemvalidasian data barang, dimana data barang dambil dari data gudang yang nantinya akan divalidasi kemudian disimpan di database barang, seperti ditunjukkan pada gambar 3.5.

Pemvalidasian proses pembelian dengan menginputkan data transaksi pembelian dengan disertakan data pembayaran dan data supplier, kemudian data akan disimpan pada database barang dan pembelian, seperti pada gambar 3.6.

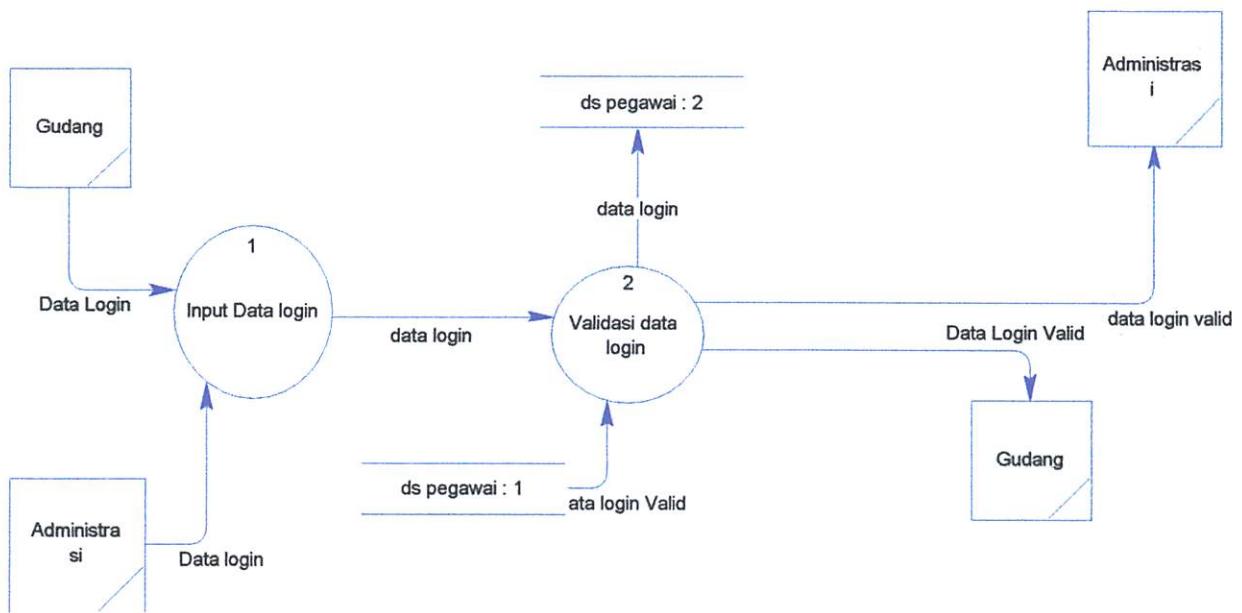
Pada gambar 3.7 ditunjukkan pemrosesan data transaksi penjualan yang melibatkan data pelanggan, data pembelian serta data penjualan yang akan divalidasi kemudian disimpan pada database penjualan.



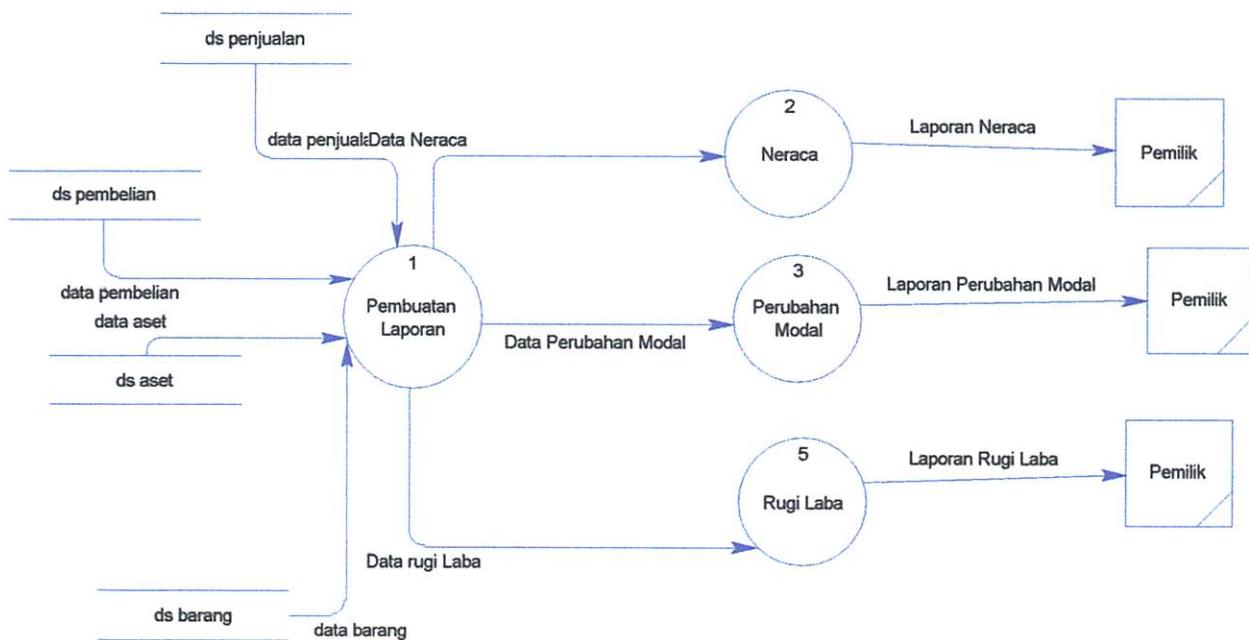
Gambar 3.1. *Conteks Diagram* Sistem Informasi Akuntansi



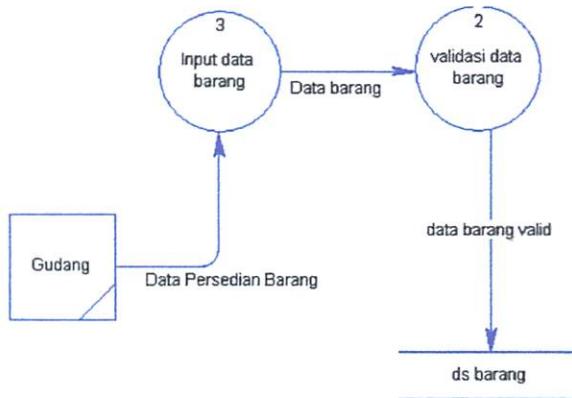
Gambar 3.2. DFD level 0 Sistem Informasi Akuntansi



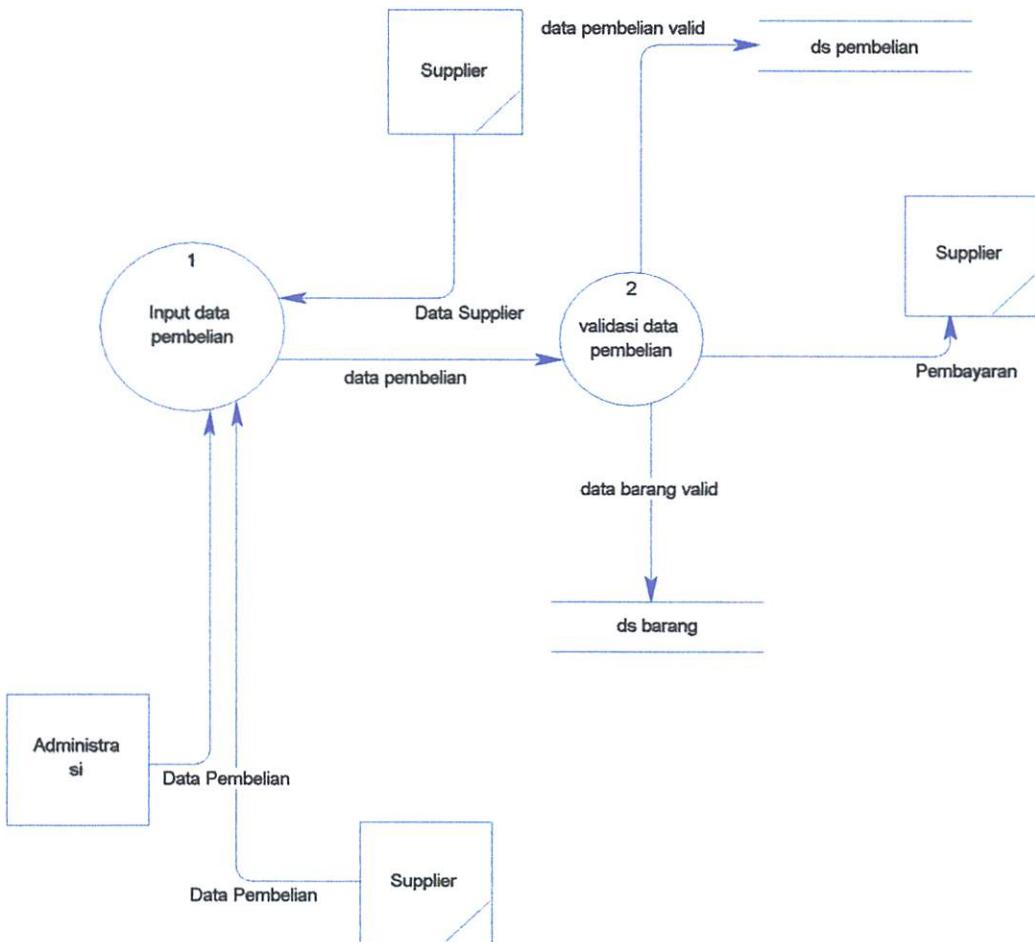
Gambar 3.3. DFD level 1 Validasi Login



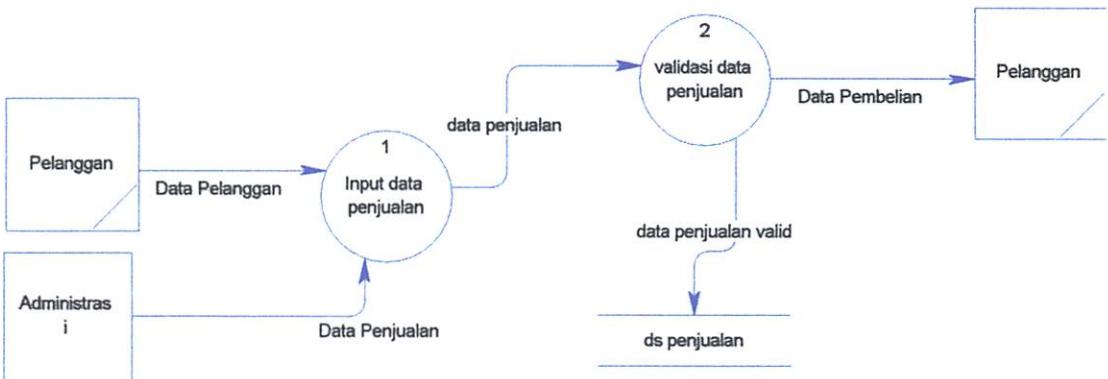
Gambar 3.4. DFD level1 Pembuatan Laporan



Gambar 3.5. DFD level1 Pemvalidasian Data Barang



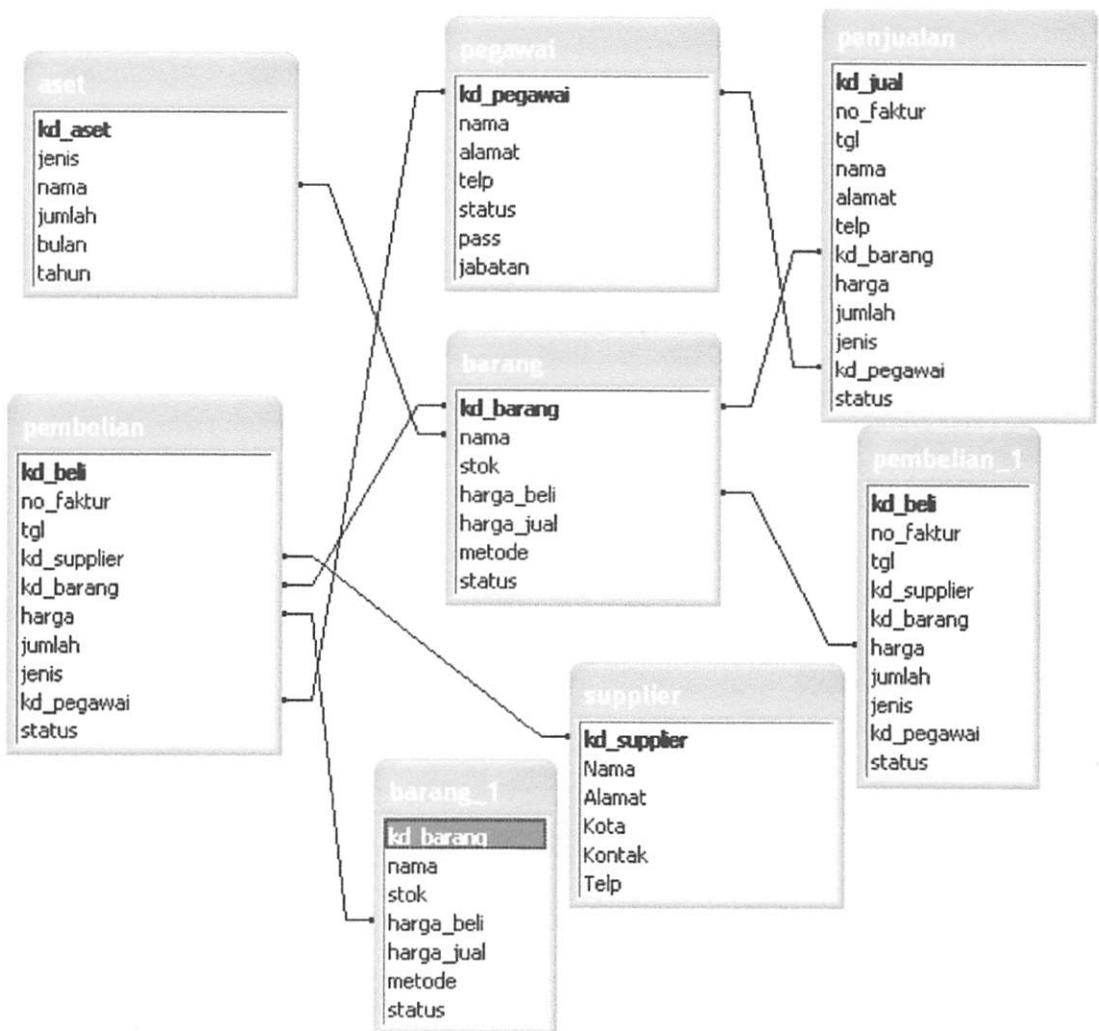
Gambar 3.6. DFD level 1 Proses Pembelian



Gambar 3.7. DFD level 1 Proses Penjualan

3.3.2. Pemodelan Data Relasional

Relasi atau hubungan antar tabel pada *database* Aplikasi Akuntansi dapat digambarkan dalam bentuk fisik yang paling umum adalah dengan menggunakan Diagram Relasi Entitas (*Entity Relationship Diagram/ERD*) dimana pada dasarnya diagram ini dibuat sebagai gambaran tentang hubungan antar atribut/field dalam data fisik/tabel/entitas yang terlibat pada setiap modul pemrograman yang sangat berguna sebagai referensi dalam pembuatan program. dibawah ini digambarkan relasi antar tabel yang satu dengan yang lain.



Gambar 3.8. Entity Relationship Diagram (ERD)

3.4. Desain Tabel Pembentukan Sistem

Dalam perancangan sistem ini, ada beberapa tabel yang digunakan berikut ini adalah struktur tabel yang digunakan pada *database*:

1. Tabel Aset

Tabel ini digunakan untuk menyimpan data aset yang ada selama usaha berjalan, mulai dari awal bisnis hingga akhir periode. Berisi informasi yang diperlukan tentang asset yang ada dalam usaha dagang.

Tabel 3.1. Tabel Aset

Field	Tipe Data	Ukuran	Key
kd_Aset	AutoNumber	LongInteger	Primari Key
jenis	Text	20	
nama	Text	30	
jumlah	Currency	Fixed	
bulan	Number	LongInteger	
tahun	Number	LongInteger	

2. Tabel Barang

Tabel ini digunakan untuk menyimpan data barang baik yang masuk maupun pengurangan stok barang, berisi informasi yang diperlukan tentang data barang yang tersedia beserta harga jual, harga beli dan stoknya.

