

**PENGEMBANGAN SISTEM INFORMASI PENJUALAN  
SEMBAKO PADA CV. SURYA MALANG BERBASIS  
CLIENT SERVER**

**SKRIPSI**



**Disusun Oleh :**

**GUNAWAN SASMITO ADI WIBOWO**

**NIM : 06.12.574**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2011**

3043

UNITED STATES DEPARTMENT OF JUSTICE  
FEDERAL BUREAU OF INVESTIGATION  
COMMUNICATIONS SECTION  
WASHINGTON, D. C.

NOV 1 08 1954  
COMMUNICATIONS SECTION  
WASHINGTON, D. C.

SECRET

COMMUNICATIONS SECTION  
UNITED STATES DEPARTMENT OF JUSTICE  
WASHINGTON, D. C.

**LEMBAR PERSETUJUAN**

**PENGEMBANGAN SISTEM INFORMASI PENJUALAN  
SEMBAKO PADA CV. SURYA MALANG  
BERBASIS CLIENT SERVER**

**SKRIPSI**

**Disusun dan Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh  
Gelar Sarjana Teknik Elektro Strata Satu (S-1)**

**Disusun Oleh :**

**GUNAWAN SASMITO ADI WIBOWO  
NIM : 06.12.574**



**Mengetahui  
Ketua Jurusan Teknik Elektro S-1**

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



**Diperiksa dan Disetujui**

**Dosen Pembimbing I**

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

**Dosen Pembimbing II**

**A. Faisal, ST  
NIP.P.1031000431**

**JURUSAN TEKNIK ELEKTRO S-1  
KONSENTRASI TEKNIK KOMPUTER & INFORMATIKA  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI NASIONAL MALANG  
2011**

GUNAWAN THANKS TO:

ALLAH SWT & ROSULNYA (MUHAMMAD SAW) ATAS RAHMAT DAN KARUNIA YANG TELAH DIBERIKAN KEPADAKU. DAN SEMUA YANG UDAH DOAIN, KASIH SEMANGAT, MOTIVASI DAN DUKUNGANNYA.

MY FAMILY

TERUTAMA BUAT BAPAK, IBU, KAKAKQ BUAT DOA, DUKUNGAN (MORIL N MATERIL), SEMANGAT, POKOKX SEMUANYA DEH. THANKS A LOT....!!!!

BUAT KELUARGA LEK HERI...TERIMAKASIH BANYAK ATAS MOTIVASI DAN MAKAN SIANGNYA YANG ENAK2,,!!!

MAKSIH BUAT MEGARIA FIRLANASARI YANG UD MEMBERIKAN MOTIVASI DAN DUKUNGANNYA SELAMA INI. AYO KMU JUGA SEMNAGT BIAR CEPET LULUS JUGA...HEEHEE!!!

MY FRIEND

TEMAN2 PANGESTU (KAPAN NGE BAND MANEH??? GORO'ANKU WES GATEL...HAHAHAH..).

TIM EMPIRE : AYO GOLEK MUSUH N NDANG DI SEBUL TROMPET'E REK...N TERIMAKASIH BUAT PERANG YANG TIADA HENTI-HENTINYANYA DAN KOMPAK SELALU UNTUK TIM KITA.

BWT TEMEN2 K211..AYO REMI MANEH REK..KAPAN FUTSAL MENEH...!!!!

BWT TEMEN SEPERJUANGAN YANG MASIH NGERJAIN SKRIPSI TETEP SEMANGAT N HAJAR TERUSS....!!!!

## ABSTRAKSI

### PENGEMBANGAN SISTEM INFORMASI PENJUALAN SEMBAKO PADA CV. SURYA MALANG BERBASIS CLIENT SERVER

**Nama : Gunawan Sasmito Adi Wibowo**

**Dosen Pembimbing : M. Ibrahim Ashari, ST, MT dan A. Faisol, ST**

Melalui Sistem Informasi Manajemen berbasis teknologi informasi pekerjaan menjadi mudah, efektif dan efisien. Sistem Informasi itu sendiri terdiri dari dua kata yaitu Sistem dan Informasi. Sistem sendiri berarti gabungan dari beberapa sub sistem yang bertujuan untuk mencapai satu tujuan. Informasi berarti sesuatu yang mudah dipahami oleh si penerima. Sistem Informasi memiliki makna sistem yang bertujuan menampilkan informasi.

Sebagai toko yang sedang berkembang, tentunya CV. SURYA harus memiliki Sistem Informasi guna mempercepat proses transaksi baik dengan *customer* maupun *supplier*. Namun transaksi-transaksi yang dilakukan seperti penjualan atau pembelian barang masih bersifat manual, ini yang menjadi kendala dimana disisi lain transaksi harus segera diproses dan laporannya bisa tersimpan dengan baik dan aman.

Oleh karena itu untuk memenuhi kebutuhan transaksi yang cepat dan aman tersebut, maka penulis mengusulkan sistem informasi Toko Sembako Pada CV. SURYA berbasis *Client Server* yang dapat memberikan kemudahan kepada customer, supplier, pemilik dan karyawan untuk melakukan proses transaksi.

**Kata Kunci:** Sistem Informasi, *client server*.

## KATA PENGANTAR

Alhamdulillah, puji syukur kehadiratMu Ya Allah yang telah memberikan Rahmat dan HidayahNya, sehingga penulis dapat menyelesaikan skripsi dengan judul **“Pengembangan Sistem Informasi Penjualan Sembako pada CV. Surya Malang Berbasis client Server”** dengan lancar. Skripsi merupakan persyaratan kelulusan Studi di Jurusan Teknik Elektro S-1 Konsentrasi Teknik Komputer dan Informatikan ITN Malang dan untuk mencapai gelar Sarjana Teknik.

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

1. Allah SWT yang selalu melimpahkan segala rahmatNya.
2. Bapak Prof. Dr. Ir. Abraham Lomi, MSEE selaku Rektor Institut Teknologi Nasional Malang.
3. Bapak Ir. Sidik Noetjahjono, MT selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang
4. Bapak Yusuf Ismail Nahkoda ST, MT selaku Ketua Jurusan Teknik Elektro S-1
5. Bapak Eng Aryuanto ST, MT selaku Sekretaris Jurusan Teknik Elektro S-1.
6. Bapak M.Ibrahim Ashari, ST.MT selaku Dosen Pembimbing 1.
7. Bapak A.Faisol, ST selaku Dosen Pembimbing II.
8. Ayah dan Ibu yang selalu memberikan do'a restu, dorongan dan semangat.
9. Kakak, Adik dan saudara-saudara yang senantiasa memberikan dorongan dan semangat.
10. Teman-teman dan semua yang telah membantu dalam penyelesaian penyusunan skripsi ini.

Penulis telah berusaha semaksimal mungkin dan menyadari sepenuhnya akan keterbatasan pengetahuan dalam menyelesaikan laporan ini. Untuk itu penyusun mengharapkan saran dan kritik yang membangun dari pembaca demi kesempurnaan laporan ini.

Harapan penulis semoga laporan skripsi ini memberikan manfaat bagi perkembangan ilmu pengetahuan bangsa dan negara.

Malang, Februari 2011

**Penulis**

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
KATA PENGANTAR.....	ii
LEMBAR PERSEMBAHAN.....	iii
ABSTRAKSI.....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
BAB I : PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	2
1.3. Tujuan Penelitian.....	2
1.4. Batasan Masalah.....	2
1.5. Metode Penelitian.....	3
1.6. Sistematika Penulisan.....	4
BAB II : LANDASAN TEORI.....	5
2.1. Pengertian Sistem, Informasi dan Sistem Informasi... 5	5
2.1.1. Pengertian Sistem.....	5
2.1.2. Pengertian Informasi.....	6
2.1.3. Pengertian Sistem Informasi.....	9
2.1.4. Pengertian Sistem Informasi Manajemen.....	10
2.2. Hierarki Data.....	11

2.3. Pengertian Database.....	12
2.3.1. Pengertian Umum Database.....	12
2.3.2. Pengertian Database Relasional.....	14
2.3.3. Tujuan Sistem Database.....	17
2.3.4. Normalisasi.....	19
2.4. Dokumentasi Sistem.....	21
2.4.1. Data Flow Diagram (DFD).....	21
2.4.2. Entity Relationship Diagram (ERD).....	22
2.5. Metode Pengembangan Sistem Informasi.....	22
2.5.1. System Development Life Cycle (SDLC).....	23
2.5.2. Metode Information Engineering.....	26
2.6. Software Yang Digunakan.....	30
2.6.1. Borland Delphi 7.0.....	30
2.6.2. Borland Interbase Delphi 7.0.....	30
2.7. Client Server.....	33
2.7.1. Arsitektur Client Server.....	34
2.7.2. Metode Akses Data.....	36
2.7.3. Komponen Perangkat Lunak Sistem Client Server.....	36
2.7.4. Keuntungan Sistem Informasi Berbasis Client Server.....	37
2.7.5. Kelemahan Sistem Informasi	



Client Server.....	38
<b>BAB III : ANALISIS DAN PERANCANGAN SISTEM.....</b>	<b>39</b>
3.1. Analisis Sistem.....	39
3.2. Perancangan Sistem Database.....	41
3.2.1. Pembuatan Data Flow Diagram (DFD).....	41
3.2.1.1. Context Diagram.....	41
3.2.1.2. DFD Level 0.....	42
3.2.1.3. DFD Level 1.....	43
3.2.1.4. DFD Level 2 Proses Penjualan.....	44
3.2.1.5. DFD Level 2 Proses Retur.....	46
3.2.1.6. DFD Level 2 Proses Pembelian.....	48
3.2.1.7. DFD Level 2 Retur Pembelian.....	49
3.2.2. Pembuatan Diagram Hubungan Entitas (ERD). ..	50
3.2.2.1. Gambar Diagram Hubungan Entitas (ERD).....	59
3.2.2.1.1. Conceptual ERD.....	59
3.2.2.1.2. Detailed ERD.....	60
<b>BAB IV : IMPLEMENTASI DAN PENGUJIAN SISTEM.....</b>	<b>61</b>
4.1. Form Login.....	61
4.2. Form Utama.....	62
4.3. Form Level User.....	62
4.4. Menu Master.....	63
4.4.1. Form Data Barang.....	63

4.4.2. Form Data Supplier.....	64
4.4.3. Form Data Customer.....	64
4.5. Menu Pembelian.....	65
4.5.1. Form Barang.....	65
4.5.2. Form Supplier.....	66
4.5.3. Form Pembelian.....	68
4.5.4. Form Retur Pembelian.....	70
4.6. Menu Penjualan.....	71
4.6.1. Form Penjualan.....	71
4.6.2. Form Customer.....	72
4.6.3. Form Laporan Rugi Laba.....	73
4.6.4. Form Bayar Piutang.....	74
BAB V : PENUTUP.....	76
5.1. Kesimpulan.....	76
5.2. Saran.....	76
DAFTAR PUSTAKA.....	77
LAMPIRAN.....	79

## DAFTAR TABEL

3.8. Tabel User name.....	51
3.9. Tabel Customer.....	51
3.10. Tabel Supplier.....	52
3.11. Tabel Barang.....	53
3.12. Tabel Pembelian Master.....	53
3.13. Tabel Pembelian Detail.....	54
3.14. Tabel Piutang.....	54
3.15. Tabel Penjualan Master.....	55
3.16. Tabel Penjualan Detail.....	55
3.17. Tabel Lanjutan Tabel Penjualan Detail.....	56
3.18. Tabel Retur Penjualan Master.....	56
3.19. Tabel Retur Penjualan Detail.....	57
3.20. Tabel Retur Pembelian Master.....	57
3.21. Tabel Retur Pembelian Detail.....	58

