**RANCANG BANGUN SISTEM INFORMASI PENJUALAN DAN MANAJEMEN TOKO BERBASIS CLIENT SERVER**

**(STUDI KASUS CAHAYA PHONECELL YOGYAKARTA)**

TUGAS AKHIR

Disusun guna memenuhi salah satu persyaratan

untuk menyelesaikan Program Studi Strata 1 Teknik Informatika

Disusun oleh :

**PUTRA ADIGUNA NARENDRA**

**3065111018**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS BISNIS DAN TEKNOLOGI INFORMASI**

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**

**2016**

**TUGAS AKHIR**

**RANCANG BANGUN SISTEM INFORMASI PENJUALAN DAN MANAJEMEN TOKO BERBASIS CLIENT SERVER**

**(STUDI KASUS CAHAYA PHONECELL YOGYAKARTA)**

Disusun oleh :

**Putra Adiguna Narendra**

**3065111018**

Telah dipertahankan di depan tim penguji

Pada tanggal \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Pembimbing

**Arif Pramudwiatmoko,S.T., M.eng.**

|  |  |
| --- | --- |
| Penguji 1 | Penguji 2 |
| (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) | (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_) |

Proyek Tugas Akhir ini telah diterima sebagai salah satu syarat untuk mencapai derajat Sarjana S-1 Program Studi Teknik Informatika

Yogyakarta,\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ketua Program Studi Teknik Informatika

Fakultas Bisnis dan Teknologi Informasi, Universitas Teknologi Yogyakarta

**Yuli Asriningtias, S.Kom., M.KomPERNYATAAN**

Dengan ini Penulis menyatakan bahwa laporan penelitian ini tidak merupakan karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi lain, dan sepanjang pengetahuan Penulis juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang tertulis sebagai acuan dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, Agustus 2016

Putra Adiguna Narendra

**MOTTO**

*Apa masalahnya dengan terjatuh jika hanya tinggal bangun sekali lagi .*

*(Kitao Aya)*

*Que Sera Sera.*

*(Rendra)*

*Jika kamu tidak bisa membuat itu bagus. Paling tidak buat itu kelihatan bagus.*

*(Bill Gates)*

**PERSEMBAHAN**

Alhamdulillah puji syukur atas kehadirat Allah SWT atas segala rahmat dan karunianya. Sholawat serta salam penulis curahkan kepada Nabi Muhammad SAW. Penulis persembahkan laporan Tugas Akhir ini kepada:

1. Allah SWT yang telah memberi rahmat, hidayah, dan kesehatan yang selalu mengingtakan akan sesulit apapun usaha yang dilakukan akan selalu memberikan hasil yang terbaik.
2. Ibu, Bapak, kakak, dan adik terkasih yang selalu mendoakan keberhasilan dan kemudahan bagi penulis, yang senantiasa memberikan semangat dan motivasi dalam pengerjaan Tugas Akhir.
3. Sahabat seangkatan dan seperjuangan Teknik Informatika angkatan 2006, keluarga besar HsC tidak akan terlupakan.
4. Teman-teman yang tidak dapat penulis sebutkan satu-persatu yang ikut mendukung penulis dalam penyusunan laporan Tugas Akhir ini.

**KATA PENGANTAR**

Puji dan syukur penulis sampaikan kehadirat Allah SWT, Tuhan Yang Maha Pemurah lagi Maha Penyayang. Berkat rahmat, hidayah, dan inayah-Nya, akhirnya penulis dapat menyelesaikan laporan Proyek Tugas Akhir dengan judul Rancang Bangun Sistem Informasi Penjualan dan Manajemen Toko Berbasis Client Server.

Penyusunan Laporan Tugas Akhir ini diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Teknik Informatika Fakultas Bisnis Dan Teknologi Informasi Universitas Teknologi Yogyakarta. Selain itu dengan adanya Tugas Akhir ini diharapkan mampu menjadi tempat mendekatkan jarak antara teori dan aplikasi dalam bentuk yang lebih nyata, sehingga apa yang kita dapat selama menuntut ilmu dapat diimplementasikan kedunia sosial masyarakat.

Penulisan laporan Tugas Akhir ini terselesaikan karena bantuan dari berbagai pihak. Untuk itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu penulis dalam menyelesaikan laporan penelitian.

1. Bapak Prof. Bambang Hartadi, MM., Ph.D., Akt. selaku Rektor Universitas Teknologi Yogyakarta.
2. Bapak Arif Pramudwiatmoko, S.T., M.eng. selaku dosen pembimbing yang berkenan membimbing dan mengarahkan penulis, sehingga Laporan Tugas Akhir ini dapat terselesaikan.
3. IbuYuli Asriningtias, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Informatika Fakultas Bisnis dan Teknologi Informasi Universitas Teknologi Yogyakarta.
4. Teman–teman program Studi Teknik Informatika Universitas Teknologi Yogyakarta atas dukungan dan semangat dalam penyelesaian Laporan Tugas Akhir.
5. Semua pihak yang tidak dapat penulis sebutkan satu per satu, baik secara langsung maupun tidak langsung membantu dalam penulisan Tugas Akhir ini.

Semoga atas bantuan dan kerjasama yang telah diberikan menjadi amal baik dan mendapat balasan dari Allah SWT. Penulis menyadari bahwa penelitian ini jauh dari sempurna. Untuk itu diperlukannya saran dan masukan demi sempurnanya penyusunan Laporan Tugas Akhir ini. Akhir kata, penulis berharap semoga laporan Tugas Akhir ini dapat bermanfaat bagi semua yang membutuhkan.

Yogyakarta, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Penulis

INTISARI

Permasalahan yang sering dialami dalam pengelolaan sebuah toko tidak hanya pada proses transaksi jual dan beli saja, tetapi juga meliputi manajemen pegawai dan persediaan barangnya.Saat ini sistem pencatatan laporan dan transaksi dilakukan dengan cara konvensional, mencatat kedalam buku. Hal tersebut dirasa kurang efisien danakan menimbulkan beberapa masalah pada saat buku hilang, Pemilik Toko akan kesulitan untuk mendapatkan laporan laporan transaksi yang terjadi. Penelitian dilakukan melalui beberapa tahap, diantaranya pengumpulan data melalui observasi dan wawancara. Data tersebut kemudian dianalisis untuk mendapatkan gambaran dalam menentukan desain yang akan dikembangkan pada perancangan sistem selanjutnya dilakukan pengujian terhadap sistem yang telah diimplementasikan untuk mengetahui fungsi dari setiap komponen sistem. Hasil dari pengembangan sistem berupa aplikasi Penjualan dan Manajemen Toko, dan SQL Server 2005 sebagai basis datanya. Dengan adanya sistem ini Pemilik Toko dapat memanajemen kegiatan transaksi toko, dan resiko kehilangan laporan laporan transaksi lebih kecil dibanding cara pencatatan sebelumnya.

Kata Kunci : Transaksi, Tok, Manajemen, Penjualan, Client Server .

ABSTRACT

About problem which often been experienced deep management one shop not at only sell transaction processing and even buy, but also cover clerk and stock management it. Now write-up registry system and transactions did by conventional, note into book. Reducing perceives that thing efficient and will evoke severally problem upon binds books to get lost, Tradesman will handicap to get happening transactions reporting. Research is done through many phases, amongst those data collecting via observation and interview. That data then dianalisis to get picture in determine design who will be developed on succeeding system scheme done by examination to system already diimplementasikan to know function of each system component. Result of system development as application of Sell and Shop Management, and SQL IS server 2005 as basis of its data. With marks sense this system Tradesman that memanajemen can shop transactions activity, and reporting loss jeopardies smaller transactionses to be appealed previous registry tricks.

keywords: Transactions, Shop, Management, Sell, Client’s Server.

**DAFTAR ISI**

**HALAMAN JUDUL**  i

**HALAMAN PENGESAHAN**  ii

**HALAMAN PERNYATAAN**  iii

**HALAMAN SURAT KETERANGAN INSTANSI** iv

**HALAMAN MOTTO**  v

**HALAMAN PERSEMBAHAN**  vi

**KATA PENGANTAR**  vii

**INTISARI**  ix

**ABSTRACT**  x

**DAFTAR ISI**  xi

**DAFTAR GAMBAR**  xvi

**DAFTAR TABEL**  xviii

**BAB I PENDAHULUAN**

1. Latar Belakang Masalah 1
2. Rumusan Masalah 2
3. Batasan Masalah 2
4. Tujuan Penelitian 2
5. Manfaat Penelitian 3
6. Sistematika Penulisan 3

**BAB II LANDASAN TEORI**

1. Tinjauan Pustaka 5
2. Sistem 6

2.2.1 Pengertian Sistem 6

2.2.2 Elemen Sistem 7

2.2.3 Klasifikasi Sistem 8

1. Sistem Informasi 9

2.3.1 Definisi Sistem Informasi 9

2.3.2 Komponen Dasar Sistem Informasi 10

1. Penjualan 11

2.4.1 Pengertian Penjualan 11

2.4.2 Harga Pokok Penjualan 11

1. Analisis Sistem 12

2.5.1 Pengertian Anlisis Sistem 12

2.5.1 Langkah Langkah Anlisis Sistem 13

1. Entity Relationship Diagram 13
2. Basis Data 15

2.7.1 Konsep Dasar Basis Data 15

2.7.2 Prinsip dan Tujuan Basis Data 16

2.7.3 Komponen Basis Data 16

2.7.4 Elemen Elemen Sistem Manajemen Basis Data 16

2.7.5 Persyaratan Basis Data 17

2.7.6 Abstraksi Data 19

2.8. DBMS 19

2.8.1 Tujuan DBMS 21

2.8.2 DBA 21

* + 1. Komponen Software DBMS 21

2.9 DAD (Diagram Arus Data) 22

2.9.1 Pengertian DAD 22

2.9.2 Komponen DAD 22

2.9.3 Pedoman Menggambar DAD 26

2.10 Arsitektur Basis Data 26

2.10.1 Basis Data SQL Server2005 26

2.10.2 Tipe Data 27

2.10.3 Fungsi SQL Server2005 27

2.11 Borland Delphi 7 27

**BAB III METODE PENELITIAN**

1. Metode Pengambilan Data 29
2. Analisis Sistem 30
3. Perancangan Sistem 30
4. Implementasi Sistem 31
5. Perangkat Pendukung 31

**BAB IV ANALISIS DAN PERANCANGAN SISTEM**

1. Analisis Sistem 33
2. Perancangan Sistem 34
   * 1. Diagram Konteks 34
     2. Diagram Berjenjang 35
     3. DAD (Diagram Alir Data) 36
        1. Diagram Alir Data Level 1 36
        2. Diagram Alir Data Level 2 Proses 1 37
        3. Diagram Alir Data Level 2 Proses 2 38
        4. Diagram Alir Data Level 2 Proses 3 39
3. Gambaran Sistem 40
4. Identifikasi Entitas 40
5. Entity Relationship Diagram (ERD) 41
6. Perancangan Tabel 41
7. Relasi Antar Tabel 47
8. Perancangan Antarmuka 47

**BAB V IMPLEMENTASI SISTEM**

1. Cara Kerja Program 61
2. Pembahasan Unjuk Kerja Sistem 62
3. Pengaturan Basis Data 62
4. Form Login 63
5. Form Utama 64
6. Form Data Barang 64
7. Form Data Pulsa 65
8. Form Data Supplier 67
9. Form Data Pegawai 68
10. Form Pembelian 69
11. Form Penjualan 71
12. Form Retur Barang 72
13. Form Laporan Pembelian Barang 73
14. Form Grafik Penjualan 74

**BAB VI PENUTUP**

1. Simpulan 75
2. Saran 76

**DAFTAR PUSTAKA**

**LAMPIRAN**

**DAFTAR GAMBAR**

**Gambar 2.1** Analisis Sistem 12

**Gambar 2.2** One to One Relationship 14

**Gambar 2.3** One to Many Relationship 14

**Gambar 2.4** Many to Many Relationship 14

**Gambar 2.5** Notasi Kestuan Luar 23

**Gambar 2.6** Alur Data 23

**Gambar 2.7** Konsep Paket Dari Data 24

**Gambar 2.8** Konsep Alur Data Mengumpul 24

**Gambar 2.9** Konsep Alur Data Menyebar 24

**Gambar 2.10** Sumber dan Tujuan 25

**Gambar 2.11** Notasi Proses 25

**Gambar 2.12** Simbol Simpanan Data 26

**Gambar 4.1** Diagram Konteks 35

**Gambar 4.2** Diagram Berjenjang 35

**Gambar 4.3** DAD Level 1 36

**Gambar 4.4** DAD Level 2 Proses 1 37

**Gambar 4.5** DAD Level 2 Proses 2 38

**Gambar 4.6** DAD Level 2 Proses 3 39

**Gambar 4.7** Relasi Antar Entitas 41

**Gambar 4.8** Relasi Antar Tabel 47

**Gambar 4.9** Rancangan Form Menu Utama 48

**Gambar 4.10** Rancangan Form Pengaturan Basis Data 49

**Gambar 4.11** Rancangan Form Menu Login 49

**Gambar 4.12 R**ancangan Form Data Barang 50

**Gambar 4.13 R**ancangan Form Data Pulsa 50

**Gambar 4.14 R**ancangan Form Data Supplier 51

**Gambar 4.15 R**ancangan Form Data Pegawai 51

**Gambar 4.16 R**ancangan Form Data Petugas Toko 52

**Gambar 4.17 R**ancangan Form Kategori Barang 52

**Gambar 4.18 R**ancangan Form Satuan Barang 53

**Gambar 4.19 R**ancangan Form Stok Minimal 53

**Gambar 4.20 R**ancangan Form Data Gaji Pokok 53

**Gambar 4.21 R**ancangan Form Tunjangan Gaji Pegawai 54

**Gambar 4.22 R**ancangan Form Pembelian 54

**Gambar 4.23 R**ancangan Form Penjualan 55

**Gambar 4.24 R**ancangan Form Retur Barang 55

**Gambar 4.25 R**ancangan Form Penggajian 56

**Gambar 4.26 R**ancangan Form Penjualan Pulsa 56

**Gambar 4.27 R**ancangan Form Laporan Data Barang 57

**Gambar 4.28 R**ancangan Form Laporan Data Supplier 57

**Gambar 4.29 R**ancangan Form Laporan Pembelian 58

**Gambar 4.30** Rancangan Form Laporan Penjualan 59

**Gambar 4.31** Rancangan Form Laporan Penggajian 59

**Gambar 4.32** Rancangan Form Laporan Penjualan Pulsa 59

**Gambar 4.33** Rancangan Form Laporan Barang Retur 59

**Gambar 5.1** Tampilan Pengaturan Basis Data 62

**Gambar 5.2** Tampilan Form Login 63

**Gambar 5.3** Pesan Validasi Gagal 63

**Gambar 5.4** Tampilan Form Utama 64

**Gambar 5.5** Tampilan Form Data Barang 65

**Gambar 5.6** Tampilan Form Data Pulsa 66

**Gambar 5.7** Tampilan Form Data Supplier 67

**Gambar 5.8** Tampilan Form Pegawai 68

**Gambar 5.9** Tampilan Form Pembelian 69

**Gambar 5.10** Tampilan Form Penjualan 71

**Gambar 5.11** Tampilan Form Detail Penjualan 72

**Gambar 5.12** Tampilan Form Retur Barang 73

**Gambar 5.13** Tampilan Form Laporan Pembelian Barang 74

**Gambar 5.14** Tampilan Form Grafik Penjualan 74

**DAFTAR TABEL**

**Tabel 4.1** Tabel Pegawai 41

**Tabel 4.2** Tabel Barang 42

**Tabel 4.3** Tabel Penjualan 42

**Tabel 4.4** Tabel Pembelian 43

**Tabel 4.5** Tabel Supplier 43

**Tabel 4.6** Tabel Detail Pembelian 43

**Tabel 4.7** Tabel Detail Penjualan 44

**Tabel 4.8** Tabel Kategori 44

**Tabel 4.9** Tabel Satuan 44

**Tabel 4.10** Tabel Retur 44

**Tabel 4.11** Tabel Penggajian 45

**Tabel 4.12** Tabel Tunjangan 45

**Tabel 4.13** Tabel Gaji Pokok 46

**Tabel 4.14** Tabel Login 46

**Tabel 4.15** Tabel Pulsa 46

**Tabel 4.16** Tabel Jual Pulsa 46

**BAB I**

**PENDAHULUAN**

1. Latar Belakang

Cahaya Phonecell, merupakan salah satu toko yang bergerak dibidang jual beli telepon seluler, kartu perdana, pulsa dan aksesoris dengan berbagai macam model untuk setiap jenisnya seperti casing, charger, pelindung layar.

Pada saat ini Cahaya Phonecell masih menggunakan sistem pencatatan transaksi secara konvensional pada setiap kegiatan transaksi yang dilakukan, seringkali timbul permasalahan yang terjadi baik dalam kegiatan operasional jual beli maupun pengarsipan data transaksi dan pendataan.

Dalam perkembangan usahanya, Cahaya Phonecell memiliki banyak faktor yang perlu mendapat perhatian dan dukungan yang maksimal dari segi pengelolaannya untuk mencapai tujuan operasional, terutama dalam kegiatan manajemen toko dan manajemen gaji pegawai.

Teknologi informasi merupakan salah satu alternatif utama untuk mendukung kegiatan operasional toko dalam menyelesaikan masalah usaha jual beli yang meliputi transaksi penjualan, transaksi pembelian, perhitungan harga pokok penjualan, arsip laporan penjualan dan pembelian berikut penghitungan laba dan rugi, pendataan persediaan barang, serta manajemen gaji pegawai.

Dengan menerapkan teknologi informasi dalam kegiatan manajemen toko, maka proses transaksi dapat dilaksanakan dengan lebih cepat, cermat dan efisien, sehingga diharapkan akan meningkatkan eksistensi dan memudahkan dalam pengelolaan toko.

Berdasar latar belakang tersebut di atas mendasari penulis untuk memilih judul penelitian tugas akhir “Rancang Bangun Sistem Informasi Penjualan dan Manajemen Toko Berbasis Client Server (Studi Kasus Cahaya Phonecell Yogyakarta)”.

1. Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, maka permasalahaan yang dapat dirumuskan sebagai berikut :

1. Bagaimana merancang dan membangun sebuah sistem pengelolaan data yang dapat menyelesaikan permasalahan dalam kegiatan operasional jual beli?
2. Bagaimana Sistem informasi yang dibangun dapat memberikan kemudahaan bagi pengarsipan data transaksi yang meliputi, data transaksi penjualan, transaksi pembelian, dan pendataan barang?
3. Bagaimana Sistem informasi yang dibangun dapat membantu memanajemen gaji pegawai pada toko Cahaya Phonecell?.
4. Batasan Masalah

Dengan beberapa permasalahan yang terdapat pada objek penelitian ini, maka penulis membatasi ruang lingkup masalah, dan batasan masalah tersebut adalah :

* 1. Sistem yang dibangun berbasis client server.
  2. Sistem ini membahas mengenai penjualan secara tunai.
  3. Laporan berupa laporan penjualan, laporan persediaan barang, dan laporan data gaji pegawai.
  4. Sistem akan menampilkan ketika ada persedian barang yang hampir habis.
  5. Sistem akan menampilkan produk terlaris dalam bentuk grafis.

1. Tujuan Penelitian

Adapun tujuan dari penilitian ini adalah :

1. Merancang dan membangun sebuah sistem pengelolaan data yang dapat menyelesaikan permasalahan dalam kegiatan operasional jual beli.
2. Sistem informasi yang dibangun dapat memberikan kemudahaan bagi pengarsipan data transaksi yang meliputi, data transaksi penjualan, transaksi pembelian, dan pendataan barang.
3. Sistem informasi yang dibangun dapat membantu memanajemen gaji pegawai pada Toko Cahaya Phonecell.
4. Manfaat Penelitian

Dengan penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

* 1. Dapat memberikan pemahaman tentang aplikasi manajemen toko berbasis client server.
  2. Dapat mempermudah dan meningkatkan kinerja pada instansi yang bersangkutan.
  3. Sistem informasi yang dibangun dapat membantu memanajemen gaji pegawai pada toko Cahaya Phonecell.

1. Sistematika Penulisan

Untuk memberikan gambaran yang jelas serta memudahkan pemahaman laporan yang akan disusun, maka sistematika penulisan disusun sebagai berikut:

**BAB I PENDAHULUAN**

Merupakan bab pembuka yang berisi sub-sub informasi mengenai hal-hal yang mendasari penyusunan Proyek Tugas Akhir ini. Sub-sub bagian ini terdiri dari latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Bab ini berisi tinjauan pustaka dan akan diuraiakan dasar teori yang mengarahkan pada analisa data yang ada, yang terdiri dari konsep dasar sistem, konsep sistem informasi dan pembahasan dasar sistem informasi.

**BAB III METODE PENELITIAN**

Dalam bab ini menguraikan mengenai teknik dan metode penelitian yang dilakukan untuk memperoleh data dan informasi mengenai topik yang penulis ambil.

**BAB IV ANALISA DAN PERANCANGAN SISTEM**

Dalam bab ini akan dibahas tentang analisa sistem serta perancangan sistem mencakup diagram arus data, desain basis data, alur program dan rancangan masukandan keluaran.

**BAB V IMPLEMENTASI SISTEM**

Bab ini berisi tentang bagaimana cara kerja sistem, pengujian sistem dan membahas secara lebih terinci tentang unjuk kerja dari sistem secara keseluruhan.

**BAB VI PENUTUP**

Bab ini sebagai penutup dari bab-bab sebelumnya yang berisi simpulan dan saran dari penulis tentang Implementasi Sistem Penjualan dan Stok Barang.

**DAFTAR PUSTAKA**

**LAMPIRAN**

**BAB II**

**LANDASAN TEORI**

* + - 1. **Tinjauan Pustaka**

Penelitian ini mengambil dari beberapa refrensi Proyek Tugas Akhir Dormawati Siburian (2012) dengan judul “Implementasi Pengolahan data penjualan aksesoris computer berbasis client server” dengan Studi Kasus Wara Komputer Yogyakarta, penelitian tersebut membahas tentang sistem penjualan barang secara client server, dengan detail transaksi meliputi transaksi penjualan, pembelian dan persediaan barang, laporan berupa laporan penjualan dan laporan persediaan barang.

Laporan hasil penelitian kedua yang dilakukan oleh saudari Anna Fenny Dasmasela (2012) tentang sistem informasi penjualan dan pembelian pada Koperasi Mahasiswa Universitas Merdeka Malang. Sistem informasi penjualan yang dibangun berbasis desktop, sistem penjualan tersebut lebih kompleks dalam proses transaksinya, dimulai dari proses transaksi penjualan, transaksi pembelian, persediaan barang, hingga data kepegawaian termasuk didalamnya perhitungan gaji pegawai.

Laporan penelitan yang terakhir oleh saudara Fredy Nixen Kanam (2012) yang membahas sistem manajemen penjualan dan produksi pada distro ”N” Pearlee, penelitian tersebut membahas mengenai permasalahanyang dihadapi dalam mengatasi transaksi penjualan, pembelian, pemesananserta produksi.

Berdasarkan beberapa referensi tersebut di atas sebagai acuan penulis untuk mengembangkan penelitian tersebut dengan membangun sistem informasi manajemen toko denganan menambahkan beberapa fungsi baru seperti penentuan harga pokok, diskon harga barang, laporan penjualan, pembelian, persediaan barang, laporan laba rugi yang dapat dilihat secara berkala, serta laporan kepegawaian.

* + - 1. **Sistem**

**2.2.1 Pengertian Sistem**

Pengertian sistem menurut para ahli dalam mendefinisikan konsep dasar sistem. Indrajit seperti yang dikutip Hutahaean (2014) mendefinisaikan sistem mengandung arti kumpulan kumpulandari komponen komponen yang dimiliki unsur keterkaitan antara satu dengan lainya. Sedang menurut Jogiyanto yang dikutip Hutahaean (2014), sistem sistem yang menekankan pada komponen penyusun sistem, didefinisikan sebagai kumpulan baik abstrak maupun fisik yang saling berinteraksi untuk mencapai tujuan tertentu.

Ada beberapa definisi tentang sistem, yaitu:

**Menurut Jerry FithGerald** ;

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

**Menurut Ludwig Von Bartalanfy** ;

Sistem merupakan seperangkat unsur yang saling terikatdalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.

**Menurut Anatol Raporot** ;

Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.

**Menurut L. Ackof** ;

Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiri dari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.

Terdapat dua kelompok pendekatan di dalam mendefinisikan sistem, yaitu yang menekankan pada prosedurnya dan yang menekankan pada komponen atau elemennya.

* 1. Pendekatan sistem yang lebih menekankan pada prosedur mendefinisikan sistem sebagai berikut :

Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu masalah guna mencapai sasaran tertentu.

* 1. Pendekatan sistem yang lebih menekankan pada komponen atau elemennya mendefinisikan sistem sebagai berikut :

Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu oleh Jogiyanto yang dikutip Hutahaean (2014).

Dari kedua definisi ini benar dan tidak bertentangan satu sama lainnya sementara yang membedakan adalah pendekatan pembahasan mengenai sistem itu sendiri. Suatu sistem akan lebih mudah untuk dipelajari bila kita mengetahui apa arti dan fungsi dari sistem itu sendiri. Dengan demikian pengertian dari definisi sistem akan mempunyai peranan yang penting dalam mempelajari cara kerja suatu sistem.