Tabel 3.2. Tabel Barang

Field	Tipe Data	Ukuran	Key
kd_barang	AutoNumber	LongInteger	Primari Key
Nama	Text	50	
Stok	Number	LongInteger	
Harga_beli	Currency	Fixed	
Harga_jual	Currency	Fixed	
Metode	Text	5	
Status	Text	1	

3. Tabel Pegawai

Tabel ini digunakan untuk menyimpan data pengguna sistem, berisi informasi yang diperlukan tentang pengguna sistem aplikasi

Tabel 3.3. Tabel Pegawai

Field	Tipe Data	Ukuran	Key
kd_pegawai	Text	30	Primari Key
nama	Text	255	
alamat	Text	255	
telp	Text	255	
status	Text	1	
pass	Text	255	

4. Tabel Supplier

Tabel ini digunakan untuk menyimpan data pemasok barang dagang, berisi informasi yang diperlukan tentang profil pemasok barang.

Tabel 3.4. Tabel Supplier

Field	Tipe Data	Ukuran	Key
kd_supplier	AutoNumber	LongInteger	Primari Key
nama	Text	25	
alamat	Text	50	
kota	Text	15	
kontak	Text	20	
telp	Text	25	

5. Tabel Pembelian

Tabel ini digunakan untuk menyimpan data pembelian barang selama transaksi, berisi informasi yang diperlukan tentang transaksi pembelian.

Tabel 3.5. Tabel Pembelian

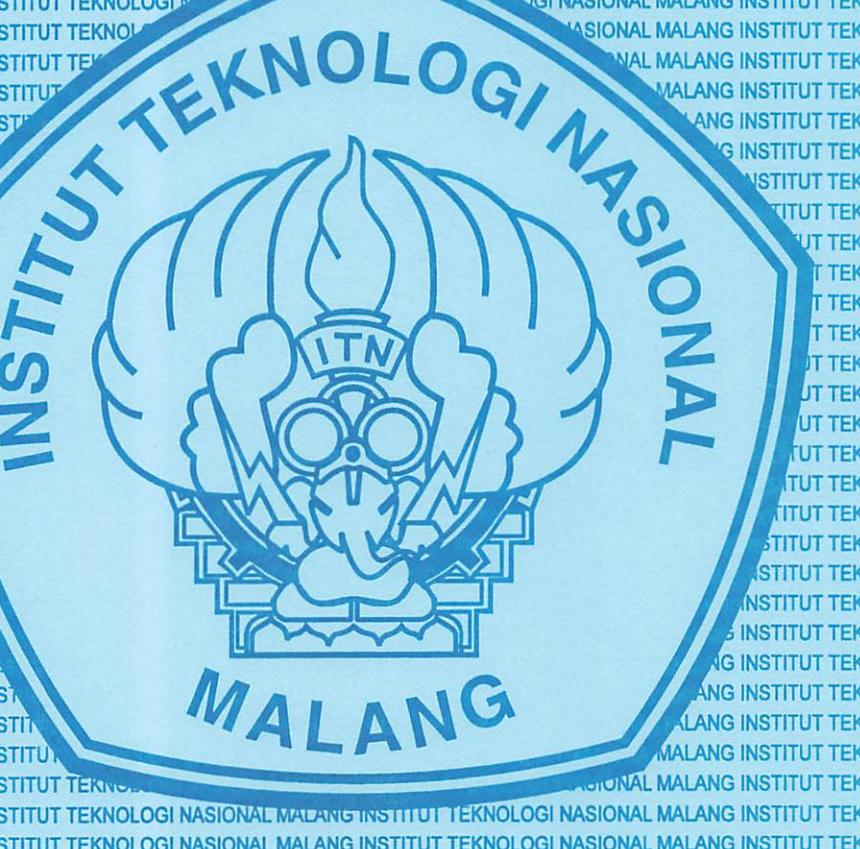
Field	Tipe Data	Ukuran	Key
kd_beli	AutoNumber	LongInteger	Primari Key
no_faktur	Text	30	
tgl	Date/Time	Short Date	
kd_supplier	Number	LongInteger	
kd_barang	Number	LongInteger	
harga	Currency	Fixed	
jumlah	Number	LongInteger	GeneralNumber
jenis	Text	255	
kd_pegawai	Text	30	
status	Text	1	

6. Tabel Penjualan

Tabel ini digunakan untuk menyimpan data pembelian barang selama transaksi, berisi informasi yang diperlukan tentang transaksi pembelian.

Tabel 3.6. Tabel Penjualan

Field	Tipe Data	Ukuran	Key
kd_jual	AutoNumber	LongInteger	Primari Key
no_faktur	Text	30	
Tgl	Date/Time	Short Date	
Nama	Text	50	
Alamat	Text	70	
Telp	Text	20	
kd_barang	Number	LongInteger	
Harga	Currency	Fixed	
Jumlah	Number	LongInteger	
Jenis	Text	255	



BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

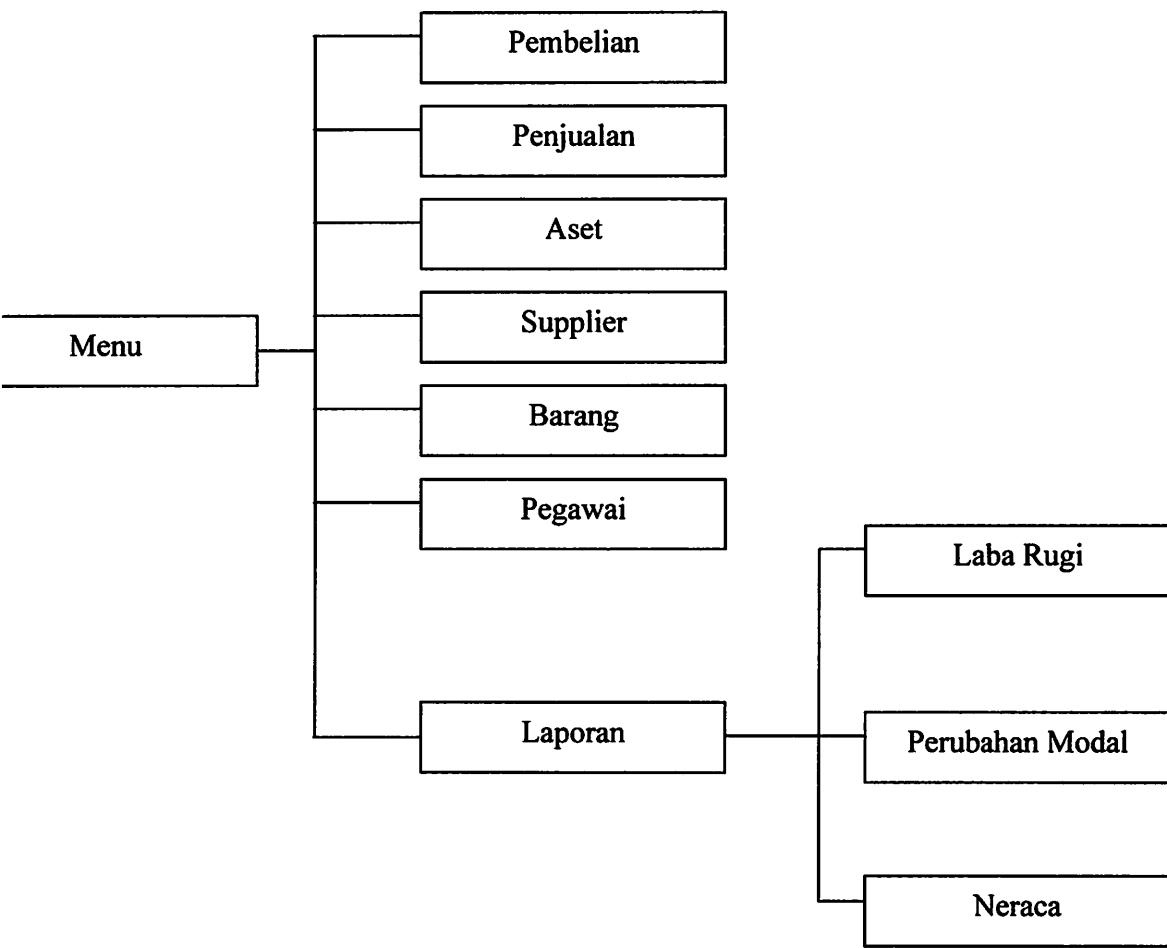
4.1. Hasil Perancangan Sistem

Implementasi sistem dilakukan dengan menggunakan *Borland Delphi 7*. Sedangkan untuk *database*, digunakan *Microsoft Office Access 2003*. Program ini dirancang dengan penampilan yang menarik dan dengan tingkat pemakaian yang sangat mudah bagi pengguna yang belum bisa memahaminya, sehingga memungkinkan sumber daya manusia yang belum paham tentang teknologi untuk memakai dan mengoperasikannya.

Dengan perancangan aplikasi yang mudah dipahami dan dijalankan, aplikasi ini mempunyai fitur atau struktur menu yang sangat mudah dipahami dalam pengisian dan pengoperasiannya, diantaranya :

1. Menu Pembelian, pada menu pembelian tersedia cara pembelian baik tunai atau kredit, juga telah tersedia faktur pembelian.
2. Menu Penjualan, pada menu penjualan tersedia cara penjualan baik tunai atau kredit, juga telah tersedia faktur penjualan.
3. Menu Aset, tersedia jenis-jenis aset yang dipergunakan untuk menjalankan usaha dagang beserta totalnya.
4. Menu Supplier, sebagai tempat penyimpanan data Supplier.
5. Menu Barang, tersedia penyimpanan data barang dengan penentuan harga jual yang telah diperhitungkan dari harga beli barang.
6. Menu Pegawai, sebagai tempat penyimpanan data pegawai.

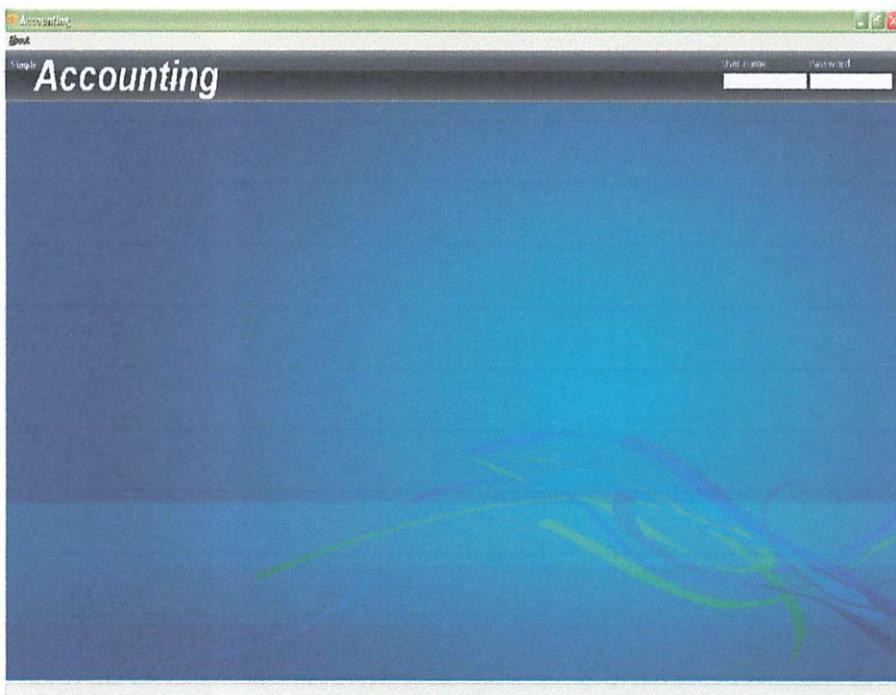
7. Menu Laporan yang terdiri dari laporan laba rugi, laporan perubahan modal, dan laporan neraca. Pada laporan ini akan diperhitungkan periode dari akun tersebut.



Gambar 4.1. Struktur Menu Aplikasi

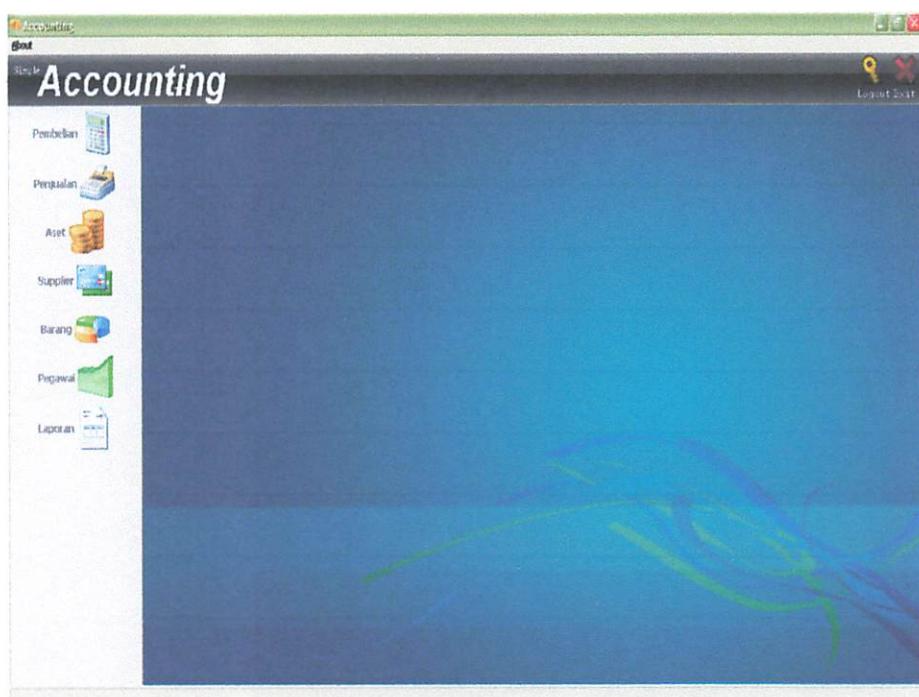
4.2. Pengujian Aplikasi Akuntansi

Berikut ini adalah pengujian terhadap sistem yang dikembangkan. Pada saat aplikasi pertama kali dijalankan, maka pengguna akan diminta melakukan pengisian terhadap *username* dan *password* yang disediakan. Pengisian pertama kali adalah pengisian nama dan password pengguna seperti terlihat pada Gambar 4.2.



Gambar 4.2. Pengisian User dan Password

Pengisian berikutnya adalah masuk menu utama setelah pengisian pada *username* dan *password* telah berhasil, dimana pengguna dapat memilih untuk menggunakan menu-menu yang telah disediakan oleh aplikasi. Pada tahap ini pengguna bisa langsung dapat memulai menjalankan aplikasi seperti terlihat pada Gambar 4.3.

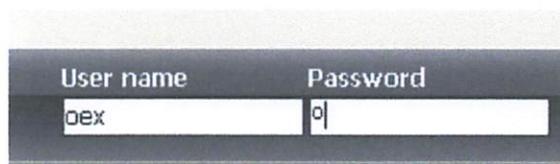


Gambar 4.3. Struktur Menu Utama

4.3. Pengujian Entri data

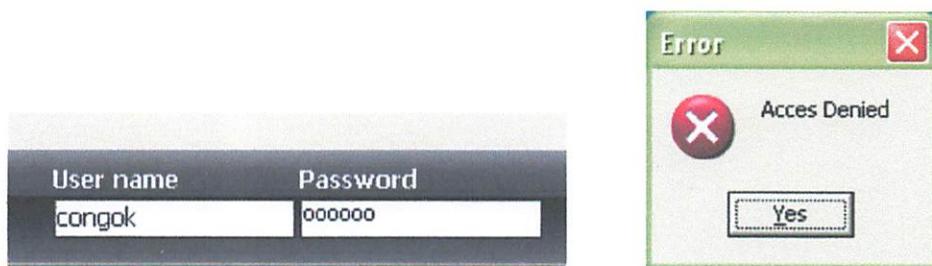
4.3.1. Form Login

Halaman utama pemakaian aplikasi ini akan langsung ditampilkan *form login*, yang bertujuan untuk menentukan hak akses pemakai. Pada *form* ini pemakai akan memasukkan *username* dan *password* sesuai dengan data pegawai yang ada seperti terlihat pada gambar 4.4.



Gambar 4.4. *form login*

Apabila terjadi kesalahan pengisian, maka akan muncul pesan seperti gambar di bawah ini :



Gambar 4.5. Pesan *Error* pada kesalahan

Aplikasi akan langsung menyatakan *Error* jika *username* atau *password* yang dimasukkan salah atau asal-asalan, karena tidak sama atau tidak ditemukan di dalam data pegawai pada *database*.

4.3.2. Form Pegawai

Data Pegawai akan diinputkan melalui *form* kosong yang ditunjukkan pada gambar dibawah ini.