## DAFTAR GAMBAR

2.1. Model Umum Sebuah Sistem.....	51
2.2. Hubungan Data dan Informasi.....	51
2.3. Simbol-simbol Dalam DFD.....	52
2.4. IB Console.....	31
2.5. Arsitektur Model 2 tier.....	35
2.6. Arsitektur Client server.....	35
3.1. Context Diagram.....	42
3.2. DFD Level 0.....	43
3.3. DFD level 1.....	44
3.4. DFD Level 2 Proses penjualan.....	46
3.5. DFD level 2 Retur Penjualan.....	47
3.6. DFD Level 2 Proses Pembelian.....	49
3.7. DFD level 2 Retur Pembelian.....	50
3.8. Gambar Tabel User Name.....	51
3.9. Gambar Tabel Customer.....	51
3.10. Gambar Tabel Supplier.....	52
3.11. Gambar Tabel Barang.....	53
3.12. Gambar Tabel Pembelian Master.....	53
3.13. Gambar Tabel Pembelian Detail.....	54
3.14. Gambar Tabel Piutang.....	54
3.15. Gambar Tabel Penjualan Master.....	55

3.16. Gambar Tabel Penjualan Detail.....	55
3.17. Gambar Tabel Lanjutan Tabel Penjualan Detail...	56
3.18. Gambar Tabel Retur Penjualan Master.....	56
3.19. Gambar Tabel Retur Penjualan Detail.....	57
3.20. Gambar Tabel Retur Pembelian Master.....	57
3.21. Gambar Tabel Retur Pembelian Detail.....	58
3.22. Conceptual ERD.....	59
3.21. Detail ERD.....	60
4.1. Tampilan form login User.....	61
4.2. Tampilan Form Nama User dan Password Salah....	61
4.3. Tampilan Form Utama.....	62
4.4. Tampilan Form Set Login.....	63
4.5. Tampilan Form Laporan data barang.....	63
4.6. Tampilan Form laporan Data Supplier.....	64
4.7. Tampilan Form Data Customer.....	64
4.8. Tampilan Form Barang Pada Menu Isi Data.....	65
4.9. Tampilan Form Barang Pada Menu Laporan.....	66
4.10. Tampilan Form Supplier Pada Menu Isi Data.....	67
4.11. Tampilan Form Suppler Pada Menu Laporan Data.	67
4.12. Tampilan Form Pembelian Pada Menu Isi Data....	68
4.13. Tampilan Form Pembelian Pada Menu Laporan Data.....	69
4.14. Tampilan Cetak Struk Pembelian.....	69

<b>4.15. Tampilan Form Retur Pembelian Pada</b>	
Menu Update.....	70
<b>4.16. Tampilan Form Retur Pembelian Pada Menu Read.</b>	<b>70</b>
<b>4.17. Tampilan Form Penjualan.....</b>	<b>71</b>
<b>4.18. Tampilan Cetak Struk Penjualan.....</b>	<b>71</b>
<b>4.19. Tampilan Customer Pada Menu Isi Data.....</b>	<b>72</b>
<b>4.20. Tampilan Form Customer Pada Menu Laporan Data</b>	<b>73</b>
<b>4.21. Tampilan Form Laporan Rugi Laba.....</b>	<b>73</b>
<b>4.22. Tampilan Form Daftar Piutang.....</b>	<b>74</b>
<b>4.23. Tampilan Form Bayar Piutang.....</b>	<b>75</b>
<b>4.24. Tampilan Form Jatuh Tempo.....</b>	<b>75</b>

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan ilmu pengetahuan dan teknologi kian pesat, hal tersebut dapat kita lihat dan kita rasakan di sekeliling kita secara langsung maupun tidak langsung. Perkembangan tersebut tengah berdampak pada segala aspek kehidupan manusia. Salah satu jenis perkembangan ilmu pengetahuan dan teknologi adalah perkembangan dunia komputasi, satu diantaranya adalah kemajuan Sistem Informasi Manajemen. Hampir tidak ada batas ruang dan waktu sehubungan dengan Sistem Informasi Manajemen tersebut, Informasi dari tempat yang jauh secara fisik dapat dengan cepat dan mudah diketahui oleh kita. Melalui Sistem Informasi Manajemen berbasis teknologi informasi pekerjaan menjadi mudah, efektif dan efisien. Sistem Informasi itu sendiri terdiri dari dua kata yaitu Sistem dan Informasi. Sistem sendiri berarti gabungan dari beberapa sub sistem yang bertujuan untuk mencapai satu tujuan. Informasi berarti sesuatu yang mudah dipahami oleh si penerima. Sistem Informasi memiliki makna sistem yang bertujuan menampilkan informasi.

Sebagai toko yang sedang berkembang, tentunya CV. SURYA harus memiliki Sistem Informasi guna mempercepat proses transaksi baik dengan *customer* maupun *supplier*. Namun transaksi-transaksi yang dilakukan seperti penjualan atau pembelian barang masih bersifat manual, ini yang menjadi kendala dimana disini lain transaksi harus segera diproses dan laporannya bisa terekap dengan baik dan aman.

Oleh karena itu untuk memenuhi kebutuhan transaksi yang cepat dan aman tersebut, maka penulis mengusulkan sistem informasi Toko Sembako Pada CV. SURYA Malang berbasis *Client Server* yang dapat memberikan kemudahan kepada *customer*, *supplier*, pemilik dan karyawan untuk melakukan proses transaksi. Sistem informasi yang diusulkan dapat menangani transaksi-transaksi yang berkaitan dengan kegiatan pada toko seperti: pengelolaan sirkulasi barang/stok (pencarian, penambahan, penjualan, pemesanan) rugi laba dan pelaporan (hasil penjualan, hasil pembelian stok, daftar pembeli dan masih banyak lagi ). Oleh karena itu, permasalahan tersebut

dianggap representative untuk diangkat menjadi judul Tugas Akhir sebagai berikut :  
“ Pengembangan Sistem Informasi Penjualan Sembako Pada CV. SURYA Malang berbasis *client server* ”.

Terdapat beberapa manfaat yang dapat diraih dalam pengembangan sistem informasi *Client Server* ini, yakni :

- Meningkatkan Efektivitas dan Efisiensi Pengelolaan Toko
- Membuat Sistem Informasi yang berbasiskan *client server* yang Fleksibel
- Merangsang Pertumbuhan dan Perkembangan Toko

## 1.2. Rumusan Masalah

Rumusan masalah yang dapat diambil dari pembuatan tugas akhir ini adalah :

1. Bagaimana mengembangkan Aplikasi Sistem Informasi Penjualan Sembako berbasis *client server* menggunakan Borland Delphi 7 dan Interbase sebagai Databasenya.
2. Bagaimana cara mengatur sistem (merubah data, menghapus data, ataupun menambahkan data) di dalam Sistem Informasi.
3. Bagaimana cara menyajikan informasi tersebut secara optimal.

## 1.3. Tujuan Penelitian

Tugas Akhir ini bertujuan untuk membangun sebuah Sistem Informasi berbasis *client server*, sehingga kita bisa tukar-menukar file dalam suatu network yang mensupport TCP/IP protocol dan memungkinkan kita bisa memanfaatkan jaringan untuk mengendalikan komputer lain melalui komputer kita.

## 1.4. Batasan Masalah

Batasan masalah yang diambil pada penulisan skripsi ini diharapkan mampu membatasi pembahasan agar sesuai dengan tujuan penelitian itu sendiri. Adapun batasan masalah yang diajukan adalah sebagai berikut :

1. Pengembangan Aplikasi Sistem Informasi Penjualan sembako pada CV. SURYA Malang berbasis *client server* menggunakan model *two tier*.



2. Bahasa pemrograman yang digunakan adalah Delphi 7 dan Interbase sebagai Databasenya.
3. Yang mendapatkan hak akses pada aplikasi ini hanya *administrator*, pegawai dan pemilik.

### 1.5. Metode Penelitian

Adapun metode penelitian yang digunakan adalah sebagai berikut:

#### 1. Studi literatur

Pengumpulan data yang dilakukan dengan mencari bahan-bahan kepustakaan dan referensi dari berbagai sumber sebagai landasan teori yang ada hubungannya dengan permasalahan yang dijadikan objek penelitian.

#### 2. Analisa Kebutuhan Sistem

Data dan informasi yang telah diperoleh akan dianalisa agar didapatkan kerangka global yang bertujuan untuk mendefinisikan kebutuhan sistem baik *hardware* maupun *software*, dimana nantinya akan digunakan sebagai acuan perancangan sistem.

#### 3. Perancangan sistem

Berdasarkan data dan informasi yang telah diperoleh serta analisa kebutuhan untuk membangun sistem ini, akan dibuat rancangan kerangka global yang menggambarkan mekanisme dari sistem yang akan dibuat.

#### 4. Coding

Tahapan ini menerjemahkan hasil perancangan spesifikasi program dari tahapan sebelumnya kedalam baris-baris kode program yang dapat dimengerti oleh komputer.

#### 5. Eksperimen dan Evaluasi

Pada tahap ini, sistem yang telah selesai dibuat akan diuji coba, yaitu pengujian berdasarkan fungsionalitas program, dan akan dilakukan koreksi dan penyempurnaan program jika diperlukan.

## **1.6. Sistematika Penulisan**

Untuk mempermudah dan memahami pembahasan penulisan skripsi ini, maka sistematika penulisan disusun sebagai berikut :

### **Bab I : Pendahuluan**

Berisi Latar Belakang, Rumusan Masalah, Tujuan Penelitian, Pembatasan Permasalahan, Metode Penelitian dan Sistematika Penulisan.

### **Bab II : Tinjauan Pustaka**

Berisi tentang landasan teori mengenai permasalahan yang berhubungan dengan penelitian yang dilakukan.

### **Bab III : Perancangan dan Analisa Sistem**

Bab ini menjelaskan tentang perancangan umum maupun uraian lebih lanjut mengenai perancangan sistem dalam pembuatan perangkat lunak. Uraian perancangan sistem ini meliputi perancangan data mengenai data input dan output sistem, perancangan proses mengenai bagaimana sistem akan bekerja dengan proses-proses tertentu, maupun perancangan antar muka dalam desain dan implementasi yang akan digunakan dalam pembuatan Tugas Akhir ini.

### **Bab IV: Pembuatan dan Pengujian Sistem**

Berisi tentang implementasi dari perancangan sistem yang telah dibuat serta pengujian terhadap sistem tersebut.

### **Bab V : Penutup**

Merupakan bab terakhir yang memuat intisari dari hasil pembahasan yang berisikan kesimpulan dan saran yang dapat digunakan sebagai pertimbangan untuk pengembangan penulisan selanjutnya.

## **BAB II**

### **LANDASAN TEORI**

Bab ini merupakan landasan teori yang akan mendasari penulisan Tugas Akhir. Landasan teori diperlukan agar pembahasan Tugas Akhir ini nantinya memiliki dasar teori yang kuat, sehingga implementasinya dapat dipertanggungjawabkan. Teori-teori yang dibahas berkaitan dengan masalah yang akan dipecahkan antara lain :

Pengertian Sistem, Informasi, dan Sistem Informasi, Konsep dasar *Database*, Pengertian *Database*, Dokumentasi Sistem, Metode pembuatan Sistem Informasi, *software* yang digunakan, *Client Server*.