Pendekatan suatu sistem yang akan menekankan pada suatu komponen agar lebih mudah untuk dipelajari untuk tujuan analisis dan perancangan suatu sistem. Untuk menganalisis dan merencanakan suatu sistem, analis dan perancang sistem harus mengerti terlebih dahulu mengenai komponen-komponen atau elemen-elemen atau subsistem-subsistem dari sistem tersebut.

Syarat-Syarat Sistem :

1. Sistem harus dibentuk untuk menyelesaikan tujuan.
2. Elemen sistem harus mempunyai rencana yang ditetapkan.
3. Adanya hubungan di antara elemen sistem.
4. Unsur dasar dari proses (arus informasi, energi dan material) lebih penting dari pada elemen sistem.
5. Tujuan organisasi lebih penting dari pada tujuan elemen.

**2.2.2 Elemen Sistem**

Sistem terdiri dari beberapa elemen – elemen sistem yang meliputi :

1. Batasan sistem

Merupakan sesuatu yang membatasi sistem dalam mencapai tujuan sistem. Batasan sistem menunjukkan ruang lingkup dari sistem tersebut.

1. Kontrol sistem

Merupakan pengawasan terhadap pelaksanaan pencapaian tujuan dari sistem tersebut. Kontrol sistem dapat berupa kontrol terhadap pemasukan data, kontrol terhadap keluaran data, kontrol terhadap pengolahan data, kontrol terhadap umpan balik dan sebagainya.

1. Masukan

Merupakan elemen dari sistem yang bertugas untuk menerima seluruh masukan data, dimana masukan tersebut berupa jenis data, frekuensi pemasukan data dan sebagainya.

1. Proses

Merupakan elemen dari sistem yang bertugas untuk mengolah atau memproses seluruh masukan data menjadi suatu informasi yang lebih berguna. misalnya sistem produksi akan mengolah bahan baku yang berupa bahan mentah menjadi bahan jadi siapa untuk digunakan.

1. Keluaran

Merupakan hasil dari masukan yang telah diproses oleh bagian pengolah dan merupakan tujuan akhir sistem.

1. Umpan balik

Merupakan elemen dalam sistem yang bertugas mengevaluasi bagian output yang dikeluarkan, dimana elemen ini sangat penting demi kemajuan sebuah sistem. Umpan balik ini dapat berupa perbaikan sistem, pemeliharaan sistem dan sebagainya.

**2.2.3 Klasifikasi Sistem**

Menurut sudut pandangnya, dibagi beberapa bagian diantaranya adalah sebagai berikut:

1. Sistem abstrak dan sistem fisik.

Sistem abstrak merupakan sistem yang tidak bisa dilihat secara mata biasa dan biasanya sistem ini berupa pemikiran atau ide-ide. Sedangkan sistem fisik merupakan sistem yang bisa dilihat secara mata biasa dan biasanya sering digunakan oleh manusia.

1. Sistem alamiah dan sistem buatan manusia

Sistem alamiah merupakan sistem yang terjadi karena pengaruh alam, tidak dibuat manusia. Sedangakan sistem buatan manusia adalah sistem yang dirancang dan dibuat oleh manusia.

1. Sistem tertutup dan sistem terbuka

Sistem tertutup merupakan sistem yang tidak berhubungan dengan bagian luar sistem dan biasanya tidak terpengaruh eleh kondisi di luar sistem. Sedangkan sistem terbuka merupakan sistem yang berhubungan dengan bagian luar sistem.

**2.3 Sistem Informasi**

**2.3.1 Definisi Sistem Informasi**

Sistem **informasi adalah** sebuah sistem informasi yang mempunyai fungsi mengumpulkan, memproses, menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik (Kadir,2014).

Sistem **informasi menurut Alter (Kadir,2014) adalah** kombinasi antara prosedur kerja, informasi, orang, dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah perusahaan.

Adapun menurut Leitch seperti yang dikutip Kadir(2014) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan- laporan yang diperlukan.

Ada empat operasi dasar dari sistem informasi yaitu mengumpulkan, mengolah, menyimpan dan menyebarkan informasi. Informasi mungkin dikumpulkan dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau ke luar organisasi.

**2.3.2 Komponen Dasar Sistem Informasi**

1.  Blok Masukan

Input yang mewakili data yang masuk ke dalam sistem informasi. Input ini termasuk metode-metode dan media untuk menangkap data yang akan dimasukan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di dasar data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

3.   Blok Keluaran

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4.   Blok Teknologi

Teknologi merupakan kotak alat (*tool box*) dari sistem informasi. Teknologi digunakan untuk menerima input, mrnjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem informasi. Teknologi terdiri atas 2 bagian utama, yaitu perangkat lunak (*software*) dan perangkat keras *(hardware).*

5.  Blok Basis Data

Basis data merupakan kumpulan dari data yang saling berhubungan satu sama lainnya, tersimpan di perangkat keras dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan dalam dasar data untuk keperluan penyediaan informasi lebih lanjut.

6.   Blok Kendali

Supaya sistem informasi dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian-pengendalian di dalamnya. Banyak hal yang dapat merusak sistem informasi, seperti bencana alam, api, temperatur, air, debu, kecurangan-kecurangan, kegagalan-kegagalan sistem itu ssendiri, kesalahan-kesalahan, ketidakefisienan sabotase dan lain-lain.

**2.4 Penjualan**

**2.4.1 Pengertian Penjualan**

Penjualan merupakan pembelian sesuatu (barang atau jasa) dari suatu pihak kepada pihak lainnya dengan mendapatkan ganti uang dari pihak tersebut. penjualan yang diharapkan tidak tercapai dan pendapatan pun akan berkurang.

Pengertian penjualan menurut Simamora(2010) adalah pendapatan lazim dalam perusahaan dan merupakan jumlah kotor yang dibebankan kepada pelanggan atas barang dan jasa.

Sedang pengertian penjualan menurut Marom(2010) penjualan artinya penjualan barang dagangan sebagai usaha pokok perusahaan yang biasanya dilakukan secara teratur.

Berdasarkan pengertian di atas, maka dapat disimpulkan bahwa penjualan adalah persetujuan kedua belah pihak antara penjual dan pembeli, dimana penjual menawarkan suatu produk dengan harapan pembeli dapat menyerahkan sejumlah uang sebagai alat ukur produk tersebut sebesar harga jual yang telah disepakati.

**2.4.2 Harga Pokok Penjualan**

Seluruh biaya yang dikeluarkan untuk memperoleh barang yang dijual atau harga perolehan dari barang yang dijual. Apabila harga jual lebih besar dari harga pokok penjualan maka akan diperoleh laba, dan sebaliknya apabila harga jual lebih rendah dari harga pokok penjualan akan diperoleh kerugian.

Adapun manfaat dari harga pokok penjualan sebagai patokan untuk menentukan harga jualdan untuk mengetahui laba yang diinginkan perusahaan.

**2.5 Analisis Sistem**

**2.5.1 Pengertian Analisis Sistem**

Ada yang mendefinisikan Sistem Analis sebagai :

1. Seorang yang menggunakan pengetahuan aplikasi komputer yang dimilikinya untuk memecahkan masalah-masalah bisnis, dibawah petunjuk Manajer Sistem.
2. Seorang yang bertanggung jawab menterjemahkan kebutuhan - kebutuhan

si pengguna sistem (user) ke dalam spesifikasi teknik yang diperlukan oleh Programmer dan diawasi oleh Manajemen.

Pengertian sistem analis ini dapat digambarkan sebagai berikut :

****

**Gambar 2.1** Analisis Sistem

Jadi Analisis Sistem dapat didefinisikan sebagai berikut:

“Penguraian dari suatu sistem informasi yang utuh keadaan bagian-bagian komponennnya dengan maksud untuk mengidenifikasikan dan mengevaluasi permasalahan, kesempatan, hambatan yang terjadi, dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya”.

Fungsi Analisis Sistem adalah

1. Mengidentifikasikan masalah-masalah dari user.
2. Menyatakan secara spesifik sasaran yang harus dicapai untuk memenuhi kebutuhan user.
3. Memilih alternatif-alternatif metode pemecahan masalah.
4. Merencanakan dan menerapkan rancangan sistemnya sesuai dengan permintaan user.

**2.5.2 Langkah - Langkah Analisis Sistem**

Tahap analis merupakan tahap yang kritis dan sangat penting, karena kesalahan di tahap ini akan menyebabkan juga kesalahan pada tahap selanjutnya.

Tugas dari analisis sistem yaitu menganalisis sistem untuk untuk menemukan beberapa kelemahan sehingga dapat diusulkan perbaikannya.

Di dalam tahap analis sistem terdapat langkah-langkah dasar yang harus dilakukan oleh analis sebagai berikut:

1. *Identify,* yaitu mengidentifikasikan masalah.
2. *Understand,* yaitu memahami kerja dari sistem yang ada.
3. *Analyze,* yaitu menganalisis sistem.
4. *Report,* yaitu membuat laporan hasil analisis.

**2.6 Entity Relationship Diagram (ERD).**

Metode ini menggambarkan hubungan antar tabel yang ada dalam suatu database. Pertama kali yang dilakukan adalah membagi database menjadi beberapa tabel yang disebut object atau entity. Masing-masing entity ini mempunyai beberapa atribut, yang merupakan bagian dari entity tersebut. Antar beberapa entity tersebut dapa digambarkan suatu hubungan (relationship) yang biasanya dipakai dalam ERD. ERD digunakan untuk menggambarkan hubungan antar entity. Untuk menghubungkan satu entity dengan entity yang lain digunakan entity key, yaitu suatu atribut tertentu atau sekelompok atribut tertentu yang bersifat unik, sehingga dapat digunakan untuk membedakan suatu anggota entity dengan anggota yang lain pada entity yang sama. Selain itu harus ada pula yang disebut relationship key, yaitu suatu relationship yang digunakan untuk menyatakan hubungan antar entity key.

Macam-macam hubungan antar entity:

1. One to one relationship

Merupakan suatu hubungan dimana satu anggota *entity* mempunyai hubungan dengan satu anggota *entity* pada *entity* yang berbeda. Ada 2 macam hubungan, yaitu obligatory dan non obligatory. Obligatory adalah semua anggota dari suatu *entity* harus berpartisipasi atau mempunyai hubungan dengan *entity* yang lain.



**Gambar 2.2** One to One Relationship

1. One to Many Relationship

Merupakan suatu hubungan antara suatu anggota *entity* yang satu dengan beberapa anggota *entity* pada *entity* yang berbeda. Hubungan ini juga bias dua macam yaitu *obligatory* dan *non obligatory*.



**Gambar 2.3** One to Many Relationship

1. Many to Many Relationship

Merupakan hubungan antara beberapa anggota *entity* yang satu dengan beberapa anggota *entity* pada *entity* yang lain. Jadi kedua belak pihak bias mempunyai hubungan lebih dari satu dengan beberapa anggota *entity*. Hubungan *entity* juga bias dua macam yaitu *obligatory* dan *non obligatory*. Contoh satu orang mahasiswa mempunyai mata kuliah yang lebih dari satu dan mata kuliah dapat diikuti oleh lebih dari satu orang mahasiswa.



**Gambar 2.4** Many to Many Relationship

**II.7 Basis Data (Database)**

Secara sederhana Basis Data *(Database)* dapat diungkapkan sebagai suatu pengorganisasian data dengan bantuan komputer yang memungkinkan data dapat diakses dengan mudah dan cepat. Dalam hal ini, pengertian akses dapat mencakup perolehan data, seperti menambah data dan menghapus data (Kadir, 2014).

Dengan adanya komputer, data dapat disimpan dalam media pengingat yang disebut *harddisk*. Dengan menggunakan media ini, kehadiran kertas yang digunakan untuk menyimpan data dapat dikurangi. Selain itu data menjadi lebih cepat untuk diakses terutama kalau dikemas dalam bentuk *database*.

DBMS singkatan dari *Database Management System*. DBMS merupakan perangkat lunak atau program komputer yang dirancang secara khusus untuk memudahkan pengelolaan *database* (Kadir, 2014).

**2.7.1 Konsep Dasar Basis Data**

Basis data merupakan kumpulan data yang saling berhubungan (Kadir,2014). Relasi biasanya ditunjukan dengan kunci dari tiap file yang ada. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, yang merupakan satu kumpulan entitas yang seragam. Satu record terdiri dari *field* yang saling berhubungan menunjukan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. Dari pengertian diatas dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu:

1. Bersifat *data oriented* dan bukan *program oriented*
2. Dapat digunakan oleh beberapa program aplikasi tanpa mengubah basis datanya
3. Dapat dikembangkan dengan mudah, baik *volume* maupun strukturnya
4. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah
5. Dapat digunakan dengan cara-cara yang berbeda.

**2.7.2 Prinsip dan Tujuan Basis Data**

Prinsip utama basis data adalah pengaturan data dengan tujuan utama fleksibelitas dan kecepatan dalam pengambilan data kembali. Adapun tujuan basis data diantaranya adalah sebagai berikut:

1. Efisiensi meliputi *speed, space* dan *accurancy*
2. Menangani data dalam jumlah besar
3. Kebersamaan pemakaian *(Sharebility)*
4. Meniadakan duplikasi dan *inkonsistensi* data.

**2.7.3 Komponen Basis Data**

Sistem basis data merupakan perpaduan antara basis data dan SMBD (Sistem Manajemen Basis Data). Komponen-komponen sistem basis data meliputi:

1. Perangkat Keras *(Hardware)* sebagai pendukung operasi pengolahan data
2. Sistem Operasi *(Operating System)* atau perangkat lunak untuk mengelola basis datanya
3. Basis data *(Database)* sebagai inti dari sistem basis data
4. Sistem Manajemen Basis Data (SMBD)
5. Pemakai *(User)*
6. Aplikasi lain.

**2.7.4 Elemen-Elemen Sistem Manajemen Basis Data**

Elemen-elemen sistem manajemen basis data relational adalah:

1. *Entitas*

Entitas adalah orang, tempat, kejadian atau konsep yang informasinya direkam.

1. *Atribut*

Atribut biasa disebut juga *data elemen*, *data field*, atau *data item* yang digunakan untuk menerangkan suatu entitas dan mempunyai harga tertentu.

1. *Data Value* (Nilai Data)

*Data Value* adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut.

1. *File*/Tabel

Kumpulan *record* sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

1. *Record/Tuple* (Baris)

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap atau *record* bisa juga didefenisikan sebagai kumpulan kolom yang menyatakan suatu data yang saling terkait. Satu *record* mewakili satu data atau informasi.

1. *Field* (Kolom)

Didefenisikan sebagai satuan data terkecil dalam sebuah tabel.

1. *Database* (Basis Data)

Merupakan kumpulan *file-file* yang mempunyai kaitan antara satu file dengan yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.

1. Query

Merupakan perintah yang dirancang untuk memenggil kelompok *record* tertentu dari satu file atau lebih untuk melakukan opeasi pada file.

1. View

Data yang terdiri atas jumlah *record* yang tampak atau diproses dalam urutan penampilan.

1. *Indeks*

Merupakan suatu mekanisme dalam *database* yang memungkinkan pencarian data dapat dilakukan dengan cepat.

**2.7.5 Persyaratan Basis Data**

Ada beberapa ketentuan yang harus diperhatikan pada pembuatan *file* basis data agar dapat memenuhi kriteria sebagai suatu basis data, yaitu:

1. Redudansi dan inkonsistensi data: penyimpanan data yang sama dibeberapa tempat disebut redundansi, hal ini menyebabkan pemborosan dan menimbulkan inkonsistensi data (data tidak konsisten) karena bila terjadi perubahan terhadap data maka data harus dirubah dibeberapa tempat, hal ini tentunya tidak efisien.
2. Pengaksesan data dalam basis data harus siap diakses oleh siapa saja yang membutuhkan dan mempunyai hak untuk mengaksesnya. Oleh karena itu perlu dibuat suatu program pengelolaan atau suatu aplikasi untuk mengakses data yang dikenal sebagai DBMS.
3. Data terisolasi untuk standarisasi, jika data tersebar dalam beberapa *file* dalam bentuk format yang tidak sama, maka akan menyulitkan dalam menulis program aplikasi untuk mengambil dan menyimpan data, oleh karena itu data dalam satu *database* harus dibuat satu format sehingga mudah dibuat program aplikasinya.
4. Masalah keamanan (*security*), tidak setiap pemakai sistem basis data diperbolehkan untuk mengakses semua data, misalnya data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan dan personalia, sedang bagian gudang tidak diperkenankan untuk membukanya. Keamanan dapat diatur dan disesuaikan baik ditingkat basis data atau aplikasinya.
5. Masalah integritas *(Integrity)*, basis data berisi *file/*tabel yang saling berhubungan, masalah utama adalah bagaimana kaitan antar file tersebut terjadi meski diketahui bahwa file A terkait dengan file B, namun secara teknis ada field yang mengaitkan kedua filetersebut oleh karena itu field kunci tidak dapat diabaikan dalam merancang suatu basis data.
6. *Multiple user*, salah satu alasan basis data dibangun adalah karena nantinya data tersebut digunakan oleh banyak orang, baik dalam waktu berbeda maupun bersamaan sehingga kebutuhan akan basis data handal yang mendukung banyak pemakai perlu dipertimbangkan.

**2.7.6 Abstraksi Data**

Kegunaan utama sistem basis data adalah agar pemakai mampu menyusun suatu pandangan abstraksi dari data. Bayangan mengenai data tidak lagi memperhatikan kondisi sesungguhnya bagaimana satu data masuk ke basis data disimpan dalam sektor mana, tetapi menyangkut secara menyeluruh bagaimana data tersebut dapat diabstraksikan mengenai kondisi yang dihadapi oleh pemakai sehari-hari.

Sistem yang sesungguhnya, tentang teknis bagaimana data disimpan dan dipelihara seakan-akan disembunyikan kerumitannya dan kemudian diungkapkan dalam bahasa dan gambar yang mudah dimengerti oleh orang awam. Ada tiga kelompok pemakai dalam tingkatan abstraksi saat memandang suatu basis data, yaitu :

1. Level Fisik: level ini merupakan level abstraksi paling rendah karena menggambarkan bagaimana data disimpan dalam kondisi sebenarnya.
2. Level Konseptual: level ini menggambarkan data apa yang disimpan dalam basis data dan hubungan relasi yang terjadi antara data dari keseluruhan basis data. Pemakai tidak mempedulikan kerumitan dalam struktur level fisik lagi, penggambaran cukup dengan memakai kotak, garis, dan hubungan secukupnya.
3. Level Pandangan Pemakai *(View level)*: level ini merupakan level abstraksi data tertinggi yang menggambarkan hanya sebagian saja yang dilihat dan digunakan dari keseluruhan basis data, hal ini disebabkan beberapa pengguna tidak membutuhkan semua isi basis data.

**2.8 DBMS (*Database Management System*)**

Suatu DBMS berisi satu koleksi data dan satu set program untuk mengakses data tersebut, jadi DBMS terdiri dari basis data dan set program untuk menambah data, manghapus data, mengambil dan membaca data. Set program pengelola merupakan suatu paket program yang dibuat agar memudahkan dan mengefisienkan pemasukan atau perekam informasi dan pengambilan atau pembacaan informasi kedalam basis data. Jadi lingkungan SMBD (Sistem Management Basis Data) terdiri dari :

1. Komponen perangkat lunak yang membentuk sistem manajemen basis data.
2. Perangkat lunak sistem yang berinteraksi dengan sistem manajemen basis data.
3. Utilitas, *loading* dan *backup*, reorganisasi *file*, *report generation,* maupun *performance monitoring.*

Paket bahasa dalam DBMS dibagi menjadi beberapa definisi, yaitu:

1. DDL *(Data Definition Language)*: DDL merupakan satu paket bahasa DBMS yang berguna untuk melakukan spesifikasi terhadap skema basis data. Hasil kompilasi dari DDL adalah satu set tabel yang disimpan dalam file khusus yang disebut *Data Directory/Dictionery.* Contoh perintah DDL misalnya, *Create Table, Create Index, Alter, Drop*.
2. DML *(Data Manipulation Language)*: DML merupakan satu paket DBMS yang memperbolehkan pemakai untuk mengakses atau memanipulasi data sebagaimana yang telah diorganisasikan sebelumnya dalam model data yang tepat, dengan DML dapat dilakukan:
3. Mengambil informasi yang tersimpan dalam basis data.
4. Menyisipkan informasi baru dalam basis data.
5. Menghapus informasi dari tabel.

Terdapat dua tipe DML yaitu prosedural dan non prosedural:

1. Prosedural DML

membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan dan bagaimana cara mendapatkannya. Misalnya: dBase III, FoxBase.

1. Non prosedural DML membutuhkan pemakai untuk menspesifikasikan data apa yang dibutuhkan tanpa tahu bagaimana cara mendapatkannya. Misalnya: SQL *(Structured Query Language)* atau QBE *(Query By Example).*

**2.8.1 Tujuan DBMS**

Mendapatkan data spesifik dalam sebuah format yang mempertemukan informasi dengan pengguna yang berbeda sama sekali merupakan unsur yang berlainan. Adapun poin yang sangat penting mengenai basis data ialah bahwa data dan basis data haruslah menyatu dan saling berhubungan dengan data yang lain menggunakan *field-field* kunci guna mempertemukan kepentingan pengguna.

**2.8.2 DBA (Database Administrator)**

Dalam lingkungan pemprosesan *file* tradisional, pengintegrasian data dikerjakan dengan program-program yang tidak terintegrasi oleh programer-programer yang individual. Cukup lama, orang-orang tidak melakukan komunikasi dengan yang lain untuk mendiskusikan struktur data yang terbaik untuk organisasi mereka, bahkan struktur data individual untuk satu aplikasi atau kumpulan aplikasi.

Sebuah lingkup basis data harusnya mempunyai satu orang atau sekelompok orang, dalam bagian struktur basis data untuk menangani administrasi basis data yang biasa disebut administrator basis data/DBA *(DataBase Administrator)*. DBA bertanggung jawab untuk bekerjasama dengan analis dan pengguna untuk melengkapi berbagai tugas seperti mendefinisikan data, pemodelan data, disain basis data, menjamin keserasian integritas data, memonitor efisiensi basis data dan mengevaluasi perbedaan teknologi DBMS. Realitas menunjukkan bahwa pemberian kewenangan terhadap seorang khusus sebagai admin basis data sangat menambah efektivitas dari kinerja DBMS.

**2.8.3 Komponen Software DBMS**

Sistem manajemen basis data ialah sekumpulan program yang didesain khusus untuk mendeskripsikan, melindungi, dan mengakses basis data serta mengakhiri keterbatasan pemrosesan *file* tradisional.

Sebuah DBMS bisa membantu suatu organisasi guna memudahkan kemungkinan merespon dan mengganti informasi-informasi yang diperlukan. Sebuah DBMS mempunyai dua komponen *software* utama:

1. SKBD (Sistem Kontrol Basis Data), s*oftware* ini menjadi antarmuka program aplikasi pengguna untuk mengambil data dari basis data*.* Proses memperoleh data seperti baca dan tulis diperintah dari program. Isi dalam *software* tersebut adalah bahasa manipulasi data (*Data Manipulation Language*/DML).
2. SPBD (Sistem Penyimpanan Basis Data), *software* ini memanipulasi *file-file* data yang diperlukan untuk data yang akan dimasukkan dalam basis data. Hal ini untuk menyeimbangkan dan menjaga semua organisasi data dan *link* antara data-data tersebut. Isi sistem penyimpanan basis data ini adalah bahasa pendefinisian data (*Data Definition Language*/DDL). Bahasa ini merupakan kosa kata untuk mendefinisikan basis data yang berisi terminologi pendefinisian *record*, *fields*, *field-field* kunci, dan hubungan antara *record*. SKBD dan SPBD diatas bekerjasama menyediakan semua fungsi DBMS agar efisien dan efektif.

**2.9 DAD (Diagram Arus Data)**

**2.9.1 Pengertian DAD**

Diagram Arus Data (DAD) adalah suatu diagram yang sangat membantu dalam komunikasi dengan pemakaian sistem untuk memahami sistem secara logika. DAD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir (misalnya lewat telepon, surat dan sebagainya) atau dimana data tersebut akan disimpan (misalnya *file* kartu, *hard disk,* tape, diskettedan lain sebagainya). DAD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur, merupakan dokumentasi dari sistem yang baik seperti yang dikutip Hutahaean(2014) dari jogiyanto.

**2.9.2 Komponen DAD**

Sebuah DAD hendaknya tercakup di dalam satu halaman. Hal ini berarti bahwa orang dapat melihat pada diagram tanpa merasa kesulitan dan bila sistem tidak terlalu kompleks maka dapat dilaksanakan, namun bila sistem tersebut kompleks maka perlu dilakukan usaha untuk menguasai sistem dalam beberapa DAD. Adapun simbol-simbol yang digunakan oleh DAD adalah sebagai berikut:

1. **Kesatuan Luar (*enternal entity*) atau batas sistem (*boundary*).**

Setiap sistem pasti mempunyai batas sistem (boundary) yang memisahkan suatu sistem dengan lingkungan luarnya. Sistem akan menerima input dan menghasilkan output kepada lingkungan luarnya. Kesatuan luar (*external entity)* merupakan kesatuan (*entity*) di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.