Username	Nama	Alamat	Telp	Jabatan
oex	udiq	jl.kh.agus/664477	664477	Administrator
nuns	nunik	jl.bengawa	578343	Gudang
noedx	mohammad	jl.menuju	8976362	Manager

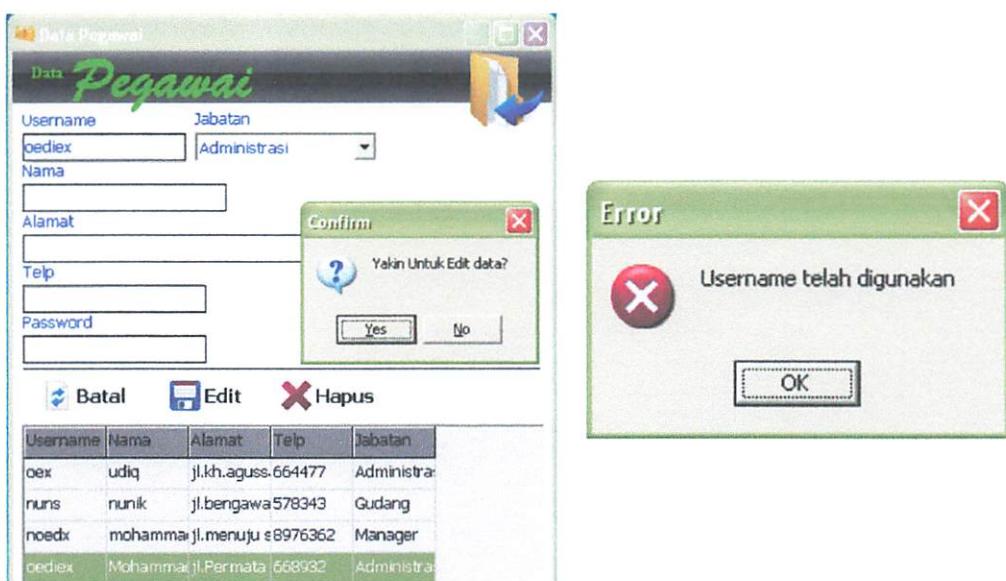
Gambar 4.6. Penginputan Data Pegawai

Pada *form* ini data pegawai bisa ditambah, diedit dan dihapus sesuai dengan permintaan pemakai, dicontohkan pada gambar di bawah ini :

Username	Nama	Alamat	Telp	Jabatan
oex	udiq	jl.kh.agus/664477	664477	Administrator
nuns	nunik	jl.bengawa	578343	Gudang
noedx	mohammad	jl.menuju	8976362	Manager

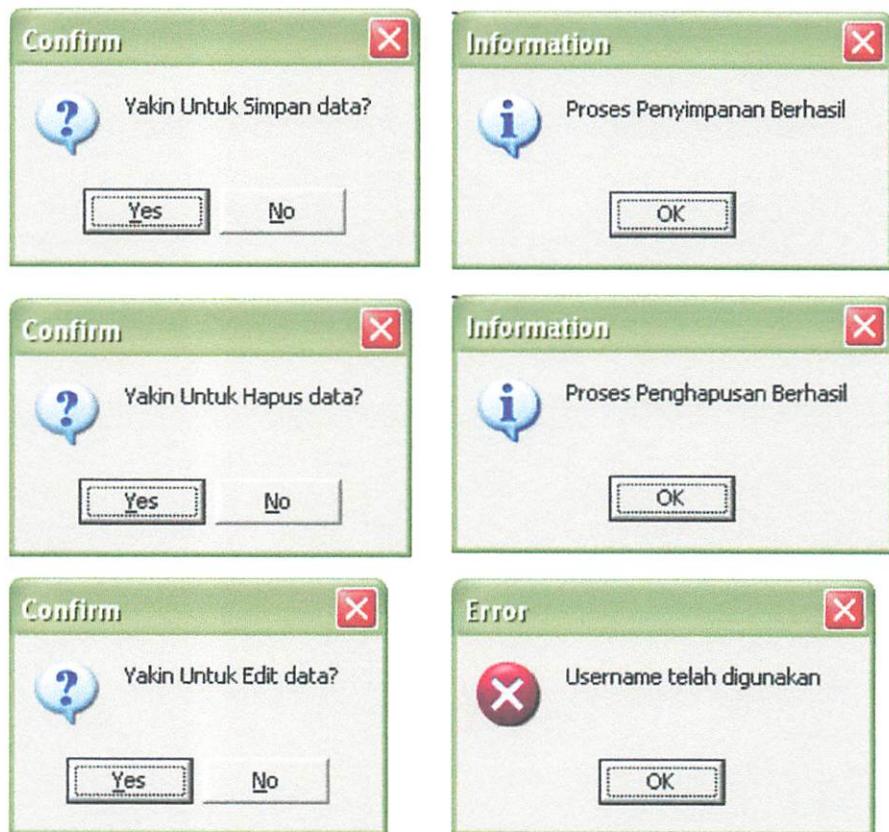
Gambar 4.7. Menyimpan data pegawai

Proses penyimpanan data pada gambar di atas akan menampilkan pesan penyimpanan setelah semua data sudah diisikan ke dalam *form*. Apabila pada pengan format data tidak sesuai dengan format pada *form* maka *form* akan langsung menghapus *inputan* tersebut secara otomatis. Pengeditan dan penghapusan data dilakukan dengan memilih terlebih dahulu data mana yang akan *diedit* atau dihapus. Di saat pemakai memilih data, *form* akan memunculkan tombol untuk mengedit atau menghapus data tersebut kemudian diikuti perintah atau pesan pengeditan atau penghapusan data. Jika ada *username* yang dikan sama atau telah ada pada *database* maka akan muncul peringatan bahwa *username* telah digunakan seperti pada gambar di bawah ini :



Gambar 4.8. Mengedit data dan peringatan penggunaan *username*

Pesan-pesan yang ditampilkan pada setiap *event* akan ditunjukkan pada gambar di bawah ini :



Gambar 4.9. Pesan Penyimpanan, penghapusan dan pengeditan data

4.3.3. Form Supplier

Data supplier diinputkan dan ditampilkan melalui *form* pada gambar dibawah ini. Pada isian nama diisikan nama suppliernya dan pada *contact* diisikan nama yang bisa dihubungi selama pemesanan atau pemasukan barang. Proses *penginputan*, *pengeditan* dan penghapusan data, sama dengan proses yang dilakukan pada *form* pegawai, begitu juga dengan pesan-pesan yang ditampilkan pada penyimpanan data, *pengeditan* data, penghapusan data serta jika ada data yang telah ada, seperti terlihat dalam gambar 4.10 dan gambar 4.11.

The screenshot shows a Windows application window titled "Data Supplier". Inside, there are input fields for "Nama" (Jaja Plastik), "Alamat" (Jl. Kota Baru 125), "Kota" (Pamekasan), "Contact" (Budi Hartono), and "Telp" (0324548932). Below these fields is a "Confirm" dialog box with the question "Yakin Untuk Simpan data?" (Are you sure you want to save the data?). Two buttons are visible: "Yes" and "No". At the bottom of the main window are two buttons: "Batal" (Cancel) and "Simpan" (Save). A table below lists 12 supplier entries with columns: Kode, Nama, Alamat, Kota, Contact, and Telp. The last entry in the table is highlighted.

Kode	Nama	Alamat	Kota	Contact	Telp
8	coca cola	Jl.mahakam 93 surabaya	obi	03133789	
9	sprite	Jl.mahlkota 10	sumenep	rosi	668793
10	bolovia souvenir	Jl.kasuar i blok	sumenep	ragil	665457
11	restu bunda	Jl.imam bonjol	sumenep	abi heidar	667654
12	Jajanan	Jl.Meranggi 45	Sumenep	Mamat	662891

Gambar 4.10. Form Supplier

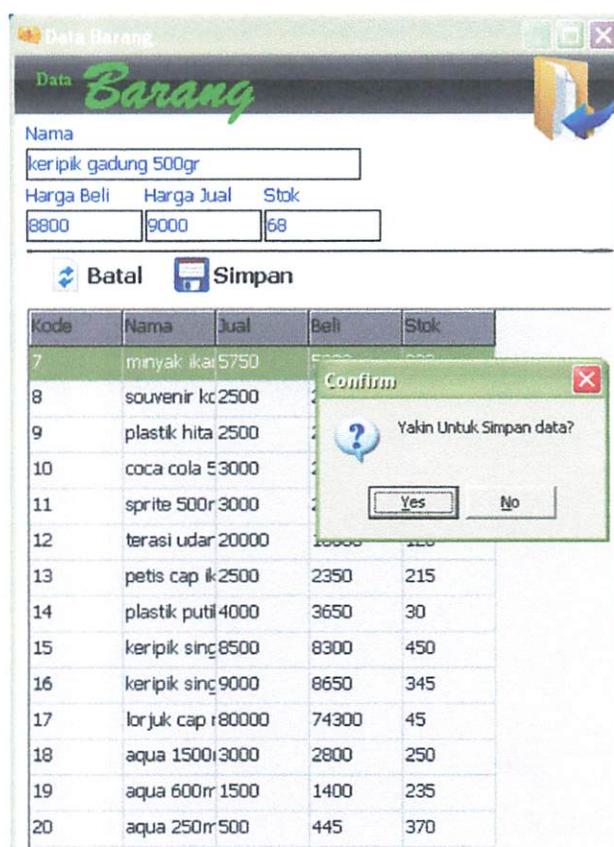
The screenshot shows a Windows application window titled "Data Supplier". Inside, there are input fields for "Nama" (Jaja Plastik), "Alamat" (Jl.pandian timur 75), "Kota" (sumenep), "Contact" (anshor), and "Telp" (665431). Below these fields is an "Error" dialog box with the message "Nama sudah digunakan" (Name is already used). Two buttons are visible: "OK" and "Batal". At the bottom of the main window is a "Batal" button. A table below lists 13 supplier entries with columns: Kode, Nama, Alamat, Kota, Contact, and Telp. The last entry in the table is highlighted.

Kode	Nama	Alamat	Kota	Contact	Telp
8	coca cola	Jl.mahakar surabaya	obi	03133789	
9	sprite	Jl.mahlkota sumenep	rosi	668793	
10	bolovia sou	Jl.kasuar i bsumenep	ragil	665457	
11	restu bund	Jl.imam boisumenep	abi heidar	667654	
12	Jajanan	Jl.Meranggi Sumenep	Mamat	662891	
13	Jaja Plastik	Jl. Kota Bar Pamekasan	Budi Hartono	032454893	

Gambar 4.11. Peringatan nama sama

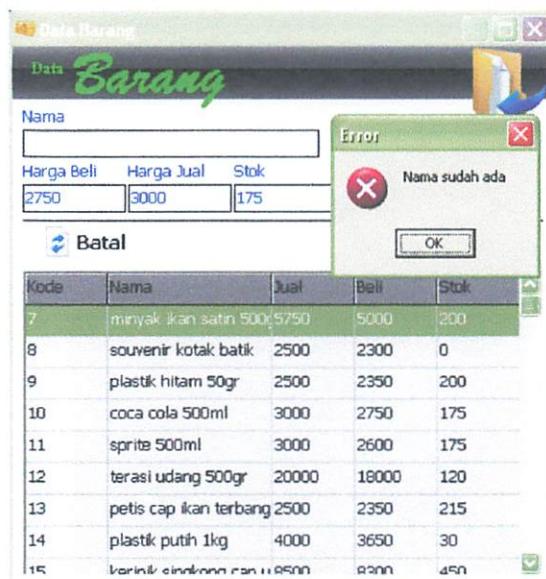
4.3.4. Form Data Barang

Barang dagang akan diinputkan pada *form* data barang, proses penginputan, pengeditan dan penghapusan dibuat sama dengan *form* pegawai dan supplier untuk memudahkan pemahaman jalannya aplikasi, serta jika telah ada data yang ada pada data yang telah diinputkan maka tersebut terdapat kesalahan, seperti terlihat dalam gambar 4.12.



Gambar 4.12. Form Data Barang

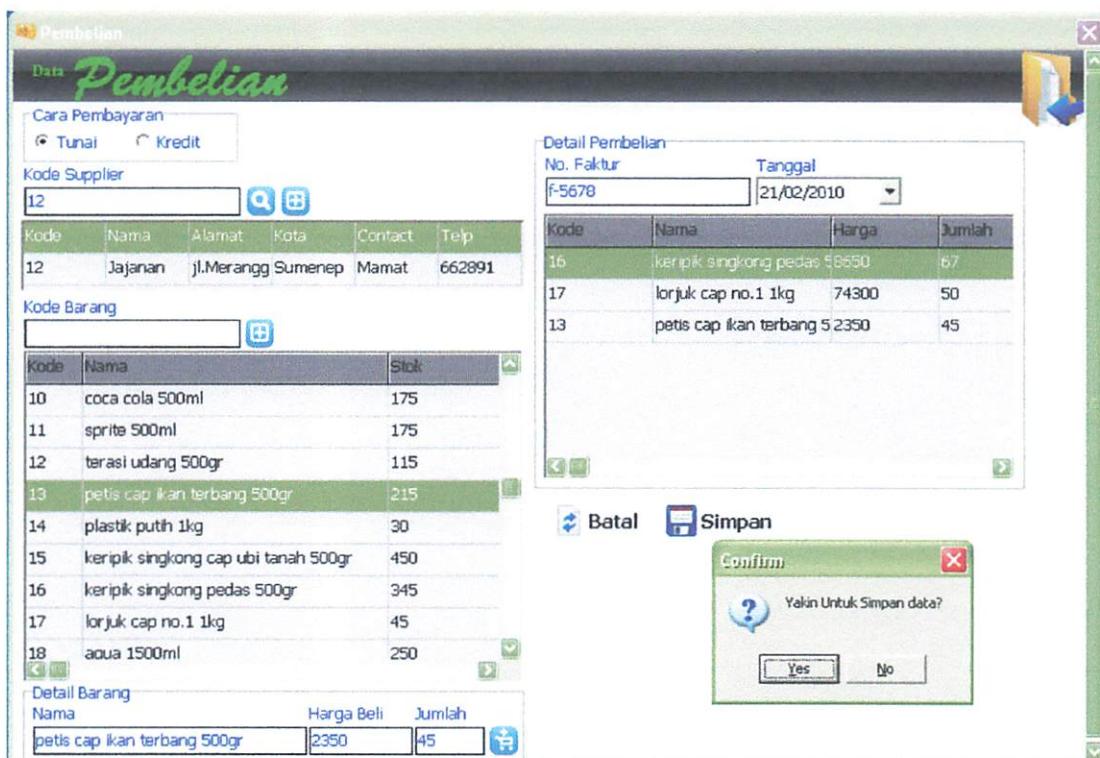
Jika nama barang yang dimasukkan sudah ada, maka muncul peringatan bahwa barang sudah ada, seperti pada gambar di bawah ini :



Gambar 4.13. Peringatan penggunaan nama yang sama

4.3.5. Form Pembelian

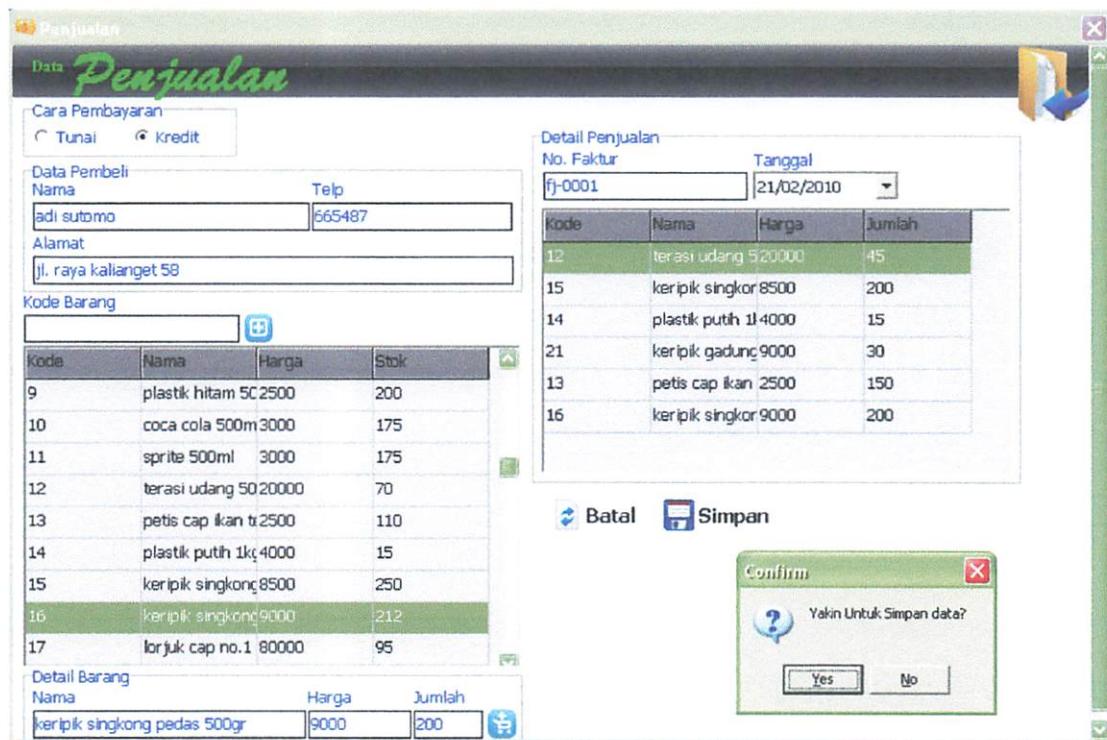
Pada form pembelian tersedia pembayaran secara tunai atau kredit, penampilan daftar supplier dan daftar barang juga faktur pembeliannya, pesan dan peringatan akan muncul bila tidak sesuai dengan sistem yang dijalankan, seperti terlihat dalam gambar 4.14.