#### **2.1 Pengertian Sistem, Informasi dan Sistem Informasi**

##### **2.1.1 Pengertian Sistem <sup>[1]</sup>**

Pengertian Sistem adalah :

*A group of interrelated components working together toward a common goal by accepting inputs and producing outputs in an organized transformation process.*

Yang artinya apabila diterjemahkan kurang lebih :

*Sistem adalah kumpulan dari beberapa komponen yang saling berhubungan dan bekerja sama untuk mencapai suatu tujuan tertentu dengan menerima masukan dan menghasilkan keluaran dalam suatu proses perubahan yang terstruktur.*

Sistem memiliki dua pengertian utama<sup>[2]</sup>, yaitu sebagai suatu kesatuan dari beberapa elemen atau subsistem dan sebagai suatu prosedur untuk mencapai tujuan. Jadi menurutnya pendekatan system adalah :

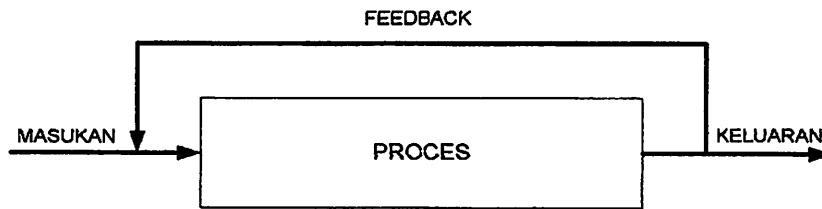
*Komponen-komponen atau subsistem-subsistem yang saling berinteraksi, dimana masing-masing bagian tersebut dapat bekerja sendiri-sendiri (independen) dan bersama-sama serta saling berhubungan membentuk satu kesatuan sehingga tujuan atau sasaran system tersebut tercapai secara keseluruhan dalam lingkungan yang kompleks.*

---

<sup>1</sup> James A. O'Brien, *Management Information System – Managing Information Technology in the Business Enterprise*, McGraw-Hill, Inc., 2004, halaman 8.

<sup>2</sup> Hasta Dewa Putranta ST, MM, *Pengantar Sistem dan Teknologi Informasi*, Amus Yogyakarta, 2004, halaman 15.

Dari definisi diatas maka dapat diketahui manfaat system yaitu untuk menyatukan atau mengintegrasikan semua unsure yang ada dalam suatu lingkup, sehingga setiap informasi yang ada akan dapat dimanfaatkan oleh banyak pihak yang ada dalam lingkup tersebut dan mempermudah pencapaian tujuan yang diinginkan.



Gambar 2.1 Model umum sebuah system

### 2.1.2 Pengertian Informasi

Informasi sangat berhubungan dengan data. Oleh karena itu sebelum dijelaskan pengertian informasi, ada baiknya diketahui definisi data terlebih dahulu. Pengertian data menurut L. Whitten, Lonnie D. Bentley, dan Kevin C. Dittman adalah :

*Raw facts about people, places, event and things that are importance in an organization. Each fact is, by itself, relatively meaningless.*

Apabila diterjemahkan secara bebas artinya kurang lebih :

*Data adalah fakta-fakta yang masih mentah tentang orang-orang, tempat, kejadian, dan lain sebagainya, yang penting bagi organisasi. Akan tetapi setiap fakta tersebut biasanya tidak berarti (karena masih mentah).*

Menurut Jeffrey L. Whitten, Lonnie D. Bentley, dan Kevin C. Dittman<sup>[3]</sup>, definisi informasi adalah :

*Data that has been processed or reorganized into a more meaningful form for someone. Information is farmed from combinations of data that hopefully have meaning to the recipient.*

Yang apabila diterjemahkan artinya kurang lebih :

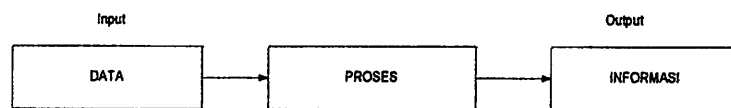
*Informasi adalah data yang telah diproses atau diatur ulan sehingga menjadi lebih berarti bagi seseorang. Informasi tersusun dari kombinasi-kombinasi data yang diharapkan dapat memberikan arti tertentu bagi penerimanya.*

<sup>3</sup>Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, *System Analysis and Design Methods Sixth Edition*, McGraw-Hill, Inc., 2004, halaman 27.

Sedangkan menurut Hasta Dewa Putranta ST., MM<sup>[4]</sup>, pengertian informasi adalah :

*Data yang telah diolah menjadi suatu hasil yang lebih berguna dan berarti bagi penerima informasi yang membutuhkan.*

Dari beberapa definisi di atas, dapat diambil kesimpulan bahwa informasi merupakan keluaran (*output*) dari suatu proses pengolahan data *input*. Informasi biasanya sudah tersusun dengan baik dan mempunyai arti bagi yang menerimanya sehingga dapat dijadikan dasar untuk pengambilan keputusan oleh manajemen.



Gambar 2.2 Hubungan Data dan Informasi

Karena informasi adalah suatu data yang disajikan dalam bentuk tertentu yang berguna untuk membuat keputusan, maka informasi harus memenuhi suatu standar kualitas tertentu. Kualitas tersebut antara lain :

- a. Kemengertian (*understandability*) membuat pemakai menyadari arti pentingnya informasi.
- b. Informasi dapat mempengaruhi pembuat keputusan, yaitu dengan menurunkan ketidakpastian atau menaikkan pengetahuan tentang hal yang mempunyai hubungan (*relevance*) dengan pembuat keputusan tersebut.
- c. Informasi yang mempunyai ketepatan waktu (*timeliness*) yaitu suatu informasi yang tersedia sebelum informasi tersebut kehilangan kemampuannya dalam hal mempengaruhi pembuatan keputusan.
- d. Nilai perkiraan (*predivtive value*) dan nilai umpan balik (*feedback value*) mempengaruhi pembuat keputusan untuk memperkirakan, memastikan, atau mengoreksi tujuan-tujuan atau keinginan-keinginannya.
- e. Informasi yang mempunyai kebenaran (*validity*) adalah suatu informasi yang mencerminkan kejadian-kejadian atau obyek-obyek yang sesungguhnya.

<sup>4</sup>Hasta Dewa Putranta ST., MM, *Pengantar Sistem dan Teknologi Informasi*, Amus Yogyakarta, 2004, halaman 22.

- f. Keakuratan (*accuracy*) adalah kecocokan antar informasi dengan kejadian-kejadian atau obyek-obyek yang diwakilinya.
- g. Kelengkapan (*completeness*) adalah derajat sampai seberapa jauh informasi menyediakan kejadian-kejadian atau obyek-obyek yang berhubungan.
- h. Informasi yang mempunyai kepastian yaitu informasi yang dapat diperoleh dengan menggunakan cara-cara pengukuran yang independent, tetapi menggunakan metode pengukuran yang sama.
- i. Kenetralan (*neutrality*) mempunyai arti bahwasannya suatu informasi tidak berat sebelah.
- j. Kemampuan untuk dapat membandingkan (*comparability*) adalah kualitas informasi yang memungkinkan pemakai untuk mengidentifikasi persamaan dan perbedaan diantara dua informasi.

Jika dapat membandingkan informasi dari kejadian-kejadian atau obyek-obyek yang sama yang dikumpulkan pada waktu yang berbeda-beda, informasi tersebut disebut sebagai informasi yang konsisten.

### 2.1.3 Pengertian Sistem Informasi

Sistem Informasi sangat berhubungan dengan Teknologi Informasi. Oleh karena itu perlu diketahui pula definisi Teknologi Informasi. Menurut Jeffrey L. Whitten, Lonnie D. Bentley, dan Kevin C. Dittman<sup>5</sup> Pengertian Teknologi Informasi adalah :

*A contemporary term that describes the combination of computer technology (hardware and software) with telecommunication technology (data, image, and voice network).*

Apabila diterjemahkan artinya kurang lebih :

*Teknologi Informasi adalah istilah kontemporer yang menggambarkan kombinasi antara teknologi computer (perangkat keras dan perangkat lunak) dengan teknologi telekomunikasi (data, gambar, dan jaringan suara).*

Sedangkan pengertian Sistem Informasi menurut Kenneth C. Laudon dan Jane P. Laudon<sup>6</sup> adalah :

*Satuan komponen yang saling berhubungan yang mengumpulkan ( atau mendapatkan Kembali), memproses, menyimpan, dan mendistribusikan informasi untuk mendukung pengambilan keputusan dan kendali dalam suatu organisasi.*

Setelah mempelajari dari definisi diatas dapat dikatakan bahwa system informasi adalah system manusia atau mesin yang terpadu untuk menyajikan informasi guna mendukung fungsi operasi dan pengambilan keputusan dalam organisasi. Adapun sebagian dari kegunaan sistem informasi yaitu :

- a. Menyajikan informasi guna mendukung pengambilan keputusan
- b. Menyajikan informasi guna mendukung operasi harian
- c. Menyajikan informasi yang berkenaan dengan kepengurusan

---

<sup>5</sup> Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, *System Analysis and Design Methods Sixth Edition*, McGraw-Hill, Inc., 2004, halaman 12.

<sup>6</sup> Kenneth C. Laudon dan Jane P. Laudon, *Sistem Informasi Manajemen, Mengelola Perusahaan Digital*, penerbit Andi Yogyakarta, 2005, halaman 9-10.

#### **2.1.4 Pengertian Sistem Informasi Manajemen**

Pengertian Sistem Informasi Manajemen (SIM) menurut Jeffrey L. Whitten, Lonnie D. Bentley, dan Kevin C. Dittman adalah :

*An Information system that provides for management-oriented reporting based on transaction processing and operations of the organization.*

Apabila diterjemahkan secara bebas kurang lebih :

*SIM adalah sistem informasi yang menyediakan pelaporan untuk level manajemen berdasarkan proses transaksi dan operasi dari suatu organisasi.*

Sedangkan pengertian SIM menurut Kenneth C. Laudon dan Jane P. Laudon<sup>7</sup> adalah :

*Sistem Informasi pada level manajerial organisasi yang melayani fungsi perencanaan, pengendalian, dan pengambilan keputusan dengan cara menyediakan rangkuman rutin dan laporan perkembangan.*

---

<sup>7</sup>Kenneth C. Laudon dan Jane P. Laudon, *Sistem Informasi Manajemen, Mengelola Perusahaan Digital*, penerbit Andi Yogyakarta, 2005, halaman 58.



## 2.2 Hierarkhi Data

Data harus disusun secara teratur agar pengolahannya dapat dilakukan dengan baik dan efisien. Menurut Budi Sutedjo Dharma Oetomo, S.Kom, MM<sup>8</sup>, hierarkhi data dapat dibagi menjadi :

1. **Database**, yaitu kumpulan *file* yang saling terkait satu sama lain, misalnya *file* data induk karyawan, *file* jabatan, dan lain sebagainya. Kumpulan *file* yang tidak saling terkait satu sama lain tidak dapat disebut *database*. Misalnya *file* induk karyawan dengan *file* tamu undangan.
2. **File**, adalah kumpulan dari record yang saling terkait dan memiliki format *field* yang sama dan sejenis.
3. **Record**, merupakan *field* yang menggambarkan suatu unit data individu tertentu. *Record* terdiri dari kumpulan item yang secara logis saling berhubungan. Setiap *record* dapat dikenali dengan menggunakan *field* kunci.
4. **Field**, adalah atribut dari *record* yang menunjukkan suatu item dari data. Misalnya *record* mahasiswa memiliki atribut NIP, nama, alamat, dan telepon.
5. **Byte**, adalah atribut dari *field* yang berupa huruf, yang membentuk nilai sebuah *field*. Huruf tersebut dapat berupa numeric maupun abjad atau karakter khusus.
6. **Bit**, adalah bagian terkecil dari data keseluruhan, berupa karakter ASCII nol atau satu yang merupakan komponen pembentuk *byte*.

---

<sup>8</sup>Budi Sutedjo Dharma Oetomo, S. Kom, MM, *Perencanaan dan Pembangunan Sistem Informasi*, Andi Yogyakarta, 2002, halaman 102.

## 2.3 Pengertian Database

### 2.3.1 Pengertian Umum Database

Suatu Sistem Informasi Manajemen menggambarkan ketersediaan suatu rangkaian data yang cukup lengkap yang disimpan agar dapat menyediakan informasi untuk mendukung operasi, manajemen, dan pembuatan keputusan dalam suatu organisasi. Media untuk penyimpanan rangkaian data tersebut adalah suatu *database*.