Suatu kesatuan luar dapat disimbolkan dengan suatu notasi kotak sebagai berikut :

**Gambar 2.5** Notasi kesatuan luar di DAD.

1. **Alur Data (*Data flow* )**

Alur data digunakan untuk perpindahan data atau paket data dari satu bagian ke bagian yang lain. Alur data) di DAD diberi simbol suatu panah. Arus data ini mengalir di antara proses*,* Simpanan data dan kesatuan luar (*external entity*). Arus data sebaiknya di beri nama yang jelas dan mempunyai arti. Nama dari arus data dituliskan disamping garis panahnya.

**Gambar 2.6** Alur data yang mengalir dari kesatuan luar ke proses

Ada 4 konsep tentang alur data:

1. Packets of data (konsep paket dari data)

Apabila ada 2 data atau lebih yang mengalir dari satu sumber yang sama menuju pada tujuan yang sama dan mempunyai hubungan digambarkan dengan satu alur data.

**Gambar2.7** Konsep paket dari data

1. Diverging data flow (konsep alur data menyebar)

Apabila ada sejumlah paket data yang dari sumber yang sama menuju pada tujuan yang berbeda atau paket data yang kompleks dibagi menjadi beberapa elemen data yang dikirim ke tujuan yang berbeda.

Data Flow Diagram

**Gambar 2.8** Konsep alur data menyebar

1. Converging data flow (konsep alur data mengumpul)

Apabila ada beberapa alur data yang berbeda sumber menuju ke tujuan yang sama.

**Gambar 2.9** Konsep arus data mengumpul

1. Sumber dan Tujuan

Arus data harus dihubungkan pada proses, baik dari maupun yang menuju proses.

*Dari proses ke bukan proses Dari bukan proses menuju proses*

*Dari proses ke proses*

**Gambar 2.10** Sumber dan tujuan

1. **Proses (*process)***

Suatu proses adalah kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang masuk ke dalam proses untuk di hasilkan arus data yang akan keluar dari proses. Suatu proses dapat ditunjukkan dengan simbol lingkaran atau dengan simbol empat persegi panjang tegak dengan sudut-sudutnya tumpul.

**Gambar 2.11** Notasi proses di DAD

1. **Simpanan Data (Data store)**

Simpanan data merupakan simpanan dari data yang dapat berupa suatu file atau database di sistem komputer. Simpanan data di DAD dapat disimbolkan dengan sepasang garis horizontal paralel yang tertutup di salah satu ujungnya.



**Gambar 2.12** Simbol dari simpanan data di DAD

**2.9.3 Pedoman Menggambar DAD**

Pedoman-pedoman dalam menggambarkan DAD adalah sebagai berikut:

1. Mengidentifikasi terlebih dahulu semua kesatuan luar yang terlibat di sistem.
2. Mengidentifikasikan semua input dan output dengan kesatuan luar.
3. Mengambarkan terlebih dahulu suatu diagram konteks.

**2.10 Arsitektur *Basis Data***

**2.10.1 Basis Data SQL Server2005**

Karena *software* untuk basis data server yang dipakai dalam pembuatan sistem ini adalah SQL Server2005, maka penyusun membahas sedikit mengenai SQL Server2005 untuk acuan pembuatan *database*.

SQL Server2005 adalah sebuah sistem berasitektur terbuka yang memungkinkan para pengembang program memperluas dan menambahkan fungsi-fungsi ke dalam database tersebut. Contoh Sql Server: objek *server*, *database, table, stored procedure, role, login, user, dan job*.

**2.10.2 Tipe Data**

Tabel merupakan bentuk fisik data yang disusun dalam *database* terdiri atas baris dan kolom. Setiap kolom selalu menyimpan data yang sejenis (setipe).

Tipe data secara umum dapat dikelompokan menjadi data:

1. Nilai numerik adalah semua data yang berupa bilangan yang dapat dimanipulasi dengan menggunakan operasi aritmatika.
2. Nilai string adalah semua data berupa teks (huruf dan tanda baca) dan angka.
3. Nilai tanggal dan jam dalam SQL Server2005 merupakan nilai data khusus yang digunakan untuk untuk menyimpan data tanggal dan waktu.

**2.10.3 Fungsi SQL Server2005**

Fungsi *(function)* merupakan fasilitas yng disediakan oleh database secara umum untuk memanipulasi data yang kompleks, seperti mencari jumlah data, mencari rata-rata nilai yang ada dalam suatu tabel, dan lain-lain, bukan sekedar membaca data dari tabel saja.

**2.11 Borland Delphi 7**

Borland Delphi 7 merupakan suatu bahasa pemragraman yang memberikan berbagai fasilitas pembuatan aplikasi visual seperti *visual basic*. Keunggulan bahasa pemrograman ini terletak pada kemudahan dalam menggunakan kode program, produktivitas, kualitas, pengembangan perangkat lunak, penggunaan file unit ganda untuk pemrograman modular, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur dalam bahasa pemrograman *Object Pascal*. Keunggulan lain dari Delphi adalah dapa digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis Windows. Khusus untuk pemograman database. Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan *programmer* dalam membuat program. Delphi memiliki tampilan khusus yang didukung suatu lingkup kerja komponen Delphi untuk membangun suatu aplikasi dengan menggunakan *Visual Component Library* (VCL).

**BAB III**

**METODE PENELITIAN**

1. **Metode Pengambilan Data**

Data-data yang diperoleh sebagai dasar dalam penyusunan laporan ini, diambil dari berbagai sumber dengan menggunakan beberapa metode. Metode-metode yang digunakan antara lain sebagai berikut:

Metode Pengamatan *(Observasi)*

Metode pengumpulan data yang dilakukan secara sistematis dengan cara melakukan pengamatan langsung terhadap objek yang diteliti

Metode Wawancara *(Interview)*

Metode pengumpulan data yang dilakukan dengan cara mengadakan wawancara pada narasumber yang menjadi responden berkaitan dengan topik masalah yang diambil sehingga data yang didapat lebih akurat.

Studi pustaka *(library research)*

Metode pengumpulan data yang diperoleh dari berbagai buku, literatur dan media lain sebagai referensi penulis dalam menyusun laporan Proyek Tugas Akhir

Metode Analisis

Metode ini dilakukan untuk menganalisis terhadap hasil-hasil observasi dan hasil wawancara untuk mendapatkan kesimpulan akhir dan membuat rencana pengembangan selanjutnya untuk dapat mengatasi permasalahan yang ada.

Metode Desain

Metode ini dilakukan untuk mengetahui langkah apa saja yang harus dibuat.

Metode Konstruksi

Metode ini adalah kelanjutan dari metode desain yang bertujuan untuk menerapkan langkah-langkah yang telah di buat ke dalam bentuk aplikasi komputer.

1. **Analisis Sistem**

Analisis Sistem dilakukan untuk menganalisis terhadap hasil-hasil observasi dan hasil wawancara untuk mendapatkan simpulan akhir dan membuat rencana pengembangan selanjutnya untuk mengatasi permasalahan yang ada. Analisis sistem diperlukan dalam rangka mengkaji aplikasi sistem yang akan dibangun.

Adapun langkah-langkah yang digunakan untuk menganalisis sistem adalah :

1. Pengumpulan data yang diperoleh untuk perancangan sistem.
2. Melakukan pengelompokkan data sesuai dengan permasalahan yang dihadapi oleh sistem aplikasi yang akan dibuat.
3. Mendeskripsikan data untuk memperoleh gambaran data dan menemukan bahasa pemrograman yang tepat untuk diterapkan dalam pembuatan program.
4. **Perancangan Sistem**

Tahap perancangan sistem dilakukan setelah tahap analisis sistem. Perancangan sistem menggunakan notasi untuk membantu dalam memberikan gambaran atau penjelasan pada pemakai tentang sistem yang akan dibuat, yaitu mengenai aliran data dan informasi mulai dari masukan data sampai dengan keluaran data dari sistem yang dibuat.

1. Perancangan Diagram Alur Data (DAD)

Untuk dapat memahami sistem pengolahan data, maka dapat dibuat suatu diagram arus data yang dapat menggambarkan jalannya sistem yang akan dikembangkan

1. Perancangan Antarmuka

Perancangan antarmuka diperlukan untuk memberikan tampilan yang menarik agar mudah dipahami dan tidak membosankan bagi pengguna. Perancangan tampilan terdiri dari :

1. Rancangan Masukan

Rancangan ini dibuat untuk memasukkan data agar dapat memberikan hasil yang sesuai dengan sistem yang akan dibangun

1. Rancangan Proses

Rancangan ini dilakukan untuk mengelola masukan yang telah diberikan oleh pengguna untuk mendapatkan hasil keluaran

1. Rancangan Keluaran

Hasil akhir dari keluaran yang telah diproses adalah keluaran. Perancangan keluaran ini beruapa laporan hasil penelitian

1. **Implementasi Sistem**

Analisis dan perancangan sistem yang dilakukan menjadi basis bagi proses implementasi sistem. Implementasi sistem merupakan transformasi konsep rancangan menjadi wujud sistem yang utuh dan dapat digunakan. Metode yang digunakan dalam implementasi sistem yaitu konstruksi sistem.

Konstruksi sistem merupakan aktivitas pemrograman atau pengkodean dalam pembuatan sistem dengan menggunakan alat-alat (*tools*) penelitian sehingga terbentuk sistem informasi pendapatan sesuai dengan tujuan penelitian.

1. **Perangkat Pendukung**

Alat-alat (*tools*)yang digunakan dalam penelitian ini yaitu:

Perangkat Keras

Syarat perangkat keras yang digunakan memiliki kemampuan *processing* yang memadai dalam menjalankan program aplikasi yang digunakan.

Sistem komputer yang digunakan pada tahap pengembangan adalah :

* Intel Core i5-2450M Cpu 2.50 GHz
* 2GB DDR2 Memory
* Kapasitas Harddiks 320 GB

Perangkat Lunak

Adapun perangkat lunak yang digunakan adalah :

1. Borland Delphi 7.0

Merupakan program untuk membuat aplikasi–aplikasi berbasis windows yang dijalankan pada sistem operasi Microsoft Windows, dan juga mendukung dalam pembuatan aplikasi basis data yang dapat digunakan untuk berinteraksi dengan informasi yang disimpan dalam basis data.

SQL Server 2008

SQL Server 2008 Merupakan perangkat lunak yang berfungsi sebagai pengolah basis data MySQL berbasis GUI (Graphical User Interface). Kemudahan dalam pengelolaan basisdata MySQL menjadi hal yang penting dalam pembangunan sistem. Adanya fasilitas diagram yang berfungsi menentukan relasi antartabel berbasis grafik memudahkan kita melakukan penambahan kata kunci dalam suatu tabel dengan kesalahan yang minimal.

**BAB IV**

**ANALISIS DAN PERANCANGAN**

1. **Analisis Sistem**

Sistem Informasi Penjualan dan Manajemen Toko dibangun untuk mempermudah proses pengolahan data transaksi yang diperoleh toko Cahaya Phonecell. Analisis sistem pada instansi, sebagai berikut :

1. Toko Cahaya Phonecell memperoleh pendapatan dari hasil penjualan telepon seluler, penjualan pulsa dan penjualan aksesoris.
2. Pendapatan hasil dari penjualan oleh Pegawai toko dilaporkan pertiap hari setelah jam kerja selesai kepada bagian Admin, yang kemudian diproses secara manual dengan cara mencatatnya pada buku laporan.
3. Seluruh hasil pendapatan yang tercatat pada buku laporan tersebut kemudian oleh Pegawai bagian Admin dilaporkan kepada pemilik Toko pertiap minggu.

Permasalahan yang timbul sebagian besar terjadi karena semua proses pengolahan datanya masih dilakukan secara manual yang kemudian diarsipkan kedalam buku, masalah yang sering timbul :

1. Pegawai sering kali melakukan kesalahan dalam pencatatan transaksi sehingga kadang harus dicatat ulang.
2. Penyimpanan data transaksi pada berkas-berkas yang dibukukan dapat menghambat proses pelaporan kepada pihak pemilik Toko, karena harus mencari berkas-berkas yang harus dilaporkan tersebut diantara berkas-berkas lain, dan pastinya membutuhkan tenaga dan waktu yang lama.
3. Berkas-berkas yang sudah dilaporkan kepada pemilik Toko nantinya akan terbengkalai karena tidak digunakan lagi dan tidak tersimpan dengan baik, karena petugas harus menyiapkan berkas-berkas baru yang nantinya akan digunakan untuk menyimpan data-data transaksi yang baru.

Permasalahan-permasalahan yang timbul tesebut diharapkan dapat teratasi dengan adanya sistem informasi secara terkomputerisasi, sehingga diharapkan dapat membantu kinerja para Pegawai diantaranya :

1. Dapat meningkatkan sistem kerja Pegawai menjadi cepat, lebih baik dan meningkatkan efektifitas serta efisiensi waktu maupun tenaga. Serta mempermudah dan mempercepat proses penyimpanan data transaksi, sehingga Pegawaitidak membutuhkan waktu ber jam-jam dalam melakukan penyimpanan data pendapatan.
2. Dengan adanya sistem secara terkomputerisasi dapat mempercepat proses pencarian data jika dibutuhkan sewaktu-waktu, sehingga proses pelaporan pun tidak terhambat dan tidak membutuhkan waktu yang lama dalam pencarian data.
3. Tersimpanya data dengan baik, sehingga jika Pegawai ingin meyimpan data yang baru dapat disimpan kedalam sistem yang sama tanpa perlu menghapus data yang sudah tidak terpakai.
4. **Perancangan Sistem**

Sebelum pembuatan sistem secara keseluruhan, terlebih dahulu dibuat rancangan untuk mempermudah dalam pengerjaannya. Pada tahap perancangan sistem bertujuan untuk memberikan gambaran tentang sistem yang akan dibuat, yaitu mengenai aliran data, proses dari sistem mulai dari masukkan serta keluaran yang akan dibuat.

**4.2.1 Diagram Konteks**

Diagram konteks merupakan bentuk global dari proses pengolahan data. Secara umum proses pengolahan data yang digambarkan pada diagram kontek hanya sebatas input, proses dan output data. Bentuk diagram kontek Sistem Informasi Penjualan dan Manajemen Tokoseperti pada gambar 4.1.

****

**Gambar 4.1** Diagram Konteks

**4.2.2 Diagram Berjenjang**

Dari diagram konteks maka dapat digambarkan diagram yang lebih terinci lagi atau diagram berjenjang. Gambar diagram berjenjang Sistem Informasi Penjualan dan Manajemen Toko :



**Gambar 4.2** Diagram Berjenjang

**4.2.3 Diagram Alir Data (DAD)**

Untuk dapat memahami sistem secara logika, maka dibuat suatu diagram alir data yang dapat menggambarkan jalanya sistem yang akan dikembangkan.Diagram alir data ini juga menampilkan kegiatan sistem lengkap dengan komponen-komponen yang menunjukkan secara tegas file-file yang dipakai, unsur sumber atau tujuan data, serta aliran data dari suatu proses ke proses lainnya.

**4.2.3.1 DAD Level 1**

****

**Gambar 4.3** DAD level 1

**4.2.3.2 DAD Level 2 Proses 1**

**Gambar 4.4** DAD level 2 Proses 1

**4.2.3.3 DAD Level 2 Proses 2**

****

**Gambar 4.5** DAD level 2 Proses 2

**4.2.3.4 DAD Level 2 Proses 3**

****

**Gambar 4.6** DAD level 2 Proses 3

1. **Gambaran Sistem**

Sistem Informasi Penjualan dan Manajemen Toko dapat direferensikan dalam Entity Relationship Diagram (ERD) untuk menggambarkan sistem yang ada.

**Indentifikasi Entitas**

Berikut ini daftar entitas yang akan digunakan :

* + Entitas Pegawai
  + Entitas Barang
  + Entitas Penjualan
  + Entitas Pembelian
  + Entitas Suplier
  + Entitas Detail Pembelian
  + Entitas Detail Penjualan
  + Entitas Kategori
  + Entitas Satuan
  + Entitas Retur
  + Entitas Penggajian
  + Entitas Tunjangan
  + Entitas Gaji pokok
  + Entitas Login
  + Entitas Jual Pulsa
  + Entitas Pulsa

1. **ERD (Entity Relationship Diagram)**



**Gambar 4.7** Relasi Antar Entitas

1. **Perancangan Tabel**
2. Tabel Pegawai

Tabel ini berisikan tentang data diri karyawan

Nama tabel : tb\_karyawan

Field Kunci : nip

**Tabel 3.1** Tabel Pegawai

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | Nip | Varchar(30) |
|  | Nama | Varchar(50) |
|  | tempat\_lhr | Varchar(30) |
|  | tgl\_lhr | Datetime |
|  | jns\_kel | Varchar(10) |
|  | Pnd\_akhir | Varchar(30) |
|  | Alamat | Varchar(80) |
|  | Tgl\_masuk | Datetime |

1. Tabel Barang

Tabel ini berisikan tentang data barang

Nama tabel : barang

Field Kunci : kd\_brg

**Tabel 3.2** Tabel Barang

|  |  |  |
| --- | --- | --- |
| Fk | Nama kolom | Tipe data |
|  | kd\_brg | Varchar(10) |
|  | nama\_brg | Varchar(100) |
|  | kd\_satuan | Varchar(20) |
|  | kd\_kategori | Varchar(20) |
|  | stok | Int |
|  | harga\_beli | Money |
|  | harga\_jual | Money |

1. Tabel Penjualan

Tabel ini berisikan tentang data barang yang dijual

Nama tabel : penjualan

Field Kunci : nota\_jual

**Tabel 3.3** Tabel Penjualan

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | Nota\_jual | Varchar(10) |
|  | id | Varchar(20) |
|  | Tgl\_jual | Datetime |
|  | Total\_harga | Money |

1. Tabel Pembelian

Tabel ini berisikan tentang data barang yang dibeli toko

Nama tabel : Pembelian

Field Kunci : nota\_beli

**Tabel 3.4** Tabel Pembelian

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | nota\_beli | Varchar(10) |
|  | id | Varchar(20) |
|  | Kd\_sup | Varchar(10) |
|  | tgl\_beli | Datetime |
|  | total\_beli | Money |

1. Tabel Suplier

Tabel ini berisikan tentang data yang menyuplai barang

Nama tabel : supplier

Field Kunci : kd\_sup

**Tabel 3.5** Tabel Supplier

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | kd\_sup | Varchar(10) |
|  | Nama | Varchar(50) |
|  | Telp | Varchar(20) |
|  | Alamat | Varchar(100) |

1. Tabel Detail Pembelian

Tabel ini berisikan tentang data transaksi barang yang dibeli toko

Nama tabel : det\_beli

Field Kunci : kd\_det\_beli

**Tabel 3.6** Tabel Detail Pembelian

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | Kd\_det\_beli | Int |
|  | Nota\_beli | Varchar(10) |
|  | Kd\_brg | Varchar(10) |
|  | Harga | Money |
|  | Jumlah | Int |
|  | Total | Money |

1. Tabel Detail Penjualan

Tabel ini berisikan tentang data transaksi penjualan

Nama tabel : det\_jual

Field Kunci : kd\_det\_jual

**Tabel 3.7** Tabel Detail Penjualan

|  |  |  |
| --- | --- | --- |
|  | Nama kolom | Tipe data |
| Fk | kd\_det\_jual | Int |
|  | nota\_jual | Varchar(10) |
|  | kd\_brg | Varchar(10) |
|  | hrg\_jual | Money |
|  | jumlah\_brg | Int |
|  | subtotal | Money |

1. Tabel Kategori

Tabel ini berisikan tentang data kategori barang

Nama tabel : kategori

Field Kunci : kd\_kategori

**Tabel 3.8** Tabel Kategori

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | Kd\_kategori | Varchar(20) |
|  | Kategori | Varchar(30) |

1. Tabel Satuan

Tabel ini berisikan tentang data satuan ba rang

Nama tabel : satuan

Field Kunci : kd\_satuan

**Tabel 3.9** Tabel Satuan

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | kd\_satuan | Varchar(20) |
|  | satuan | Varchar(30) |

1. Tabel Retur

Tabel ini berisikan tentang data barang yang diretur

Nama tabel : retur

Field Kunci : kd\_retur

**Tabel 3.10** Tabel Retur

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | Kd\_retur | Int |
|  | Nota\_beli | Varchar(10) |
|  | Id | Varchar(20) |
|  | Tgl\_retur | Datetime |
|  | Kd\_brg | Varchar(10) |
|  | Jml\_beli | Int |
|  | Jml\_retur | Int |
|  | Status | Varchar(10) |

1. Tabel Penggajian

Tabel ini berisikan tentang data transaksi penggajian pegawai

Nama tabel : penggajian

Field Kunci : no\_slip

**Tabel 3.11** Tabel Penggajian

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | No\_slip | Varchar(10) |
|  | Tgl\_gaji | Datetime |
|  | Nip | Varchar(30) |
|  | Kd\_gapok | Int |
|  | Kd\_tunjangan | Int |
|  | Tot\_gaji | Money |
|  | Status | Varchar(15) |

1. Tabel Tunjangan

Tabel ini berisikan tentang data tunjangan gaji pegawai

Nama tabel : tunjangan

Field Kunci : kd\_tunjangan

**Tabel 3.12**Tabel Tunjangan

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | Kd\_tunjangan | Int |
|  | nama\_tunj | Varchar(30) |
|  | besar\_tunj | Money |

1. Tabel Gaji Pokok

Tabel ini berisikan tentang data gaji pokok pegawai

Nama tabel : gapok

Field Kunci : kd\_gapok

**Tabel 3.13** Tabel Gaji Pokok

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | Kd\_gapok | Int |
|  | gapok | Money |

1. Tabel Login

Tabel ini berisikan tentang data pengguna dan hak akses

Nama tabel : login

Field Kunci : \_login

**Tabel 3.14** Tabel Login

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | Id | Varchar(20) |
|  | Nip | Varchar(30) |
|  | pass | Varchar(20) |
|  | Akses | Varchar(20) |

1. Tabel Pulsa

Tabel ini berisikan tentang data pulsa dan nominal pulsa

Nama tabel : pulsa

Field Kunci : id\_pulsa

**Tabel 3.15** Tabel Pulsa

|  |  |  |
| --- | --- | --- |
|  | Nama Kolom | Tipe Data |
| Fk | id\_pulsa | Int |
|  | Niai | Money |
|  | Margin | Money |

1. Tabel Jual Pulsa

Tabel ini berisikan tentang data transaksi penjualan pulsa dan nominal pulsa

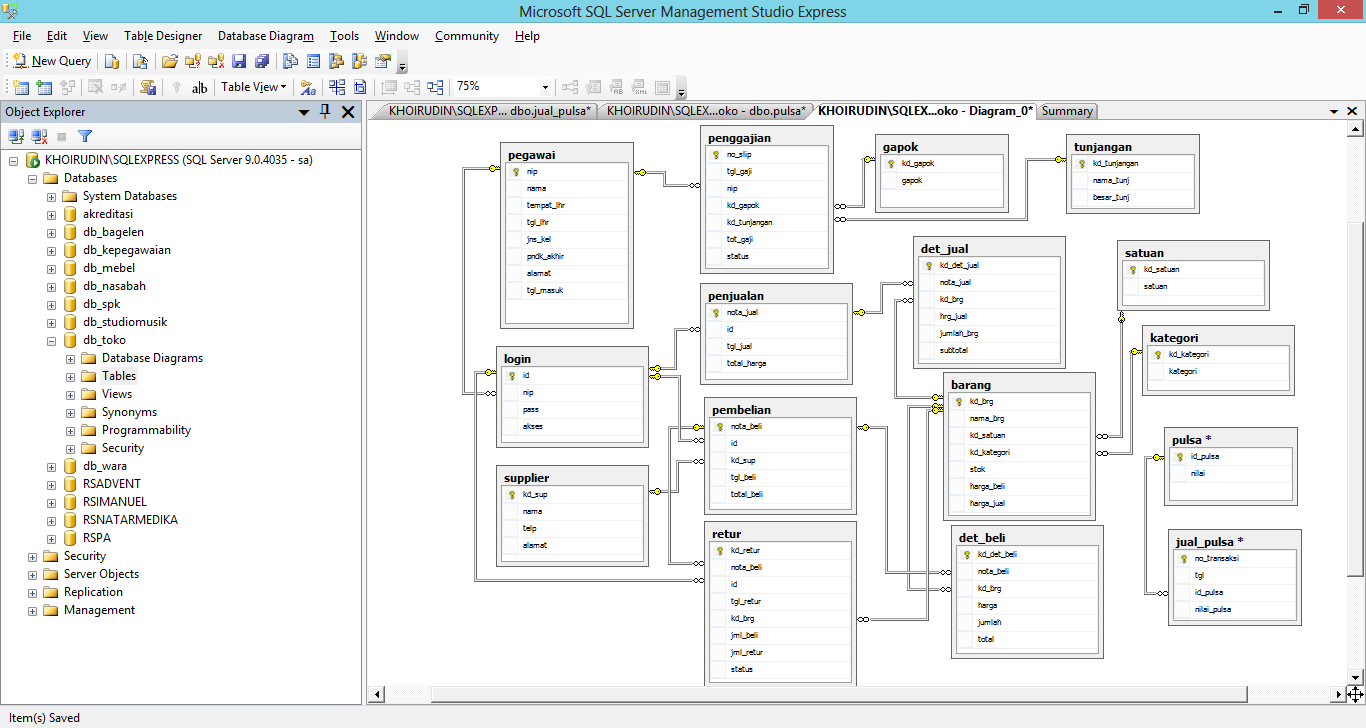
Nama tabel : jual\_pulsa

Field Kunci : no\_transaksi

**Tabel 3.16** Tabel Jual Pulsa

|  |  |  |
| --- | --- | --- |
| Fk | Nama Kolom | Tipe Data |
|  | no\_transaksi | Int |
|  | Tgl | Datetime |
|  | id\_pulsa | Int |
|  | nilai\_pulsa | Money |
|  | Total | Money |

1. **Relasi Antartabel**

****

**Gambar 4.8** Relasi Antartabel

**IV.3 Perancangan Antarmuka**

Perancangan antarmuka diperlukan sebagai sarana masukan dan keluaran data. Perancangan tampilan terdiri dari beberapa tampilan yaitu aturan, pemroses data dan bantuan. Untuk keluar dari tampilan awal atau akan menuju tampilan lainya ada juga menu keluar, berikut adalah rancangan antarmuka Sistem Penjualan dan Manajemen Toko pada Toko Cahaya Phonecell :

1. **Menu Utama**

Pada form menu utama disediakan fasilitas menu file yang berisikan login, logout, keluar. Menu master data yang berisikan menu data barang, data pulsa, data supplier, data pegawai, setting dan data petugas. Menu transaksi yang berisikan transaksi pembelian, transaksi penjualan, transaksi retur dan transaksi penggajian. Menu laporan yang terdiri dari menu laporan pembelian, laporan penjualan, laporan penggajian, laporan pendapatan bagihasil koperasi, laporan grafik pendapatan penyewaan, serta laporan grafik penjualan.