Gambar 4.14. Form Pembelian

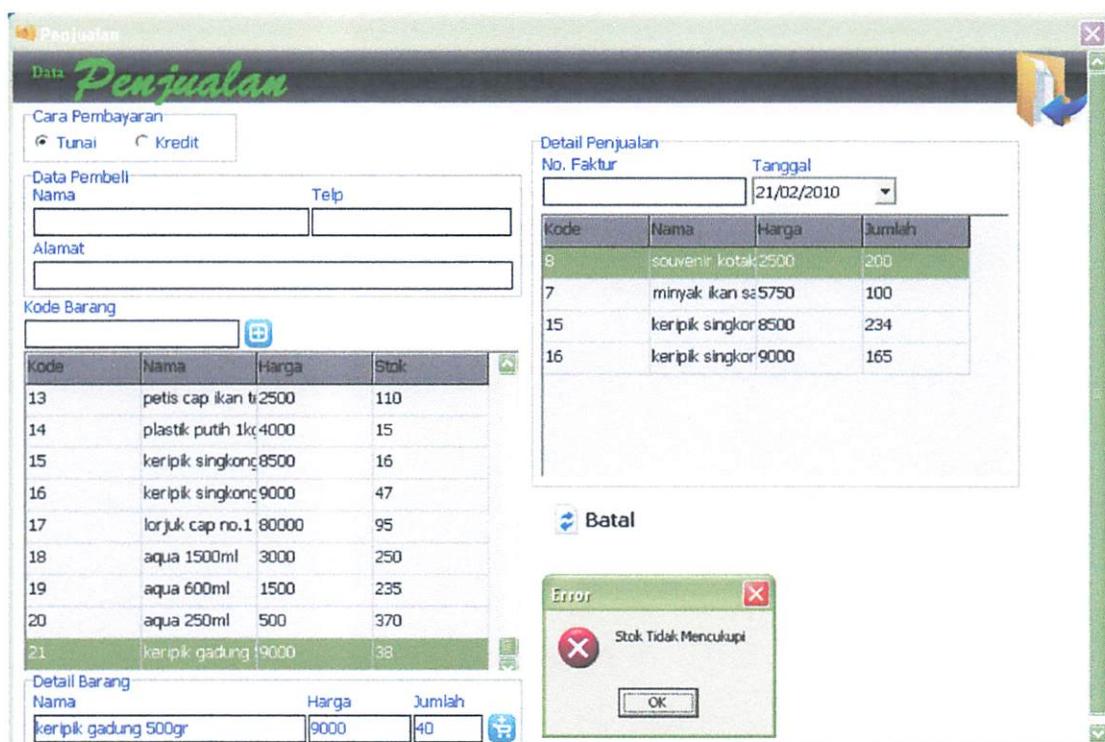
4.3.6. Form Penjualan

Form penjualan hampir sama dengan form pembelian yang akan menampilkan faktur penjualan, cara pembayaran, data pembeli, serta daftar barang yang telah dibeli seperti terlihat dalam gambar 4.15.



Gambar 4.15. Form Penjualan

Jika pada penjualan barang melebihi batas maksimal stok barang yang ada, maka pada *form* ini akan menampilkan peringatan stok tidak memenuhi seperti ditunjukkan pada gambar 4.16.



Gambar 4.16. Peringatan Stok

4.3.7. Aset Usaha Dagang

Pada form asset, akan menampilkan semua aset yang dipakai dalam pelaksanaan akuntansi pada usaha dagang. Apa saja aset yang dipergunakan dalam pelaksanaan selama pergerakan bisnis dapat disimpan dan ditampilkan di *form* ini, seperti terlihat dalam gambar 4.17.

The screenshot shows a Windows application window titled "Data Aset Perusahaan". The main form has fields for "Jenis Aset" (Aset Lancar), "Nama" (Perlengkapan Kantor), and "Jumlah" (7500000). Below the form is a "Confirm" dialog box asking "Yakin Untuk Simpan data?" (Are you sure to save data?) with "yes" and "No" buttons. At the bottom are "Batal" (Cancel) and "Simpan" (Save) buttons. To the right is a table listing assets:

Kode	Jenis	Nama	Jumlah
14	Aset Tetap	Gedung	20000000
15	Aset Tetap	Kendaraan Roda Empat	30000000
16	Aset Tetap	Kendaraan Roda Dua	20000000
17	Aset Lancar	Lapangan Parkir	3000000
18	Modal Awal	Modal Sendiri	100000000
19	Aset Lancar	Pick-up Carteran	20000000

Gambar 4.17. *Form* Aset Usaha Dagang

Pada proses pengeditan, penghapusan, dan penyimpanan sama dengan *form* lainnya. Pesan yang ditampilkan juga sama pada proses pengeditan, penghapusan dan penyimpanan data.

4.3.8. Laporan Laba Rugi

Form ini akan menampilkan bentuk laporan laba rugi beserta hasilnya, juga media print untuk mencetak hasilnya, seperti terlihat dalam gambar 4.18.

The screenshot shows a software application window titled "Laporan Laba Rugi". The main content is a profit and loss statement for the period of January 2010, dated May 16, 2010, in Malang. The statement details revenue and expenses, resulting in a profit of 106.340.

PT. ACCOUNTING		
LAPORAN PERUBAHAN RUGI LABA		
Periode Januari 2010		
Malang, 16/05/2010		
Pendapatan		
Penjualan	112.940	
Total Pendapatan	<u>112.940</u>	
Beban		
Harga Pokok Penjualan	6.600	
Total Beban	<u>6.600</u>	
Total Laba	<u><u>106.340</u></u>	

Gambar 4.18. Laporan Laba Rugi

4.3.9. Laporan Perubahan Modal

Form ini akan menampilkan bentuk laporan laba rugi beserta hasilnya, seperti terlihat dalam gambar 4.19.

The screenshot shows a Microsoft Word document window with the title bar 'Laporan Perubahan Modal'. The main content is a balance sheet for 'PT. ACCOUNTING' for the period 'Januari 2010'. The document includes a header with the company name and period, a date 'Malang, 16/05/2010', and a table with the following data:

PT. ACCOUNTING	
LAPORAN PERUBAHAN MODAL	
Periode Januari 2010	
Modal Awal	100.000.000
Laba Bersih	106.340
Penarikan Prive	0
Modal Akhir	100.106.340

Gambar 4.19. Laporan Perubahan Modal

4.3.10. Laporan Neraca

Form ini akan menampilkan bentuk laporan laba rugi beserta hasilnya, seperti terlihat dalam gambar 4.20.

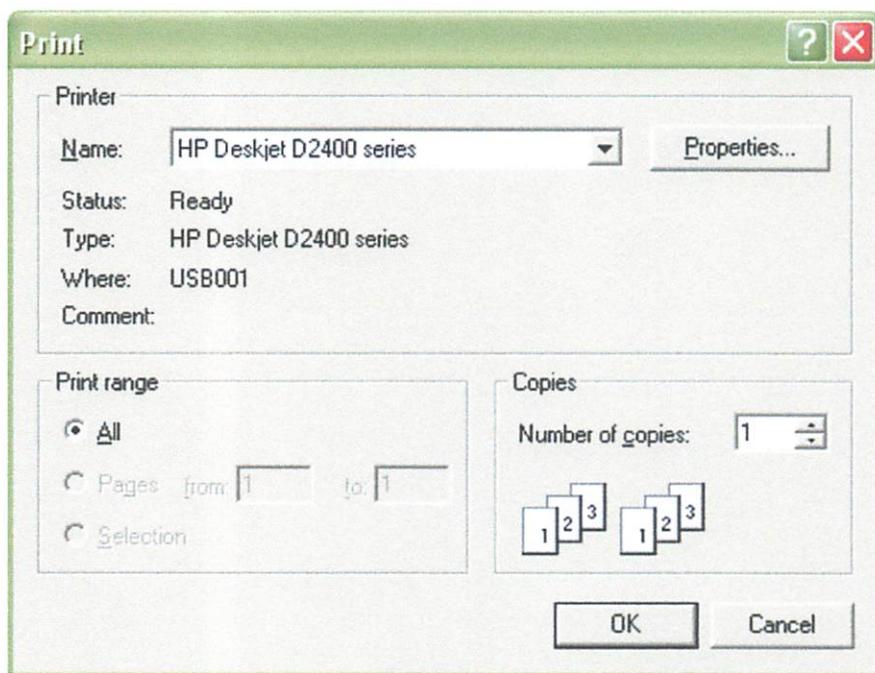
The screenshot shows a Microsoft Word document window with the title 'Laporan Neraca'. The document contains a balance sheet for PT. ACCOUNTING for the period of 2010. The title is centered at the top. Below it, the heading 'HARTA' is underlined. The balance sheet is divided into two main sections: 'Aset Lancar' (Current Assets) and 'Aset Tetap' (Fixed Assets). Each section lists individual assets with their respective values and a total for each section.

PT. ACCOUNTING	
LAPORAN NERACA	
PERIODE TAHUN 2010	
HARTA	
Aset Lancar	
Lapangan Parkir	3000000
Pick-up Carteran	2000000
Perlengkapan Kantor	7500000
Plutang	
TOTAL =	30.500.000
Aset Tetap	
Gedung	2000000
Kendaraan Roda Empat	3000000
Kendaraan Roda Dua	2000000
TOTAL =	70.000.000

Gambar 4.20. Laporan Neraca

4.3.11. Printer Setup

Pada setiap form laporan akan ada menu *setup printer* yang berguna untuk mengatur pencetakan laporan sesuai dengan keinginan. Pengaturan juga bisa pada jenis *printer* dan jenis kertas, seperti pada gambar 4.21.



Gambar 4.21. *Setup Printer*

4.4. Perbandingan Sistem *Paperbase* dengan Sistem Komputerisasi

Table 5.1. Perbandingan *Paperbase* dan Komputerisasi

No	Sistem <i>Paperbase</i>	Komputerisasi
1.	Pencatatan transaksi dan stok barang membutuhkan waktu $\pm 1,5$ jam dengan memperhatikan aspek data ganda dan stok barang yang tersedia.	Pencatatan transaksi dan stok barang membutuhkan waktu ± 15 menit dengan adanya system peringatan data ganda dan perhitungan stok barang, sehingga tidak mengkhawatirkan adanya nama barang ganda dengan melebihi stok barang yang tersedia.
2.	Perhitungan laba rugi serta pelaporannya membutuhkan waktu ± 1 hari dengan tingkat kesalahan berkisar 10%.	Dengan perhitungan serta pelaporan yang otomatis, hanya membutuhkan ± 5 menit dengan hampir tidak terjadi kesalahan dan bisa langsung dicetak sehingga mempermudah dalam pembacaan laporan tersebut yang nantinya bisa dipakai sebagai alat pengambilan keputusan terhadap usaha dagang yang dijalankan.
3.	Penyimpanan data kurang terjamin karena masih ada data yang hilang dan cacat, seperti pencarian data supplier yang membutuhkan waktu lama karena data yang disimpan hilang atau cacat	Data tersimpan dengan rapi di dalam sebuah database yang sewaktu-waktu bisa diperbarui, semua data yang disimpan bisa lebih mudah ditemukan. Seperti data supplier dan data barang.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian sistem yang dilakukan, dapat diambil kesimpulan beberapa hal antara lain :

1. Dengan aplikasi ini pekerjaan lebih cepat dibandingkan dengan sistem paperbase yang banyak membutuhkan waktu lama. Seperti pencatatan transaksi dan stok barang yang membutuhkan waktu $\pm 1,5$ jam, dengan menggunakan sistem komputerisasi hanya membutuhkan ± 15 menit. Perhitungan laba rugi serta pelaporan yang membutuhkan waktu ± 1 hari dengan tingkat kesalahan 10 %, dengan aplikasi ini hanya membutuhkan waktu ± 5 menit dengan hampir tidak terjadi kesalahan dan dapat langsung dicetak sehingga memudahkan dalam pembacaan dan pengambilan keputusan tehadap usaha dagang yang dijalankan.
2. Dengan disediakannya fitur-fitur dan menu-menu yang mudah dipahami, aplikasi ini mudah dan gampang dalam pemakaian dan pengaplikasiannya dalam badan usaha dagang level menengah ke bawah. Memungkinkan semua pengguna dapat dengan mudah memahami jalan kerja aplikasi ini.
3. Personalia tidak perlu khawatir lagi akan terjadi kesamaan data serta nama dan jumlah barang yang masuk atau keluar.

4. Hasil laporan pada aplikasi bisa langsung dicetak dengan fitur penyesuaian printer pada aplikasi sehingga nantinya memudahkan dalam pembacaan laporan.

5.2 Saran

1. Sistem informasi akuntansi ini masih dapat dikembangkan lebih jauh lagi karena dalam pembuatannya masih banyak menggunakan batasan yang mempertimbangkan banyak hal yaitu luasnya sistem, sumber daya manusia, serta badan usaha dagang yang akan menggunakan sistem ini.
2. Aplikasi ini dapat dikembangkan lagi dengan menambahkan software untuk kapasitas penyimpanan data (database) yang lebih besar.
3. Untuk pengembangan lebih lanjut dari aplikasi ini juga dapat ditambahkan fasilitas untuk *backup* data, sehingga jika terjadi kerusakan data atau sistem pada aplikasi, data dapat dikembalikan seperti sebelumnya.

DAFTAR PUSTAKA

- Ali, Muhamad. 1999. *Mengakses Database pada Delphi dengan ADO.* 02 Desember 2007. www.ilmukomputer.com/umum/Ali-Delphi.pdf.
- Andi, MADCOMS. *Pemrograman Borland Delphi 7.* Yogyakarta : ANDI; Madiun : MADCOMS. 2007.
- Away, Abdia, Gunaidi, S.Si. *The Shortcut of Delphi for Accounting.* Bandung : Informatika. 2008.
- Heryanto, Imam. *Membuat Database dengan Microsoft Office Access.* Bandung : Informatika. 2008.
- Oktiar, Gita. 2006. *Sistem akuntansi penjualan kredit suku cadang pada P.T. Astra international tbk. hso Cabang Semarang.* 22 Agustus 2008. <http://media.diknas.go.id/media/document/doc-2.pdf>.
- Rine. 2006. *Perancangan Sistem Informasi.* 31 Januari 2010. <http://rine06.wordpress.com/2008/11/22/perancangan-sistem-informasi.html>
- Solusindo, Parama. *Buku Petunjuk SIAP Sistem Informasi Akuntansi Perusahaan.* 19 Januari 2010. www.ilmukomputer.com/umum/Buku-Petunjuk-SIAP.pdf.
- Zakaria, Marcus, Teddy. *Berlatih Tipe Data di Delphi.* 02 Desember 2007. www.ilmukomputer.com/umum/teddy-tipedata.pdf.

LAMPIRAN



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAN TEKNOLOGI NASIONAL MALANG

INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI

FAKULTAS TEKNIK SIPIL DAN PERENCANAAN

PROGRAM PASCASARJANA MAGISTER TEKNIK

T. BNI (PERSERO) MALANG
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145

Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

NAMA : MOHAMMAD NURUL SHIDDIEQIIN
NIM : 04.12.664
JURUSAN/KONSENTRASI : TEKNIK ELEKTRO S-1 / KOMPUTER DAN INFORMATIKA
JUDUL SKRIPSI : PENGEMBANGAN APLIKASI AKUNTANSI UNTUK HOME INDUSTRY PADA STUDI KASUS DI KABUPATEN SUMENEP

Dipertahankan dihadapan Tim Penguji Ujian Skripsi jenjang program Strata Satu (S-1)

Pada Hari : Sabtu

Tanggal : 21 Agustus 2010

Dengan nilai : 83 (A) *Zay*

PANITIA UJIAN SKRIPSI

KETUA

Ir. Yusuf Ismail Nakhoda, MT
NIP.Y. 1018800189

ANGGOTA PENGUJI

Dosen Penguji I

Dr. Eng Aryuanto S, ST, MT.
NIP.Y.1030800417

Dosen Penguji II

Sotyoahadi, ST
NIP.Y.1039700309



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : MOHAMMAD NURUL SHIDDIEQIIN
Nim : 04.12.664
Masa Bimbingan : 10 FEBRUARI 2010 s/d 10 AGUSTUS 2010
Judul Skripsi : PENGEMBANGAN APLIKASI AKUNTANSI UNTUK HOME INDUSTRY PADA STUDI KASUS DI KABUPATEN SUMENEP

No.	Tanggal	Uraian	Paraf Pembimbing
1.	26/09/2009	Ucapan terimakasih pada kata pengantar (Bab I)	
2.	26/09/2009	Tata letak gambar dan pengurutannya (Bab II)	
3.	26/09/2009	Tata letak gambar, tabel serta pengurutannya (Bab III)	
4.	01/10/2009	Pengujian Sistem (Bab IV)	
5.	23/10/2009	Penjelasan form (Bab IV)	
6.	14/11/2009	Acc Bab I, II, dan III	
7.	18/02/2010	Acc Bab IV	
8.	20/02/2010	Acc Bab V	
9.			
10.			