Pengertian *database* menurut Budi Sutedjo Dharma Oetomo, S. Kom, MM<sup>9</sup> adalah :

*Tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam system, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Database merupakan himpunan kelompok data yang saling berkaitan, dimana data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi.*

Sedangkan pengertian *database* menurut Thomas M. Connolly dan Carolyn E. Begg<sup>10</sup> adalah :

*A shared collection of logically related data, and a description of this data, designed to meet the information needs of an organization.*

Yang apabila diterjemahkan kurang\ng lebih artinya :

*Database adalah suatu kumpulan data yang terhubung secara logis, dan sekaligus deskripsinya, yang dirancang untuk memenuhi kebutuhan informasi sebuah organisasi.*

---

<sup>9</sup>Budi Sutedjo Dharma Oetomo, S. Kom, MM, *Perencanaan dan Pembangunan Sistem Informasi*, Andi Yogyakarta, 2002, halaman 99.

<sup>10</sup>Thomas M. Connolly and Carolyn E. Begg, *Database System A Practical Approach to Design, Implementation, and Management*, University of Paisley, 2001, halaman 14.

Menurut James A. O'Brien<sup>11</sup>, pengertian *database* adalah :

*An integrated collection of logically related data elements that consolidates records previously stored in separate files into a common pool of data elements that provides data for many applications. The data stored in a database are independent of application program using them and the type of storage devices on which they are stored.*

Apabila diterjemahkan artinya kurang lebih :

*Database adalah suatu kumpulan terintegrasi dari elemen-elemen data yang terhubung secara logis, dimana kumpulan tersebut menghubungkan record-record yang telah disimpan sebelumnya dalam file-file yang terpisah, ke dalam satu wadah yang akan menghasilkan data untuk digunakan oleh program aplikasi. Data yang disimpan dalam database tergantung dari program aplikasi yang menggunakannya dan tipe media penyimpanannya.*

Data dalam suatu *database* perlu diatur sedemikian rupa sehingga *database* dapat dikontrol, dijaga keamanannya, dan dimodifikasi agar dapat bekerja secara optimal. Oleh karena itu diperlukan perangkat lunak yang dikenal sebagai DBMS (*Database Management System*). Pengertian DBMS menurut Thomas M. Connolly dan Carolyn E. Begg<sup>12</sup> adalah :

*A software system that enables users to define, create, maintain, and control access to the database.*

Apabila diterjemahkan artinya kurang lebih :

*DBMS adalah suatu sistem perangkat lunak yang memungkinkan pengguna untuk mendefinisikan, membuat, merawat, dan mengontrol akses ke suatu database.*

---

<sup>11</sup> James A. O'Brien, *Management Information System – Managing Information Technology in the Business Enterprise*, McGraw-Hill, Inc., 2004, halaman 73.

<sup>12</sup> Thomas M. Connolly and Carolyn E. Begg, *Database System A Practical Approach to Design, Implementation, and Management*, University of Paisley, 2001, halaman 16.

DBMS memiliki beberapa fasilitas yang dapat mendukung pengorganisasian data dalam *database* agar sesuai dengan yang diharapkan. Fasilitas tersebut antara lain :

- Memungkinkan pengguna untuk mendefinisikan suatu *database* dengan menggunakan *Data Definition Language* (DDL). DDL memungkinkan pengguna untuk menentukan tipe, struktur, dan bagaimana data disimpan dalam suatu *database*.
- Memungkinkan pengguna untuk memasukkan, mengubah, menghapus, dan menampilkan data dari *database*. Hal tersebut biasanya dilakukan menggunakan *Data Manipulation Language* (DML). DML menyediakan fasilitas penyaringan data yang disebut bahasa *query* (*query language*). Tujuan utama bahasa *query* adalah untuk memudahkan penyaringan data yang kompleks, atau apabila ada pengembangan program di kemudian hari. Bahasa *query* yang paling umum digunakan adalah *Structured Query Language* (SQL), yang sekarang telah menjadi bahasa standar untuk DBMS relasional.
- Menyediakan pengaturan akses *database*. Misalnya DBMS dapat memberikan fasilitas pengamanan sistem yang dapat mencegah pengguna yang tidak berhak mengakses *database*, sistem integritas yang dapat menjaga konsistensi data, sistem pengontrol konkurensi yang memungkinkan akses secara bersamaan ke dalam suatu *database*, sistem recovery yang dapat mengembalikan *database* ke kondisi semula apabila terjadi kerusakan pada perangkat keras atau perangkat lunaknya, dan menyediakan katalog bagi pengguna yang berisi deskripsi data yang ada dalam *database*.

### 2.3.2 Pengertian Database Relasional

Pengertian *database* relasional menurut Mark Whitehorn dan Bill Marklyn<sup>13</sup> adalah :

*Suatu database yang menggunakan model relasional, dimana model tersebut menyediakan mekanisme-mekanisme yang kita perlukan untuk menyimpan dan memanipulasi data yang kompleks dengan cara yang sedemikian rupa sehingga kita*

dapat meng-query data itu nanti dan memperoleh jawaban-jawaban yang kita perlukan.

Ada beberapa komponen pokok dalam *database* relasional yaitu<sup>14</sup> :

- *Entity*, adalah objek-objek terpisah (orang, tempat, benda, konsep, atau peristiwa dalam suatu organisasi yang akan direpresentasikan ke dalam *database*.
- *Attribute*, adalah property yang menggambarkan aspek-aspek yang dimiliki suatu objek yang akan dicatat.
- *Relationship* (relasi), adalah hubungan- hubungan yang dapat terjadi antara dua *entity*.

Secara umum suatu *database* relasional dibuat untuk meniru dunia nyata. Dalam dunia nyata ada banyak objek yang saling berhubungan satu sama lain. Misalnya seorang karyawan toko akan berurusan dengan banyak pelanggan, satu pelanggan dapat membeli banyak item barang, dan seterusnya. Oleh karena itu *database* harus menyimpan informasi tentang objek-objek yang berbeda tersebut dan juga tentang hubungan antar objek-objeknya.

Setiap tabel dalam *database* akan mewakili satu jenis objek di dunia nyata. Untuk menentukan hubungan-hubungan apa saja yang perlu ditetapkan antar table, maka perlu diketahui jenis-jenis hubungan yang dapat terjadi antar objek dalam dunia nyata.

---

<sup>13</sup>Mark Whitehorn dan Bill Marklyn, *Seluk beluk Database Relasional*, Penerbit Erlangga, 2003, halaman 143.

<sup>14</sup>Thomas M. Connolly and Carolyn E. Begg, *Database System A Practical Approach to Design, Implementation, and Management*, University of Paisley, 2001, halaman 15

Ada empat jenis hubungan yang mungkin terjadi antara setiap pasangan objek, yaitu<sup>15</sup> :

- Satu ke banyak (*one to many*)

Hubungan satu ke banyak dapat dimisalkan seperti hubungan antara pelanggan dan pemesanan yang dapat mereka lakukan. Artinya seorang pelanggan mungkin saja tidak melakukan pemesanan sama sekali, melakukan hanya satu kali pemesanan, atau melakukan lebih dari satu kali pemesanan. Hubungan ini bersifat asimetris karena setiap pemesanan hanya akan dilakukan oleh satu pelanggan saja, akan tetapi setiap pelanggan dapat melakukan lebih dari satu pemesanan. Hubungan satu ke banyak sering terjadi dalam dunia bisnis.

- Satu ke satu (*one to one*)

Sebagai contoh dianggap bahwa setiap staf dalam suatu kantor harus memiliki ruangan tersendiri. Hubungan yang terjadi antara staf dengan ruangnya adalah hubungan satu ke satu, karena seorang staf hanya akan memiliki satu ruangan, dan satu ruangan hanya ditempati oleh seorang staf. Hubungan satu ke satu jarang dijumpai dalam merancang suatu *database*.

- Banyak ke banyak (*many to many*)

Contoh hubungan banyak ke banyak adalah hubungan antara karyawan toko dengan pelanggan. Seorang pelanggan bias dilayani oleh satu atau lebih karyawan dan seorang karyawan bias saja melayani banyak pelanggan. Hubungan banyak ke banyak sangat sering dijumpai.

- Tidak ada (*none*)

Pada kenyataannya ada beberapa objek yang memang tidak memiliki hubungan sama sekali satu dengan yang lainnya. Sebagai contoh informasi tentang sewa kantor tidak berhubungan sama sekali dengan informasi tentang pelanggan.

---

<sup>15</sup>Mark Whitehorn dan Bill Marklyn, *Seluk beluk Database Relasional*, Penerbit Erlangga, 2003, halaman 76-77.

Setelah mengetahui bentuk-bentuk relasi yang mungkin terjadi, maka relasi-relasi tersebut perlu dimodelkan dalam *database*. Untuk memodelkan relasi dalam *database* dapat digunakan *key* (*primary key* dan *foreign key*). *Primary key* adalah suatu pengenal unik dalam suatu table yang berfungsi untuk mengidentifikasi table tersebut. Setiap tabel dalam *database* relasional harus memiliki *primary key*. Suatu *Primary key* dapat terdiri atas satu atau lebih *field*. Tidak boleh ada nilai dalam suatu *primary key* yang kosong (*null*).

*Foreign key* adalah pengenal yang berisi nilai-nilai yang diambil dari *primary key* tabel yang berhubungan dengan tabel bersangkutan. *Foreign key* tidak harus ada dalam setiap tabel. Tetapi apabila terdapat suatu hubungan antara dua tabel, maka salah satu tabelnya akan memiliki *foreign key*, dan sebuah tabel sangat mungkin memiliki lebih dari satu *foreign key*. Jika sebuah tabel memiliki lebih dari satu *foreign key*, tabel tersebut harus memiliki lebih dari satu hubungan dengan tabel lainnya.

### 2.3.3 Tujuan Sistem Database

Suatu DBMS berisi koleksi data yang saling berhubungan dan suatu program untuk mengakses data tersebut. Jadi DBMS terdiri dari *database* dan program pengelola untuk menambah data, menghapus data, mengambil, membaca data dan modifikasi data.

Peran suatu sistem *database* dalam mendukung kegiatan Sistem Informasi Manajemen adalah sebagai suatu alat untuk mengadakan ketersediaan data lengkap dan menyediakan informasi-informasi untuk mendukung kegiatan manajemen dan pengambilan keputusan.

Tujuan dari manajemen *database* adalah sebagai berikut :

1. Menyediakan tempat menyimpan untuk data yang relevan.
2. Memudahkan pemakai dalam mengakses data.
3. Memungkinkan respon yang segera atas permintaan data dari pemakai.
4. Melakukan modifikasi terakhir dengan segera pada *database*.
5. Memungkinkan pemakaian secara serentak dan bersamaan yang berarti juga meningkatkan kebebasan data sehingga berguna untuk beberapa program.
6. Memungkinkan pengembangan lebih lanjut dalam system *database*.

7. Meminimalkan duplikasi dan redundansi dalam penyimpanan data.
8. Melindungi data dari gangguan kerusakan atau pemakaian oleh orang yang tidak berhak.

Untuk mendapatkan data yang optimal dan stabil, diperlukan kriteria sebagai berikut :

1. Kebenaran struktur (*consistency*)  
Konsistensi dengan jalur definisi dan informasi organisai.
2. Kemudahan  
Mudah untuk dimengerti oleh orang yang tidak mempunyai pendidikan khusus atau pemakai sistem yang bukan ahli.
3. Tidak ada *redundancy*  
Tidak ada informasi yang berlebihan atau ditulis berkali-kali, dengan demikian satu set informasi hanya disimpan di satu tempat yang benar.
4. Dapat dipakai bersama (*sharebility*)  
Kondisi data tidak spesifik dan hanya dapat dipaki oleh satu aplikasi, namun diharapkan dapat diakses oleh beberapa aplikasi, sesuai dengan kebutuhan.
5. Mudah dikembangkan (*extentability*)  
Mampu mengakomodasi kebutuhan baru dari suatu aplikasi dengan akibat yang ringan terhadap perubahan data dasarnya.
6. Kesatuan (*Integrity*)  
Konsistensi dengan jalur bisnis yang digunakan dan pengelolaan informasi.