****

**Gambar 4.9** Form Menu Utama

1. **Form Pengaturan Database**

Form Pengaturan Databse digunakan untuk menghubungkan apliksi program dengan database server.



**Gambar 4.10** Form Pengaturan Database

1. **Form Login**

Form Login digunakan untuk masuk kedalam sistem, dengan memasukan username dan password, tombol login digunakan untuk login, dan tombol batal digunakan untuk membatalkan login kedalam sistem. Dapat dilihat pada Gambar Desain Form login dibawah ini:



**Gambar 4.11** Form Menu Login

1. **Form Data Barang**

Form Data Barang berfungsi untuk melakukan pendataan barang . Data yang di inputkan meliputi Kode Barang, Nama Barang, Kategori, Satuan, jumlah stok bila ada, harga beli, dan harga jual . Pada form ini disediakan fasilitas simpan, ubah, dan hapus



**Gambar 4.12** Form Data Barang

1. **Form Data Pulsa**

Form data pulsa berfungsi untuk melakukan pendataan pulsa yang dijual. Data yang di masukan meliputi Keterangan Provider pulsa, Jumlah Saldo pulsa, dan margin atau keuntungan yang diinginkan.



**Gambar 4.13** Form Data Pulsa

1. **Form Data Supplier**

Form data pulsa berfungsi untuk melakukan pendataan supplier yang menyuplai barang. Data yang di masukan meliputi kode supplier, nama, alamat, dan nomer telepon.



**Gambar 4.14** Form Data Supplier

1. **Form Data Pegawai**

Form data Pegawai berfungsi untuk melakukan pendataan Pegawai. Data yang di masukan yaitu meliputi Nip, Nama, Tempat lahir, Tanggal lahir, Jenis kelamin, Alamat, Tanggal masuk bekerja.



**Gambar 4.15** Form Data Pegawai

1. **Form Data Petugas Toko**

Form data petugas berfungsi untuk melakukan pendataan pegawai yang bertugas. Data yang dimasukkan meliputi nama pegawai, username, password, dan hak akses.



**Gambar 4.16** Form Menu Data Petugas Toko

1. **Form Data Kategori Barang**

Form setting data kategori barang berfungsi untuk melakukan pendataan jenis kategori barang. Data yang masukan jenis kategori.



**Gambar 4.17** Form Kategori Barang

1. **Form Data Satuan Barang**

Form setting data satuan barang berfungsi untuk melakukan pendataan jenis satuan barang. Data yang dimasukan jenis satuan.



**Gambar 4.18** Form Satuan Barang

1. **Form Stok Minimal**

Berfungsi untuk melakukan pengaturan terhadap batas jumlah stok minimal barang yang akan ditampilkan.



**Gambar 4.19** Form Stok Minimal

1. **Form Gaji Pokok**

Form gaji pokok berfungsi untuk melakukan pengaturan gaji pokok pegawai, data yang dimasukkan meliputi masa kerja pegawai.



**Gambar 4.20** Form Data Gaji Pokok

1. **Form Tunjangan Pegawai**

Form tunjangan pegawai berfungsi mengatur data tunjangan pegawai,meliputi nama tunjangan, dan besar tunjangan.



**Gambar 4.21** Form Tunjangan Gaji Pegawai

1. **Form Pembelian**

Form pembelian berfungsi untuk memproses transaksi terhadap pembelian barang, data-data yang dimasukkan meliputi data petugas, data supplier, dan data barang yangakan dibeli, untuk nota beli, dan tanggal transaksi dimasukkan secara otomatis oleh sistem.



**Gambar 4.22** Form Pembelian

1. **Form Penjualan**

Berfungsi untuk memproses transaksi penjualan barang, data-data yang dimasukkan meliputi data petugas, dan data barang yang akan dijual, untuk nota jual, dan tanggal transaksi dimasukkan secara otomatis oleh sistem.



**Gambar 4.23** Form Penjualan

1. **Form Retur Barang**

Form retur berfungsi untuk memproses data barang yang akan diajukan untuk diretur kepada supplier.



**Gambar 4.24** Form Retur Barang

1. **Form Penggajian**

Berfungsi untuk memproses transaksi penggajian pegawai, data-data yang dimasukkan meliputi data pegawai, data gaji dan data tunjangan gaji.



**Gambar 4.25** Form Penggajian

1. **Form Penjualan Pulsa**

Form penjualan pulsa berfungsi untuk melakukan proses transaksi penjualan pulsa kepada konsumen.



**Gambar 4.26** Form Penjualan Pulsa

1. **Form Laporan Data Barang**

Form laporan data barang merupakan form tampilan yang berisikan informasi tentang data barang, mulai dari kode, nama, dan jenis barang hingga informasi data harga jual dan beli barang.



**Gambar 4.27** Form Laporan Data Barang

1. **Form Laporan Data Supplier**

Form laporan data supplier merupakan form tampilan yang berisikan informasi tentang data supplier toko.



**Gambar 4.28** Form Laporan Data Supplier

1. **Form Laporan Pembelian**

Form laporan pembelian merupakan form tampilan yang berisi informasi tentang transaksi pembelian barang, meliputi informasi tanggal transaksi pembelian hingga total harga beli barang.



**Gambar 4.29** Form Laporan Pembelian

1. **Form Laporan Penjualan**

Form laporan penjualan merupakan form tampilan yang berisikan informasi tentang data barang, mulai dari kode, nama, dan jenis barang hingga informasi data harga jual dan beli barang.



**Gambar 4.30** Form Laporan Penjualan

1. **Form Laporan Penggajian**

Form laporan penggajian merupakan form tampilan yang berisikan informasi tentang data penggajian pegawai, mulai dari nip, nama, gaji pokok, tunjangan dan total gaji pegawai.



**Gambar 4.31** Form Laporan Penggajian

1. **Form Laporan Penjualan pulsa**

Form laporan penjualan pulsa merupakan form tampilan yang berisikan informasi tentang data transaksi penjualan pulsa.



**Gambar 4.32** Form Laporan Penjualan Pulsa

1. **Form Laporan Barang Retur**

Form laporan penjualan pulsa merupakan form tampilan yang berisikan informasi tentang data barang yang diretur.



**Gambar 4.33** Form Laporan Barang Retur

**BAB V**

**IMPLEMENTASI SISTEM**

**5.1 Cara Kerja Sistem**

Implemestasi dari perancangan pada penelitian ini adalah sebuah aplikasi penjualan dan manajemen toko berbasis client server yang menerapkan metode pemrograman bahasa pascal.

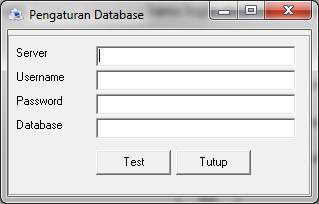
Untuk dapat mengakses aplikasi penjualan dan manajemen toko ini pengguna harus terdaftar sebagai user yang mempunyai hak akses tersendiri. Setelah proses registrasi akunmaka pengguna dapat melakukan login pada aplikasi.

Tampilan aplikasi program akan terlihat berbeda menurut hak akses pengguna. Bagian pertama adalah login untuk petugas transaksi yang berisi form pembelian, penjualan barang dan pulsa. Bagian kedua adalah login untuk petugas administrator yang bisa mengakses keseluruhan form, termasuk laporan-laporan transaksi dan login admin ini bisa juga untuk menambahkan pengguna baru.

**5.2 Pembahasan Unjuk Kerja Sistem**

**5.2.1 Form Pengaturan Database**

Pada saat aplikasi penjualan barang diakses pertama muncul adalah form pengaturan database, karena pengguna harus terhubung dengan database sistem terlih dahulu.

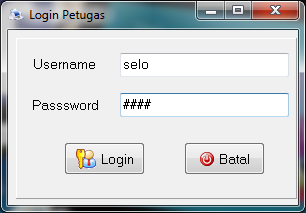
  
**Gambar 5.1** Tampilan form pengaturan database

**Potongan syntax program pengaturan database**

|  |
| --- |
| PathServer:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'conf.cfg');  db:=txtdb.Text;  serv:=txtser.Text;  user:=txtuser.Text;  pass:=txtpass.Text;  conSet.Connected:=False;  conSet.ConnectionString:='Provider=SQLOLEDB.1;Persist Security Info=False;User ID='+user+';password='+pass+';Initial Catalog='+db+';Data Source='+serv;  conSet.Connected:=True;  PathServer.WriteString('confdb','db',db);  PathServer.WriteString('confdb','server',serv);  PathServer.WriteString('confdb','user',user);  PathServer.WriteString('confdb','pass',pass);  PathServer.Free;  Application.MessageBox('Koneksi Kedatabase Sukses. Program Harus Direstart !','Informasi',MB\_OK+MB\_ICONINFORMATION);  Application.Terminate;  except  Application.MessageBox('Koneksi database Gagal','Kesalahan',MB\_OK+MB\_ICONERROR); |

**5.2.2 Form Login**

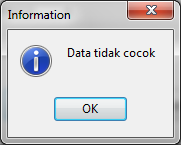
Pada saat aplikasi Toko diakses pertama muncul adalah form login, karena pengguna harus login dulu sebelum mengakses form utama.

  
**Gambar 5.2** Tampilan form login

**Potongan syntax program login**

|  |
| --- |
| with dm.Qlogin do  begin  close;  SQL.Clear;  SQL.Add('select \* from login where id='+QuotedStr(Edid.Text)+ 'and pass='+QuotedStr(Edpass.Text));  ExecSQL;  Active:=true;  end;  if dm.Qlogin.RecordCount=0 then  begin  MessageDlg('Data tidak cocok',mtInformation,[mbok],0); |

Proses validasi login terdiri dari dua proses. Yang pertama proses validasi elemen masukan yang tidak sesuai. Jika terjadi kesalahan maka akan tampil pesan informasi berikut :

  
**Gambar 5.3** Pesan validasi gagal

Yang kedua adalah status pengguna yang aktif, maksudnya status pengguna sebagai login administrator, atau sebagai login petugas.

**5.2.3 Form Utama**

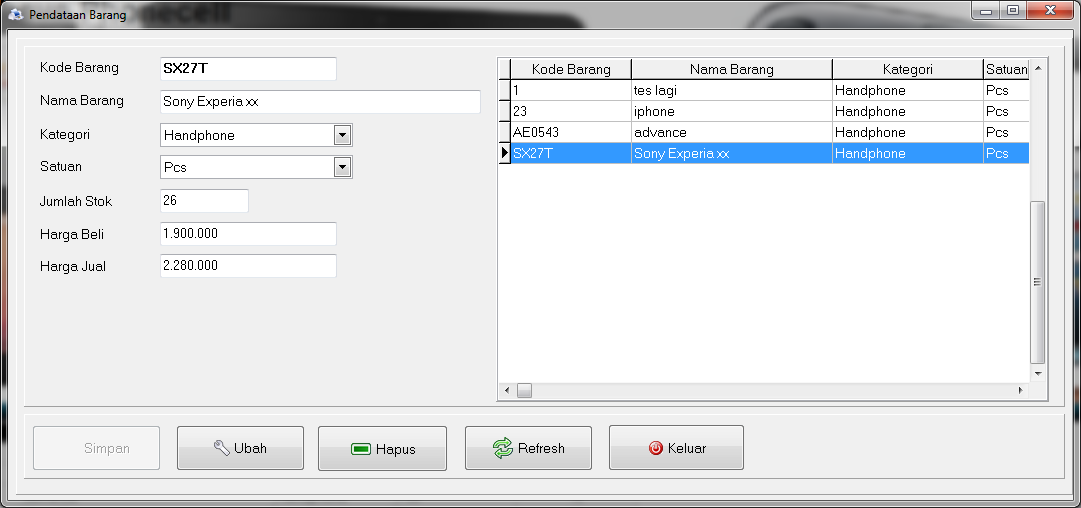
Halaman form utama adalah halaman yang telah sukses mengakses login setiap pengguna. Halaman form utama ini dibagi menjadi beberapa proses, untuk setiap menu pengguna yang memiliki hak akses sebagai petugas, dan hak akses sebagai administrator.



**Gambar 5.4** Tampilan form utama

**5.2.4 Form Data Barang**

Pada form data barang pengguna dapat menambahkan data barang baru dengan stok yang sesuai dengan ketentuan, menghapus data barang, mengubah data barang yang sudah ada, serta menambahkan atau merubah harga jual.



**Gambar 5.5** Tampilan form data barang

Pada saat tombol ubah ditekan oleh pengguna, maka proses perubahan data barang akan berjalan, berganti dengan data baru yang telah dimasukan.

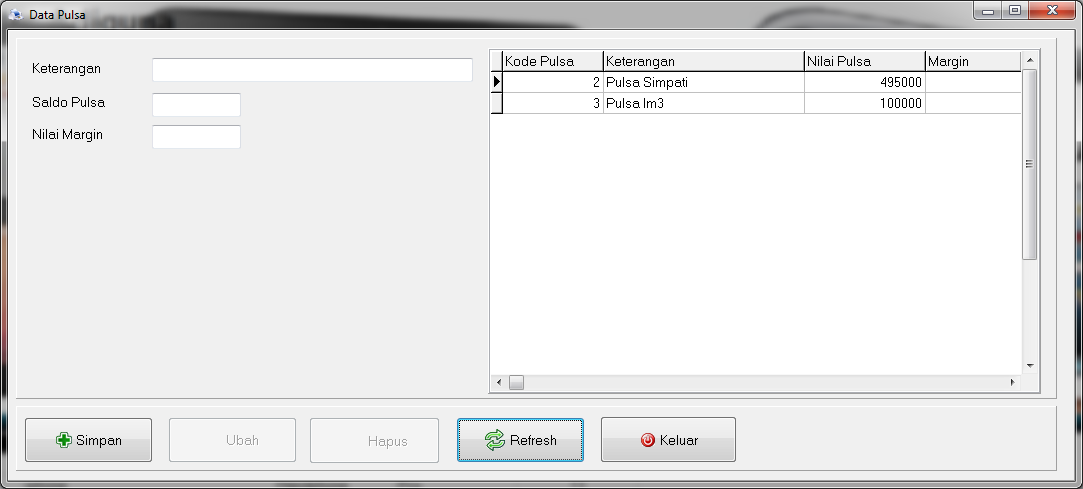
Untuk proses ubah data barang terdapat validasi potongan program berikut :

**Potongan program ubah data barang**

|  |
| --- |
| with dm.Qumum do  begin  close;  SQL.Clear;  sql.Add('update barang set');  sql.Add('nama\_brg = '+quotedstr(ednama\_brg.Text)+',');  sql.Add('kd\_satuan = '+quotedstr(dblkcbbsatuan.KeyValue)+',');  sql.Add('kd\_kategori = '+quotedstr(dblok\_kat.KeyValue)+',');  sql.Add('harga\_beli = '+CurrToStr(a)+',');  sql.Add('harga\_jual = '+CurrToStr(b)+',');  sql.Add('stok = '+(edstok.Text));  SQL.Add('where kd\_brg ='+quotedstr(edkd\_brg.Text));  execsql;  end;  dm.Koneksi.CommitTrans;  ShowMessage('Data Berhasil Diubah'); |

**5.2.5 Form Data Pulsa**

Pada form data pulsa pengguna dapat menambahkan data pulsa baru dengan saldo yang sesuai dengan ketentuan, mengubah data pulsa, dan menentukan nilai margin tau selisih keuntungan penjualan pulsa.

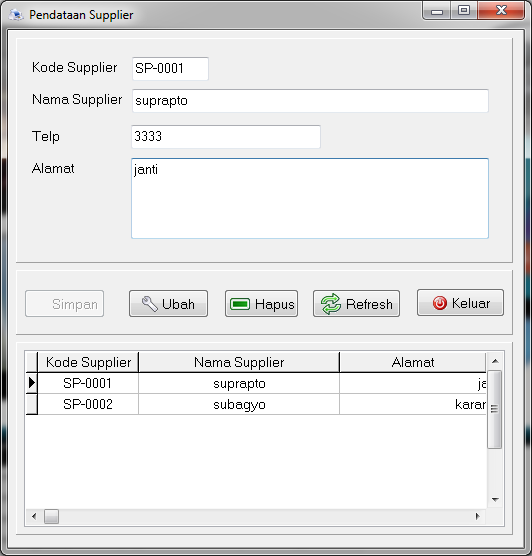
****

**Gambar 5.6** Tampilanform data pulsa

**Potongan syntax program simpan data pulsa**

|  |
| --- |
| with dm.qrypulsa do  begin  sql.Clear;  SQL.Add('select \* from pulsa where ket='+quotedstr(edtket.text)+'');  open;  end;  if dm.qrypulsa.Recordset.RecordCount > 0 then  ShowMessage('Data sudah ada silahkan isi yang lain!!')  else  begin  try dm.Koneksi.BeginTrans;  with dm.Qumum do  begin  close;  SQL.Clear;  sql.Add('insert into pulsa values('  +quotedstr(edtket.Text)+','  +(edtnilai.Text)+','  +(edtmargin.Text)+')'); |

**5.2.6 Form Data Supplier**

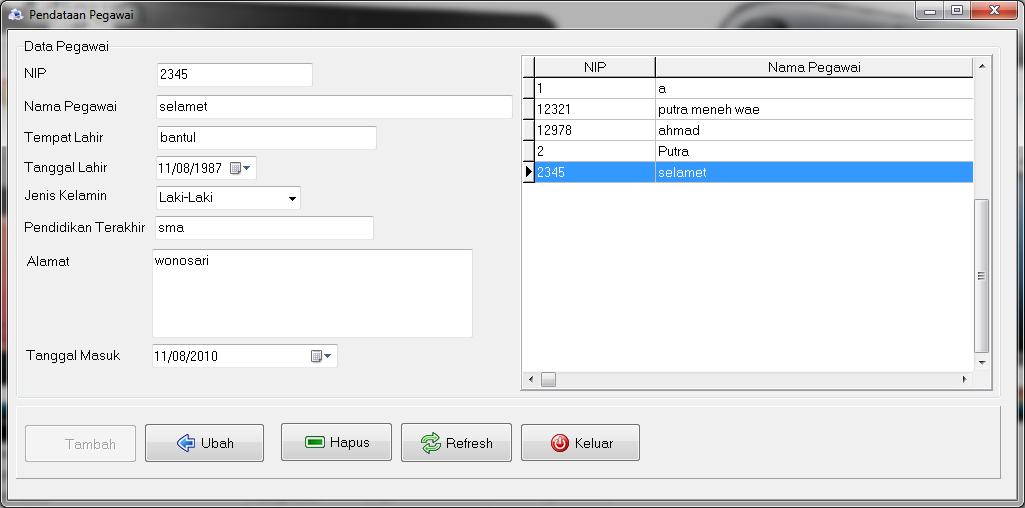
**  
Gambar 5.7** Tampilanform data supplier

Pada tampilan form data supplier ini pengguna dapat melihat daftar supplier, menambahkan, merubah, serta menghapus data supplier.

**Potongan syntax program data supplier**

|  |
| --- |
| with dm.Qsupplier do  begin  sql.Clear;  SQL.Add('select \* from supplier where kd\_sup='+quotedstr(Edkd\_sup.text)+'');  open;  end;  if dm.Qsupplier.Recordset.RecordCount > 0 then  ShowMessage('Data sudah ada silahkan isi yang lain!!')  else  begin  try dm.Koneksi.BeginTrans;  with dm.Qumum do  begin  close;  SQL.Clear;  sql.Add('insert into supplier values('+quotedstr(Edkd\_sup.Text)+','  +quotedstr(Ednama\_sup.Text)+','  +quotedstr(edtelp.Text)+','  +quotedstr(Mealamat.Text)+')');  execsql;  end;  dm.Koneksi.CommitTrans;  ShowMessage('Data Berhasil Disimpan'); |

**5.2.7 Form Data Pegawai**

**Gambar 5.8** TampilanForm Data Pegawai

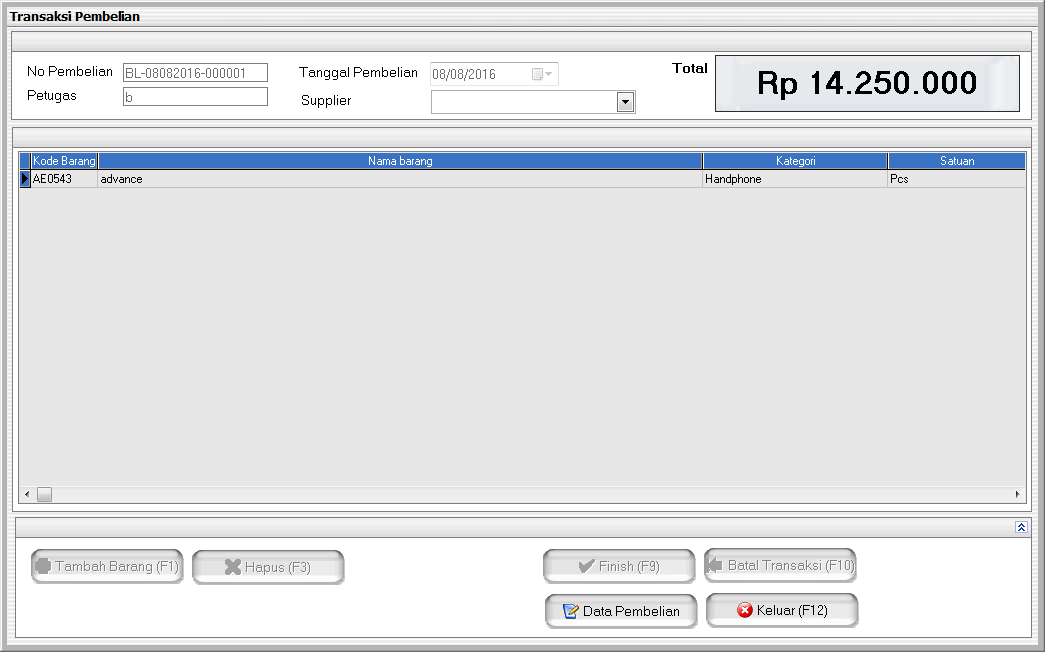
Form data pegawai merupakan form yang di gunakan untuk mengelola atau melakukan pendataan kepegawaian toko.

**Potongan syntax program tambah pegawai**

|  |
| --- |
| if not (key in [#8,'0'..'9']) then  Key:=#0;  end;  procedure TFpegawai.btnsimpanClick(Sender: TObject);  var  foto : TMemoryStream;  begin  if (ednip.Text='') or (ednama\_peg.Text='') or (edtmp\_lhr.Text='') or  (edpndk\_akhir.Text='') or  (cbjns\_kel.Text='') or (mealamat.Text='') then  ShowMessage('Data Kurang Lengkap!') else  begin  with dm.Qpegawai do  begin  sql.Clear;  SQL.Add('select \* from pegawai where nip='+quotedstr(ednip.text)+'');  open;  end;  if dm.Qpegawai.Recordset.RecordCount > 0 then  ShowMessage('Data sudah ada silahkan isi yang lain!!')  else  begin  try dm.Koneksi.BeginTrans;  with dm.Qumum do  begin  close;  SQL.Clear;  sql.Add('insert into pegawai values('+quotedstr(ednip.Text)+','  +quotedstr(ednama\_peg.Text)+','  +quotedstr(edtmp\_lhr.Text)+','  +quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_lhr.Date))+','  +quotedstr(cbjns\_kel.Text)+','  +quotedstr(edpndk\_akhir.Text)+','  +quotedstr(mealamat.Text)+','  +quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_masuk.Date))+')'); execsql;  end;  dm.Koneksi.CommitTrans;  ShowMessage('Data Pegawai Berhasil Disimpan'); |

**5.2.8 Form Pembelian**

Pada form pembelian pengguna dapat menambahkan data transaksi pembelian baru yang sesuai dengan transaksi yang telah dilakukan,pengguna juga dapat mengubah atau menghapus data pembelian.