Malang, 20 Februari 2010
Dosen Pembimbing

M. Ibrahim Ashari, ST.,MT.
NIP. P. 1030100358



INSTITUT TEKNOLOGI NASIONAL MALANG
FAKULTAS TEKNOLOGI INDUSTRI
JURUSAN TEKNIK ELEKTRO

FORMULIR BIMBINGAN SKRIPSI

Nama : MOHAMMAD NURUL SHIDDIEQIIN
Nim : 04.12.664
Masa Bimbingan : 10 FEBRUARI 2010 s/d 10 AGUSTUS 2010
Judul Skripsi : PENGEMBANGAN APLIKASI AKUNTANSI UNTUK HOME INDUSTRY PADA STUDI KASUS DI KABUPATEN SUMENEP

No.	Tanggal	Uraian	Paraf Pembimbing
1.	28/09/2009	Penulisan Abstrak dan Latar Belakang (Bab I)	
2.	29/09/2009	Uraian Tinjauan Pustaka Ms.Access dan Delphi (Bab II)	
3.	29/09/2009	Keterangan Tabel pada pembentukan Sistem (Bab III)	
4.	01/10/2009	Pengujian Sistem (Bab IV)	
5.	23/10/2009	Kesimpulan Sistem (Bab V)	
6.	18/11/2009	Acc Bab I, II, dan III	
7.	20/02/2010	Acc Bab IV	
8.	20/02/2010	Acc Bab V	
9.			
10.			

Malang, 20 Februari 2010
Dosen Pembimbing

I Komang Somawirata, ST.,MT.
NIP. P. 1030100361

INSTITUT TEKNOLOGI NASIONAL
Jl. Sigura-gura No 2
M A L A N G

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : MOHAMMAD NURUL SHIDDIEQIIN

Nim : 04.12.664

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / ~~tidak bersedia~~ *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**“PENGEMBANGAN APLIKASI AKUNTANSI
UNTUK HOME INDUSTRY PADA STUDY KASUS
DI KABUPATEN SUMENEP”**

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 01 Juli 2009
Kami yang membuat pernyataan,



M. Ibrahim Ashari, ST., MT.
NIP.P. 1030100358

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
*) coret yang tidak perlu

PERNYATAAN KESEDIAAN DALAM PEMBIMBINGAN SKRIPSI

Sesuai permohonan dari mahasiswa :

Nama : MOHAMMAD NURUL SHIDDIEQIIN

Nim : 04.12.664

Jurusan : Teknik Elektro S-1

Konsentrasi : Teknik Komputer & Informatika

Dengan ini Menyatakan (bersedia / tidak bersedia *) Membimbing Skripsi dari mahasiswa tersebut, dengan judul :

**“PENGEMBANGAN APLIKASI AKUNTANSI
UNTUK HOME INDUSTRY PADA KASUS INDUSTRY
DI KABUPATEN SUMENEP”**

Demikian surat Pernyataan ini kami buat agar dapat dipergunakan seperlunya.

Malang, 01 Juli 2009
Kami yang membuat pernyataan,



I Komang Somawirata, ST., MT.
NIP/E 1030100361

Catatan :

Setelah disetujui agar formulir ini
Diserahkan mahasiswa/i yang bersangkutan
Kepada Jurusan untuk diproses lebih lanjut.
*) coret yang tidak perlu

Menu Utama

```
unit UMain;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls, Menus, StdCtrls, Dialogs,
Buttons, Messages, ExtCtrls, ComCtrls, StdActns, ActnList, ToolWin, ImgList,
jpeg,UBarang,Usupplier,Upegawai,Uaset,Upembelian,Upenjualan, Grids, DB, ADODB;

type
  TMainForm = class(TForm)
    StatusBar: TStatusBar;
    ActionList1: TActionList;
    EditCut1: TEditCut;
    EditCopy1: TEditCopy;
    EditPaste1: TEditPaste;
    FileNew1: TAction;
    FileSave1: TAction;
    FileExit1: TAction;
    FileOpen1: TAction;
    FileSaveAs1: TAction;
    WindowCascade1: TWindowCascade;
    WindowTileHorizontal1: TWindowTileHorizontal;
    WindowArrangeAll1: TWindowArrange;
    WindowMinimizeAll1: TWindowMinimizeAll;
    HelpAbout1: TAction;
    FileClose1: TWindowClose;
    WindowTileVertical1: TWindowTileVertical;
    ImageList1: TImageList;
    EditSelectAll1: TEditSelectAll;
    Image1: TImage;
    Panel1: TPanel;
    Image2: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    btnpembelian: TSpeedButton;
    btnaset: TSpeedButton;
    btnsupplier: TSpeedButton;
    btnlaporan: TSpeedButton;
    btnpegawai: TSpeedButton;
    btnbarang: TSpeedButton;
    btnpenjualan: TSpeedButton;
    SpeedButton11: TSpeedButton;
    SpeedButton10: TSpeedButton;
    MainMenu1: TMainMenu;
    Abut1: TMenuItem;
    laporan: TPanel;
    GroupBox1: TGroupBox;
```

```
GroupBox2: TGroupBox;
pilihan: TComboBox;
view: TBitBtn;
data: TStringGrid;
con: TADOQuery;
bulan: TComboBox;
tahun: TComboBox;
username: TLabeledEdit;
pass: TLabeledEdit;
procedure FileNew1Execute(Sender: TObject);
procedure HelpAbout1Execute(Sender: TObject);
procedure FileExit1Execute(Sender: TObject);
procedure SpeedButton11Click(Sender: TObject);
procedure btnbarangClick(Sender: TObject);
procedure btnsupplierClick(Sender: TObject);
procedure btpegawaiClick(Sender: TObject);
procedure btasetClick(Sender: TObject);
procedure btppembelianClick(Sender: TObject);
procedure btppenjualanClick(Sender: TObject);
procedure btlaporanMouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure btlaporanClick(Sender: TObject);
procedure viewClick(Sender: TObject);
procedure btppembelianMouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
procedure SpeedButton10Click(Sender: TObject);
procedure passKeyPress(Sender: TObject; var Key: Char);
procedure usernameKeyPress(Sender: TObject; var Key: Char);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
  procedure CreateMDIChild(const Name: string);
public
  fbarang :TFormBarang;
  fsupplier : Tformsupplier;
  fpegawai : TFormPegawai;
  faset : TFormAset;
  fPembelian : TformPembelian;
  fpenjualan:TFormPenjualan;
end;

var
  MainForm: TMainForm;

implementation
{$R *.dfm}
uses data, about, URrugilaba, QuickRpt, UModal, UNeraca, Ulogin;
```

```

procedure TMainForm.CreateMDIChild(const Name: string);
var
  Child: TMDIChild;
begin
  { create a new MDI child window }
  Child := TMDIChild.Create(Application);
  Child.Caption := Name;
  if FileExists(Name) then Child.Memo1.Lines.LoadFromFile(Name);
end;

procedure TMainForm.FileNew1Execute(Sender: TObject);
begin
  CreateMDIChild('NONAME' + IntToStr(MDIChildCount + 1));
end;

procedure TMainForm.HelpAbout1Execute(Sender: TObject);
begin
  AboutBox.ShowModal;
end;

procedure TMainForm.FileExit1Execute(Sender: TObject);
begin
  Close;
end;

procedure TMainForm.SpeedButton11Click(Sender: TObject);
begin
  if (MessageDlg( 'Yakin Keluar dari
Program?',mtConfirmation,[mbYes,mbNo],0)=mrYes) then
    begin
      Application.Terminate;
    end;
end;

procedure TMainForm.btnbarangClick(Sender: TObject);
begin
  if fbarang=nil then
    fbarang := TFormbarang.Create(Application)
  else fbarang.Show
end;

procedure TMainForm.btnsupplierClick(Sender: TObject);
begin
  if fsupplier=nil then
    fsupplier := TFormSupplier.Create(Application)
  else fsupplier.Show
end;

procedure TMainForm.btnpegawaiClick(Sender: TObject);
begin

```

```

if fpegawai=nil then
  fpegawai := TFormPegawai.Create(Application)
else fpegawai.Show
end;

procedure TMainForm.btnasetClick(Sender: TObject);
begin
  if faset=nil then
    faset := TFormAset.Create(Application)
  else faset.Show
end;

procedure TMainForm.btpembelianClick(Sender: TObject);
begin
  if fPembelian=nil then
    fPembelian := TFormPembelian.Create(Application)
  else fPembelian.Show
end;

procedure TMainForm.btnpenjualanClick(Sender: TObject);
begin
  if fpenjualan=nil then
    fpenjualan:= TFormpenjualan.Create(Application)
  else fpenjualan.Show
end;

procedure TMainForm.bnllaporanMouseUp(Sender: TObject;
  Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
  laporan.Visible := true;
end;

procedure TMainForm.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
begin
  laporan.Left := -600;
end;

procedure TMainForm.bnllaporanClick(Sender: TObject);
begin
  laporan.Left := Image1.Left;
end;

procedure TMainForm.viewClick(Sender: TObject);
var i,n,j:integer;
  biaya,beban,modal :double;
  rptNerc : Trptneraca;
begin
  if (bulan.Text="") and (tahun.Text="") then exit;

```

```

case pilihan.ItemIndex of
4: begin
    con.close;
    con.sql.clear;
    con.sql.add('select jumlah,harga from penjualan Where jenis="Kredit" and
month(tgl) ='+IntToStr(bulan.ItemIndex+1)+' and year(tgl) ='+tahun.Text+' ');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
    begin
        biaya:=0;
        data.Cells[1,1] := con.fieldByName('Jenis').AsString;
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i+1] := con.fieldByName('kd_aset').AsString;
                Cells[2,i+1] := con.fieldByName('Nama').AsString;
                Cells[3,i+1] := con.fieldByName('Jumlah').AsString;
                biaya := biaya + con.fieldByName('Jumlah').AsCurrency;
                RowCount :=i+1;
                con.Next;
            end;
        end
    end;
    con.close;
    con.sql.clear;
    con.sql.add('select jumlah,harga from penjualan Where jenis="Kredit"
and month(tgl) ='+IntToStr(bulan.ItemIndex+1)+' and year(tgl) ='+tahun.Text+' ');
    con.Prepared;
    con.open;
    beban:=0;
    if con.recordcount >0 then
    begin
        for j := 1 to con.RecordCount do
        begin
            with data do
            begin
                beban :=beban+
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
                con.Next;
            end;
        end;
    end;
    data.Cells[2,i+1] := 'Piutang ';
    data.Cells[4,i+1] := FormatCurr('#,',beban);
    data.Cells[2,i+2] := 'TOTAL = ';
    data.Cells[4,i+2] := FormatCurr('#,',biaya+beban);
    beban :=biaya+beban;
    n:= data.RowCount+4;

```

```

con.sql.clear;
con.sql.add('select * from asset where jenis="Aset Tetap" and bulan =' +
IntToStr(bulan.ItemIndex+1)+' and tahun=' + tahun.Text +');
con.Prepared;
con.open;
if con.recordcount >0 then
begin
  biaya :=0;
  data.RowCount :=n+con.recordcount+2;
  data.Cells[1,n] := con.fieldByName('Jenis').AsString;
  for i := 1+n to con.RecordCount+n do
  begin
    with data do
    begin
      Cells[0,i] := con.fieldByName('kd_asset').AsString;
      // Cells[1,i] := con.fieldByName('Jenis').AsString;
      Cells[2,i] := con.fieldByName('Nama').AsString;
      Cells[3,i] := con.fieldByName('Jumlah').AsString;
      biaya := biaya + con.fieldByName('Jumlah').AsCurrency;
      con.Next;
    end;
  end
end;
data.Cells[2,i] := 'TOTAL = ';
data.Cells[4,i] := FormatCurr('#,',biaya);
data.RowCount := data.RowCount +1;
data.Cells[1,i+1] := 'TOTAL = ';
data.Cells[4,i+1] := FormatCurr('#,',biaya+beban);
con.close;
con.sql.clear;
con.sql.add('delete from `temp` ');
con.ExecSQL;
data.RowCount := data.RowCount +1;
for i:= 1 to data.RowCount-1 do
begin
  with data do
  begin
    con.close;
    con.sql.clear;
    con.sql.add('INSERT INTO `temp` '+
      '(k1, k2, k3,k4) values("'+trim(Cells[1,i]
)+"',"'"+trim(Cells[2,i])+"','"+"+trim(Cells[3,i])+"',"+trim(Cells[4,i])+") ');
    con.ExecSQL;
  end;
end;
rptNerc := Trptneraca.Create(self);
with rptNerc do
begin
  con.close;
  con.sql.clear;

```

```

        con.sql.add('Select * from temp');
        con.Prepared;
        con.open;
        Preview;
        Free;
    end;
end;
3: begin
    con.close;
    con.sql.clear;
    con.sql.add('select * from asset where jenis="Aset Lancar" and bulan =' +
        IntToStr(bulan.ItemIndex+1)+' and tahun=' +tahun.Text+');
    con.Prepared;
    con.open;
    if con.recordcount >0 then
    begin
        biaya:=0;
        data.Cells[1,1] := con.fieldByName('Jenis').AsString;
        for i := 1 to con.RecordCount do
        begin
            with data do
            begin
                Cells[0,i+1] := con.fieldByName('kd_aset').AsString;
                Cells[2,i+1] := con.fieldByName('Nama').AsString;
                Cells[3,i+1] := con.fieldByName('Jumlah').AsString;
                biaya := biaya + con.fieldByName('Jumlah').AsCurrency;
                RowCount :=i+1;
                con.Next;
            end;
        end
    end;
    con.close;
    con.sql.clear;
    con.sql.add('select jumlah,harga from penjualan Where jenis="Kredit" and
month(tgl) =' +IntToStr(bulan.ItemIndex+1)+ ' and year(tgl) =' +tahun.Text+' ');
    con.Prepared;
    con.open;
    beban:=0;
    if con.recordcount >0 then
    begin
        for j := 1 to con.RecordCount do
        begin
            with data do
            begin
                beban :=beban+(
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
                con.Next;
            end;
        end;
    end;
end;

```

```

data.Cells[2,i+1] := 'Piutang ';
data.Cells[4,i+1] := FormatCurr('#,',beban);
data.Cells[2,i+2] := 'TOTAL = ';
data.Cells[4,i+2] := FormatCurr('#,',biaya+beban);
beban := biaya+beban;
n:= data.RowCount+4;
con.sql.clear;
con.sql.add('select * from aset where jenis="Aset Tetap" and bulan
=' + IntToStr(bulan.ItemIndex+1) +' and tahun=' + tahun.Text + ');
con.Prepared;
con.open;
if con.recordcount >0 then
begin
  biaya :=0;
  data.RowCount :=n+con.recordcount+2;
  data.Cells[1,n] := con.fieldByName('Jenis').AsString;
  for i := 1+n to con.RecordCount+n do
  begin
    with data do
    begin
      Cells[0,i] := con.fieldByName('kd_aset').AsString;
      // Cells[1,i] := con.fieldByName('Jenis').AsString;
      Cells[2,i] := con.fieldByName('Nama').AsString;
      Cells[3,i] := con.fieldByName('Jumlah').AsString;
      biaya := biaya + con.fieldByName('Jumlah').AsCurrency;
      con.Next;
    end;
  end
end;
data.Cells[2,i] := 'TOTAL = ';
data.Cells[4,i] := FormatCurr('#,',biaya);
data.RowCount := data.RowCount +1;
data.Cells[1,i+1] := 'TOTAL HARTA = ';
data.Cells[4,i+1] := FormatCurr('#,',biaya+beban);
con.close;
con.sql.clear;
con.sql.add('delete from `temp` ');
con.ExecSQL;
for i:= 1 to data.RowCount-1 do
begin
  with data do
  begin
    con.close;
    con.sql.clear;
    con.sql.add('INSERT INTO `temp` +
      '(k1, k2, k3,k4) values("'+trim(Cells[1,i]
)+'"","'+trim(Cells[2,i])+'","'+trim(Cells[3,i])+'"","'+trim(Cells[4,i])+'") ');
    con.ExecSQL;
  end;
end;