### 2.3.4 Normalisasi

Normalisasi adalah alat yang digunakan untuk melakukan proses pengelompokan data menjadi tabel-tabel yang menunjukkan entitas dan relasinya<sup>16</sup>. Dalam proses normalisasi, persyaratan suatu tabel harus dipecah didasarkan adanya kesulitan pengorganisasian data misalnya untuk menambah atau menyisipkan, menghapus atau mengubah data, serta pembacaan data dari tabel tersebut. Apabila masih ada kesulitan, maka tabel tersebut harus dipecah lagi dan dilakukan proses normalisasi kembali sampai didapat tabel yang optimal.

Tujuan utama normalisasi adalah<sup>17</sup>:

- Mencari dan mengelompokkan seluruh sifat yang berhubungan dengan objek tertentu.
- Untuk menghilangkan informasi yang berlebihan.
- Untuk memberikan identifikasi unik bagi *record* individu.

Normalisasi memiliki beberapa tingkatan. Mulai dari tingkatan pertama, kedua, dan seterusnya. Akan tetapi untuk membuat suatu *database* yang optimal, pada umumnya hanya diperlukan sampai tingkat ke tiga saja. Tingkatan normalisasi tersebut antar lain :

- Bentuk normal pertama (1NF)

Bentuk 1NF pada dasarnya hanya uraian tabel dasar. Pada tahap ini suatu tabel berada dalam bentuk 1NF apabila nilai datanya hanya memiliki komponen tunggal. Artinya dalam tahap ini dibuat tabel-tabel yang menampung data dan dikelompokkan berdasarkan karakteristik tertentu, dan setiap tabel diusahakan tidak memiliki *field* yang berulang.

- Bentuk normal kedua (2NF)

Dalam bentuk 2NF suatu tabel harus memiliki semua *field non- key* (*field* yang tidak termasuk dalam *primary key*) yang tergantung penuh secara fungsional pada *primary key*. Dengan kata lain dalam tahap ini dilakukan penentuan *field* kunci yang unik dan dapat mewakili untuk masing-masing tabel.

- Bentuk normal ketiga (3NF)

Pada tahap 3NF dilakukan penentuan relasi antar tabel sehingga ditemukan adanya *field-field* kunci skunder (*foreign key*) pada tabel-tabel tertentu.

---

<sup>16</sup> Budi Sutedjo Dharma Oetomo, S.Kom, MM, *Perencanaan dan pembangunan Sistem Informasi*, Andi Yogyakarta, 2002, halaman 131.

<sup>17</sup> Mark Whitehorn dan Bill Marklyn, *Seluk beluk Database Relasional*, Penerbit Erlangga, 2003, halaman 213.

## 2.4 Dokumentasi Sistem

Dokumentasi sistem bertujuan untuk membantu memahami dan mengevaluasi suatu system informasi. Ada beberapa komponen dalam dokumentasi system yang biasa dipakai antara lain : *Data Flow Diagram* (DFD), *Flowchart* dan *Entity Relationship Diagram* (ERD).

### 2.4.1 Data Flow Diagram

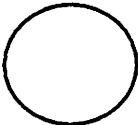



*Data Flow Diagram* (DFD) adalah representasi grafik dari sebuah sistem. DFD menggambarkan komponen-komponen sebuah sistem, aliran-aliran data diantara komponen-komponen tersebut dan asal, tujuan dan penyimpanan dari data tersebut. Ada tiga tipe DFD yaitu *Context Diagram* (CD) atau diagram konteks, DFD fisik dan DFD logis.

*Context Diagram* (CD) adalah diagram tingkat atas, yaitu diagram yang paling tidak detil, dari sebuah sistem informasi yang menggambarkan aliran-aliran data ke dalam dan keluar entitas-entitas eksternal. Entitas-entitas eksternal adalah entitas-entitas yang terletak diluar system yang mengirim data ke atau menerima data dari sistem tersebut. Lingkaran dalam CD mendefinisikan batas system. Batas sistem adalah batas antara “daerah kepentingan sistem” dan “lingkungan sistem”. Lingkungan sistem adalah segala sesuatu yang berhubungan atau mempengaruhi sistem tersebut, yaitu digambarkan dengan kotak-kotak entitas. Terminologi sistem terakhir adalah *Interface*. *Interface* adalah aliran yang menghubungkan sebuah sistem dengan lingkungan tersebut.

*Data Flow Diagram* (DFD) fisik adalah representasi grafik dari sebuah system yang menunjukkan entitas-entitas internal dan eksternal dari system tersebut, dan aliran –aliran data ke dalam dan keluar dari entitas-entitas tersebut. Entitas-entitas internal adalah personel, tempat atau mesin dalam sistem tersebut yang mentransformasikan data. DFD fisik tidak menunjukkan apa yang dilakukan, tetapi menunjukkan dimana, bagaimana, dan oleh siapa proses-proses dalam sebuah sistem dilakukan.

DFD logis menggambarkan kegiatan-kegiatan yang dilakukan oleh sistem, sedang DFD fisik menggambarkan infrastruktur yang dimiliki oleh sebuah sistem. DFD

digunakan untuk dua hal utama, yaitu untuk membuat dokumentasi dari sistem informasi yang ada atau menyusun dokumentasi untuk sistem informasi yang baru.

SIMBOL	NAMA	KETERANGAN
	Proses Data	Gambar lingkaran menunjukkan proses transformasi data atau dapat mewakili suatu pekerjaan atau proses
	Aliran Data	Gambar anak panah menunjukkan aliran suatu berkas elemen data.
	Penyimpan Data	Menunjukkan tempat dalam menyimpan data
	Entity	Menunjukkan sumber data ( <i>source</i> ) atau tujuan aliran data dari atau ke sistem

Gambar 2.3 Simbol-simbol dalam DFD

(Sumber : Leman. 1998. Metodologi Pengembangan Sistem Informasi. Jakarta : PT. Elex Media Komputindo)

## 2.4.2 Entity Relationship Diagram (ERD)

ERD adalah representasi data secara logis dari suatu organisasi atau area bisnis tertentu, yang disajikan dalam bentuk gambar. ERD menggambarkan *entity* yang terdapat dalam lingkungan bisnis tersebut, relasi antar *entity-entity* nya, dan atribut-atribut yang dimiliki oleh *entity* dan relasi tersebut. ERD dibuat pada tahapan analisis dari proses perancangan dan pembuatan sistem.

## 2.5 Metode Pengembangan Sistem Informasi

Pengembangan sistem informasi berbasis komputer adalah tugas yang kompleks dan memerlukan waktu yang lama. Hal ini disebabkan karena

pengembangan sistem harus melewati beberapa taha, mulai dari perencanaan system, penetapan, pengoperasian, sampai pemeliharaan system.

Apabila setelah kurun waktu tertentu system tersebut dalam pengoperasiannya mengalami permasalahan yang serius, dan tidak dapat ditangani lagi dalam tahap pemeliharaan system, maka system tersebut perlu dikembangkan lagi mulai dari tahap awal, agar dapat memenuhi kebutuhan informasi saat ini. Siklus ini disebut dengan siklus hidup suatu sistem (*system life cycle*).

Ada banyak metode yang dikemukakan pakar sistem informasi tentang tahapan pada daur pengembangan sistem. Akan tetapi yang dibahas disini hanya metode SDLC(*system development life cycle*) dan metode *Information Engineering* menurut McFadden.

### 2.5.1 System Development Life Cycle (SDLC)

SDLC memiliki enam fase untuk mendesain dan menganalisis suatu sistem, memecahkan masalah yang timbul, dan cara pengembangan sistem tersebut. SDLC banyak digunakan oleh organisasi atau perusahaan untuk menganalisis dan mendesain sistem. Enam fase yang tercakup dalam SDLC adalah<sup>18</sup> :

#### 1. *Preliminary Investigation* (penelitian pendahuluan)

Pada fase ini tahap-tahap yang harus dilakukan adalah :

- Analisis awal

Yaitu mencari tahu apa saja kegiatan klien atau pelanggan, dan menemukan permasalahan yang dihadapinya.

- Usulan solusi alternative

Setelah diketahui kegiatan dan permasalahan yang dihadapi, maka dapat dicari solusi permasalahan yang sesuai. Solusi dapat diperoleh melalui hasil wawancara dengan klien, atau mempelajari data-data yang sudah dimiliki oleh klien. Kemudian dapat diberikan pilihan apakah akan meninggalkan sistem lama, meningkatkan sistem lama, atau membuat sistem baru.

- Penjelasan biaya dan keuntungan

Dari beberapa solusi tersebut, dapat dihitung biaya yang diperlukan dan keuntungan yang didapat untuk masing-masing solusi.

- Rencana awal

Rencana awal harus disampaikan dalam bentuk laporan, agar klien dapat memilih rencana yang sesuai. Dalam tahap ini pengembang sistem juga dapat membantu memberikan rekomendasi, agar klien dapat memilih solusi yang tepat untuk mendapatkan hasil optimal.

## 2. *System Analysis* (analisis sistem)

Tahap-tahap yang harus dilakukan pada fase ini antara lain :

- Mengumpulkan data

Data yang diperlukan dapat diperoleh dari dokumen-dokumen yang telah ada sebelumnya, mengadakan wawancara, atau membagikan kuisioner.

- Analisis data

Berdasarkan data-data yang diperoleh, sistem dapat digambarkan dengan menggunakan *Data Flow Diagram* (DFD) yang merepresentasikan aliran data yang terdapat dalam sistem.

- Menulis laporan

Setelah selesai melakukan analisis, perlu dibuat laporan yang dapat menjelaskan bagaimana sistem berjalan, masalah yang dihadapi dalam sistem tersebut, dan apa yang dibutuhkan oleh sistem baru, serta memberikan rekomendasi yang sesuai.

## 3. *Design System* (perancangan sistem)

Fase ini memiliki tahap-tahap antara lain :

- Desain awal

Desain awal menjelaskan kebutuhan sistem dan komponen-komponen yang terlibat dalam sistem. Desain yang dibuat harus sesuai dengan biaya yang diperlukan. Desain awal dapat berbentuk *prototype* yang dapat diujicobakan pada lingkungan yang kecil untuk mengetahui perubahan sistem yang terjadi dan keuntungan yang dapat diperoleh dari sistem yang baru.

- Rincian desain

Rincian desain menjelaskan hal-hal yang diperlukan sistem untuk melakukan *input*, *output*, proses, penyimpanan, dan lain sebagainya.

- Penulisan laporan

Desain awal dan rincian yang telah dibuat harus dituliskan dalam bentuk laporan untuk memantau sejauh mana sistem telah dikerjakan.

#### 4. *System Development* (pengembangan sistem)

- Mengembangkan atau membuat *software*

Dalam tahap ini terdapat dua pilihan yaitu, membuat *software* yang baru mengembangkan *software* yang sudah ada sebelumnya.

- Membuat *hardware* yang sesuai

Dalam tahap ini harus dicari dan ditentukan *hardware* yang sesuai dengan spesifikasi *software* yang telah dibuat.

- Menguji sistem

Apabila *software* dan *hardware* sudah siap, maka dapat dilakukan pengujian sistem untuk mengetahui apakah sistem berjalan sesuai yang direncanakan.

#### 5. *System Implementation* (penerapan sistem)

- Konversi sistem baru

Untuk menggantikan sistem yang lama dengan sistem yang baru, perlu dilakukan konversi. Ada beberapa cara untuk melakukan konversi, antara lain :

- *Direct Implementation*, yaitu tidak memakai sistem lama dan langsung menggunakan sistem yang baru.

- *Parallel Implementation*, sistem lama dan baru dipakai bersama-sama sampai sistem baru benar-benar berjalan dengan baik.