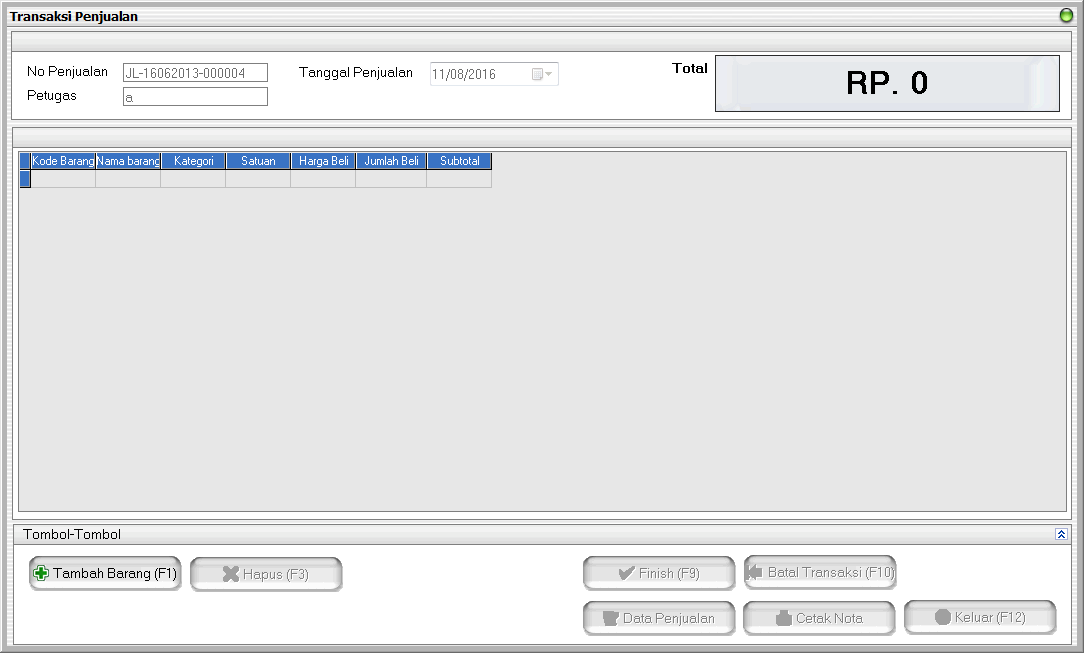
**Gambar 5.8** TampilanForm Pembelian

**Potongan syntax program tambah barang pembelian**

|  |
| --- |
| with dm.Q1 do  begin  close;  sql.Clear;  sql.Add('select stok from barang where kd\_brg='+QuotedStr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));  open;  b:=FieldValues['stok'];  end;  with dm.Q2 do  begin  close;  sql.Clear;  sql.Add('select jml\_beli from det\_beli where kd\_brg='+QuotedStr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));  open;  end;  a:=dm.Q2.FieldValues['jml\_beli'];  c:=b-a;  try dm.Koneksi.BeginTrans;  with dm.Qumum do  begin  close;  SQL.Clear;  sql.Add('delete from det\_beli');  SQL.Add('where nota\_beli ='+quotedstr(edno\_beli.Text)+'and kd\_brg='+QuotedStr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));  execsql;  SQL.Clear;  sql.Add('update barang set');  sql.Add('stok = '+(CurrToStr(c)));  SQL.Add('where kd\_brg ='+quotedstr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));  execsql;  end;  dm.Koneksi.CommitTrans;  ShowMessage('Data Berhasil Dihapus');  except dm.Koneksi.RollbackTrans  end;  dm.Qdet\_beli.Close;  dm.Qdet\_beli.Open;  with dm.Qumum do  begin  Close;  SQL.Clear;  SQL.Add('Select sum(subtotal) as total from det\_beli where nota\_beli ='+QuotedStr(Fpembelian.edno\_beli.Text));  Open;  if dm.Qdet\_beli.RecordCount=0 then  total.Caption:='0' else  begin  total.Caption:=FieldValues['total'];  total.Caption:=FormatCurr('Rp. ##,###,###',FieldValues['total']);  end;end;  btnhapus.Enabled:=false;  with dm.Qdet\_beli do  begin  close;  sql.Clear;  sql.Add('select det\_beli.nota\_beli,det\_beli.jumlah,det\_beli.total,barang.harga\_jual '+  ' ,barang.kd\_brg,barang.nama\_brg,kategori.kategori,det\_beli.harga,satuan.satuan from '+  ' det\_beli,barang,kategori,satuan '+  ' where barang.kd\_brg=det\_beli.kd\_brg and kategori.kd\_kategori=barang.kd\_kategori and satuan.kd\_satuan=barang.kd\_satuan '+  ' and det\_beli.nota\_beli ='+QuotedStr(edno\_beli.Text));  open;  end; |

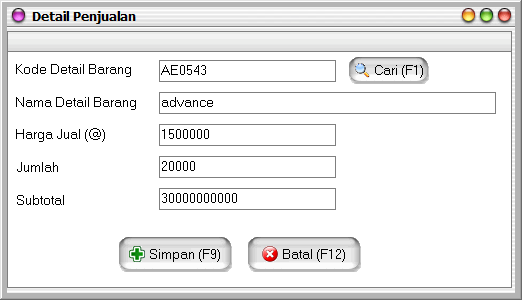
**5.2.9 Form Penjualan**

Pada form penjualan digunakan untuk melakukan proses transaksi penjualan, pengguna dapat menambahkan data transaksi baru yang sesuai dengan transaksi yang dilakukan. Proses terakhir dalam form penjualan ini akan memunculkan form pembayaran.



**Gambar 5.9** Tampilan Form Penjualan

Saat pengguna menambahkan data barang yang dijual, akan muncul form detail penjualan dimana pengguna diminta untuk memasukkan data jumlah barang yang dijual.

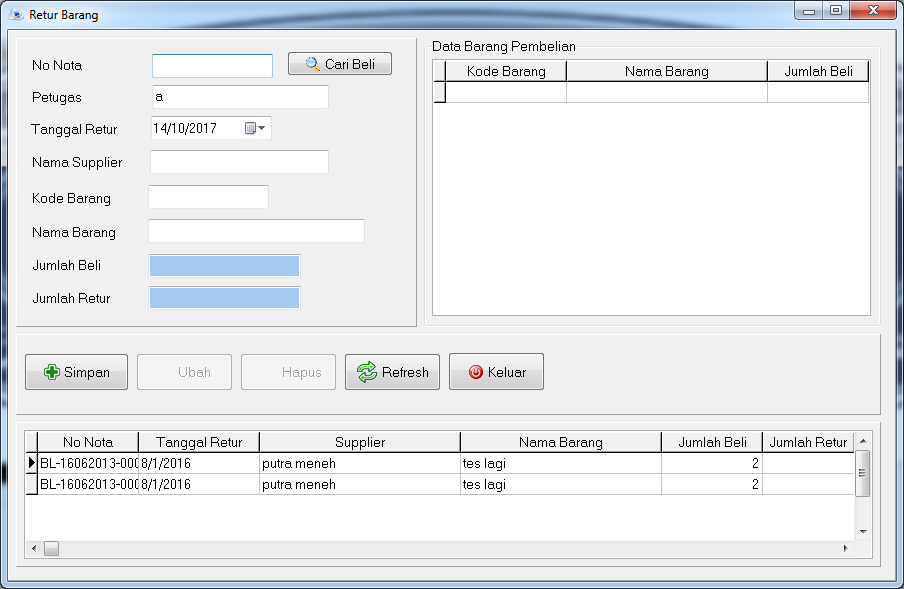


**Gambar 5.10** Tampilan Form Detail Penjualan

penggantian barang yang rusak atau cacat produksi dari semua daftar pelanggan yang yang sudah membeli barang ataupun memesan barang.

**5.2.10 Form Retur Barang**

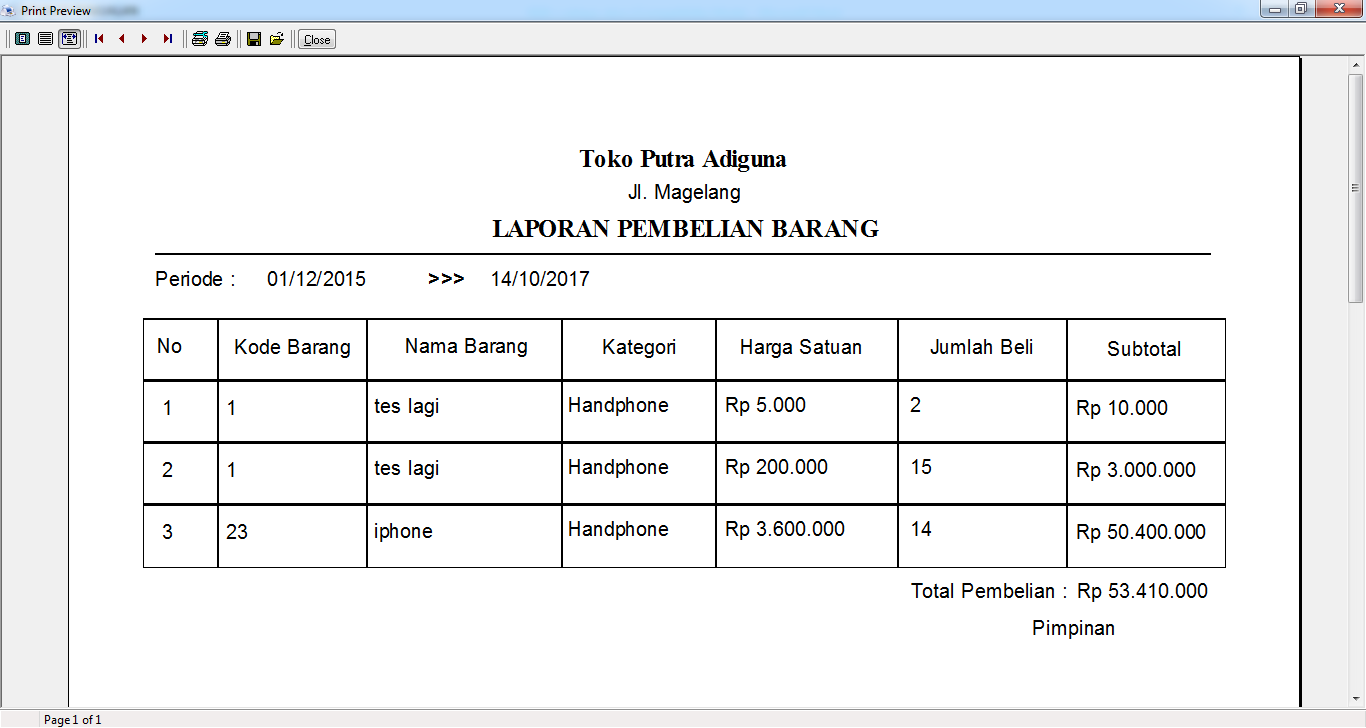
Pada form retur barang digunakan untuk melakukan proses retur, pengguna dapat meminta penggantian barang yang rusak atau cacat produksi dari semua daftar supplier yang sesuai dengan detail pembelian.

 **Gambar 5.11** Tampilan Form Retur Barang

Tampilan retur barang merupakan bagian dari seluruh data barang yang sudah diganti dengan masing-masing no nota yang ditentukan.

**5.2.11 Form Laporan Pembelian Barang**

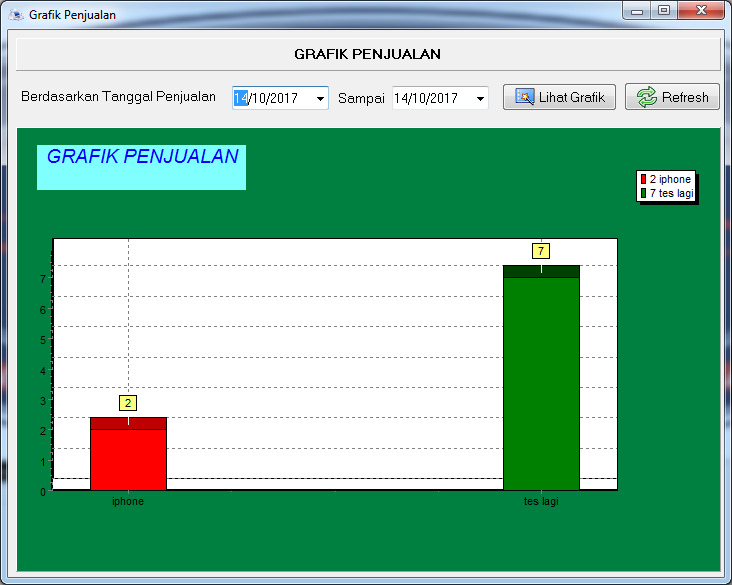
Pada form laporan pembelian berisi informasi tentang transaksi pembelian barang, meliputi informasi tanggal transaksi pembelian hingga total harga beli barang.



**Gambar 5.12** Tampilan Form Laporan Pembelian Barang

**5.2.12 Form Grafik Penjualan**

Pada form grafik penjualan berisi informasi penjualan barang dalam bentuk grafik, grafik penjualan bisa dilihat berdasarkan pertanggal transaksi.



**Gambar 5.13** Tampilan Form Grafik Penjualan

**BAB VI  
PENUTUP**

**6.1 Simpulan**

Berdasarkan hasil penelitian, percobaan dan yang telah dilakukan, maka ditarik beberapa kesimpulan, antara lain :

1. Dengan menggunakan sistem aplikasi penjualan dan manajemen toko ini diharapkan dapat memberikan suatu alternatif baru yang dapat dijadikan pertimbangan untuk kelanjutan operasional sistem yang akan datang.
2. Dengan membangun aplikasi penjualan dan manajemen toko ini, maka telah memudahkan bertransaksi lebih cepat. Serta pendataan semua jenis transaksi dapat di hitung lebih akurat dan efisien waktu.
3. Pada aplikasi penjualan dan manajemen toko ini pengguna dapat dibatasi dengan group login tersendiri serta dapat terjamin keamanan secara client server.
4. Dengan akses login yang diberikan kepada pengguna, maka pengguna akan merasa lebih nyaman bahwa bagian transaksi sudah ditentukan masing-masing.
5. Pengunaan aplikasi penjualan barang ini masih terdapat kekurangan yaitu belum untuk transaksi pulsa online.

**6.2 Saran**

Berdasarkan hasil penelitian dan simpulan diatas, maka dapat diberikan saran sebagai berikut :

1. Penelitian dapat dikembangkan lagi dengan metode yang sama seperti menambahkan metode *sms gateway* untuk transksi pulsa.
2. Aplikasi ini masih dapat dikembangkan lagiuntuk menambahkan beberapa fitur baru seperti menambahkan foto ataupun gambar barang.

**DAFTAR PUSTAKA**

Dasmasela, A.F., 2012, ”Implementasi Sistem Informasi Penjualandan Pembelian Barang pada Koperasi Mahasiswa Universitas Merdeka Malang”, *Proyek Tugas Akhir*, Teknik Informatika, Universitas Teknologi Yogyakarta, Yogyakarta.

Hutahaean, J., 2014,” Konsep Sistem Informasi”, Deepublish, Yogyakarta.

Kadir, A., 2014, ”Pengenalan Sistem Informasi”, Edisi Revisi, Andi Offset, Yogyakarta.

Kanam, F.N., 2012, ”Sistem Manajemen Produksi Dan Penjualan Barang pada Distro ’N’Pearlee”, *Proyek Tugas Akhir*, Teknik Informatika, Universitas Teknologi Yogyakarta, Yogyakarta

Marom, C., 2010, ”Sistem Akutansi Perusahaan Dagang”, Grasindo, Jakarta.

Siburian, D., 2012, ”Implementasi Pengolahan Data Penjualan Aksesoris Komputer Berbasis Client Server pada Wara Komputer Yogyakarta”, *Proyek Tugas Akhir*,Teknik Informatika,Universitas Teknologi Yogyakarta, Yogyakarta.

Simamora, H., 2010, ”Akutansi Basis Pengambilan Keputusan Bisnis”, Salemba Empat, Jakarta.

**Lampiran**

unit UUtama;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, XPMan, ImgList, ExtCtrls, Menus, ComCtrls, ToolWin, StdCtrls, IniFiles,

jpeg, Grids, DBGrids, SUIDBCtrls;

type

TFutama = class(TForm)

StatusBar1: TStatusBar;

ToolBar1: TToolBar;

ToolButton1: TToolButton;

ToolButton2: TToolButton;

ToolButton3: TToolButton;

ToolButton4: TToolButton;

ToolButton5: TToolButton;

ToolButton6: TToolButton;

ToolButton11: TToolButton;

ToolButton10: TToolButton;

ToolButton9: TToolButton;

ToolButton8: TToolButton;

ToolButton7: TToolButton;

ToolButton12: TToolButton;

MainMenu1: TMainMenu;

File1: TMenuItem;

Login1: TMenuItem;

Logout1: TMenuItem;

Keluar1: TMenuItem;

Data1: TMenuItem;

DataBarang1: TMenuItem;

DataSupplier1: TMenuItem;

Setting2: TMenuItem;

DataJenisBarang1: TMenuItem;

HargaJualBarang1: TMenuItem;

NamaPimpinanToko1: TMenuItem;

DataPetugas1: TMenuItem;

ransaksi1: TMenuItem;

ransaksiPembelian1: TMenuItem;

ransaksiPenjualan1: TMenuItem;

ransaksiRetur1: TMenuItem;

Laporan1: TMenuItem;

LaporanDataBarang1: TMenuItem;

LaporanDataSupplier1: TMenuItem;

LaporanPembelian1: TMenuItem;

LaporanPenjualan1: TMenuItem;

LaporanReturBarang1: TMenuItem;

LaporanLabaRugi1: TMenuItem;

GrafikPenjualan1: TMenuItem;

Help1: TMenuItem;

PetunjukPenggunaan1: TMenuItem;

Setting1: TMenuItem;

LokasiDatabase1: TMenuItem;

Timer1: TTimer;

ImageList1: TImageList;

XPManifest1: TXPManifest;

AlamatToko1: TMenuItem;

NamaToko1: TMenuItem;

KeteranganToko1: TMenuItem;

entangProgram1: TMenuItem;

Image1: TImage;

nama\_toko: TLabel;

alamat\_toko: TLabel;

ReturKeSupplier1: TMenuItem;

PenggantianBarangDariSupplier1: TMenuItem;

ToolButton13: TToolButton;

ToolButton14: TToolButton;

DataSatuan1: TMenuItem;

GajiPokok1: TMenuItem;

DataPegawai1: TMenuItem;

unjangan1: TMenuItem;

DataPulsa1: TMenuItem;

Penggajian1: TMenuItem;

LaporanPenggajian1: TMenuItem;

StokMinimal1: TMenuItem;

tmr1: TTimer;

suiDBGrid1: TsuiDBGrid;

btnjualpulsa: TToolButton;

btn2: TToolButton;

procedure Timer1Timer(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure AlamatToko1Click(Sender: TObject);

procedure NamaPimpinanToko1Click(Sender: TObject);

procedure NamaToko1Click(Sender: TObject);

procedure ToolButton12Click(Sender: TObject);

procedure DataPetugas1Click(Sender: TObject);

procedure DataSupplier1Click(Sender: TObject);

procedure DataBarang1Click(Sender: TObject);

procedure ransaksiPembelian1Click(Sender: TObject);

procedure ransaksiPenjualan1Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure DataJenisBarang1Click(Sender: TObject);

procedure HargaJualBarang1Click(Sender: TObject);

procedure ToolButton4Click(Sender: TObject);

procedure ToolButton6Click(Sender: TObject);

procedure ToolButton10Click(Sender: TObject);

procedure ToolButton8Click(Sender: TObject);

procedure Login1Click(Sender: TObject);

procedure ToolButton2Click(Sender: TObject);

procedure Logout1Click(Sender: TObject);

procedure Keluar1Click(Sender: TObject);

procedure LaporanDataBarang1Click(Sender: TObject);

procedure LaporanDataSupplier1Click(Sender: TObject);

procedure LaporanPembelian1Click(Sender: TObject);

procedure LaporanPenjualan1Click(Sender: TObject);

procedure LaporanReturBarang1Click(Sender: TObject);

procedure GrafikPenjualan1Click(Sender: TObject);

procedure entangProgram1Click(Sender: TObject);

procedure ReturKeSupplier1Click(Sender: TObject);

procedure PenggantianBarangDariSupplier1Click(Sender: TObject);

procedure ToolButton14Click(Sender: TObject);

procedure LaporanLabaRugi1Click(Sender: TObject);

procedure DataSatuan1Click(Sender: TObject);

procedure GajiPokok1Click(Sender: TObject);

procedure DataPegawai1Click(Sender: TObject);

procedure unjangan1Click(Sender: TObject);

procedure DataPulsa1Click(Sender: TObject);

procedure Penggajian1Click(Sender: TObject);

procedure LaporanPenggajian1Click(Sender: TObject);

procedure StokMinimal1Click(Sender: TObject);

procedure tmr1Timer(Sender: TObject);

procedure btnjualpulsaClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Futama: TFutama;

iniData:TIniFile;

Palmt:String;

Ppimpinan:string;

Pnama:string;

ifOk:Boolean;

status,laporan : string;

Pstokmin:String;

stokminimal:Integer;

implementation

uses Ubarang, Ulogin, Upembelian, Udm, UPenjualan, Upetugas, Uretur,

Usupplier, Ukategori, Udata\_barang, Uset\_harga\_jual, Ugrafik, Unota,

Ulap\_barang, Ulap\_supplier, Uabout, Uretur\_ganti, Uperiode, Usatuan,

Ugaji\_pokok, Upegawai, Utunjangan, Uharga\_semua, Upulsa, Upenggajian,

Uslip\_gaji, Ulap\_penggajian, Ustokminimal, ADODB, Ujualpulsa;

{$R \*.dfm}

procedure TFutama.Timer1Timer(Sender: TObject);

begin

StatusBar1.Panels[1].Text := 'Tanggal: '+FormatDateTime('dd mmmm yyyy',Now);;

StatusBar1.Panels[2].Text :='Jam : '+timetostr(now);

end;

procedure TFutama.FormClose(Sender: TObject; var Action: TCloseAction);

begin

if MessageDlg('Yakin Mau Menutup Aplikasi????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

Application.Terminate

else

Action:=caNone;

end;

procedure TFutama.AlamatToko1Click(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Palmt:=iniData.ReadString('setting','p\_alamat','');

ifOk:=InputQuery('Masukkan Alamat Perusahaan!','Alamat',Palmt);

if ifOk then

begin

if Palmt = '' then

begin

ShowMessage('Alamat Kosong !');

Exit;

end;

iniData.WriteString('setting','p\_alamat',Palmt);

iniData.Free;

alamat\_toko.Caption:=Palmt;

MessageDlg('Alamat Perusahaan Telah Diubah !',mtInformation,[mbOK],0);

end;

except

end;

end;

procedure TFutama.NamaPimpinanToko1Click(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Ppimpinan:=iniData.ReadString('setting','p\_pimpinan','');

ifOk:=InputQuery('Masukkan Nama Pimpinan Toko!','Alamat',Ppimpinan);

if ifOk then

begin

if Ppimpinan = '' then

begin

ShowMessage('Nama Pimpinan Kosong !');

Exit;

end;

iniData.WriteString('setting','p\_pimpinan',Ppimpinan);

iniData.Free;

MessageDlg('Nama Pimpinan Toko Telah Diubah !',mtInformation,[mbOK],0);

end;

except

end;

end;

procedure TFutama.NamaToko1Click(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Pnama:=iniData.ReadString('setting','p\_nama','');

ifOk:=InputQuery('Masukkan Nama Toko!','Nama',Pnama);

if ifOk then

begin

if Pnama = '' then

begin

ShowMessage('Nama Toko Kosong !');

Exit;

end;

iniData.WriteString('setting','p\_nama',Pnama);

iniData.Free;

nama\_toko.Caption:=Pnama;

MessageDlg('Nama Toko Telah Diubah !',mtInformation,[mbOK],0);

end;

except

end;

end;

procedure TFutama.ToolButton12Click(Sender: TObject);

begin

close;

end;

procedure TFutama.DataPetugas1Click(Sender: TObject);

begin

Fpetugas.Show;

end;

procedure TFutama.DataSupplier1Click(Sender: TObject);

begin

Fsupplier.Show;

end;

procedure TFutama.DataBarang1Click(Sender: TObject);

begin

FBarang.Show;

end;

procedure TFutama.ransaksiPembelian1Click(Sender: TObject);

begin

Fpembelian.Show;

end;

procedure TFutama.ransaksiPenjualan1Click(Sender: TObject);

begin

FPenjualan.Show;

end;

procedure TFutama.FormShow(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Pnama:=iniData.ReadString('setting','p\_nama','');

Palmt:=iniData.ReadString('setting','p\_alamat','');

Ppimpinan:=iniData.ReadString('setting','p\_pimpinan','');

alamat\_toko.Caption:=Palmt;

nama\_toko.Caption:=Pnama;

Fnota.nama\_toko.Caption:=Pnama;

Fnota.alamat.Caption:=Palmt;

Fslip\_gaji.nama\_toko.Caption:=Pnama;

Fslip\_gaji.alamat.Caption:=Palmt;

Flap\_penggajian.nama\_toko.Caption:=Pnama;

Flap\_penggajian.alamat.Caption:=Palmt;

except

end;

status :='login';

Login1.Caption:='Login';

Logout1.Visible:=False;

Data1.Visible:=false;

ransaksi1.Visible:=false;

Laporan1.Visible:=false;

Login1.Visible:=true;