```

```

rptNerc := Trptneraca.Create(self);
with rptNerc do
begin
  con.close;
  con.sql.clear;
  con.sql.add('Select * from temp');
  con.Prepared;
  con.open;
  Preview;
  Free;
end;
end;
1: begin
  con.close;
  con.sql.clear;
  con.sql.add('select jumlah,harga from penjualan where jenis="Tunai" and
month(tgl) ='+IntToStr(bulan.ItemIndex+1)+' and year(tgl) ='+tahun.Text+' ');
  con.Prepared;
  con.open;
  biaya :=0;
  if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    with data do
    begin
      biaya :=biaya+(
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
      con.Next;
    end;
  end;
end;
reprugilaba.p1.Caption:= FormatFloat('#,0',biaya);
reprugilaba.t1.Caption:= FormatFloat('#,0',biaya);
con.close;
con.sql.clear;
con.sql.add('SELECT jumlah, harga FROM pembelian where month(tgl)
='+IntToStr(bulan.ItemIndex+1)+' and year(tgl) ='+tahun.Text+' ');
con.Prepared;
con.open;
beban :=0;
if con.recordcount >0 then
begin
  for i := 1 to con.RecordCount do
  begin
    with data do
    begin
      beban :=beban+(
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
      con.Next;
    end;
  end;
end;

```

```

        end;
    end;
end;
reprugilaba.b1.Caption:= FormatFloat('#,0',beban);
reprugilaba.t2.Caption:= FormatFloat('#,0',beban);
reprugilaba.total.Caption:= FormatFloat('#,0',abs(biaya-beban));
reprugilaba.periode.Caption:='Periode '+bulan.Text+' '+tahun.Text;
if biaya>beban then
    reprugilaba.note.Caption:= 'Total Laba'
else
    reprugilaba.note.Caption:= 'Total Rugi';
reprugilaba.Preview;
end;
2:
begin
con.close;
con.sql.clear;
con.sql.add('select jumlah,harga from penjualan where jenis="Tunai" and
month(tgl) =' +IntToStr(bulan.ItemIndex+1)+ ' and year(tgl) =' +tahun.Text+ ' ');
con.Prepared;
con.open;
biaya :=0;
if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
    begin
        with data do
        begin
            biaya :=biaya+(
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
            con.Next;
        end;
    end;
end;
con.close;
con.sql.clear;
con.sql.add('SELECT jumlah, harga FROM pembelian where month(tgl)
=' +IntToStr(bulan.ItemIndex+1)+ ' and year(tgl) =' +tahun.Text+ ' ');
con.Prepared;
con.open;
beban :=0;
if con.recordcount >0 then
begin
    for i := 1 to con.RecordCount do
    begin
        with data do
        begin
            beban :=beban+(
con.fieldByName('harga').AsCurrency+con.fieldByName('jumlah').AsCurrency);
            con.Next;
        end;
    end;
end;

```

```

        end;
    end;
end;
con.close;
con.sql.clear;
con.sql.add('select jumlah from aset where jenis="Modal Awal" and bulan
='+IntToStr(bulan.ItemIndex+1)+' and tahun='+tahun.Text+' ');
con.Prepared;
con.open;
modal:=con.fieldByName('jumlah').AsCurrency;
repmodal.p0.Caption:= FormatFloat('#,0',modal);
repmodal.p3.Caption:= FormatFloat('#,0',abs(biaya-beban));
repmodal.total.Caption:= FormatFloat('#,0',abs(modal+(biaya-beban)));
repmodal.periode.Caption:='Periode '+bulan.Text+' '+tahun.Text;
repmodal.Preview;
end;
else
    exit;
end;
end;

procedure TMainForm.btnpembelianMouseMove(Sender: TObject;
Shift: TShiftState; X, Y: Integer);
begin
    laporan.Left := -600;
end;

procedure TMainForm.SpeedButton10Click(Sender: TObject);
begin
    if (MessageDlg( 'Yakin Untuk Logout ?',mtConfirmation,[mbYes,mbNo],0)=mrYes)
then
begin
    begin
        SpeedButton11.Visible :=false;
        SpeedButton10.Visible := false;
        Panel1.Visible :=false;
        pass.Visible:= true;
        username.Visible:=true;
        username.Clear ;
        pass.SetFocus;
    end;
end;
end;

procedure TMainForm.passKeyPress(Sender: TObject; var Key: Char);
begin
if key<>#13 then exit;
    con.close;
    con.sql.clear;
    con.sql.add('SELECT kd_pegawai, pass  FROM pegawai where
kd_pegawai="'+trim(username.Text)+" and pass='"+trim(pass.Text)+" ');
    con.Prepared;

```

```

con.open;
if con.recordcount >0 then
begin
  SpeedButton11.Visible :=True;
  SpeedButton10.Visible := true;
  Panel1.Visible := true;
  pass.Visible:= false;
  username.Visible:=false;
end
else
begin
  MessageDlg( 'Acces Denied',mtError,[mbYes],0);
  SpeedButton11.Visible :=false;
  SpeedButton10.Visible := false;
  Panel1.Visible := false;
  pass.Clear;
  pass.SetFocus;
end;
pass.Clear;
end;

procedure TMainForm.usernameKeyPress(Sender: TObject; var Key: Char);
begin
  if key=#13 then pass.SetFocus;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
  username.Clear
end;
end.

```

Pembelian

```

unit UPembelian;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, ADODB, Grids, StdCtrls, ExtCtrls, Buttons, ComCtrls;

type
TFormPembelian = class(TForm)
  Image2: TImage;
  jenis: TLabel;
  Label1: TLabel;
  batal: TSpeedButton;
  simpan: TSpeedButton;
  kode: TLabel;
  con: TADOQuery;
  Image1: TImage;

```

```
kd_supplier: TLabeledEdit;
cari: TSpeedButton;
carisupplier: TPanel;
dtcari: TStringGrid;
keyword: TLabeledEdit;
btncariok: TSpeedButton;
btncaribatal: TSpeedButton;
cara: TRadioGroup;
hasilsupplier: TStringGrid;
kd_barang: TLabeledEdit;
dtbarang: TStringGrid;
btnviewbarang: TSpeedButton;
gbJenis: TGroupBox;
faktur: TLabeledEdit;
GroupBox2: TGroupBox;
namabarang: TLabeledEdit;
harga: TLabeledEdit;
jumlah: TLabeledEdit;
data: TStringGrid;
tgl: TDateTimePicker;
Label3: TLabel;
btnbatal: TSpeedButton;
btnpesan: TSpeedButton;
tempkdbarang: TLabel;
tempkd: TLabel;
addsupplier: TSpeedButton;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure cariClick(Sender: TObject);
procedure keywordChange(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btncaribatalClick(Sender: TObject);
procedure dtcariSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure btncariokClick(Sender: TObject);
procedure kd_barangChange(Sender: TObject);
procedure btnviewbarangClick(Sender: TObject);
procedure dtbarangSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure namabarangChange(Sender: TObject);
procedure btnpesanClick(Sender: TObject);
procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure btnbatalClick(Sender: TObject);
procedure fakturChange(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure batalClick(Sender: TObject);
procedure addsupplierClick(Sender: TObject);
private
  { Private declarations }
public
```

```

{ Public declarations }
end;

var
  FormPembelian: TFormPembelian;

implementation
uses UMain, UBarang, USupplier;
{$R *.dfm}

procedure TFormPembelian.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Action := caFree;
  MainForm.fpembelian:= nil;
end;

procedure TFormPembelian.cariClick(Sender: TObject);
begin
  carisupplier.Visible := True ;
  carisupplier.Left :=cari.Left;
  btncariok.Visible :=False;
  keywordChange(self);
end;

procedure TFormPembelian.keywordChange(Sender: TObject);
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from supplier where kd_supplier like "%'+keyword.Text+'%" ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do
    begin
      with dtcari do
      begin
        Cells[0,i] := con.fieldByName('kd_supplier').AsString;
        Cells[1,i] := con.fieldByName('nama').AsString;
        Cells[2,i] := con.fieldByName('alamat').AsString;
        Cells[3,i] := con.fieldByName('kota').AsString;
        Cells[4,i] := con.fieldByName('kontak').AsString;
        Cells[5,i] := con.fieldByName('telp').AsString;
        RowCount :=i+1;
        con.Next;
      end;
    end;
  end;
end;

```

```

simpan.Visible := false;
kode.Caption := "";
simpan.Caption :='Simpan';
end;

procedure TFormPembelian.FormCreate(Sender: TObject);
begin
  with data do
begin
  Cells[0,0] := 'Kode';
  Cells[1,0] := 'Nama';
  Cells[2,0] := 'Harga';
  Cells[3,0] := 'Jumlah';
end;
  with dtbarang do
begin
  Cells[0,0] := 'Kode';
  Cells[1,0] := 'Nama';
  Cells[2,0] := 'Stok';
end;
  with dtcari,hasilsupplier do
begin
  Cells[0,0] := 'Kode';
  Cells[1,0] := 'Nama';
  Cells[2,0] := 'Alamat';
  Cells[3,0] := 'Kota';
  Cells[4,0] := 'Contact';
  Cells[5,0] := 'Telp';
end;
  kd_barangChange(self);
end;

procedure TFormPembelian.bnncaribatalClick(Sender: TObject);
begin
  carisupplier.Visible := false;
  kd_supplier.Clear;
  hasilsupplier.RowCount :=1;
end;

procedure TFormPembelian.dtcariSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
  if dtcari.Cells[0,ARow]<>" then
begin
  btncariok.Visible := true;
  kd_supplier.Text := dtcari.Cells[0,ARow];
  hasilsupplier.RowCount :=2;
  with dtcari do
begin
  hasilsupplier.Cells[0,1] := cells[0,ARow];

```

```

    hasilsupplier.Cells[1,1] := Cells[1,ARow];
    hasilsupplier.Cells[2,1] := Cells[2,ARow];
    hasilsupplier.Cells[3,1] := Cells[3,ARow];
    hasilsupplier.Cells[4,1] := Cells[4,ARow];
    hasilsupplier.Cells[5,1] := Cells[5,ARow];
  end;
end;
end;

procedure TFormPembelian.btncariokClick(Sender: TObject);
begin
  carisupplier.Visible := false;
end;

procedure TFormPembelian.kd_barangChange(Sender: TObject);
var i:integer;
begin
  btnpesan.Visible := false;
  con.close;
  con.sql.clear;
  con.sql.add('select * from barang where kd_barang like
%"'+trim(kd_barang.Text)+'%"');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do
    begin
      with dtbarang do
      begin
        Cells[0,i] := con.fieldByName('kd_barang').AsString;
        Cells[1,i] := con.fieldByName('nama').AsString;
        Cells[2,i] := con.fieldByName('stok').AsString;
        RowCount :=i+1;
        con.Next;
      end;
    end
  end;
end;
end;

procedure TFormPembelian.btnviewbarangClick(Sender: TObject);
begin
  if MainForm.fbarang=nil then
    MainForm.fbarang := TFormbarang.Create(Application)
  else MainForm.fbarang.Show
end;

procedure TFormPembelian.dtbarangSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin

```

```

if dtbarang.Cells[0,ARow]<>" then
begin
    namabarang.Text := dtbarang.Cells[1,Arow];
    jumlah.Text :='1';
    harga.Text :='0';
    tempkdbarang.Caption := dtbarang.Cells[0,arow];
end;
end;

procedure TFormPembelian.namabarangChange(Sender: TObject);
begin
    if namabarang.Text <>" then btnpesan.Visible := true;
end;

procedure TFormPembelian.btnpesanClick(Sender: TObject);
var i: integer;
begin
    i:= data.RowCount;
    data.RowCount := i+1;
    with data do
    begin
        Cells[0,i] := tempkdbarang.Caption;
        Cells[1,i] := namabarang.Text;
        Cells[2,i] := harga.Text;
        Cells[3,i] := jumlah.Text;

        FixedRows :=1;
    end;
    fakturChange(self);
end;

procedure TFormPembelian.dataSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
    if dATA.Cells[0,ARow]<>" then
begin
    tempkd.Caption := IntToStr(arow);
    btnbatal.Visible := true;
end;
end;

procedure TFormPembelian.btnbatalClick(Sender: TObject);
var i:integer;
begin
    if (data.RowCount >2) and (StrToInt(tempkd.Caption)<>data.RowCount) then
begin
    for i:= 0 to data.ColCount -1 do
begin
    data.Cells[i,StrToInt(tempkd.Caption)]:=data.Cells[i,data.RowCount-1];

```

```

    end;
    data.RowCount := data.RowCount -1;
end else
if (strToInt(tempkd.Caption)=data.RowCount) or (data.RowCount =2)then
begin
    for i:= 0 to data.ColCount -1 do
    begin
        data.Cells[i,StrToInt(tempkd.Caption)]:= "";
    end;
    data.RowCount := data.RowCount -1;
end;
btnbatal.Visible := false;
tempkd.Caption:="";
fakturChange(self);
end;

procedure TFormPembelian.fakturChange(Sender: TObject);
begin
    if (trim(faktur.Text)<>"")and(trim(data.Cells[0,1]<>")) and(trim(hasilsupplier.Cells[0,1]<>")) then
begin
    simpan.Visible :=true;
end else simpan.Visible := false;
end;

procedure TFormPembelian.simpanClick(Sender: TObject);
var i:integer;
begin
    if (MessageDlg('Yakin Untuk '+ simpan.Caption +' '+
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
begin
    try
        if simpan.Caption='Simpan' then
begin
        with data do
        for i:= 1 to data.RowCount-1 do
        begin
            con.close;
            con.sql.clear;
            con.sql.add('INSERT INTO `pembelian` '+
            '(kd_barang, kd_pegawai, kd_supplier, no_faktur, tgl, jumlah, jenis,
            '+
            'harga) values(""+trim(Cells[0,i]
            )+"""+trim('0001)+"""+trim(hasilsupplier.Cells[0,1])+"""+trim(faktur.Text)+"""+DateT
            oStr(tgl.Date)+"""+Cells[3,i]+"""+cara.Items.Strings[cara.itemindex] +"""+Cells[2,i]+""")+
            ');
            con.ExecSQL;
            con.sql.clear;
        end;
    end;
end;

```

```

        con.sql.add('Update barang set stok='+trim(Cells[3,i])+' where
kd_barang= '+Cells[0,i]+' ');
        con.ExecSQL;

    end;
end;
MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
batalClick(self);

except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;
begin
    batalClick(self);
end;
end;

procedure TFormPembelian.batalClick(Sender: TObject);
begin
    data.RowCount := 1;
    faktur.Clear;
end;

procedure TFormPembelian.addsupplierClick(Sender: TObject);
begin
    if MainForm.fsupplier=nil then
        MainForm.fsupplier := TFormSupplier.Create(Application)
    else MainForm.fsupplier.Show
end;

end.