- *Pilot Implementation*, sistem yang baru hanya digunakan oleh beberapa orang saja.

#### 6. *System Maintenance* (perawatan sistem)

Dalam fase ini yang dilakukan adalah mengamati sistem agar dapat diperbaharui secara berkala sesuai dengan kebutuhan.

---

<sup>18</sup> Brian K. Williams and Stacey C. Sawyer, *Using Information Technology, A Practical Introduction to Computers & Communications*, McGraw-Hill, 2005, halaman 458-465.

### 2.5.2 Metode *Information Engineering*

Menurut McFadden, untuk mencapai tujuan, suatu organisasi memerlukan arsitektur sistem informasi dan metode yang mampu mendukung pembuatan arsitektur tersebut. Metode tersebut dinamakan *Information Engineering*. *Information Engineering* adalah metode umum yang digunakan untuk membuat dan merawat suatu sistem informasi. Metode ini merupakan metode *top-down* yang dimulai dengan membuat model bisnisnya, kemudian dilanjutkan dengan pembuatan model data dan prosesnya sehingga dapat mendukung dan berhubungan dengan model bisnisnya.

Ada beberapa tahapan dalam *Information Engineering* yaitu<sup>19</sup> :

- Perencanaan (*Planning*)

Kekuatan utama *Information Engineering* adalah penekanannya pada strategi perencanaan sistem informasi. Tujuan dari tahap perencanaan adalah untuk menyesuaikan teknologi informasi dengan strategi bisnis suatu organisasi, Suatu organisasi akan mampu bersaing apabila dapat mengembangkan rancangan sistem informasi yang tepat, dan menerapkan rancangan tersebut ke dalam suatu proyek sistem informasi yang nyata. Ada tiga tahap dalam fase perencanaan, yaitu :

a. Mengidentifikasi factor strategi perencanaan (*strategic planning factor*)

Yang termasuk kedalam factor strategi perencanaan antara lain tujuan organisasi, factor-faktor penting untuk mencapai kesuksesan, dan permasalahan yang dihadapi organisasi. Tujuan dari tahap ini adalah membuat rencana awal dan menghubungkan rencana sistem informasi dengan rencana strategi bisnis.

b. Mengidentifikasi rencana objek organisasi (*corporate planning objects*)

Tujuan utama tahap ini adalah untuk menentukan objek-objek penting yang akan berhubungan dengan perencanaan sistem informasi. Ada beberapa objek penting yang harus diidentifikasi dalam tahap ini, yaitu :

- Unit-unit organisasi

Terdiri dari bermacam-macam departemen yang dapat diperoleh dari struktur organisasi.

- Lokasi organisasi

Objek ini hanya dibuat untuk organisasi yang beroperasi di lebih dari satu lokasi.



- Fungsi-fungsi bisnis  
Adalah kumpulan dari proses bisnis yang saling berhubungan satu sama lain yang dapat mendukung tercapainya misi suatu organisasi. Suatu fungsi bisnis dapat dihubungkan dengan beberapa unit organisasi, dan begitu pula sebaliknya, suatu unit organisasi dapat memiliki beberapa fungsi bisnis.
  - Tipe-tipe *entity*  
*Entity* pada tahap ini dapat diperoleh dari hasil wawancara dengan beberapa manajer. Daftar *entity* yang diperoleh nantinya akan bertambah seiring dengan tahapan analisis selanjutnya.
- c. Membuat *enterprise model*  
Tahap ini terdiri dari tiga fase, yaitu :
- Dekomposisi Fungsional  
Merupakan proses pemecahan fungsi-fungsi dalam suatu organisasi menjadi tingkatan yang lebih rinci. Selama dalam tahap perencanaan, setiap fungsi-fungsi tingkat tinggi akan dipecah menjadi bagian-bagian yang lebih kecil untuk mendukung masing-masing fungsi tersebut.
  - *Entity-Relationship Diagram*  
*Entity-Relationship Diagram* tingkat tinggi mencakup tipe-tipe *entity* yang telah ditemukan pada fase perencanaan tahap kedua. Diagram ini akan terus berkembang apabila ditemukan tipe-tipe *entity* yang baru.
  - Matriks Perencanaan  
Tahap ini berguna untuk membuat hubungan antara berbagai macam komponen yang telah dikembangkan pada tahap-tahap sebelumnya.
- Analisis (*analysis*)  
Tujuan tahapan ini adalah untuk membuat spesifikasi yang lebih rinci tentang sistem informasi yang diperlukan suatu organisasi. Tahap analisis meliputi studi tentang sistem informasi dan situasi bisnis saat ini, serta penentuan kebutuhan sistem baru. Ada dua tahapan utama dalam fase analisis, yaitu membuat model data konseptual dan membuat model prosesnya.

a. Membuat model data konseptual

Model data konseptual adalah model rincian yang menggambarkan keseluruhan struktur data yang terorganisasi. Yang termasuk dalam model data konseptual adalah *entity*, relasi, dan atribut, termasuk aturan-aturan bisnis dan pembatas yang mendefinisikan bagaimana data digunakan.

b. Membuat model proses

Model proses menggambarkan deskripsi logis dari proses yang dilakukan oleh fungsi-fungsi organisasi, dan aliran data dalam proses –proses tersebut. Yang dimaksud dengan proses adalah himpunan tugas logis yang dilakukan berulang-ulang untuk mendukung satu atau lebih fungsi-fungsi bisnis. Proses ini memiliki batasan yang jelas dan akan mengubah *input* menjadi *output*.

Ada dua macam tipe proses, yaitu proses fisik (*physical proses*) yang akan mengubah masukan nyata menjadi keluaran nyata dan proses informasi (*information process*) yang mengubah data menjadi informasi.

Setelah proses-proses berhasil diidentifikasi dan dipecah menjadi level yang lebih rinci, langkah selanjutnya adalah membuat model yang menggambarkan aliran data dalam proses-proses tersebut. Alat yang biasa dipakai untuk menggambarkan model ini adalah DFD.

- Perancangan (*Design*)

Tujuan utama tahap ini adalah mengubah model informasi yang telah dikembangkan dalam tahap analisis menjadi model yang sesuai dengan teknologi yang akan digunakan untuk penerapan sistem informasi.

a. Merancang *database*

Tujuan perancangan *database* adalah untuk memetakan model data konseptual menjadi model implementasi sehingga DBMS dapat memprosesnya agar dapat berguna bagi semua *user* dalam organisasi. Perancangan *database* dapat dibagi menjadi dua tahap, yaitu :

- Perancangan *database* logis (*logical database design*)

Adalah proses pemetaan model data konseptual menjadi struktur *database* yang sesuai dengan DBMS. Jika tujuannya adalah menggunakan *database*

relasional, maka model data konseptual (dari tahap analisis) harus dipetakan menjadi relasi yang ternormalisasi.

- Perancangan *database* fisik (*physical database design*)

Adalah proses pemetaan struktur *database* dari desain logis menjadi struktur penyimpanan secara fisik, seperti *files* atau *table*. Pada tahap ini juga ditetapkan indeks yang digunakan, metode pengaksesan *database*, keamanan, fungsi *backup* data, dan pemulihan (*recovery*) data.

b. Merancang Proses

Tujuan perancangan proses adalah menentukan rincian logika untuk masing-masing proses. Ada dua tahapan dalam perancangan proses, yaitu :

- Penetapan rincian logika proses, termasuk logika penggunaan *database* secara bersamaan.
- Merancang antarmuka pengguna untuk memudahkan *user* berinteraksi dengan aplikasi *database*, misalnya dengan menggunakan *window*, *icon*, atau produk GUI (*graphical user interface*) yang lain.

- Penerapan (*Implementation*)

Tujuan tahap ini adalah untuk membangun dan menginstall sistem informasi sesuai dengan rencana dan rancangan dari tahap-tahap sebelumnya. Implementasi mencakup serangkaian tahap yang mengarah pada operasional sistem informasi, seperti membuat definisi *database*, *coding* program, pengujian sistem, membuat prosedur operasional dan dokumentasi, pelatihan untuk pengguna, dan perawatan *database*. Dalam konteks *Information Engineering* penekanannya hanya pada pembuatan definisi *database* dan program aplikasinya.

Definisi *database* biasa dibuat dalam bentuk skema, yang merupakan deskripsi menyeluruh dari struktur logis atau *database*. Skema dibuat dengan menggunakan bahasa DDL (*data definition language*). Sedangkan program aplikasi dapat dibuat dengan menggunakan bahasa pemrograman procedural seperti C, *Delphi*, *Visual Basic*. Apabila menggunakan *database* relasional, *query*nya menggunakan bahasa standard SQL. Pernyataan SQL dapat juga dimasukkan dalam *coding* program untuk mengakses *record-record* dalam *database*.

---

<sup>19</sup>Fred R. Mcfadden and Jeffrey A. Hoffer, *Modern Database Management*, Benjamin/Cummings Publishing Company, Inc. 1994, halaman 79-97

## **2.6 Software yang Digunakan**

Penggunaan *software* untuk Pengembangan Sistem Informasi Penjualan Sembako Pada CV. Surya berbasis *Client server* ini menggunakan Borland Delphi 7.0 dan Borland Interbase 7.0 sebagai databasenya. Keduanya saling terintegrasi satu sama lain, sehingga dapat memudahkan membuat program aplikasi yang kompleks dan handal.

### **2.6.1 Borland Delphi 7.0**

Borland Delphi 7 adalah paket bahasa pemrograman yang bekerja dalam system operasi Windows, yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Beberapa jenis aplikasi yang dapat kita buat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web.

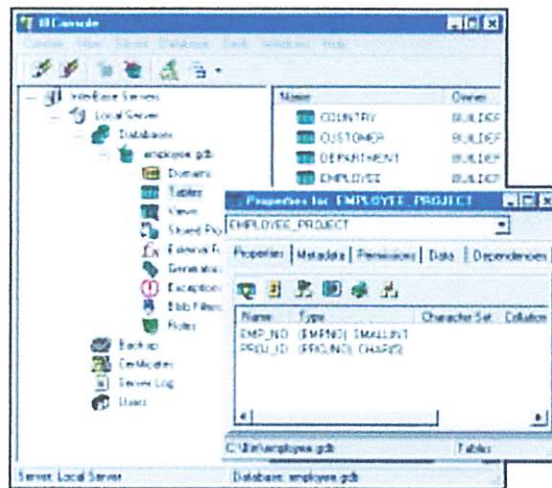
Secara umum kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan kita untuk membuat aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah pemrogram dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam dua kelompok yaitu object dan bahasa pemrograman. Secara ringkas, object adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapat dilihat (*visual*). Object biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa Object Pascal yang sudah sangat dikenal dikalangan pemrograman profesional. Gabungan dari object dan bahasa pemrograman ini disebut sebagai bahasa pemrograman berorientasi object atau *Object Oriented Programming (OOP)*.

### **2.6.2 Borland Interbase Delphi 7.0**

Borland Interbase 7 adalah database transaksional yang ekonomis dan memiliki performa yang tinggi, dan banyak digunakan oleh jutaan pengguna di seluruh dunia. Dengan mengkombinasi instalasi yang mudah, *crash-recovery* secara otomatis, dan

perawatan yang sangat minim, Interbase menjadi sangat sesuai untuk *embedding* dalam aplikasi terdistribusi. Dukungan terhadap *multiprosesor* dan arsitektur yang canggih menjadikannya sebuah pilihan yang tepat untuk aplikasi bisnis dengan kekuatan tinggi yang memiliki banyak pengguna yang terkoneksi. Kekuatannya, yaitu kemudahan penggunaan, dukungan terhadap platform Windows, Linux, dan Solaris (termasuk pengembangan dalam lingkungan seperti Borland Delphi, C++ Builder, dan Kylix), membuat Interbase menjadi favorit bagi para pengembang.