Futama.tmr1.Enabled:=False;

FUtama.ToolButton4.Visible:=false;

FUtama.ToolButton6.Visible:=false;

FUtama.ToolButton8.Visible:=false;

FUtama.ToolButton10.Visible:=false;

FUtama.ToolButton5.Visible:=false;

FUtama.ToolButton11.Visible:=false;

FUtama.ToolButton9.Visible:=false;

FUtama.ToolButton13.Visible:=false;

FUtama.ToolButton14.Visible:=false;

FUtama.ToolButton7.Visible:=false;

FUtama.btnjualpulsa.Visible:=false;

end;

procedure TFutama.DataJenisBarang1Click(Sender: TObject);

begin

fkategori.show;

end;

procedure TFutama.HargaJualBarang1Click(Sender: TObject);

begin

Fharga\_semua.Show;

end;

procedure TFutama.ToolButton4Click(Sender: TObject);

begin

Fpembelian.Show;

end;

procedure TFutama.ToolButton6Click(Sender: TObject);

begin

FPenjualan.Show;

end;

procedure TFutama.ToolButton10Click(Sender: TObject);

begin

Fretur.Show;

end;

procedure TFutama.ToolButton8Click(Sender: TObject);

begin

Fgrafik.Show;

end;

procedure TFutama.Login1Click(Sender: TObject);

begin

Flogin.Show;

end;

procedure TFutama.ToolButton2Click(Sender: TObject);

begin

if ToolButton2.Caption='Login' then

begin

if Flogin=nil then

Application.CreateForm(TFlogin,Flogin)

else Flogin.Show;

end else

if ToolButton2.Caption='LogOut' then

LogOut1Click(sender);

end;

procedure TFutama.Logout1Click(Sender: TObject);

begin

ToolButton2.Caption:='Login';

FormShow(sender);

end;

procedure TFutama.Keluar1Click(Sender: TObject);

begin

close;

end;

procedure TFutama.LaporanDataBarang1Click(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Pnama:=iniData.ReadString('setting','p\_nama','');

Palmt:=iniData.ReadString('setting','p\_alamat','');

Ppimpinan:=iniData.ReadString('setting','p\_pimpinan','');

Flap\_barang.alamat.Caption:=Palmt;

Flap\_barang.nama\_toko.Caption:=Pnama;

Flap\_barang.pimpinan.Caption:=Ppimpinan;

except

end;

with dm.Qbarang do

begin

close;

sql.Clear;

sql.Add('SELECT barang.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, barang.stok, barang.harga\_beli, barang.harga\_jual, barang.kd\_satuan, barang.kd\_kategori ');

sql.Add('FROM barang INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan');

open;

end;

Flap\_barang.QuickRep1.Preview;

end;

procedure TFutama.LaporanDataSupplier1Click(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'konfigurasi.ini');

Pnama:=iniData.ReadString('setting','p\_nama','');

Palmt:=iniData.ReadString('setting','p\_alamat','');

Ppimpinan:=iniData.ReadString('setting','p\_pimpinan','');

Flap\_supplier.alamat.Caption:=Palmt;

Flap\_supplier.nama\_toko.Caption:=Pnama;

Flap\_supplier.pimpinan.Caption:=Ppimpinan;

except

end;

with dm.Qsupplier do

begin

close;

sql.Clear;

sql.Add('select \* from supplier');

open;

end;

Flap\_supplier.QuickRep1.Preview;

end;

procedure TFutama.LaporanPembelian1Click(Sender: TObject);

begin

laporan:='beli';

Fperiode.Show;

end;

procedure TFutama.LaporanPenjualan1Click(Sender: TObject);

begin

laporan:='jual';

Fperiode.Show;

end;

procedure TFutama.LaporanReturBarang1Click(Sender: TObject);

begin

laporan:='retur';

Fperiode.Show;

end;

procedure TFutama.GrafikPenjualan1Click(Sender: TObject);

begin

Fgrafik.Show;

end;

procedure TFutama.entangProgram1Click(Sender: TObject);

begin

Fabout.Show;

end;

procedure TFutama.ReturKeSupplier1Click(Sender: TObject);

begin

Fretur.Show;

end;

procedure TFutama.PenggantianBarangDariSupplier1Click(Sender: TObject);

begin

Fretur\_ganti.Show;

end;

procedure TFutama.ToolButton14Click(Sender: TObject);

begin

Fretur\_ganti.Show;

end;

procedure TFutama.LaporanLabaRugi1Click(Sender: TObject);

begin

laporan:='laba';

Fperiode.Show;

end;

procedure TFutama.DataSatuan1Click(Sender: TObject);

begin

fsatuan.show;

end;

procedure TFutama.GajiPokok1Click(Sender: TObject);

begin

fgaji\_pokok.show;

end;

procedure TFutama.DataPegawai1Click(Sender: TObject);

begin

fpegawai.show;

end;

procedure TFutama.unjangan1Click(Sender: TObject);

begin

ftunjangan.show;

end;

procedure TFutama.DataPulsa1Click(Sender: TObject);

begin

fpulsa.show;

end;

procedure TFutama.Penggajian1Click(Sender: TObject);

begin

fpenggajian.show;

end;

procedure TFutama.LaporanPenggajian1Click(Sender: TObject);

begin

laporan:='gaji';

Fperiode.Show;

end;

procedure TFutama.StokMinimal1Click(Sender: TObject);

begin

fstokminimal.showmodal;

end;

procedure TFutama.tmr1Timer(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'setting.ini');

Pstokmin:=iniData.ReadString('setting','stokmin','');

if Pstokmin = '' then

begin

stokminimal:=100;

end else

begin

stokminimal:=StrToInt(Pstokmin);

end;

with dm.qrystokminimal do

begin

close;

sql.clear;

sql.Add('SELECT barang.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, barang.stok, barang.harga\_beli, barang.harga\_jual ');

sql.Add('FROM barang INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan ');

sql.Add(' where barang.stok<='+IntToStr(stokminimal)+'');

ExecSQL;

Active:=true;

end;

if dm.qrystokminimal.RecordCount>0 then

suiDBGrid1.Visible:=True else

suiDBGrid1.Visible:=false;

except

end;

end;

procedure TFutama.btnjualpulsaClick(Sender: TObject);

begin

fjualpulsa.showmodal;

end;

end.

unit Upenjualan;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, SUIForm, Grids, DBGrids, SUIDBCtrls, SUIButton,

ComCtrls, StdCtrls, SUIEdit, SUIImagePanel, ActnList;

type

TFpenjualan = class(TForm)

suiForm1: TsuiForm;

suiPanel1: TsuiPanel;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

edno\_jual: TsuiEdit;

edpetugas: TsuiEdit;

tgl\_jual: TDateTimePicker;

total: TsuiButton;

suiPanel2: TsuiPanel;

suiDBGrid1: TsuiDBGrid;

suiPanel3: TsuiPanel;

btnrefresh: TsuiButton;

btnkeluar: TsuiButton;

btntambah: TsuiButton;

btnhapus: TsuiButton;

btnfinish: TsuiButton;

ActionList1: TActionList;

tambah: TAction;

hapus: TAction;

finish: TAction;

refresh: TAction;

keluar: TAction;

suiButton1: TsuiButton;

suiButton2: TsuiButton;

btnbatal\_transaksi: TsuiButton;

batal: TAction;

procedure btnfinishClick(Sender: TObject);

procedure btnrefreshClick(Sender: TObject);

procedure btnkeluarClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure edno\_jualChange(Sender: TObject);

procedure btntambahClick(Sender: TObject);

procedure btnhapusClick(Sender: TObject);

procedure suiButton2Click(Sender: TObject);

procedure suiButton1Click(Sender: TObject);

procedure btnbatal\_transaksiClick(Sender: TObject);

procedure suiDBGrid1CellClick(Column: TColumn);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fpenjualan: TFpenjualan;

a,b,c:Currency;

k:string;

implementation

uses Udm, Udet\_jual, Ubayar, Upembelian, Unota, Udata\_penjualan, ADODB;

{$R \*.dfm}

procedure TFpenjualan.btnfinishClick(Sender: TObject);

begin

Fbayar.edtotal.Text:=total.Caption;

Fbayar.ShowModal;

end;

procedure TFpenjualan.btnrefreshClick(Sender: TObject);

begin

FormShow(sender);

end;

procedure TFpenjualan.btnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFpenjualan.FormShow(Sender: TObject);

var Kd,tgl :string;

begin

total.Caption:='RP. 0';

tgl:=FormatDateTime('ddmmyyyy',tgl\_jual.Date);

DM.Qpenjualan.Active:=True;

dm.Qpenjualan.Requery();

with Dm.Qpenjualan do

begin

close;

sql.Clear;

sql.Add('select \* from penjualan ');

open;

if (Dm.Qpenjualan.RecordCount<=0) then

begin

edno\_jual.Text :='JL-'+tgl+'-000001';

end

else

begin

Dm.Qumum.Close;

Dm.Qumum.SQL.Clear;

Dm.Qumum.SQL.Add('select max(right(nota\_jual,6)) as Kode from penjualan');

Dm.Qumum.Open;

Kd := FormatFloat('JL-'+tgl+'-000000',Dm.Qumum['Kode']+1);

edno\_jual.Text := Kd;

end;

end;

suiDBGrid1.SetFocus;

tgl\_jual.Date:=now;

btnhapus.Enabled:=false;

btnfinish.Enabled:=false;

btnbatal\_transaksi.Enabled:=false;

SetWindowPos(Self.Handle,HWND\_TOPMOST,0, 0, 0, 0,SWP\_NOMOVE or SWP\_NOSIZE or SWP\_SHOWWINDOW);

end;

procedure TFpenjualan.edno\_jualChange(Sender: TObject);

begin

with dm.Qdet\_jual do

begin

close;

sql.Clear;

sql.Add('SELECT det\_jual.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, det\_jual.hrg\_jual, det\_jual.jumlah\_brg, det\_jual.subtotal, det\_jual.nota\_jual ');

sql.Add('FROM det\_jual INNER JOIN ');

sql.Add(' barang ON det\_jual.kd\_brg = barang.kd\_brg INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan and det\_jual.nota\_jual ='+QuotedStr(edno\_jual.Text));

open;

if dm.Qdet\_jual.RecordCount=0 then

begin

dm.Qdet\_jual.Close;

total.Caption:='RP. 0';

end else

begin

with dm.Qdet\_jual do

begin

close;

sql.Clear;

sql.Add('SELECT det\_jual.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, det\_jual.hrg\_jual, det\_jual.jumlah\_brg, det\_jual.subtotal, det\_jual.nota\_jual ');

sql.Add('FROM det\_jual INNER JOIN ');

sql.Add(' barang ON det\_jual.kd\_brg = barang.kd\_brg INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan and det\_jual.nota\_jual ='+QuotedStr(edno\_jual.Text));

open;

end;

with dm.Qumum do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(subtotal) as total from det\_jual where nota\_jual ='+QuotedStr(edno\_jual.Text));

Open;

if dm.Qumum.RecordCount=0 then

total.Caption:='RP. 0' else

begin

total.Caption:=FieldValues['total'] ;

total.Caption:=FormatCurr('Rp. ##,###,###',FieldValues['total']);

end;

end;

with dm.Qpenjualan do

begin

close;

sql.Clear;

sql.Add('select \* from penjualan where nota\_jual='+QuotedStr(edno\_jual.Text));

open;

if Recordset.RecordCount>0 then

begin

tgl\_jual.Date:=FieldValues['tgl\_jual'];

edpetugas.Text:=FieldValues['id'];

end;

end;

end;

end;

end;

procedure TFpenjualan.btntambahClick(Sender: TObject);

begin

suiButton1.Enabled:=false;

suiButton2.Enabled:=false;

btnkeluar.Enabled:=False;

Fdet\_jual.ShowModal;

end;

procedure TFpenjualan.btnhapusClick(Sender: TObject);

begin

if (edno\_jual.Text='') then

ShowMessage('Data Belum Dipilih,Pilih dahulu!') else

begin

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select stok from barang where kd\_brg='+QuotedStr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));

open;

b:=FieldValues['stok'];

end;

with dm.Q2 do

begin

close;

sql.Clear;

sql.Add('select jml from det\_jual where kd\_brg='+QuotedStr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));

open;

end;

a:=dm.Q2.FieldValues['jml'];

c:=b+a;

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from det\_jual');

SQL.Add('where nota\_jual ='+quotedstr(edno\_jual.Text)+'and kd\_brg='+QuotedStr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));

execsql;

SQL.Clear;

sql.Add('update barang set');

sql.Add('stok = '+(CurrToStr(c)));

SQL.Add('where kd\_brg ='+quotedstr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

dm.Qdet\_jual.Close;

dm.Qdet\_jual.Open;

with dm.Qumum do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(subtotal) as total from det\_jual where nota\_jual ='+QuotedStr(Fpenjualan.edno\_jual.Text));

Open;

if dm.Qdet\_jual.RecordCount=0 then

total.Caption:='0' else

begin

total.Caption:=FieldValues['total'];

total.Caption:=FormatCurr('Rp. ##,###,###',FieldValues['total']);

end;

end;

btnhapus.Enabled:=false;

with dm.Qdet\_jual do

begin

close;

sql.Clear;

sql.Add('SELECT det\_jual.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, det\_jual.hrg\_jual, det\_jual.jumlah\_brg, det\_jual.subtotal, det\_jual.nota\_jual ');

sql.Add('FROM det\_jual INNER JOIN ');

sql.Add(' barang ON det\_jual.kd\_brg = barang.kd\_brg INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan and det\_jual.nota\_jual ='+QuotedStr(edno\_jual.Text));

open;

end;

if dm.Qdet\_jual.RecordCount=0 then

begin

btnfinish.Enabled:=false;

btnbatal\_transaksi.Enabled:=false;

end else

begin

btnfinish.Enabled:=true;

btnbatal\_transaksi.Enabled:=false;

end;

end;

end;

procedure TFpenjualan.suiButton2Click(Sender: TObject);

begin

with dm.Qnota do

begin

close;

sql.Clear;

sql.Add('SELECT penjualan.nota\_jual, penjualan.id, penjualan.tgl\_jual, penjualan.total\_harga, barang.kd\_brg, barang.nama\_brg, kategori.kategori, det\_jual.hrg\_jual, det\_jual.jumlah\_brg, det\_jual.subtotal ');

sql.Add('FROM det\_jual INNER JOIN ');

sql.Add(' barang ON det\_jual.kd\_brg = barang.kd\_brg INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' penjualan ON det\_jual.nota\_jual = penjualan.nota\_jual ');

sql.Add(' where penjualan.nota\_jual='+(QuotedStr(Fpenjualan.edno\_jual.Text)));

Open;

end;

Fnota.bayar.Caption:='Rp. 0';

Fnota.kembali.Caption:='Rp. 0';

Fpenjualan.Close;

Fnota.QuickRep1.Preview;

end;

procedure TFpenjualan.suiButton1Click(Sender: TObject);

begin

Fdata\_penjualan.ShowModal;

end;

procedure TFpenjualan.btnbatal\_transaksiClick(Sender: TObject);

begin

if MessageDlg('Yakin Mau Membatalkan Transaksi Ini????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

begin

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select det\_jual.\*,barang.stok from det\_jual,barang where '+

' det\_jual.kd\_brg=barang.kd\_brg and det\_jual.nota\_jual='+ QuotedStr(edno\_jual.Text));

open;

a:=FieldValues['stok'];

b:=FieldValues['jumlah\_brg'];

c:=a+b;

k:=FieldValues['kd\_brg'];end;

while not dm.Q1.Eof do

begin

try dm.koneksi.BeginTrans;

with dm.Qumum do

begin

//Active:=False;

Close;

sql.Clear;

sql.add('delete from det\_jual');

sql.Add('where nota\_jual='+Quotedstr(edno\_jual.Text));

ExecSQL;

SQL.Clear;

sql.Add('update barang set');

sql.Add('stok = '+(CurrToStr(c)));

SQL.Add('where kd\_brg ='+quotedstr(k));

execsql;

end;

dm.koneksi.CommitTrans;

dm.Qdet\_jual.Close;

dm.Qdet\_jual.Open;

except dm.koneksi.RollbackTrans;

end;

dm.Q1.Next;

end;

edno\_jualChange(sender);

btnfinish.Enabled:=False;

Fpembelian.btnbatal\_transaksi.Enabled:=false;

btnhapus.Enabled:=False;

total.Caption:='RP. 0';

end;

end;

procedure TFpenjualan.suiDBGrid1CellClick(Column: TColumn);

begin

if btnfinish.Enabled=false then

btnhapus.Enabled:=false else

btnhapus.Enabled:=true;

end;

end.

unit Upembelian;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, SUIForm, Grids, DBGrids, SUIDBCtrls, ComCtrls,

StdCtrls, SUIEdit, SUIImagePanel, ActnList, SUIButton, DBCtrls;

type

TFpembelian = class(TForm)

suiForm1: TsuiForm;

suiPanel1: TsuiPanel;

Label1: TLabel;

Label2: TLabel;

edno\_beli: TsuiEdit;

edpetugas: TsuiEdit;

Label3: TLabel;

tgl\_beli: TDateTimePicker;

suiPanel2: TsuiPanel;

suiDBGrid1: TsuiDBGrid;

Label4: TLabel;

total: TsuiButton;

suiPanel3: TsuiPanel;

btnrefresh: TsuiButton;

btnkeluar: TsuiButton;

btntambah: TsuiButton;

btnhapus: TsuiButton;

btnfinish: TsuiButton;

ActionList1: TActionList;

tambah: TAction;

hapus: TAction;

finish: TAction;

refresh: TAction;

keluar: TAction;

suiButton1: TsuiButton;

btnbatal\_transaksi: TsuiButton;

batal: TAction;

Timer1: TTimer;

lbl1: TLabel;

dblkcbbsupplier: TDBLookupComboBox;

procedure FormShow(Sender: TObject);

procedure btnkeluarClick(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure btntambahClick(Sender: TObject);

procedure btnhapusClick(Sender: TObject);

procedure btnfinishClick(Sender: TObject);

procedure edno\_beliChange(Sender: TObject);

procedure suiButton1Click(Sender: TObject);

procedure suiDBGrid1CellClick(Column: TColumn);

procedure btnbatal\_transaksiClick(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fpembelian: TFpembelian;

a,b,c:Currency;

k:string;

implementation

uses Udm, Ucari\_det\_brg, Udet\_beli, DB, Udata\_pembelian, ADODB;

{$R \*.dfm}

procedure TFpembelian.FormShow(Sender: TObject);

var Kd,tgl :string;

begin

tgl:=FormatDateTime('ddmmyyyy',tgl\_beli.Date);

with Dm.Qpembelian do

begin

close;

sql.Clear;

sql.Add('select \* from pembelian ');

open;

if (Dm.Qpembelian.RecordCount<=0) then

begin

edno\_beli.Text :='BL-'+tgl+'-000001';

end

else

begin

Dm.Qumum.Close;

Dm.Qumum.SQL.Clear;

Dm.Qumum.SQL.Add('select max(right(nota\_beli,6)) as Kode from pembelian');

Dm.Qumum.Open;

Kd := FormatFloat('BL-'+tgl+'-000000',Dm.Qumum['Kode']+1);

edno\_beli.Text := Kd;

end;

end;

tgl\_beli.Date:=now;

suiDBGrid1.SetFocus;

total.Caption:='RP. 0';

btnhapus.Enabled:=false;

btnfinish.Enabled:=false;

btnbatal\_transaksi.Enabled:=false;

//SetWindowPos(Self.Handle,HWND\_TOPMOST,0, 0, 0, 0,SWP\_NOMOVE or SWP\_NOSIZE or SWP\_SHOWWINDOW);

end;

procedure TFpembelian.btnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFpembelian.FormClose(Sender: TObject; var Action: TCloseAction);

begin

Fpembelian:=nil;

Action:=caFree;

end;

procedure TFpembelian.btntambahClick(Sender: TObject);

begin

suiButton1.Enabled:=false;

btnkeluar.Enabled:=False;

Fdet\_beli.ShowModal;

end;

procedure TFpembelian.btnhapusClick(Sender: TObject);

begin

if (edno\_beli.Text='') then

ShowMessage('Data Belum Dipilih,Pilih dahulu!') else

begin

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select stok from barang where kd\_brg='+QuotedStr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));

open;

b:=FieldValues['stok'];

end;

with dm.Q2 do

begin

close;

sql.Clear;

sql.Add('select jml\_beli from det\_beli where kd\_brg='+QuotedStr(dm.Qdet\_jual.fieldbyname('kd\_brg').AsString));

open;

end;

a:=dm.Q2.FieldValues['jml\_beli'];

c:=b-a;

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from det\_beli');

SQL.Add('where nota\_beli ='+quotedstr(edno\_beli.Text)+'and kd\_brg='+QuotedStr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));

execsql;

SQL.Clear;

sql.Add('update barang set');

sql.Add('stok = '+(CurrToStr(c)));

SQL.Add('where kd\_brg ='+quotedstr(dm.Qdet\_beli.fieldbyname('kd\_brg').AsString));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

dm.Qdet\_beli.Close;

dm.Qdet\_beli.Open;

with dm.Qumum do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(subtotal) as total from det\_beli where nota\_beli ='+QuotedStr(Fpembelian.edno\_beli.Text));

Open;

if dm.Qdet\_beli.RecordCount=0 then

total.Caption:='0' else

begin

total.Caption:=FieldValues['total'];

total.Caption:=FormatCurr('Rp. ##,###,###',FieldValues['total']);

end;

end;

btnhapus.Enabled:=false;

with dm.Qdet\_beli do

begin

close;

sql.Clear;

sql.Add('select det\_beli.nota\_beli,det\_beli.jumlah,det\_beli.total,barang.harga\_jual '+

' ,barang.kd\_brg,barang.nama\_brg,kategori.kategori,det\_beli.harga,satuan.satuan from '+

' det\_beli,barang,kategori,satuan '+

' where barang.kd\_brg=det\_beli.kd\_brg and kategori.kd\_kategori=barang.kd\_kategori and satuan.kd\_satuan=barang.kd\_satuan '+

' and det\_beli.nota\_beli ='+QuotedStr(edno\_beli.Text));

open;

end;

if dm.Qdet\_beli.RecordCount=0 then

begin

btnfinish.Enabled:=false;

Fpembelian.btnbatal\_transaksi.Enabled:=false;

end else

begin

btnfinish.Enabled:=true;

Fpembelian.btnbatal\_transaksi.Enabled:=false;

end;

end;

end;

procedure TFpembelian.btnfinishClick(Sender: TObject);

begin

if (edno\_beli.Text='') or (dblkcbbsupplier.KeyValue=null) then

ShowMessage('Data Kurang Lengkap, Silahkan Dilengkapi!') else

begin

with dm.Qpembelian do

begin

sql.Clear;

SQL.Add('select \* from pembelian where nota\_beli='+quotedstr(edno\_beli.text)+'');

open;

end;

if dm.Qpembelian.Recordset.RecordCount > 0 then

ShowMessage('Data sudah ada silahkan isi yang lain!!')

else

begin

with dm.Q1 do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(total) as total from det\_beli where nota\_beli ='+QuotedStr(edno\_beli.Text));

Open;

if dm.Q1.RecordCount=0 then

a:=0 else

a:=FieldValues['total'];

end;

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('insert into pembelian values('+quotedstr(edno\_beli.Text)+','

+quotedstr(edpetugas.Text)+','

+quotedstr(dblkcbbsupplier.KeyValue)+','

+quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_beli.Date))+','

+(CurrToStr(a))+')');

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Disimpan');

except dm.Koneksi.RollbackTrans

end;

dm.Qpembelian.Close;

dm.Qpembelian.Open;

FormShow(sender);

end;

end;

end;

procedure TFpembelian.edno\_beliChange(Sender: TObject);

begin

with dm.Qdet\_beli do

begin

close;

sql.Clear;

sql.Add('select det\_beli.nota\_beli,det\_beli.jumlah,det\_beli.total,barang.harga\_jual '+

' ,barang.kd\_brg,barang.nama\_brg,kategori.kategori,det\_beli.harga,satuan.satuan from '+

' det\_beli,barang,kategori,satuan '+

' where barang.kd\_brg=det\_beli.kd\_brg and kategori.kd\_kategori=barang.kd\_kategori and satuan.kd\_satuan=barang.kd\_satuan '+

' and det\_beli.nota\_beli ='+QuotedStr(edno\_beli.Text));

open;

if dm.Qdet\_beli.RecordCount=0 then

begin

dm.Qdet\_beli.Close;

btnfinish.Enabled:=false;

end else

begin

with dm.Qdet\_beli do

begin

close;

sql.Clear;

sql.Add('select det\_beli.nota\_beli,det\_beli.jumlah,det\_beli.total,barang.harga\_jual '+

' ,barang.kd\_brg,barang.nama\_brg,kategori.kategori,det\_beli.harga,satuan.satuan from '+

' det\_beli,barang,kategori,satuan '+

' where barang.kd\_brg=det\_beli.kd\_brg and kategori.kd\_kategori=barang.kd\_kategori and satuan.kd\_satuan=barang.kd\_satuan '+

' and det\_beli.nota\_beli ='+QuotedStr(edno\_beli.Text));

open;

end;

with dm.Qumum do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(total) as total from det\_beli where nota\_beli ='+QuotedStr(edno\_beli.Text));