```

Penjualan

```

unit UPembelian;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, Grids, StdCtrls, ExtCtrls, Buttons, ComCtrls;

type
    TFormPembelian = class(TForm)
        Image2: TImage;
        jenis: TLabel;
        Label1: TLabel;

```

```
batal: TSpeedButton;
simpan: TSpeedButton;
kode: TLabel;
con: TADOQuery;
Image1: TImage;
kd_supplier: TLabeledEdit;
cari: TSpeedButton;
carisupplier: TPanel;
dtcari: TStringGrid;
keyword: TLabeledEdit;
btncariok: TSpeedButton;
btncaribatal: TSpeedButton;
cara: TRadioGroup;
hasilsupplier: TStringGrid;
kd_barang: TLabeledEdit;
dtbarang: TStringGrid;
btnviewbarang: TSpeedButton;
gbJenis: TGroupBox;
faktur: TLabeledEdit;
GroupBox2: TGroupBox;
namabarang: TLabeledEdit;
harga: TLabeledEdit;
jumlah: TLabeledEdit;
data: TStringGrid;
tgl: TDateTimePicker;
Label3: TLabel;
btnbatal: TSpeedButton;
btnpesan: TSpeedButton;
tempkdbarang: TLabel;
tempkd: TLabel;
addsupplier: TSpeedButton;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure cariClick(Sender: TObject);
procedure keywordChange(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btncaribatalClick(Sender: TObject);
procedure dtcariSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure btncariokClick(Sender: TObject);
procedure kd_barangChange(Sender: TObject);
procedure btnviewbarangClick(Sender: TObject);
procedure dtbarangSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure namabarangChange(Sender: TObject);
procedure btnpesanClick(Sender: TObject);
procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure btbatalClick(Sender: TObject);
procedure fakturChange(Sender: TObject);
procedure simpanClick(Sender: TObject);
```

```

procedure batalClick(Sender: TObject);
procedure addsupplierClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormPembelian: TFormPembelian;

implementation

uses UMain, UBarang, USupplier;

{$R *.dfm}

procedure TFormPembelian.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Action := caFree;
  MainForm.fpembelian:= nil;
end;

procedure TFormPembelian.cariClick(Sender: TObject);
begin
  carisupplier.Visible := True ;
  carisupplier.Left :=cari.Left;
  btncariok.Visible :=False;
  keywordChange(self);
end;

procedure TFormPembelian.keywordChange(Sender: TObject);
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from supplier where kd_supplier like "%'+keyword.Text+'%" ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do
    begin
      with dtcari do
      begin
        Cells[0,i] := con.fieldByName('kd_supplier').AsString;
        Cells[1,i] := con.fieldByName('nama').AsString;
        Cells[2,i] := con.fieldByName('alamat').AsString;
        Cells[3,i] := con.fieldByName('kota').AsString;
      end;
    end;
  end;
end;

```

```

        Cells[4,i] := con.fieldByName('kontak').AsString;
        Cells[5,i] := con.fieldByName('telp').AsString;
        RowCount :=i+1;
        con.Next;
    end;
end
end;

simpan.Visible := false;
kode.Caption := "";
simpan.Caption :='Simpan';

end;

procedure TFormPembelian.FormCreate(Sender: TObject);
begin
    with data do
begin

    Cells[0,0] := 'Kode';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Harga';
    Cells[3,0] := 'Jumlah';

end;
    with dtbarang do
begin

    Cells[0,0] := 'Kode';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Stok';

end;
    with dtcari,hasilsupplier do
begin

    Cells[0,0] := 'Kode';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Alamat';
    Cells[3,0] := 'Kota';
    Cells[4,0] := 'Contact';
    Cells[5,0] := 'Telp';

end;
    kd_barangChange(self);
end;

```

```

procedure TFormPembelian.bnncaribatalClick(Sender: TObject);
begin
  carisupplier.Visible := false;
  kd_supplier.Clear;
  hasilsupplier.RowCount :=1;
end;

procedure TFormPembelian.dtcariSelectCell(Sender: TObject; ACol,
  ARow: Integer; var CanSelect: Boolean);
begin
  if dtcari.Cells[0,ARow]<>" " then
  begin
    btncariok.Visible := true;
    kd_supplier.Text := dtcari.Cells[0,ARow];
    hasilsupplier.RowCount :=2;
    with dtcari do
    begin

      hasilsupplier.Cells[0,1] := Cells[0,ARow];
      hasilsupplier.Cells[1,1] := Cells[1,ARow];
      hasilsupplier.Cells[2,1] := Cells[2,ARow];
      hasilsupplier.Cells[3,1] := Cells[3,ARow];
      hasilsupplier.Cells[4,1] := Cells[4,ARow];
      hasilsupplier.Cells[5,1] := Cells[5,ARow];
    end;
    end;
  end;
end;

procedure TFormPembelian.bnncariokClick(Sender: TObject);
begin
  carisupplier.Visible := false;
end;

procedure TFormPembelian.kd_barangChange(Sender: TObject);
var i:integer;
begin
  btpesan.Visible := false;
  con.close;
  con.sql.clear;
  con.sql.add('select * from barang where kd_barang like
  "%'+trim(kd_barang.Text)+'%" ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do
    begin
      with dtbarang do
      begin

```

```

Cells[0,i] := con.fieldByName('kd_barang').AsString;
Cells[1,i] := con.fieldByName('nama').AsString;
Cells[2,i] := con.fieldByName('stok').AsString;
RowCount := i+1;
con.Next;
end;
end
end;
end;

procedure TFormPembelian.btnviewbarangClick(Sender: TObject);
begin
  if MainForm.fbarang=nil then
    MainForm.fbarang := TFormbarang.Create(Application)
  else MainForm.fbarang.Show
end;

procedure TFormPembelian.dtbarangSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
  if dtbarang.Cells[0,ARow]<>" then
  begin
    namabarang.Text := dtbarang.Cells[1,Arow];
    jumlah.Text := '1';
    harga.Text := '0';
    tempkdbarang.Caption := dtbarang.Cells[0,arow];
  end;
end;

procedure TFormPembelian.namabarangChange(Sender: TObject);
begin
  if namabarang.Text <> " then btnpesan.Visible := true;
end;

procedure TFormPembelian.btnpesanClick(Sender: TObject);
var i: integer;
begin
  i:= data.RowCount;
  data.RowCount := i+1;
  with data do
  begin
    Cells[0,i] := tempkdbarang.Caption;
    Cells[1,i] := namabarang.Text;
    Cells[2,i] := harga.Text;
    Cells[3,i] := jumlah.Text;

    FixedRows :=1;
  end;
  fakturChange(self);
end;

```

```

procedure TFormPembelian.dataSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
  if dATA.Cells[0,ARow]<>" then
    begin
      tempkd.Caption := IntToStr(arow);
      btnbatal.Visible := true;
    end;
end;

procedure TFormPembelian.btnbatalClick(Sender: TObject);
var i:integer;
begin
  if (data.RowCount >2) and (StrToInt(tempkd.Caption)<>data.RowCount) then
  begin
    for i:= 0 to data.ColCount -1 do
    begin
      data.Cells[i,StrToInt(tempkd.Caption)]:=data.Cells[i,data.RowCount-1];
    end;
    data.RowCount := data.RowCount -1;
  end else
  if (strToInt(tempkd.Caption)=data.RowCount) or (data.RowCount =2)then
  begin
    for i:= 0 to data.ColCount -1 do
    begin
      data.Cells[i,StrToInt(tempkd.Caption)]:="";
    end;
    data.RowCount := data.RowCount -1;
  end;
  btnbatal.Visible := false;
  tempkd.Caption:="";
  fakturChange(self);
end;

procedure TFormPembelian.fakturChange(Sender: TObject);
begin
  if (trim(faktur.Text)<>"")and(data.Cells[0,1]<>"") and(hasilsupplier.Cells[0,1]<>"") then
  begin
    simpan.Visible :=true;
  end else simpan.Visible := false;
end;

procedure TFormPembelian.simpanClick(Sender: TObject);
var i:integer;
begin
  if (MessageDlg('Yakin Untuk '+ simpan.Caption +' '+
data?',mtConfirmation,[mbYes,mbNo],0)= mrYes then

```

```

begin
    try
        if simpan.Caption='Simpan' then
            begin
                with data do
                    for i:= 1 to data.RowCount-1 do
                        begin
                            con.close;
                            con.sql.clear;
                            con.sql.add('INSERT INTO `pembelian` '+
                                '(kd_barang, kd_pegawai, kd_supplier, no_faktur, tgl, jumlah, jenis,
                            '+
                                'harga) values'+
                                '"'+trim(Cells[0,i])+
                                '", "'+trim('0001')+'", "'+trim(hasilsupplier.Cells[0,1])+'", "'+trim(faktur.Text)+'", "'+DateT
                                oStr(tgl.Date)+'", "'+Cells[3,i]+'"'+cara.Items.Strings[cara.itemindex]+'"'+Cells[2,i]+'"')
                            ');
                            con.ExecSQL;
                            con.sql.clear;
                            con.sql.add('Update barang set stok=stok '+trim(Cells[3,i])+' where
                            kd_barang= '+Cells[0,i]+' ');
                            con.ExecSQL;

                            end;
                        end;
                    MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
                    batalClick(self);

                except
                    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
                    exit;
                end;
            end;
        else
            begin
                batalClick(self);
            end;
        end;
    end;

procedure TFormPembelian.batalClick(Sender: TObject);
begin
    data.RowCount := 1;
    faktur.Clear;
end;

procedure TFormPembelian.addsupplierClick(Sender: TObject);
begin
    if MainForm.fsupplier=nil then
        MainForm.fsupplier := TFormSupplier.Create(Application)
    else MainForm.fsupplier.Show

```

```
end;
```

```
end.
```

Aset

```
unit UAset;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, DB, ADODB, Grids, StdCtrls, ExtCtrls, Buttons;
```

```
type
```

```
  TFormAset = class(TForm)  
    Image2: TImage;  
    a: TLabel;  
    Label1: TLabel;  
    batal: TSpeedButton;  
    simpan: TSpeedButton;  
    hapus: TSpeedButton;  
    Shape1: TShape;  
    kode: TLabel;  
    nama: TLabeledEdit;  
    jumlah: TLabeledEdit;  
    data: TStringGrid;  
    con: TADOQuery;  
    Image1: TImage;  
    jenis: TComboBox;  
    Label2: TLabel;  
    procedure FormClose(Sender: TObject; var Action: TCloseAction);  
    procedure batalClick(Sender: TObject);  
    procedure FormCreate(Sender: TObject);  
    procedure simpanClick(Sender: TObject);  
    procedure jumlahKeyUp(Sender: TObject; var Key: Word;  
      Shift: TShiftState);  
    procedure jenisChange(Sender: TObject);  
    procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;  
      var CanSelect: Boolean);  
    procedure hapusClick(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    { Public declarations }
```

```
  end;
```

```
var
```

```
  FormAset: TFormAset;
```

implementation

uses UMain;

{\$R *.dfm}

```
procedure TFormAset.FormClose(Sender: TObject; var Action: TCloseAction);
begin
```

```
    Action := caFree;
```

```
    MainForm.fsupplier := nil;
```

```
end;
```

```
procedure TFormAset.batalClick(Sender: TObject);
```

```
var i:integer;
```

```
begin
```

```
    con.close;
```

```
    con.sql.clear;
```

```
    con.sql.add('select * from asset');
```

```
    con.Prepared;
```

```
    con.open;
```

```
    if con.recordcount >0 then
```

```
        begin
```

```
            for i := 1 to con.RecordCount do
```

```
                begin
```

```
                    with data do
```

```
                    begin
```

```
                        Cells[0,i] := con.fieldByName('kd_aset').AsString;
```

```
                        Cells[1,i] := con.fieldByName('Jenis').AsString;
```

```
                        Cells[2,i] := con.fieldByName('Nama').AsString;
```

```
                        Cells[3,i] := con.fieldByName('Jumlah').AsString;
```

```
                        RowCount :=i+1;
```

```
                        con.Next;
```

```
                    end;
```

```
                end
```

```
            end;
```

```
            jumlah.Clear;
```

```
            hapus.Visible :=False;
```

```
            simpan.Visible := false;
```

```
            kode.Caption := ";
```

```
            simpan.Caption :='Simpan';
```

```
            nama.Clear;
```

```
        end;
```

```
procedure TFormAset.FormCreate(Sender: TObject);
```

```
begin
```

```
    with data do
```

```
    begin
```

```
        Cells[0,0] := 'Kode';
```

```

Cells[1,0] := 'Jenis';
Cells[2,0] := 'Nama';
Cells[3,0] := 'Jumlah';

end;
batalClick (self);

end;

procedure TFormAset.simpanClick(Sender: TObject);
label lanjut, edits;
var aRow : integer;
begin
  if (MessageDlg('Yakin Untuk '+ simpan.Caption +' data?',mtConfirmation,[mbYes,mbNo],0)= mrYes then
    begin
      try
        if (jenis.Text ='Modal Awal') and (simpan.Caption='Simpan') then
          begin
            con.close;
            con.sql.clear;
            con.SQL.Add('Delete from asset where jenis="Modal Awal" ');
            con.ExecSQL;
            con.sql.clear;
            con.sql.add('insert into asset( Nama, jenis, jumlah)
values(""+trim(nome.Text )+",""+trim(jenis.Text )+",""+trim(jumlah.Text )+"") ');
            con.ExecSQL;
            goto lanjut
          end;
        if simpan.Caption='Simpan' then
          begin
            con.close;
            con.sql.clear;
            con.sql.add('insert into asset( Nama, jenis, jumlah)
values(""+trim(nome.Text )+",""+trim(jenis.Text )+",""+trim(jumlah.Text )+"") ');
            con.ExecSQL;
          end
        else If simpan.Caption ='Edit' then
          begin
            edits:
            aRow := StrToInt(kode.Caption);
            with data do
              begin
                kode.Caption := Cells[0,aRow] ;
                jenis.Text := Cells[1,aRow] ;
                nome.Text := Cells[2,aRow] ;
                jumlah.Text :=Cells[3,aRow] ;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        end;
        simpan.Caption := 'Update';
        hapus.Visible := False;
        exit;
    end
    else if simpan.Caption='Update' then
    begin
        con.close;
        con.sql.clear;
        con.sql.add('Update aset set nama="'+trim(nama.Text
)+'" jenis="'+trim(jenis.Text )+''',jumlah="'+trim(jumlah.Text )+''', where kd_aset=
'+kode.Caption+' ');
        con.ExecSQL;
    end;
    lanjut:
    MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
    batalClick(self);

except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;
end;
else
begin
    batalClick(self);
end;
end;
procedure TFormAset.jumlahKeyUp(Sender: TObject; var Key: Word;
Shift: TShiftState);
begin
    if not (Key in [ord('0')..ord('9')]) then jumlah.Clear;
end;

procedure TFormAset.jenisChange(Sender: TObject);
begin
    If (trim(nama.Text) <>"") and (trim(jenis.Text) <>"") and (trim(jumlah.Text) <>")
        then
            simpan.Visible := true
        else simpan.Visible := false;
end;

procedure TFormAset.dataSelectCell(Sender: TObject; ACol, ARow: Integer;
var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    simpan.Caption :='Edit';
    simpan.Visible := True;
    hapus.Visible := true
end;

```

```

procedure TFormAset.hapusClick(Sender: TObject);
begin
  if (MessageDlg('Yakin Untuk '+ hapus.Caption +' '+
data?',mtConfirmation,[mbYes,mbNo],0)= mrYes then
    begin
      try
        con.close;
        con.sql.clear;
        con.sql.add('Delete from asset where kd_asset='
'+data.Cells[0,strToInt(kode.Caption)]+' ');
        con.ExecSQL;

        MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
        batalClick(self);

      except
        MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
      end;
    end
  end;
end.

```

Barang

```

unit UBarang;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Grids, DBGrids, DB, ADODB, Buttons;

type
  TFormBarang = class(TForm)
    nama: TlabeledEdit;
    beli: TlabeledEdit;
    jual: TlabeledEdit;
    stok: TlabeledEdit;
    Image2: TImage;
    Label2: TLabel;
    Label1: TLabel;
    Shape1: TShape;
    batal: TSpeedButton;
    simpan: TSpeedButton;
    hapus: TSpeedButton;
    data: TStringGrid;
  end;

```

```

con: TADOQuery;
kode: TLabel;
Image1: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure batalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure namaChange(Sender: TObject);
procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure beliKeyPress(Sender: TObject; var Key: Char);
procedure beliKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure jualKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure stokKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure hapusClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  FormBarang: TFormBarang;

implementation

uses Math, UMain;

{$R *.dfm}

procedure TFormBarang.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Action := caFree;
  MainForm.fbarang := nil;
end;

procedure TFormBarang.batalClick(Sender: TObject);
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from barang ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do

```

```

begin
  with data do
  begin
    Cells[0,i] := con.fieldByName('kd_barang').AsString;
    Cells[1,i] := con.fieldByName('nama').AsString;
    Cells[2,i] := con.fieldByName('harga_jual').AsString;
    Cells[3,i] := con.fieldByName('harga_beli').AsString;
    Cells[4,i] := con.fieldByName('stok').AsString;
    // Cells[0,i] := con.fieldByName('metode').AsString;
    RowCount :=i+1;
    con.Next;
  end;
end
beli.Clear;
jual.Clear;
stok.Clear;
hapus.Visible :=False;
simpan.Visible := false;
kode.Caption := "";
simpan.Caption :='Simpan';
nama.Clear;
end;

procedure TFormBarang.FormCreate(Sender: TObject);
begin
  with data do
  begin
    Cells[0,0] := 'Kode';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Jual';
    Cells[3,0] := 'Beli';
    Cells[4,0] := 'Stok';

    end;
    batalClick (self);

end;

procedure TFormBarang.simpanClick(Sender: TObject);
var aRow : integer;
begin
  if (MessageDlg('Yakin Untuk '+ simpan.Caption +' '+
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
  begin
    try
      if simpan.Caption='Simpan' then

```

```

begin
    con.close;
    con.sql.clear;
    con.sql.add('insert into barang( nama,harga_beli,harga_jual,stok)
values(""+trim(nama.Text)+"','"+trim(beli.Text)+"','"+trim(jual.Text)+"','"+trim(stok.Text
)+"') ");
    con.ExecSQL;
end
else If simpan.Caption ='Edit' then
begin
aRow := StrToInt(kode.Caption);
with data do
begin
    kode.Caption := Cells[0,arow] ;
    Nama.Text := Cells[1,arow] ;
    jual.Text := Cells[2,arow] ;
    beli.Text :=Cells[3,arow] ;
    stok.Text := Cells[4,arow] ;
end;
simpan.Caption := 'Update';
hapus.Visible := False;
exit;
end
else if simpan.Caption='Update' then
begin
    con.close;
    con.sql.clear;
    con.sql.add('Update barang set nama="'+trim(nama.Text
)+'",harga_beli="'+trim(beli.Text )+\"",harga_jual="'+trim(jual.Text
)+\"",stok="'+trim(stok.Text )+\"" where kd_barang=' +kode.Caption+' ');
    con.ExecSQL;
end;
MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
batalClick(self);

except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;
end
else
begin
    batalClick(self);
end;
end;
procedure TFormBarang.namaChange(Sender: TObject);
begin
    If (trim(nama.Text) <>"") and (trim(jual.Text) <>"") and (trim(beli.Text) <>"") and
(trim(stok.Text) <>"")

```

```

    then
        simpan.Visible := true
    else simpan.Visible := false;
end;

procedure TFormBarang.dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    simpan.Caption := 'Edit';
    simpan.Visible := True;
    hapus.Visible := true
end;

procedure TFormBarang.beliKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'9']) then beli.Clear;
end;

procedure TFormBarang.beliKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
    if not (Key in [ord('0')..ord('9')]) then beli.Clear;
end;

procedure TFormBarang.jualKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
    if not (Key in [ord('0')..ord('9')]) then jual.Clear;
end;

procedure TFormBarang.stokKeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
    if not (Key in [ord('0')..ord('9')]) then stok.Clear;
end;

procedure TFormBarang.hapusClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk '+ hapus.Caption +' '+
data?'',mtConfirmation,[mbYes,mbNo],0))= mrYes then
        begin
            try
                con.close;
                con.sql.clear;
                con.sql.add('Delete from barang where kd_barang='
'+data.Cells[0,strtoint(kode.Caption)]+' ');
                con.ExecSQL;
            end;
        end;
end;

```

```

MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
batalClick(self);

except
  MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
exit;
end;
end;
end;

end.