2.4 Gambar IB Console

### KEBUTUHAN SISTEM MINIMAL

- 32 MB RAM.
- 20 MB dari spasi harddisk untuk instalasi.
- Drive CD-ROM.

### **Windows**

- Microsoft Windows Server 2003, Windows XP, Windows 2000 (SP2), Windows ME, Windows 98 SE, Windows NT 4.0 (SP6 atau lebih).

## **FITUR-FITUR DAN KEUNTUNGAN**

### **KEUNGGULAN DATABASE TERINTEGRASI**

- **Kinerja yang tinggi;**  
Interbase memiliki semua fitur, kekuatan, dan skalabilitas yang diperlukan untuk aplikasi bisnis yang kompleks yang menawarkan keunggulan performa untuk jumlah pengguna yang besar.
- **Backup secara online;**  
Menjaga produktivitas saat melakukan backup, dan memastikan bahwa data di-backup dengan aman.
- **Crash Recovery secara otomatis;**  
Kembali pada pekerjaan dalam waktu yang sangat singkat setelah kehilangan sumber daya atau crash pada server tanpa atensi dari staf TI atau administrator database.
- **Instalasi yang mudah;**  
Borland Interbase didesain untuk dapat diinstall dengan mudah pada lokasi *end-user* tanpa pertolongan administrator database. Interbase dapat di-install secara terselubung, sehingga *end-user* tidak tahu jika sudah terinstal.
- **Biaya kepemilikan total yang murah;**  
Biaya penyebaran dan pemeliharaan yang rendah memberikan biaya kepemilikan total yang rendah.

### **CEPAT**

- **Dukungan untuk multiproses secara simetris;**  
Dengan adanya kemungkinan aplikasi menggunakan banyak prosesor, aplikasi bekerja lebih cepat, mendukung lebih banyak pengguna, dan menangani arsitektur aplikasi yang lebih kompleks.
- ***Native server multi-thread* dengan latar belakang koleksi sampah dan manajemen *multi-user page-buffer*;**  
Interbase memberikan performa yang lebih baik dengan membebaskan transaksi pengguna dari penghalangan performa oleh koleksi sampah. Hal ini dilakukn dengan membuat untaian latar brlakang serial yang mereduksi beban

server, menyebarkan tugas lewat periode waktu yang ditentukan pengguna. Manajemen cache sudah dioptimalkan untuk menjaga *working set* dari halaman-halaman database untuk transaksi-transaksi individual melawan operasi-operasi panjang, termasuk backup database, sweeping, tabel berukuran besar, dan scan indeks.

- Penyetelan sendiri dengan *query optimizer* berbasis biaya;

Fitur penyetelan sendiri (self-tuning) dari Interbase meningkatkan performa dan meminimkan administrasi database. Termasuk di dalamnya query optimizer berbasis biaya yang menggunakan statistik pada kekardinalan tabel, otomatisasi koleksi sampah, dan *dynamic rebalancing* struktur indeks untuk melepaskan halaman indeks kosong.

## 2.7 Client Server

*Client server* adalah istilah yang digunakan untuk pengertian proses layanan database yang di tangani oleh piranti software RDBMS yang berdiri sendiri dan terdapat pada komputer yang terpisah dari *client/* pengguna. Tentunya piranti software RDBMS yang digunakan cukup canggih, karena bukan hanya sekedar menangani database.

Latar belakang pemrograman berbasis *client server* tidak lepas dari sejarah perkembangan teknologi komputer dan kebutuhan yang muncul pada perusahaan besar ( *multinasional* ), militer maupun perguruan tinggi untuk memperoleh atau berbagi informasi antar kelompoknya. Pada masa itu, proses pengelolaan data dipusatkan pada satu komputer yang dihubungkan ke sejumlah dumb terminal pada jarak tertentu. Batasan jarak yang terlalu dekat ini tidak memungkinkan mereka untuk berbagi informasi untuk jarak yang jauh, karenanya dikembangkan berbagai cara, baik itu dari segi perangkat keras, aturan ( *protocol* ) bagaimana data dikomunikasikan sampai pada pengaturan cara memperoleh data, guna memungkinkan komunikasi data pada jarak yang jauh.

Teknologi LAN ( *local area network* ) terutama sistem operasi novell turut membantu proses tersebut sehingga dengan biaya terjangkau sebuah anak cabang perusahaan dapat memiliki sistem komputerisasi yang dimiliki oleh perusahaan induk

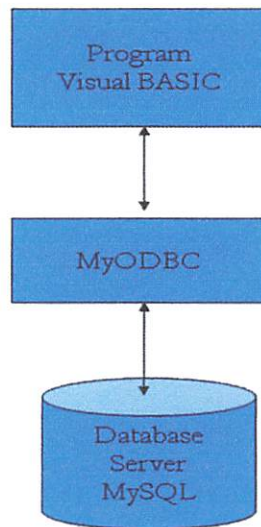
pada era tahun 1960 -1970an. Istilah *workstation/client* pun mengganti terminologi dumb terminal.

Kini *workstation/client* dapat memproses data sebelum data dikirim ke komputer *server*. Sayangnya masalah lain muncul, yakni terjadinya peningkatan arus lalu lintas data antar komputer *server* dan *workstation* sehingga mempengaruhi kinerja seluruh sistem. Kejadian ini dikenal dengan istilah *Busy/Heavy traffic* ( jalur sibuk). Lalulintas data yang kian ramai ini harus segera diatasi karena makin berkembangnya jenis data dari semula berbasis teks atau database telah di tambah oleh jenis lain seperti gambar, audio atau music dan video yang membutuhkan jalur lalulintas data lebih besar lagi. Untuk mengatasi hal tersebut mulai dikembangkan konsep *client- server* yang berbasis pada aturan bahwa komputer *server* hanya akan mengirim data yang dibutuhkan oleh *workstation* atau *client* dimana proses penyiapan data dilakukan oleh komputer *server*.

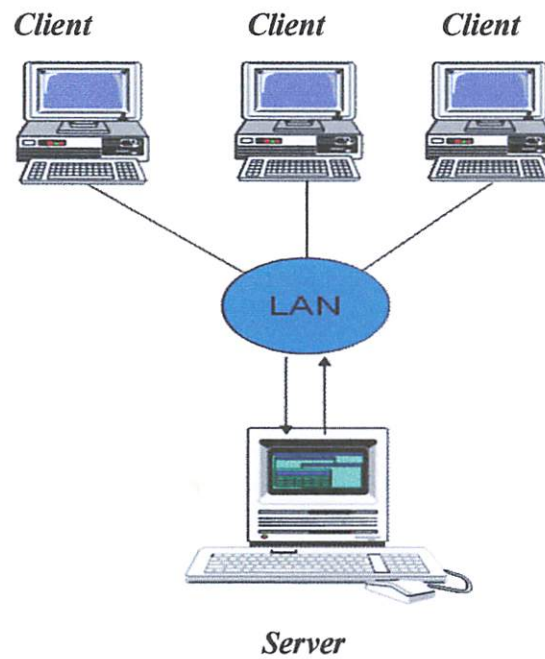
### 2.7.1 Arsitektur *Client Server*

Berdasarkan pada cara PC *client* dihubungkan ke komputer *server*, di kenal dua macam tingkatan arsitektur yaitu model dua Tier dan model tiga Tier. Dua Tier atau dua tingkatan adalah proses dimana komputer *workstation* membuat hubungan ke komputer *server* dan mempertahankan hubungan tersebut sampai proses selesai. Misalnya pada proses pemasukan data atau membuat sebuah laporan. Hubungan yang terbentuk sangat konsumtif dalam sumber daya (alokasi memori, ruang hardisk, control dll), karenanya tidak efektif jika dilakukan untuk hubungan yang melibatkan banyak pemakai (system jaringan internet). Oleh karena itu tidak heran jika konsep dua tingkatan hanya di praktekkan pada LAN, sedangkan untuk hubungan yang banyak pengguna digunakan model tiga tier. Untuk pembuatan aplikasi ini akan digunakan metode dua Tier, karena digunakan untuk jaringan skala kecil dan di praktekkan pada LAN.





Gambar 2.5 : Arsitektur Model Dua Tier



Gambar 2.6 : Arsitektur Client Server

### 2.7.2 Metode Akses Data

Istilah *Client Server* mengacu pada hubungan antara dua sistem atau proses. *Client* adalah sistem yang mengajukan permintaan (*request*) ke *server*. Dalam banyak situasi maka yang mana sistem *client* dan yang mana sistem *server* ditentukan oleh hubungan antara yang mengajukan permintaan (*client*) ke yang memberikan layanan yaitu *server* (Hall,C, 1994).

*Server* menyediakan layanan (*services*) ke *client* yang mengajukan permintaan. Dalam hal ini sangat penting untuk dapat membedakan antara *server* dengan layanan (*services*) yang disediakan. Untuk Mengakses berbagai jenis data (format file), Microsoft dan panyaedia software, menyediakan piranti file/driver file untuk keperluan tersebut. Pada saat ini terdapat dua kelompok profider file, yakni ODBC (*Open Database Connectivity*) dan OLE DB (*object Link Embedded Database*).

### 2.7.3 Komponen Perangkat Lunak Sistem *Client Server*

Arsitektur sistem client/server mempunyai beberapa subsystems yaitu user interaction/presentation subsystems, aplication subsystem, dan database management subsystems. Subsystems tersebut dapat dialokasikan ke client, server, atau terdistribusi diantara kedua sistem (Pressman, 2001)

#### *User interaction/presentation subsystem*

Merujuk atau mengimplementasikan semua fungsi yang berkaitan dengan Graphical User Interface (Pressman, 2001). GUI (Graphical User Interface ) dipergunakan untuk interface antara user dengan sistem (komputer).

#### *Aplication subsystems*

Mengimplementasikan kebutuhan-kebutuhan pada domain aplikasi. Sebagai contoh yaitu aplikasi groupware yang menyediakan fasilitas email. Aplikasi dapat dipecah sebagian di client, sebagian lain di server.(Pressman, 2001)

#### *Database management subsystems*

Melakukan pemrosesan dan pengelolaan basis data seperti yang diperlukan oleh aplikasinya. Pemrosesan dan pengolahan tersebut dapat sederhana misalnya transfer record atau kompleks misalnya memproses SQL (Pressman, 2001)

## 2.7.4 Keuntungan Sistem Informasi Berbasis *Client Server*

Secara garis besar, banyak manfaat yang dapat diambil dengan adanya sistem informasi berbasis *client server*, antara lain :

### 1. Teknologi

Tidak seperti file-based database yang menggunakan sharing file (penggunaan file secara bersama-sama) untuk mengaplikasikan multi user, database server menggunakan pendekatan client-server dimana semua user yang akan mengakses database terkoneksi dan berkomunikasi dengan database server. Semua permintaan dari user akan diatur dan diolah oleh database server secara bergantian maupun bersamaan. File-file data yang disimpan dan diatur oleh database server tidak diakses secara langsung oleh user. Hal ini merupakan penerapan dari Security yang dimiliki oleh database server.

### 2. Security

Karena semua komunikasi harus melalui server, maka pengaturan hak akses dapat dilakukan pada level server. Hak akses akan menentukan siapa saja yang dapat mengakses database dan mengolah data didalamnya. Tanpa adanya hak akses ini seorang user tidak bisa untuk megakses sebuah database. Karena user diatur pada level server dan tidak mempunyai akses langsung ke file-file data, maka kemungkinan seorang user yang tidak berwenang untuk mengcopy file data dapat dihindari.