Open;

if dm.Qumum.RecordCount=0 then

total.Caption:='Rp. 0' else

begin

total.Caption:=FieldValues['total'];

total.Caption:=FormatCurr('Rp. ##,###,###',FieldValues['total']);

end;

end;

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select \* from pembelian where nota\_beli='+QuotedStr(edno\_beli.Text));

open;

if Recordset.RecordCount>0 then

begin

edpetugas.Text:=FieldValues['id'];

tgl\_beli.Date:=FieldValues['tgl\_beli'];

end;

end;

end;

end;

end;

procedure TFpembelian.suiButton1Click(Sender: TObject);

begin

Fdata\_pembelian.ShowModal;

end;

procedure TFpembelian.suiDBGrid1CellClick(Column: TColumn);

begin

if btnfinish.Enabled=false then

btnhapus.Enabled:=false else

btnhapus.Enabled:=true;

end;

procedure TFpembelian.btnbatal\_transaksiClick(Sender: TObject);

begin

if MessageDlg('Yakin Mau Membatalkan Transaksi Ini????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

begin

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select det\_beli.\*,barang.stok from det\_beli,barang where '+

' det\_beli.kd\_brg=barang.kd\_brg and det\_beli.nota\_beli='+ QuotedStr(edno\_beli.Text));

open;

a:=FieldValues['stok'];

b:=FieldValues['jml\_beli'];

c:=a-b;

k:=FieldValues['kd\_brg'];

end;

while not dm.Q1.Eof do

begin

try dm.koneksi.BeginTrans;

with dm.Qumum do

begin

Close;

sql.Clear;

sql.add('delete from det\_beli');

sql.Add('where nota\_beli='+Quotedstr(edno\_beli.Text));

ExecSQL;

SQL.Clear;

sql.Add('update barang set');

sql.Add('stok = '+(CurrToStr(c)));

SQL.Add('where kd\_brg ='+quotedstr(k));

execsql;

end;

dm.koneksi.CommitTrans;

dm.Qdet\_beli.Close;

dm.Qdet\_beli.Open;

except dm.koneksi.RollbackTrans;

end;

dm.Q1.Next;

end;

edno\_beliChange(sender);

btnfinish.Enabled:=False;

btnbatal\_transaksi.Enabled:=false;

btnhapus.Enabled:=False;

total.Caption:='RP. 0';

end;

end;

procedure TFpembelian.Timer1Timer(Sender: TObject);

begin

if btnfinish.Enabled=false then

btnkeluar.Enabled:=true else

btnkeluar.Enabled:=false;

end;

end.

unit Upegawai;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtDlgs, ExtCtrls, ComCtrls, ADODB, JPEG, DB, StdCtrls, Buttons,

DBCtrls, Grids, DBGrids;

type

TFpegawai = class(TForm)

Panel1: TPanel;

GroupBox1: TGroupBox;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label4: TLabel;

Label5: TLabel;

Label10: TLabel;

Label11: TLabel;

Label12: TLabel;

ednip: TEdit;

ednama\_peg: TEdit;

edtmp\_lhr: TEdit;

tgl\_lhr: TDateTimePicker;

cbjns\_kel: TComboBox;

edpndk\_akhir: TEdit;

mealamat: TMemo;

tgl\_masuk: TDateTimePicker;

Panel2: TPanel;

btnsimpan: TBitBtn;

BtnUbah: TBitBtn;

Btnkeluar: TBitBtn;

Btnrefresh: TBitBtn;

pnl1: TPanel;

dbgrd1: TDBGrid;

btnBtnhapus: TBitBtn;

procedure BtnkeluarClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure BtnrefreshClick(Sender: TObject);

procedure edjml\_anakKeyPress(Sender: TObject; var Key: Char);

procedure btnsimpanClick(Sender: TObject);

procedure BtnUbahClick(Sender: TObject);

procedure FormCreate(Sender: TObject);

procedure FormKeyPress(Sender: TObject; var Key: Char);

procedure btnBtnhapusClick(Sender: TObject);

procedure dbgrd1CellClick(Column: TColumn);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fpegawai: TFpegawai;

implementation

uses Udm;

{$R \*.dfm}

procedure TFpegawai.BtnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFpegawai.FormShow(Sender: TObject);

begin

ednip.Clear;

ednama\_peg.Clear;

edtmp\_lhr.Clear;

tgl\_lhr.Date:=now;

cbjns\_kel.Text:='';

edpndk\_akhir.Clear;

mealamat.Clear;

tgl\_masuk.Date:=now;

ednip.SetFocus;

btnsimpan.Enabled:=true;

btnubah.Enabled:=false;

btnBtnhapus.Enabled:=False;

end;

procedure TFpegawai.BtnrefreshClick(Sender: TObject);

begin

FormShow(sender);

end;

procedure TFpegawai.edjml\_anakKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in [#8,'0'..'9']) then

Key:=#0;

end;

procedure TFpegawai.btnsimpanClick(Sender: TObject);

var

foto : TMemoryStream;

begin

if (ednip.Text='') or (ednama\_peg.Text='') or (edtmp\_lhr.Text='') or

(edpndk\_akhir.Text='') or

(cbjns\_kel.Text='') or (mealamat.Text='') then

ShowMessage('Data Kurang Lengkap!') else

begin

with dm.Qpegawai do

begin

sql.Clear;

SQL.Add('select \* from pegawai where nip='+quotedstr(ednip.text)+'');

open;

end;

if dm.Qpegawai.Recordset.RecordCount > 0 then

ShowMessage('Data sudah ada silahkan isi yang lain!!')

else

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('insert into pegawai values('+quotedstr(ednip.Text)+','

+quotedstr(ednama\_peg.Text)+','

+quotedstr(edtmp\_lhr.Text)+','

+quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_lhr.Date))+','

+quotedstr(cbjns\_kel.Text)+','

+quotedstr(edpndk\_akhir.Text)+','

+quotedstr(mealamat.Text)+','

+quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_masuk.Date))+')');

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Pegawai Berhasil Disimpan');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end;

end;

end;

procedure TFpegawai.BtnUbahClick(Sender: TObject);

begin

if (ednip.Text='') or (ednama\_peg.Text='') or (edtmp\_lhr.Text='') or

(edpndk\_akhir.Text='') or(cbjns\_kel.Text='') or (mealamat.Text='') then

ShowMessage('Data Kurang Lengkap!') else

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('update pegawai set');

sql.Add('nama = '+quotedstr(ednama\_peg.Text)+',');

sql.Add('tempat\_lhr = '+quotedstr(edtmp\_lhr.Text)+',');

sql.Add('tgl\_lhr = '+quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_lhr.Date))+',');

sql.Add('jns\_kel = '+quotedstr(cbjns\_kel.Text)+',');

sql.Add('pndk\_akhir = '+quotedstr(edpndk\_akhir.Text)+',');

sql.Add('alamat= '+QuotedStr(mealamat.Text)+',');

sql.Add('tgl\_masuk= '+QuotedStr(FormatDateTime('mm/dd/yyyy',tgl\_masuk.Date)));

SQL.Add('where nip ='+quotedstr(ednip.Text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Diubah');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end;

end;

procedure TFpegawai.FormCreate(Sender: TObject);

begin

KeyPreview:=true;

end;

procedure TFpegawai.FormKeyPress(Sender: TObject; var Key: Char);

begin

if Key = #13 then

begin

Key := #0; PostMessage(Handle, WM\_NEXTDLGCTL, 0, 0);

end;

end;

procedure TFpegawai.btnBtnhapusClick(Sender: TObject);

begin

if MessageDlg('Yakin Mau Menghapus Data Ini????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from pegawai');

SQL.Add('where nip='+QuotedStr(ednip.text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end

else

FormShow(sender);

end;

procedure TFpegawai.dbgrd1CellClick(Column: TColumn);

begin

btnsimpan.Enabled:=false;

btnubah.Enabled:=true;

btnBtnhapus.Enabled:=True;

with DM.Qpegawai do

begin

ednip.Text:=FieldValues['nip'];

ednama\_peg.Text:=FieldValues['nama'];

edtmp\_lhr.Text:=FieldValues['tempat\_lhr'];

tgl\_lhr.Date:=FieldValues['tgl\_lhr'];

cbjns\_kel.Text:=FieldValues['jns\_kel'];

edpndk\_akhir.Text:=FieldValues['pndk\_akhir'];

mealamat.Text:=FieldValues['alamat'];;

tgl\_masuk.Date:=FieldValues['tgl\_masuk'];

end;

end;

end.

unit Upenggajian;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtCtrls, Grids, DBGrids, ComCtrls, StdCtrls, Buttons, DBCtrls;

type

TFpenggajian = class(TForm)

Panel1: TPanel;

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Label6: TLabel;

Label7: TLabel;

edno\_slip: TEdit;

tgl\_gaji: TDateTimePicker;

ednip: TEdit;

ednama\_peg: TEdit;

edgapok: TEdit;

edmasa\_kerja: TEdit;

Label8: TLabel;

BitBtn2: TBitBtn;

GroupBox3: TGroupBox;

Label17: TLabel;

edgaji\_kotor: TEdit;

Panel5: TPanel;

btnsimpan: TBitBtn;

Btnrefresh: TBitBtn;

BitBtn1: TBitBtn;

Btnkeluar: TBitBtn;

BitBtn3: TBitBtn;

Lgapok: TLabel;

lbltotal: TLabel;

Label19: TLabel;

cbstatus: TComboBox;

lbl1: TLabel;

dblkcbbtunjangan: TDBLookupComboBox;

edtbesartunjangan: TEdit;

lbl2: TLabel;

lblkdgapok: TLabel;

procedure BtnkeluarClick(Sender: TObject);

procedure ednipChange(Sender: TObject);

procedure BitBtn2Click(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure BtnrefreshClick(Sender: TObject);

procedure edno\_slipChange(Sender: TObject);

procedure edgolChange(Sender: TObject);

procedure btntmbh\_tunjClick(Sender: TObject);

procedure Btntambah\_tunjClick(Sender: TObject);

procedure Btnhapus\_tunjClick(Sender: TObject);

procedure Btnhapus\_potClick(Sender: TObject);

procedure edtunj\_lainChange(Sender: TObject);

procedure edpot\_lainChange(Sender: TObject);

procedure btntmbh\_potClick(Sender: TObject);

procedure edtunj\_jabChange(Sender: TObject);

procedure edtunj\_anakChange(Sender: TObject);

procedure edtunj\_suamiChange(Sender: TObject);

procedure edpot\_kopChange(Sender: TObject);

procedure edtot\_tunjChange(Sender: TObject);

procedure btnsimpanClick(Sender: TObject);

procedure edtot\_potChange(Sender: TObject);

procedure BitBtn3Click(Sender: TObject);

procedure BitBtn1Click(Sender: TObject);

procedure edmasa\_kerjaChange(Sender: TObject);

procedure dblkcbbtunjanganCloseUp(Sender: TObject);

procedure edtbesartunjanganChange(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fpenggajian: TFpenggajian;

a,b,c,d,e,f,g,h,i,j,k:Currency;

cari:string;

implementation

uses Udm, DB, Ucari\_pegawai, DateUtils, Ucari\_tunj, Udata\_gaji,

Uslip\_gaji, ADODB;

{$R \*.dfm}

procedure TFpenggajian.BtnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFpenggajian.ednipChange(Sender: TObject);

begin

if ednip.Text='' then

ednip.Text:='' else

begin

with dm.Qumum do

begin

close;

sql.Clear;

sql.Add('select \* from pegawai where pegawai.nip='+QuotedStr(ednip.Text)) ;

open;

ednama\_peg.Text:=FieldValues['nama'];

end;

{with dm.Qgapok do

begin

close;

sql.Clear;

sql.Add('select gapok from tb\_gapok,tb\_golongan,tb\_jabatan where '+

' tb\_golongan.kd\_gol=tb\_gapok.kd\_gol and tb\_jabatan.kd\_jab=tb\_gapok.kd\_jab '+

' and tb\_golongan.gol='+QuotedStr(edgol.Text)+' and tb\_jabatan.jab='+QuotedStr(edjab.Text)+' and '+QuotedStr(edmasa\_kerja.Text)+' >=dari\_th and '+QuotedStr(edmasa\_kerja.Text)+' <=sampai\_th');

Open;

end;

e:=dm.Qgapok.FieldValues['gapok']; }

with dm.Qumum do

begin

close;

sql.Clear;

sql.Add('select year(tgl\_masuk) as thn from pegawai where nip='+QuotedStr(ednip.Text));

Open;

a:=YearOf(now);

b:=dm.Qumum.FieldValues['thn'];

c:=a-b;

edmasa\_kerja.Text:=CurrToStr(c);

end;

{with dm.Qumum do

begin

close;

sql.Clear;

sql.Add('select status from tb\_pegawai where nip='+QuotedStr(ednip.Text));

open;

if (dm.Qumum.FieldValues['status']='Single') or (dm.Qumum.FieldValues['status']='Janda') or (dm.Qumum.FieldValues['status']='Duda') then

begin

Ltun\_suami.Caption:='0';

edtunj\_suami.Text:='Rp. 0';

end else

if dm.Qumum.FieldValues['status']='Menikah' then

begin

d:=(dm.Qtunj\_pokok.FieldValues['suami']/100)\*(StrToCurr(Lgapok.Caption));

Ltun\_suami.Caption:=CurrToStr(d);

edtunj\_suami.Text:=FormatCurr('Rp. ##,###,###',d);

end;

end; }

{with dm.Qumum do

begin

close;

sql.Clear;

sql.Add('select jml\_anak from tb\_pegawai where nip='+QuotedStr(ednip.Text));

open;

if dm.Qumum.FieldValues['jml\_anak']=0 then

begin

Ltunj\_anak.Caption:='0';

edtunj\_anak.Text:='Rp. 0';

end else

if (dm.Qumum.FieldValues['jml\_anak'])<=2 then

begin

g:=((dm.Qumum.FieldValues['jml\_anak'])\*((dm.Qtunj\_pokok.FieldValues['anak']/100)\*(StrToCurr(Lgapok.Caption))));

Ltunj\_anak.Caption:=CurrToStr(g);

edtunj\_anak.Text:=FormatCurr('Rp. ##,###,###',g);

end else

if (dm.Qumum.FieldValues['jml\_anak'])>2 then

begin

f:=(2\*((dm.Qtunj\_pokok.FieldValues['anak']/100)\*(StrToCurr(Lgapok.Caption))));

Ltunj\_anak.Caption:=CurrToStr(f);

edtunj\_anak.Text:=FormatCurr('Rp. ##,###,###',f);

end;

end; }

{

with dm.Qtunj\_jab do

begin

close;

sql.Clear;

sql.Add('select tb\_tunj\_jab.tunj\_jab from tb\_tunj\_jab,tb\_golongan,tb\_jabatan where '+

' tb\_golongan.kd\_gol=tb\_tunj\_jab.kd\_gol and tb\_jabatan.kd\_jab=tb\_tunj\_jab.kd\_jab and '+

' tb\_golongan.gol='+QuotedStr(edgol.Text)+' and tb\_jabatan.jab='+QuotedStr(edjab.Text));

open;

if dm.Qtunj\_jab.RecordCount=0 then

begin

ShowMessage('Tunjangan Jabatan Untuk Jabatan Dan Golongan Tersebut Belum Ada!');

edtunj\_jab.Text:='Rp. 0';

Ltunj\_jab.Caption:='0';

end else

begin

Ltunj\_jab.Caption:=FieldValues['tunj\_jab'];

edtunj\_jab.Text:=FormatCurr('Rp. ##,###,###',FieldValues['tunj\_jab']);

end;

end;

}

{with dm.Qumum do

begin

close;

sql.Clear;

sql.Add('select angsr\_perbulan from tb\_koperasi,tb\_pegawai where '+

' tb\_koperasi.nip=tb\_pegawai.nip and tb\_koperasi.nip='+QuotedStr(ednip.Text)+' and tb\_koperasi.status='+QuotedStr('Belum Lunas'));

open;

if dm.Qumum.RecordCount=0 then

begin

edpot\_kop.Text:='Rp. 0';

Lpot\_kop.Caption:='0';

end else

begin

Lpot\_kop.Caption:=FieldValues['angsr\_perbulan'];

edpot\_kop.Text:=FormatCurr('Rp. ##,###,###',FieldValues['angsr\_perbulan']);

end;

end; }

end;

end;

procedure TFpenggajian.BitBtn2Click(Sender: TObject);

begin

cari:='gaji';

Fcari\_pegawai.show;

end;

procedure TFpenggajian.FormShow(Sender: TObject);

var Kd :string;

begin

with Dm.Qpenggajian do

begin

if (Dm.Qpenggajian.RecordCount<=0) then

begin

edno\_slip.Text := 'S-000001';

end

else

begin

Dm.Qumum.Close;

Dm.Qumum.SQL.Clear;

Dm.Qumum.SQL.Add('select max(right(no\_slip,6)) as Kode from penggajian');

Dm.Qumum.Open;

Kd := FormatFloat('S-000000',Dm.Qumum['Kode']+1);

edno\_slip.Text := Kd;

end;

end;

cbstatus.Text:='';

ednip.Clear;

ednama\_peg.Clear;

edmasa\_kerja.Clear;

edgapok.Clear;

edtbesartunjangan.Clear;

edgaji\_kotor.Clear;

dblkcbbtunjangan.KeyValue:=null;

tgl\_gaji.Date:=now;

Lgapok.Caption:='0';

end;

procedure TFpenggajian.BtnrefreshClick(Sender: TObject);

begin

FormShow(sender);

end;

procedure TFpenggajian.edno\_slipChange(Sender: TObject);

begin

with dm.Qumum do

begin

sql.Clear;

SQL.Add('select \* from penggajian where no\_slip ='+quotedstr(Edno\_slip.text)+'');

open;

if dm.Qumum.Recordset.RecordCount = 0 then

begin

//dm.Qdet\_tunj.Close;

//dm.Qdet\_pot.Close;

ednip.Clear;

end else

begin

tgl\_gaji.Date:=FieldValues['tgl\_gaji'];

ednip.Text:=FieldValues['nip'];

{with dm.Qdet\_pot do

begin

close;

sql.Clear;

sql.Add('select distinct tb\_det\_pot.kd\_pot,tb\_det\_pot.no\_slip ,tb\_det\_pot.pot ,tb\_potongan.nama\_pot from tb\_det\_pot,tb\_potongan '+

'where tb\_det\_pot.kd\_pot=tb\_potongan.kd\_pot and tb\_det\_pot.no\_slip ='+QuotedStr(Edno\_slip.Text));

open;

end;

}

{with dm.Qdet\_tunj do

begin

close;

sql.Clear;

sql.Add('select distinct tb\_det\_tunj.no\_slip ,tb\_det\_tunj.kd\_tunj ,tb\_det\_tunj.tunj ,tb\_tunjangan.nama\_tunj from tb\_det\_tunj,tb\_tunjangan where '+

'tb\_det\_tunj.kd\_tunj=tb\_tunjangan.kd\_tunj and tb\_det\_tunj.no\_slip ='+QuotedStr(Edno\_slip.Text));

open;

end; }

{with dm.Qumum do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(tunj) as tunjangan from tb\_det\_tunj where no\_slip ='+QuotedStr(Edno\_slip.Text));

Open;

if dm.Qdet\_tunj.RecordCount=0 then

edtunj\_lain.Text:='0' else

edtunj\_lain.Text:=FormatCurr('Rp. ##,###,###',FieldValues['tunjangan']);

end; }

{with dm.q1 do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(pot) as potongan from tb\_det\_pot where no\_slip ='+QuotedStr(Fpenggajian.Edno\_slip.Text));

Open;

if dm.Q1.RecordCount=0 then

edpot\_lain.Text:='0' else

edpot\_lain.Text:=FormatCurr('Rp. ##,###,###',FieldValues['potongan']);

end; }

end;

end;

end;

procedure TFpenggajian.edgolChange(Sender: TObject);

begin

{if Edgol.Text='' then

Edgol.Text:='' else

begin

with dm.Qgapok do

begin

close;

sql.Clear;

sql.Add('select gapok from tb\_gapok,tb\_golongan,tb\_jabatan where '+

' tb\_golongan.kd\_gol=tb\_gapok.kd\_gol and tb\_jabatan.kd\_jab=tb\_gapok.kd\_jab '+

' and tb\_golongan.gol='+QuotedStr(edgol.Text)+' and tb\_jabatan.jab='+QuotedStr(edjab.Text)+' and '+QuotedStr(edmasa\_kerja.Text)+' >=dari\_th and '+QuotedStr(edmasa\_kerja.Text)+' <=sampai\_th');

Open;

if dm.Qgapok.RecordCount=0 then

ShowMessage('Setingan Gaji Pokok Untuk Jabatan, Golongan, Dan Masa Kerja Tersebut Belum Ada') else

Lgapok.Caption:=FieldValues['gapok'];

Edgapok.Text:=FormatCurr('Rp. ##,###,###',FieldValues['gapok']);

end;

end; }

end;

procedure TFpenggajian.btntmbh\_tunjClick(Sender: TObject);

begin

Fcari\_tunj.Show;

end;

procedure TFpenggajian.Btntambah\_tunjClick(Sender: TObject);

begin

//Fcari\_pot.Show;

end;

procedure TFpenggajian.Btnhapus\_tunjClick(Sender: TObject);

begin

{try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from det\_tunj');

SQL.Add('where no\_slip ='+quotedstr(Edno\_slip.Text)+'and kd\_tunj='+QuotedStr(dm.Qdet\_tunj.fieldbyname('kd\_tunj').AsString));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Tunjangan Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

dm.Qdet\_tunj.Close;

dm.Qdet\_tunj.Open;

with dm.Q1 do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(tunj) as tunjangan from det\_tunj where no\_slip ='+QuotedStr(Fpenggajian.Edno\_slip.Text));

Open;

if dm.Q1.RecordCount=0 then

begin

Fpenggajian.edtunj\_lain.Text:='Rp. 0';

Ltunj\_lain.Caption:='0';

end else

Ltunj\_lain.Caption:=FieldValues['tunjangan'];

Fpenggajian.edtunj\_lain.Text:=FormatCurr('Rp. ##,###,###',FieldValues['tunjangan']);

end; }

end;

procedure TFpenggajian.Btnhapus\_potClick(Sender: TObject);

begin

{try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from tb\_det\_pot');

SQL.Add('where no\_slip ='+quotedstr(Edno\_slip.Text)+'and kd\_pot='+QuotedStr(dm.Qdet\_pot.fieldbyname('kd\_pot').AsString));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Potongan Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

dm.Qdet\_pot.Close;

dm.Qdet\_pot.Open;

with dm.Q1 do

begin

Close;

SQL.Clear;

SQL.Add('Select sum(pot) as potongan from tb\_det\_pot where no\_slip ='+QuotedStr(Fpenggajian.Edno\_slip.Text));

Open;

if dm.Q1.RecordCount=0 then

begin

Lpot\_lain.Caption:='0';

edpot\_lain.Text:='Rp. 0';

end else

edpot\_lain.Text:=FormatCurr('Rp. ##,###,###',FieldValues['potongan']);

Lpot\_lain.Caption:=FieldValues['potongan'];

end; }

end;

procedure TFpenggajian.edtunj\_lainChange(Sender: TObject);

begin

{if edtunj\_lain.Text='' then

edtunj\_lain.Text:='' else

begin

a:=StrToCurr(Ltunj\_jab.Caption);

b:=StrToCurr(Ltun\_suami.Caption);

c:=StrToCurr(Ltunj\_anak.Caption);

d:=StrToCurr(Ltunj\_lain.Caption);

e:=a+b+c+d;

Ltot\_tunj.Caption:=CurrToStr(e);

edtot\_tunj.Text:=FormatCurr('Rp. ##,###,###',e);

end; }

end;

procedure TFpenggajian.edpot\_lainChange(Sender: TObject);

begin

{if edpot\_lain.Text='' then

edpot\_lain.Text:='' else

begin

a:=StrToCurr(Lpot\_kop.Caption);

b:=StrToCurr(Lpot\_lain.Caption);

c:=a+b;

Ltot\_pot.Caption:=CurrToStr(c);

edtot\_pot.Text:=FormatCurr('Rp. ##,###,###',c);

end; }

end;

procedure TFpenggajian.btntmbh\_potClick(Sender: TObject);

begin

//Fcari\_pot.Show;

end;

procedure TFpenggajian.edtunj\_jabChange(Sender: TObject);

begin

{if edtunj\_jab.Text='' then

edtunj\_jab.Text:='' else

begin

a:=StrToCurr(Ltunj\_jab.Caption);

b:=StrToCurr(Ltun\_suami.Caption);

c:=StrToCurr(Ltunj\_anak.Caption);

d:=StrToCurr(Ltunj\_lain.Caption);

e:=a+b+c+d;

Ltot\_tunj.Caption:=CurrToStr(e);

edtot\_tunj.Text:=FormatCurr('Rp. ##,###,###',e);

end; }

end;

procedure TFpenggajian.edtunj\_anakChange(Sender: TObject);

begin

{if edtunj\_anak.Text='' then

edtunj\_anak.Text:='' else

begin

a:=StrToCurr(Ltunj\_jab.Caption);

b:=StrToCurr(Ltun\_suami.Caption);

c:=StrToCurr(Ltunj\_anak.Caption);

d:=StrToCurr(Ltunj\_lain.Caption);

e:=a+b+c+d;