```

Supplier

unit USupplier;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, DB, ADODB, Grids, StdCtrls, ExtCtrls, Buttons;

type

```

TFormSupplier = class(TForm)
  Image2: TImage;
  Label2: TLabel;
  Label1: TLabel;
  batal: TSpeedButton;
  simpan: TSpeedButton;
  hapus: TSpeedButton;
  Shape1: TShape;
  nama: TLabeledEdit;
  con: TADOQuery;
  kode: TLabel;
  Alamat: TLabeledEdit;
  kota: TLabeledEdit;
  kontak: TLabeledEdit;
  telp: TLabeledEdit;
  data: TStringGrid;
  Image1: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure batalClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure simpanClick(Sender: TObject);
procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure hapusClick(Sender: TObject);
procedure namaChange(Sender: TObject);
private
  { Private declarations }

```

```
public
  { Public declarations }
end;

var
  FormSupplier: TFormSupplier;

implementation

uses UMain;

{$R *.dfm}

procedure TFormSupplier.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Action := caFree;
  MainForm.fsupplier := nil;
end;

procedure TFormSupplier.batalClick(Sender: TObject);
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from supplier ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
    begin
      for i := 1 to con.RecordCount do
        begin
          with data do
            begin
              Cells[0,i] := con.fieldByName('kd_supplier').AsString;
              Cells[1,i] := con.fieldByName('nama').AsString;
              Cells[2,i] := con.fieldByName('alamat').AsString;
              Cells[3,i] := con.fieldByName('kota').AsString;
              Cells[4,i] := con.fieldByName('kontak').AsString;
              Cells[5,i] := con.fieldByName('telp').AsString;
              RowCount :=i+1;
              con.Next;
            end;
        end;
    end;
  Alamat.Clear;
  kota.Clear;
  telp.Clear;
  kontak.Clear;
  hapus.Visible :=False;
```

```

simpan.Visible := false;
kode.Caption := "";
simpan.Caption :='Simpan';
nama.Clear;
end;

procedure TFormSupplier.FormCreate(Sender: TObject);
begin
  with data do
begin

  Cells[0,0] := 'Kode';
  Cells[1,0] := 'Nama';
  Cells[2,0] := 'Alamat';
  Cells[3,0] := 'Kota';
  Cells[4,0] := 'Contact';
  Cells[5,0] := 'Telp';

end;
batalClick (self);

end;

procedure TFormSupplier.simpanClick(Sender: TObject);
var aRow : integer;
begin
  if (MessageDlg('Yakin Untuk '+ simpan.Caption +' data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
begin
  try
    if simpan.Caption='Simpan' then
      begin
        con.close;
        con.sql.clear;
        con.sql.add('insert into supplier( Nama, Alamat, Kota, Kontak, Telp)
values(""+trim(nama.Text )+"','"+trim(Alamat.Text )+"','"+trim(kota.Text )
)+"','"+trim(kontak.Text )+"','"+trim(telp.Text )+"')');
        con.ExecSQL;
      end
    else If simpan.Caption ='Edit' then
      begin
        aRow := StrToInt(kode.Caption);
        with data do
          begin
            kode.Caption := Cells[0,aRow] ;
            Nama.Text := Cells[1,aRow] ;
            alamat.Text := Cells[2,aRow] ;
            kota.Text :=Cells[3,aRow] ;
            kontak.Text := Cells[4,aRow] ;

```

```

        telp.Text := Cells[5,arow] ;
    end;
    simpan.Caption := 'Update';
    hapus.Visible := False;
    exit;
end
else if simpan.Caption='Update' then
begin
    con.close;
    con.sql.clear;
    con.sql.add('Update supplier set nama="'+trim(nama.Text
)+'",alamat="'+trim(Alamat.Text )+'",kota="'+trim(kota.Text
)+'",kontak="'+trim(kontak.Text )+'", telp="'+trim(telp.Text)+" where kd_supplier=
'+kode.Caption+' ');
    con.ExecSQL;
end;
MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
batalClick(self);

except
    MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
exit;
end;
end
else
begin
    batalClick(self);
end;
end;

```

```

procedure TFormSupplier.dataSelectCell(Sender: TObject; ACol,
ARow: Integer; var CanSelect: Boolean);
begin
    kode.Caption := IntToStr(ARow);
    simpan.Caption :='Edit';
    simpan.Visible := True;
    hapus.Visible := true
end;

```

```

procedure TFormSupplier.hapusClick(Sender: TObject);
begin
    if (MessageDlg('Yakin Untuk '+ hapus.Caption +''
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
begin
    try

```

```

        con.close;
        con.sql.clear;

```

```

        con.sql.add('Delete from supplier where kd_supplier=
'+data.Cells[0,strtoint(kode.Caption)]+' ');
        con.ExecSQL;

        MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
        batalClick(self);

    except
        MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
        exit;
    end;
end;
end;

procedure TFormSupplier.namaChange(Sender: TObject);
begin
    If (trim(nama.Text) <>"") and (trim(Alamat.Text) <>"") and (trim(kota.Text) <>"") and
(trim(kontak.Text) <>"") and (trim(telp.Text) <> "")
        then
            simpan.Visible := true
        else simpan.Visible := false;
end;
end.

```

Pegawai

```

unit UPegawai;

interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, DB, ADODB, Grids, StdCtrls, ExtCtrls, Buttons;

```

```

type
    TFormPegawai = class(TForm)
        Image2: TImage;
        Label2: TLabel;
        Label1: TLabel;
        batal: TSpeedButton;
        simpan: TSpeedButton;
        hapus: TSpeedButton;
        Shape1: TShape;
        kode: TLabel;
        nama: TLabeledEdit;
        Alamat: TLabeledEdit;
        telp: TLabeledEdit;
        pass: TLabeledEdit;
        id: TLabeledEdit;

```

```
data: TStringGrid;
con: TADOQuery;
Image1: TImage;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure FormCreate(Sender: TObject);
procedure batalClick(Sender: TObject);
procedure dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
procedure simpanClick(Sender: TObject);
procedure hapusClick(Sender: TObject);
procedure idChange(Sender: TObject);
private
  procedure getCode();
public
  { Public declarations }
end;

var
  FormPegawai: TFormPegawai;

implementation

uses UMain;

{$R *.dfm}

procedure TFormPegawai.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
  Action := caFree;
  MainForm.fpegawai := nil;
end;

procedure TFormPegawai.FormCreate(Sender: TObject);
begin
  with data do
  begin
    Cells[0,0] := 'Kode';
    Cells[1,0] := 'Nama';
    Cells[2,0] := 'Alamat';
    Cells[3,0] := 'Telp';
  end;
  batalClick (self);
  getCode;
end;
```

```

procedure TFormPegawai.batalClick(Sender: TObject);
var i:integer;
begin
  con.close;
  con.sql.clear;
  con.sql.add('select * from pegawai ');
  con.Prepared;
  con.open;
  if con.recordcount >0 then
  begin
    for i := 1 to con.RecordCount do
    begin
      with data do
      begin
        Cells[0,i] := con.fieldByName('kd_pegawai').AsString;
        Cells[1,i] := con.fieldByName('nama').AsString;
        Cells[2,i] := con.fieldByName('alamat').AsString;
        Cells[3,i] := con.fieldByName('telp').AsString;
        RowCount :=i+1;
        con.Next;
      end;
    end
  end;
  Alamat.Clear;
  pass.Clear;
  telp.Clear;

  hapus.Visible :=False;
  simpan.Visible := false;
  kode.Caption := "";
  simpan.Caption :='Simpan';
  getCode;
  nama.Clear;
end;

procedure TFormPegawai.dataSelectCell(Sender: TObject; ACol, ARow: Integer;
  var CanSelect: Boolean);
begin
  kode.Caption := IntToStr(ARow);
  simpan.Caption :='Edit';
  simpan.Visible := True;
  hapus.Visible := true
end;
procedure TFormPegawai.getCode();
var
  Kodes : String;
  i : Integer;
  found : boolean;

begin

```

```

i := 1;
Kodes := FormatCurr('0000',1);
id.Text := Kodes;
found := false;
While Not found do
begin
  Kodes := FormatCurr('0000',i);
  con.close;
  con.sql.clear;
  con.sql.add('select * from pegawai where kd_pegawai="'+kodes+''");
  con.Prepared;
  con.open;
  if con.recordcount >0 then
    found := False
  Else
  begin
    id.Text := Kodes;
    found := True;
  end;
  i := i + 1;
end;

end;
procedure TFormPegawai.simpanClick(Sender: TObject);
var aRow : integer;
begin
  if (MessageDlg('Yakin Untuk '+ simpan.Caption +' data?',mtConfirmation,[mbYes,mbNo],0)= mrYes then
  begin
    try
      if simpan.Caption='Simpan' then
      begin
        con.close;
        con.sql.clear;
        con.sql.add('INSERT INTO `pegawai` (kd_pegawai, nama, alamat, telp, pass) values("'+trim(id.Text)+'","'+trim(nama.Text)+'","'+trim(Alamat.Text)+'","'+trim(telp.Text)+'","'+trim(pass.Text)+'")');
        con.ExecSQL;
      end
      else If simpan.Caption ='Edit' then
      begin
        aRow := StrToInt(kode.Caption);
        with data do
        begin
          kode.Caption := Cells[0,aRow] ;
          Nama.Text := Cells[1,aRow] ;
          alamat.Text := Cells[2,aRow];
          telp.Text :=Cells[3,aRow];
          id.Text :=Cells[0,aRow] ;
        end;
      end;
    end;
  end;
end;

```

```

        id.Enabled := false;
      end;
      simpan.Caption := 'Update';
      hapus.Visible := False;
      exit;
    end
  else if simpan.Caption='Update' then
  begin
    con.close;
    con.sql.clear;
    con.sql.add('Update pegawai set nama="'+trim(nome.Text
)+'",alamat="'+trim(Alamat.Text )+\"",telp="'+trim(telp.Text )+\"",pass="'+trim(pass.Text
)+" where kd_pegawai= '"+id.Text+" ' ");
    con.ExecSQL;
  end;
  MessageDlg('Proses Penyimpanan Berhasil',mtInformation,[ mbOK],0 );
  batalClick(self);

except
  MessageDlg('Proses Penyimpanan Gagal',mtError,[ mbOK],0 );
  exit;
end;
end
else
begin
  batalClick(self);
end;
end;

procedure TFormPegawai.hapusClick(Sender: TObject);
begin
  if (MessageDlg('Yakin Untuk '+ hapus.Caption +' '+
data?',mtConfirmation,[mbYes,mbNo],0))= mrYes then
  begin
    try
      con.close;
      con.sql.clear;
      con.sql.add('Delete from pegawai where kd_pegawai= "'+
data.Cells[0,stroint(kode.Caption)] +'" ');
      con.ExecSQL;
      MessageDlg('Proses Penghapusan Berhasil',mtInformation,[ mbOK],0 );
      batalClick(self);

    except
      MessageDlg('Proses Penghapusan Gagal',mtError,[ mbOK],0 );
      exit;
    end;
  end
end;

```

```

end;

procedure TFormPegawai.idChange(Sender: TObject);
begin
  If (trim(nama.Text) <> "") and (trim(Alamat.Text) <> "") and (trim(pass.Text) <> "") and
  (trim(telp.Text) <> "")
    then
      simpan.Visible := true
    else simpan.Visible := false;
end;

```

end.

Laporan Laba Rugi

```

unit URrugilaba;

interface

uses Windows, SysUtils, Messages, Classes, Graphics, Controls,
  StdCtrls, ExtCtrls, Forms, QuickRpt, QRCtrls, DB, ADODB;

type
  TReprugilaba = class(TQuickRep)
    con: TADOQuery;
    TitleBand1: TQRBand;
    QRShape1: TQRShape;
    QRLabel12: TQRLLabel;
    QRLabel13: TQRLLabel;
    p1: TQRLLabel;
    QRLabel19: TQRLLabel;
    QRLabel20: TQRLLabel;
    b1: TQRLLabel;
    QRLabel8: TQRLLabel;
    QRSSysData2: QRSSysData;
    QRShape2: TQRShape;
    QRLabel2: TQRLLabel;
    QRLabel1: TQRLLabel;
    periode: TQRLLabel;
    QRLabel3: TQRLLabel;
    QRLabel5: TQRLLabel;
    QRShape3: TQRShape;
    QRShape4: TQRShape;
    t1: TQRLLabel;
    t2: TQRLLabel;
    QRShape5: TQRShape;
    QRShape6: TQRShape;
    QRShape7: TQRShape;
    total: TQRLLabel;
    note: TQRLLabel;
  end;

```

```
private  
  
public  
end;  
  
var  
  reprugilaba: Treprugilaba;  
  
implementation  
  
{$R *.DFM}  
  
end.
```

Laporan Perubahan Modal

```
unit UModal;  
  
interface  
  
uses Windows, SysUtils, Messages, Classes, Graphics, Controls,  
  StdCtrls, ExtCtrls, Forms, QuickRpt, QRCtrls;  
  
type  
  Trepmodal = class(TQuickRep)  
    TitleBand1: TQRBand;  
    QRShape1: TQRShape;  
    QRLLabel12: TQRLLabel;  
    QRLLabel13: TQRLLabel;  
    QRLLabel19: TQRLLabel;  
    QRLLabel8: TQRLLabel;  
    QRSSysData2: TQRSysData;  
    QRShape2: TQRShape;  
    QRLLabel2: TQRLLabel;  
    QRLLabel1: TQRLLabel;  
    periode: TQRLLabel;  
    QRLLabel3: TQRLLabel;  
    p2: TQRLLabel;  
    QRShape5: TQRShape;  
    QRShape6: TQRShape;  
    QRShape7: TQRShape;  
    total: TQRLLabel;  
    p0: TQRLLabel;  
    p3: TQRLLabel;  
  private  
  
  public  
  
  end;
```

```
var
  repmodal: Trepmodal;

implementation

{$R *.DFM}

end.
```

Laporan Neraca

```
unit UNeraca;

interface

uses Windows, SysUtils, Messages, Classes, Graphics, Controls,
  StdCtrls, ExtCtrls, Forms, QuickRpt, QCRCtrls, DB, ADODB;

type
  Trptneraca = class(TQuickRep)
    QRBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel1: TQRLabel;
    periode: TQRLabel;
    QRShape1: TQRShape;
    con: TADOQuery;
    QRSubDetail1: TQRSubDetail;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRLabel3: TQRLabel;
    conk1: TWideStringField;
    conk2: TWideStringField;
    conk3: TWideStringField;
    conk4: TWideStringField;
    conk5: TWideStringField;
    conk6: TWideStringField;
  private
  public
  end;

var
  rptneraca: Trptneraca;

implementation

{$R *.DFM}

end.
```