Ltot\_tunj.Caption:=CurrToStr(e);

edtot\_tunj.Text:=FormatCurr('Rp. ##,###,###',e);

end; }

end;

procedure TFpenggajian.edtunj\_suamiChange(Sender: TObject);

begin

{if edtunj\_suami.Text='' then

edtunj\_suami.Text:='' else

begin

a:=StrToCurr(Ltunj\_jab.Caption);

b:=StrToCurr(Ltun\_suami.Caption);

c:=StrToCurr(Ltunj\_anak.Caption);

d:=StrToCurr(Ltunj\_lain.Caption);

e:=a+b+c+d;

Ltot\_tunj.Caption:=CurrToStr(e);

edtot\_tunj.Text:=FormatCurr('Rp. ##,###,###',e);

end; }

end;

procedure TFpenggajian.edpot\_kopChange(Sender: TObject);

begin

{if edpot\_kop.Text='' then

edpot\_kop.Text:='' else

begin

a:=StrToCurr(Lpot\_kop.Caption);

b:=StrToCurr(Lpot\_lain.Caption);

c:=a+b;

Ltot\_pot.Caption:=CurrToStr(c);

edtot\_pot.Text:=FormatCurr('Rp. ##,###,###',c);

end; }

end;

procedure TFpenggajian.edtot\_tunjChange(Sender: TObject);

begin

{if edtot\_tunj.Text='' then

edtot\_tunj.Text:='' else

begin

a:=StrToCurr(Ltot\_tunj.Caption);

c:=StrToCurr(Ltot\_pot.Caption);

b:=StrToCurr(Lgapok.Caption);

c:=(a+b)-c;

d:=a+b;

Lgaji\_kotor.Caption:=CurrToStr(d);

Lgaji\_bersih.Caption:=CurrToStr(c);

edgaji\_kotor.Text:=FormatCurr('Rp. ##,###,###',d);

edgaji\_bersih.Text:=FormatCurr('Rp. ##,###,###',c);

end; }

end;

procedure TFpenggajian.btnsimpanClick(Sender: TObject);

var kop:string;

begin

if (edno\_slip.Text='') or (ednip.Text='') or (cbstatus.Text='Pilih') then

ShowMessage('Data Kurang Lengkap, Silahkan Dilengkapi!') else

begin

with dm.Qpenggajian do

begin

sql.Clear;

SQL.Add('select \* from penggajian where no\_slip ='+quotedstr(edno\_slip.text)+'');

open;

end;

if dm.Qpenggajian.Recordset.RecordCount > 0 then

ShowMessage('Data sudah ada silahkan isi yang lain!!')

else

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('insert into penggajian values('+quotedstr(Edno\_slip.Text)+','

+quotedstr(FormatDateTime('mm/dd/yyyy',tgl\_gaji.Date))+','

+quotedstr(ednip.Text)+','

+QuotedStr(lblkdgapok.Caption)+','

+quotedstr(dblkcbbtunjangan.KeyValue)+','

+(lbltotal.Caption)+','

+QuotedStr(cbstatus.Text)+')');

ExecSQL;

end;

dm.koneksi.CommitTrans;

ShowMessage('Data Berhasil Disimpan');

except dm.koneksi.RollbackTrans

end;

dm.Qpenggajian.Close;

dm.Qpenggajian.Open;

FormShow(sender);

end;

end;

end;

procedure TFpenggajian.edtot\_potChange(Sender: TObject);

begin

{if edtot\_pot.Text='' then

edtot\_pot.Text:='' else

begin

a:=StrToCurr(Ltot\_tunj.Caption);

c:=StrToCurr(Ltot\_pot.Caption);

b:=StrToCurr(Lgapok.Caption);

c:=(a+b)-c;

d:=a+b;

Lgaji\_kotor.Caption:=CurrToStr(d);

Lgaji\_bersih.Caption:=CurrToStr(c);

edgaji\_kotor.Text:=FormatCurr('Rp. ##,###,###',d);

edgaji\_bersih.Text:=FormatCurr('Rp. ##,###,###',c);

end; }

end;

procedure TFpenggajian.BitBtn3Click(Sender: TObject);

begin

with DM.Qpenggajian do

begin

close;

sql.Clear;

sql.Add('SELECT penggajian.no\_slip, penggajian.tgl\_gaji, penggajian.nip, pegawai.nama, penggajian.kd\_gapok, penggajian.kd\_tunjangan, penggajian.tot\_gaji, penggajian.status, tunjangan.nama\_tunj, gapok.gapok, ');

sql.Add(' tunjangan.besar\_tunj ');

sql.Add('FROM pegawai INNER JOIN ');

sql.Add(' penggajian ON pegawai.nip = penggajian.nip INNER JOIN ');

sql.Add(' gapok ON penggajian.kd\_gapok = gapok.kd\_gapok INNER JOIN ');

sql.Add(' tunjangan ON penggajian.kd\_tunjangan = tunjangan.kd\_tunj ');

ExecSQL;

Active:=True;

end;

Fdata\_gaji.ShowModal;

end;

procedure TFpenggajian.BitBtn1Click(Sender: TObject);

begin

if (edno\_slip.Text='') then

ShowMessage('Silahkan Pilih Dahulu Data Yang Akan Dicetak SLip Gaji!') else

begin

with dm.Qslip\_gaji do

begin

close;

sql.Clear;

sql.Add('SELECT penggajian.no\_slip, penggajian.tgl\_gaji, penggajian.nip, pegawai.nama, penggajian.kd\_gapok, penggajian.kd\_tunjangan, penggajian.tot\_gaji, penggajian.status, tunjangan.nama\_tunj, gapok.gapok, ');

sql.Add(' tunjangan.besar\_tunj ');

sql.Add('FROM pegawai INNER JOIN ');

sql.Add(' penggajian ON pegawai.nip = penggajian.nip INNER JOIN ');

sql.Add(' gapok ON penggajian.kd\_gapok = gapok.kd\_gapok INNER JOIN ');

sql.Add(' tunjangan ON penggajian.kd\_tunjangan = tunjangan.kd\_tunj ');

sql.Add(' where penggajian.no\_slip='+QuotedStr(Edno\_slip.Text));

open;

end;

Fslip\_gaji.QuickRep1.Preview;

end;

end;

procedure TFpenggajian.edmasa\_kerjaChange(Sender: TObject);

begin

if edmasa\_kerja.Text='' then

edmasa\_kerja.Text:='' else

begin

with dm.Qgapok do

begin

close;

sql.Clear;

sql.Add('select gapok,kd\_gapok from gapok where '+(edmasa\_kerja.Text)+' >=dari\_th and '+(edmasa\_kerja.Text)+' <=sampai\_th');

Open;

if dm.Qgapok.RecordCount=0 then

ShowMessage('Setingan Gaji Pokok Untuk Jabatan, Golongan, Dan Masa Kerja Tersebut Belum Ada') else

begin

Lgapok.Caption:=FieldValues['gapok'];

lblkdgapok.Caption:=FieldValues['kd\_gapok'];

Edgapok.Text:=FormatCurr('Rp. ##,###,###',FieldValues['gapok']);

end;

end;

end;

end;

procedure TFpenggajian.dblkcbbtunjanganCloseUp(Sender: TObject);

begin

with DM.Qumum do

begin

close;

sql.Clear;

sql.Add('select besar\_tunj from tunjangan where kd\_tunj='+QuotedStr(dblkcbbtunjangan.KeyValue));

ExecSQL;

Active:=true;

if DM.Qumum.RecordCount>0 then

edtbesartunjangan.Text:=FieldValues['besar\_tunj'];

end;

end;

procedure TFpenggajian.edtbesartunjanganChange(Sender: TObject);

begin

if edtbesartunjangan.Text='' then

edtbesartunjangan.Text:='' else

begin

lbltotal.Caption:=CurrToStr(StrToCurr(Lgapok.Caption)+StrToCurr(edtbesartunjangan.Text));

edgaji\_kotor.Text:=FormatCurr('Rp. ##,###,###',StrToCurr(lbltotal.Caption));

end;

end;

end.

unit Ustokminimal;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, ExtCtrls, Buttons,inifiles;

type

TFstokminimal = class(TForm)

pnlstokmin: TPanel;

lbl1: TLabel;

edtsetstok: TEdit;

btnset: TBitBtn;

btnkeluar: TBitBtn;

procedure btnkeluarClick(Sender: TObject);

procedure FormShow(Sender: TObject);

procedure btnsetClick(Sender: TObject);

procedure edtsetstokKeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fstokminimal: TFstokminimal;

iniData:TIniFile;

Pstokmin:String;

implementation

{$R \*.dfm}

procedure TFstokminimal.btnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFstokminimal.FormShow(Sender: TObject);

begin

edtsetstok.Clear;

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'setting.ini');

Pstokmin:=iniData.ReadString('setting','stokmin','');

if Pstokmin = '' then

begin

pnlstokmin.Caption:=IntToStr(0);

end else

begin

pnlstokmin.Caption:=Pstokmin;

end;

except

end;

end;

procedure TFstokminimal.btnsetClick(Sender: TObject);

begin

try

iniData:=TIniFile.Create(ExtractFilePath(Application.ExeName)+'setting.ini');

Pstokmin:=edtsetstok.Text;

iniData.WriteString('setting','stokmin',Pstokmin);

iniData.Free;

MessageDlg('Setingan Stok Minimal Berhasil !',mtInformation,[mbOK],0);

except

end;

FormShow(sender);

end;

procedure TFstokminimal.edtsetstokKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in [#8,'0'..'9']) then

Key:=#0;

end;

end.

unit Upulsa;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Buttons, Grids, DBGrids, ExtCtrls;

type

TFpulsa = class(TForm)

pnl2: TPanel;

pnl3: TPanel;

lbl3: TLabel;

lbl4: TLabel;

lbl1: TLabel;

edtket: TEdit;

edtnilai: TEdit;

pnl1: TPanel;

dbgrd1: TDBGrid;

pnl4: TPanel;

btnsimpan: TBitBtn;

btnubah: TBitBtn;

btnbthrefresh: TBitBtn;

btnbrnkeluar: TBitBtn;

btnBtnhapus: TBitBtn;

edtmargin: TEdit;

edtidpulsa: TEdit;

procedure edtnilaiKeyPress(Sender: TObject; var Key: Char);

procedure edtmarginKeyPress(Sender: TObject; var Key: Char);

procedure FormShow(Sender: TObject);

procedure btnbthrefreshClick(Sender: TObject);

procedure dbgrd1CellClick(Column: TColumn);

procedure btnsimpanClick(Sender: TObject);

procedure btnubahClick(Sender: TObject);

procedure btnBtnhapusClick(Sender: TObject);

procedure btnbrnkeluarClick(Sender: TObject);

procedure edtketKeyPress(Sender: TObject; var Key: Char);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Fpulsa: TFpulsa;

implementation

uses

Udm, ADODB;

{$R \*.dfm}

procedure TFpulsa.edtnilaiKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

edtmargin.SetFocus;

if not (key in [#8,'0'..'9']) then

Key:=#0;

end;

procedure TFpulsa.edtmarginKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

btnsimpan.SetFocus;

if not (key in [#8,'0'..'9']) then

Key:=#0;

end;

procedure TFpulsa.FormShow(Sender: TObject);

begin

edtket.Clear;

edtnilai.Clear;

edtmargin.Clear;

with dm.qrypulsa do

begin

close;

sql.clear;

sql.Add('select \* from pulsa');

ExecSQL;

open;

end;

btnsimpan.Enabled:=true;

btnubah.Enabled:=false;

btnBtnhapus.Enabled:=False;

end;

procedure TFpulsa.btnbthrefreshClick(Sender: TObject);

begin

FormShow(sender);

end;

procedure TFpulsa.dbgrd1CellClick(Column: TColumn);

begin

btnsimpan.Enabled:=false;

btnubah.Enabled:=true;

btnBtnhapus.Enabled:=True;

with DM.qrypulsa do

begin

edtidpulsa.Text:=FieldValues['id\_pulsa'];

edtket.Text:=FieldValues['ket'];

edtnilai.Text:=FieldValues['nilai'];

edtmargin.Text:=FieldValues['margin'];

end;

end;

procedure TFpulsa.btnsimpanClick(Sender: TObject);

begin

if (edtket.Text='') or (edtnilai.Text='') or (edtmargin.Text='') then

ShowMessage('Data Kurang Lengkap!') else

begin

with dm.qrypulsa do

begin

sql.Clear;

SQL.Add('select \* from pulsa where ket='+quotedstr(edtket.text)+'');

open;

end;

if dm.qrypulsa.Recordset.RecordCount > 0 then

ShowMessage('Data sudah ada silahkan isi yang lain!!')

else

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('insert into pulsa values('

+quotedstr(edtket.Text)+','

+(edtnilai.Text)+','

+(edtmargin.Text)+')');

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Barang Berhasil Disimpan');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end;

end;

end;

procedure TFpulsa.btnubahClick(Sender: TObject);

begin

if (edtidpulsa.Text='') or (edtket.Text='') or (edtnilai.Text='') or (edtmargin.Text='') then

ShowMessage('Data Kurang Lengkap!') else

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('update pulsa set');

sql.Add('ket = '+quotedstr(edtket.Text)+',');

sql.Add('nilai = '+(edtnilai.Text)+',');

sql.Add('margin = '+(edtmargin.Text));

SQL.Add('where id\_pulsa ='+quotedstr(edtidpulsa.Text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Diubah');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end;

end;

procedure TFpulsa.btnBtnhapusClick(Sender: TObject);

begin

if MessageDlg('Yakin Mau Menghapus Data Ini????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from pulsa');

SQL.Add('where id\_pulsa='+QuotedStr(edtidpulsa.Text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

FormShow(sender);

end

else

FormShow(sender);

end;

procedure TFpulsa.btnbrnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFpulsa.edtketKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

edtnilai.SetFocus;

end;

end.

unit Ubarang;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, ExtDlgs, ExtCtrls, ComCtrls, ADODB, JPEG, DB, StdCtrls, Buttons,

DBCtrls, Grids, DBGrids;

type

TFBarang = class(TForm)

Panel1: TPanel;

Panel2: TPanel;

Label1: TLabel;

Label2: TLabel;

Label5: TLabel;

Label6: TLabel;

edkd\_brg: TEdit;

ednama\_brg: TEdit;

dblok\_kat: TDBLookupComboBox;

edstok: TEdit;

Label8: TLabel;

Label9: TLabel;

Panel4: TPanel;

btnsimpan: TBitBtn;

btnubah: TBitBtn;

bthrefresh: TBitBtn;

brnkeluar: TBitBtn;

lbl1: TLabel;

dblkcbbsatuan: TDBLookupComboBox;

pnl1: TPanel;

dbgrd1: TDBGrid;

btnBtnhapus: TBitBtn;

edthargabeli: TEdit;

edthargajual: TEdit;

procedure FormShow(Sender: TObject);

procedure bthrefreshClick(Sender: TObject);

procedure brnkeluarClick(Sender: TObject);

procedure dblok\_katCloseUp(Sender: TObject);

procedure btnsimpanClick(Sender: TObject);

procedure btnubahClick(Sender: TObject);

procedure FormKeyPress(Sender: TObject; var Key: Char);

procedure FormCreate(Sender: TObject);

procedure edstokKeyPress(Sender: TObject; var Key: Char);

procedure btnBtnhapusClick(Sender: TObject);

procedure dbgrd1CellClick(Column: TColumn);

procedure ednama\_brgKeyPress(Sender: TObject; var Key: Char);

procedure dblok\_katKeyPress(Sender: TObject; var Key: Char);

procedure edkd\_brgKeyPress(Sender: TObject; var Key: Char);

procedure edthargabeliKeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

procedure edthargabeliKeyPress(Sender: TObject; var Key: Char);

procedure edthargajualKeyPress(Sender: TObject; var Key: Char);

procedure edthargajualKeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

private

{ Private declarations }

public

{ Public declarations }

end;

var

FBarang: TFBarang;

sRupiah,setkode: string;

iRupiah,a,b: Currency;

implementation

uses Udm;

{$R \*.dfm}

procedure TFBarang.FormShow(Sender: TObject);

begin

//SetWindowPos(Self.Handle,HWND\_TOPMOST,0, 0, 0, 0,SWP\_NOMOVE or SWP\_NOSIZE or SWP\_SHOWWINDOW);

edkd\_brg.Clear;

ednama\_brg.Clear;

edthargabeli.Clear;

edthargajual.Clear;

edstok.Text:='0';

dblok\_kat.KeyValue:=Null;

dblkcbbsatuan.KeyValue:=Null;

ednama\_brg.SetFocus;

btnsimpan.Enabled:=true;

btnubah.Enabled:=false;

btnBtnhapus.Enabled:=False;

with dm.Qbarang do

begin

close;

sql.Clear;

sql.Add('SELECT barang.kd\_brg, barang.nama\_brg, kategori.kategori, satuan.satuan, barang.stok, barang.harga\_beli, barang.harga\_jual, barang.kd\_satuan, barang.kd\_kategori ');

sql.Add('FROM barang INNER JOIN ');

sql.Add(' kategori ON barang.kd\_kategori = kategori.kd\_kategori INNER JOIN ');

sql.Add(' satuan ON barang.kd\_satuan = satuan.kd\_satuan');

open;

end;

end;

procedure TFBarang.bthrefreshClick(Sender: TObject);

begin

FormShow(sender);

end;

procedure TFBarang.brnkeluarClick(Sender: TObject);

begin

close;

end;

procedure TFBarang.dblok\_katCloseUp(Sender: TObject);

var Kd,a :string;

begin

{

if (dblok\_kat.KeyValue='') then

edkd\_brg.Text:='' else

begin

with dm.Q1 do

begin

close;

sql.Clear;

sql.Add('select kd\_kategori as kat from kategori where kd\_kategori='+QuotedStr(dblok\_kat.KeyValue));

open;

end;

a:=dm.Q1.FieldValues['kat'];

with dm.Qbarang do

begin

SQL.Clear;

SQL.Add('select \* from barang where kd\_kategori='+QuotedStr(dblok\_kat.KeyValue));

ExecSQL;

Open;

if dm.Qbarang.RecordCount>0 then

begin

dm.Qumum.Close;

dm.Qumum.SQL.Clear;

dm.Qumum.SQL.Add('select max(right(kd\_brg,5)) as Kode from barang where kd\_kategori='+QuotedStr(dblok\_kat.KeyValue));

dm.Qumum.Open;

Kd := FormatFloat('00000',dm.Qumum['Kode']+1);

edkd\_brg.Text := a+'-'+Kd;

end else

begin

edkd\_brg.Text := a+'-'+'00001';

end;

end;

end;

}

end;

procedure TFBarang.btnsimpanClick(Sender: TObject);

begin

if (edkd\_brg.Text='') or (edthargabeli.Text='') or (edthargajual.Text='') or (ednama\_brg.Text='') or (dblok\_kat.KeyValue=null) or (edstok.Text='') or (dblkcbbsatuan.KeyValue=Null) then

ShowMessage('Data Kurang Lengkap!') else

begin

with dm.Qbarang do

begin

sql.Clear;

SQL.Add('select \* from barang where kd\_brg='+quotedstr(edkd\_brg.text)+'');

open;

end;

if dm.Qbarang.Recordset.RecordCount > 0 then

ShowMessage('Data sudah ada silahkan isi yang lain!!')

else

begin

a:= StrToCurr(StringReplace(edthargabeli.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

b:= StrToCurr(StringReplace(edthargajual.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('insert into barang values('+quotedstr(edkd\_brg.Text)+','

+quotedstr(ednama\_brg.Text)+','

+quotedstr(dblkcbbsatuan.KeyValue)+','

+quotedstr(dblok\_kat.KeyValue)+','

+(edstok.Text)+','

+(CurrToStr(a))+','

+(CurrToStr(b))+')');

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Barang Berhasil Disimpan');

except dm.Koneksi.RollbackTrans

end;

dm.Qbarang.Close;

dm.Qbarang.Open;

FormShow(sender);

end;

end;

end;

procedure TFBarang.btnubahClick(Sender: TObject);

begin

if (edkd\_brg.Text='') or (edthargabeli.Text='') or (edthargajual.Text='') or (ednama\_brg.Text='') or (dblok\_kat.KeyValue=null) or (edstok.Text='') or (dblkcbbsatuan.KeyValue=null) then

ShowMessage('Data Kurang Lengkap!') else

begin

a:= StrToCurr(StringReplace(edthargabeli.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

b:= StrToCurr(StringReplace(edthargajual.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('update barang set');

sql.Add('nama\_brg = '+quotedstr(ednama\_brg.Text)+',');

sql.Add('kd\_satuan = '+quotedstr(dblkcbbsatuan.KeyValue)+',');

sql.Add('kd\_kategori = '+quotedstr(dblok\_kat.KeyValue)+',');

sql.Add('harga\_beli = '+CurrToStr(a)+',');

sql.Add('harga\_jual = '+CurrToStr(b)+',');

sql.Add('stok = '+(edstok.Text));

SQL.Add('where kd\_brg ='+quotedstr(edkd\_brg.Text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Diubah');

except dm.Koneksi.RollbackTrans

end;

dm.Qbarang.Close;

dm.Qbarang.Open;

FormShow(sender);

end;

end;

procedure TFBarang.FormKeyPress(Sender: TObject; var Key: Char);

begin

if Key = #13 then

begin

Key := #0; PostMessage(Handle, WM\_NEXTDLGCTL, 0, 0);

end;

end;

procedure TFBarang.FormCreate(Sender: TObject);

begin

keyPreview := true;

end;

procedure TFBarang.edstokKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in [#8,'0'..'9']) then

Key:=#0;

end;

procedure TFBarang.btnBtnhapusClick(Sender: TObject);

begin

if MessageDlg('Yakin Mau Menghapus Data Ini????',mtConfirmation,

[mbYes,mbCancel],0)=mrYes then

begin

try dm.Koneksi.BeginTrans;

with dm.Qumum do

begin

close;

SQL.Clear;

sql.Add('delete from barang');

SQL.Add('where kd\_brg='+QuotedStr(edkd\_brg.Text));

execsql;

end;

dm.Koneksi.CommitTrans;

ShowMessage('Data Berhasil Dihapus');

except dm.Koneksi.RollbackTrans

end;

dm.Qbarang.Close;

dm.Qbarang.Open;

FormShow(sender);

end

else

FormShow(sender);

end;

procedure TFBarang.dbgrd1CellClick(Column: TColumn);

begin

btnsimpan.Enabled:=false;

btnubah.Enabled:=true;

btnBtnhapus.Enabled:=True;

with DM.Qbarang do

begin

edkd\_brg.Text:=FieldValues['kd\_brg'];

ednama\_brg.Text:=FieldValues['nama\_brg'];

dblok\_kat.KeyValue:=FieldValues['kd\_kategori'];

dblkcbbsatuan.KeyValue:=FieldValues['kd\_satuan'];

edstok.Text:=FieldValues['stok'];

edthargabeli.Text:=FormatCurr('#,###',FieldValues['harga\_beli']);

edthargajual.Text:=FormatCurr('#,###',FieldValues['harga\_jual']);

end;

end;

procedure TFBarang.ednama\_brgKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

dblok\_kat.SetFocus;

end;

procedure TFBarang.dblok\_katKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

dblkcbbsatuan.SetFocus;

end;

procedure TFBarang.edkd\_brgKeyPress(Sender: TObject; var Key: Char);

begin

if (key=#13) then

ednama\_brg.SetFocus;

end;

procedure TFBarang.edthargabeliKeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if edthargabeli.Text='' then

begin

edthargabeli.Text:='';

end else

begin

//ribuan --> currency ( menyesuaikan setting windows )

sRupiah := edthargabeli.Text;

sRupiah := StringReplace(sRupiah,',','',[rfReplaceAll,rfIgnoreCase]); // hilangkan char koma , pemisah //ribuan selain IDR

sRupiah := StringReplace(sRupiah,'.','',[rfReplaceAll,rfIgnoreCase]); //remove char titik . pemisah //ribuan IDR

iRupiah := StrToCurrDef(sRupiah,0); // convert srupiah ke currency

//currency --> format ribuan

edthargabeli.Text := FormatCurr('#,###',iRupiah);

edthargabeli.SelStart := length(edthargabeli.text);

//menghitung harga jual

a:= StrToCurr(StringReplace(edthargabeli.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

end;

end;

procedure TFBarang.edthargabeliKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in [#8,'0'..'9']) then

Key:=#0 else

end;

procedure TFBarang.edthargajualKeyPress(Sender: TObject; var Key: Char);

begin

if not (key in [#8,'0'..'9']) then

Key:=#0 else

end;

procedure TFBarang.edthargajualKeyUp(Sender: TObject; var Key: Word;

Shift: TShiftState);

begin

if edthargajual.Text='' then

begin

edthargajual.Text:='';

end else

begin

//ribuan --> currency ( menyesuaikan setting windows )

sRupiah := edthargajual.Text;

sRupiah := StringReplace(sRupiah,',','',[rfReplaceAll,rfIgnoreCase]); // hilangkan char koma , pemisah //ribuan selain IDR

sRupiah := StringReplace(sRupiah,'.','',[rfReplaceAll,rfIgnoreCase]); //remove char titik . pemisah //ribuan IDR

iRupiah := StrToCurrDef(sRupiah,0); // convert srupiah ke currency

//currency --> format ribuan

edthargajual.Text := FormatCurr('#,###',iRupiah);

edthargajual.SelStart := length(edthargajual.text);

//menghitung harga jual

b:= StrToCurr(StringReplace(edthargajual.Text,'.','',[rfReplaceAll,rfIgnoreCase]));

end;

end;

end.