### 3. Data Integrity

Pada sistem dengan file-based database, bila sebuah komputer crash, hang atau reboot karena sesuatu hal, ada kemungkinan besar file data akan rusak atau hilang. Hal ini dapat disamakan bila kita sedang membuka aplikasi pengolah kata, dimana ada kemungkinan dokumen yang sedang dibuka dan diedit menjadi rusak atau hilang bila komputer crash, hang atau reboot. Problem ini akan semakin tampak seiring dengan bertambah besarnya file data. Lain halnya dengan database server, pada saat sebuah komputer yang merupakan database server mengalami crash, hang atau reboot kemungkinan besar hal yang terjadi hanyalah semua koneksi ke database server akan terputus sampai database server berjalan kembali. Walaupun demikian, kemungkinan

## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1. Analisis Sistem

Pada bab ketiga ini akan dijelaskan apa saja yang dilakukan dalam menganalisa dan merancang sistem informasi, mulai dari menganalisis sistem, menuangkannya dalam DFD dan membuat ERD dari sistem ini.

Toko ini sudah mempunyai sistem awal, dimana pencatatan dilakukan menggunakan kertas, kartu dan buku sebagai dokumen. Tidak ada proses komputerisasi sama sekali. Perhitungan pembelian kepada *supplier*, penjualan kepada *customer*, retur baik pembelian maupun penjualan, hutang piutang dan perhitungan stok barang, semuanya dilakukan secara manual.

Pemesanan barang pada *supplier* didasarkan pada jumlah barang yang ada di gudang dan kondisi pasar saat itu yang diputuskan oleh pemilik, apabila terdapat barang baru. Untuk pemantauan stok barang di gudang digunakan kartu stok, tetapi tidak di-*update* secara optimal, seringkali barang keluar tanpa diikuti dengan pencatatan pada kartu stok, sehingga data kartu stok tidak akurat. Akibatnya seringkali terjadi bahwa stok barang yang habis baru diketahui saat ada pemesanan untuk barang tersebut. Hal ini menimbulkan *back-order* dan tentu saja mengurangi kepuasan pelanggan. Untuk pemesanan barang lebih lebih mengandalkan kepada ingatan pegawai dan pemilik sendiri, barang apakah yang habis dijual dan harus dipesan lagi. Pemesanan dilakukan lewat telepon atau fax tanpa adanya pendokumentasian, sehingga pada saat penerimaan barang, pemilik yang melakukan pemesanan harus memeriksa sendiri apakah barang tersebut benar-benar dipesan oleh perusahaan.

Pada saat barang yang dipesan datang, maka yang diperiksa dan diarsipkan adalah tagihan dari *supplier*. Kondisi barang tidak diperiksa karena dalam keadaan terbungkus dan dijual dalam keadaan yang sama, jadi pemeriksaan hanya dilakukan pada kuantitas barang yang dicocokkan antara wujud fisik dengan apa yang tercatat pada tagihan dari *supplier*. Karena tidak adanya dokumentasi pemesanan pembelian,

hilangnya atau rusaknya data akan selalu ada. Untuk itu, proses backup data merupakan hal yang wajib dilakukan. Database server memberikan fasilitas proses backup data dengan cara yang lebih fleksibel dibandingkan file-based database. Backup data pada database server dapat dilakukan dengan menerapkan penjadwalan, misalnya secara otomatis dilakukan setiap hari jam 10 malam. Media backup juga dapat dipilih mulai dari harddisk sampai ke Tape Backup. Selain itu, file backup bisa ditempatkan pada komputer yang ada di tempat lain.

#### **4. Performance**

Database server didesain dan dioptimasi untuk melayani permintaan dari banyak user dan tidak terpengaruh dengan besarnya data yang telah ada. Lalu lintas data pada jaringan jauh lebih rendah dibandingkan file-based desktop.

#### **2.7.5 Kelemahan Sistem Informasi *Client Server***

Disamping memiliki beberapa keuntungan, sistem informasi clien server juga memiliki beberapa kelemahan, antara lain :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Dibutuhkan satu komputer khusus yang berkemampuan lebih baik dan tinggi untuk ditugaskan sebagai server.
3. Sangat ketergantungan pada server jika mengalami gangguan atau masalah, maka secara keseluruhan jaringan akan terganggu.

maka dikhawatirkan terjadinya kesalahan pengiriman barang oleh *supplier* baik jenis barang maupun kuantitas yang tidak disadari pula oleh pemilik perusahaan.

Pada sistem penjualan, untuk pelanggan yang datang ke toko atau melalui telepon, pemilik atau pegawai akan mencatat pesanan tersebut sambil mengingatkan apakah barang yang dipesan ada di gudang dengan kuantitas yang dipesan oleh *customer*. Kemudian dibuatkan nota penjualan dan barang dikirim bersama dengan nota penjualan tersebut. *Copy* nota penjualan sebanyak dua lembar, satu sebagai arsip dan satu lagi dimasukkan kedalam arsip khusus untuk *customer* yang bersangkutan yang isinya adalah kumpulan *copy* nota penjualan kredit yang belum dilunasi ditambah dengan satu lembar kartu rekapitulasi pembayaran piutang yang dilkaskan.

Penagihan piutang dan hutang dilakukan dengan cara mengingatkan hutang dan piutang yang jatuh tempo dan sekali waktu pengecekan kumpulan arsip pelanggan. Hutang tidak terlalu bermasalah karena jumlah *supplier* tidak terlalu banyak. Tetapi penagihan piutang seringkali timbul masalah karena banyaknya *customer* sehingga pengecekan arsip piutang pelanggan jarang dilakukan secara menyeluruh. Pengecekan diprioritaskan pada pelanggan yang arsipnya telah tertumpuk banyak, sehingga seringkali terdapat pelanggan yang mempunyai sedikit piutang tetapi berlangsung sangat lama karena kelalaian dalam melakukan pemeriksaan tanggal jatuh tempo.

Retur penjualan dilakukan apabila *customer* melaporkan adanya kerusakan barang, kemudian barang tersebut disimpan dalam tempat khusus untuk barang retur yang dikelompokkan menurut *supplier* masing-masing. Kemudian untuk jangka waktu tertentu sekitar satu bulan dilakukan retur pembelian berdasarkan barang retur dari *customer* tersebut.

Laporan-laporan tentang penjualan dan pembelian hanya dapat dipantau melalui nota penjualan dan pembelian. Perhitungan total transaksi untuk tiap hari dilakukan setelah jam kerja oleh pemilik sendiri. Mengingat banyaknya transaksi setiap hari, hal ini juga cukup memakan waktu. Penentuan barang apa yang laku di pasaran dan barang apa yang kurang laku sepenuhnya diputuskan berdasarkan naluri dan pengamatan langsung pemilik, tidak ada data yang dijadikan acuan, sehingga pengambilan keputusan seringkali kurang akurat.

### 3.2. Perancangan Sistem *Database*

Perancangan *Database* adalah langkah pertama yang dilakukan dalam pembuatan sistem informasi. Langkah ini dibagi dalam tiga bagian yaitu pembuatan diagram arus data (DFD), pembuatan diagram hubungan entitas (ERD) dan pembuatan tabel-tabel yang diperlukan.

#### 3.2.1 Pembuatan Data Flow Diagram (DFD)

Perancangan sistem informasi dilakukan dengan pembuatan *Data Flow Diagram* (DFD). *Data Flow Diagram* dari sistem informasi ini terdiri dari *context diagram*, DFD level 0, DFD level 1, dan DFD level 2.

##### 3.2.1.1 *Context Diagram*

*Context diagram* dari DFD ini memperlihatkan sistem secara integral dan pihak-pihak (*external entity*) siapa saja yang terlibat dalam sistem perusahaan. Pihak-pihak yang terlibat yaitu :

##### 1. *Customer*

Arus data yang mengalir dari *customer* ke sistem adalah pesanan penjualan, nota retur penjualan dan uang pembayaran dari *customer*, sedangkan dari sistem ke *customer* terdapat arus data nota penjualan dan untuk uang retur penjualan tunai.

##### 2. *Supplier*

Arus data yang mengalir dari *supplier* ke sistem adalah pesanan pembelian, nota retur pembelian dan uang pembayaran pembelian. Sedangkan arus data sebaliknya yang menghubungkan *supplier* kepada sistem adalah nota pembelian dan uang retur pembelian tunai.

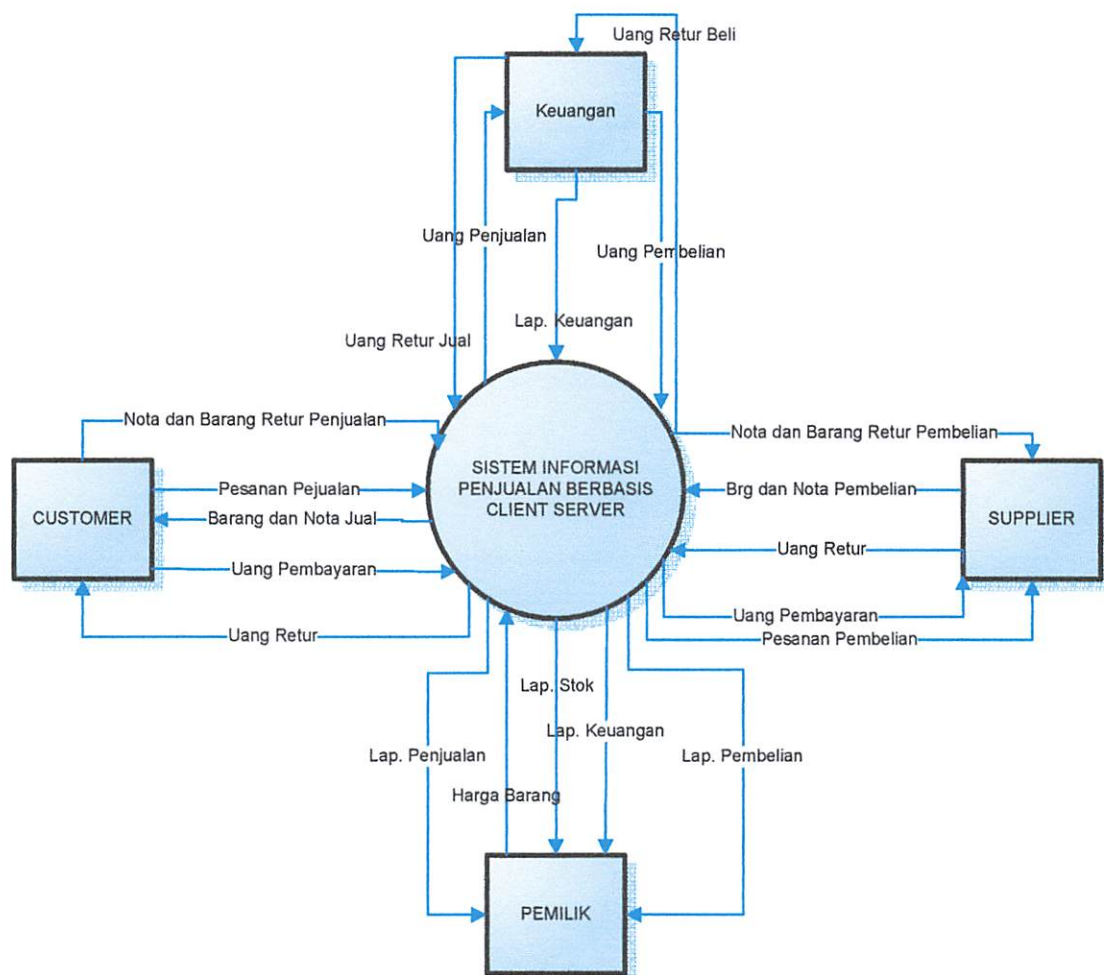
##### 3. *Owner*

Sistem terhubung dengan pemilik untuk memberikan laporan-laporan, dalam hal ini laporan yang diberikan antara lain adalah laporan-laporan yang berhubungan dengan penjualan, pembelian dan stok barang.

##### 4. *Keuangan*

Arus yang mengalir dari dari entitas keuangan ke sistem adalah uang retur jual, laporan keuangan dan uang pembelian. Sedangkan sebaliknya dari sistem ke entitas

kuangan adalah uang penjualan, laporan penjualan, laporan pembelian dan uang retur pembelian.

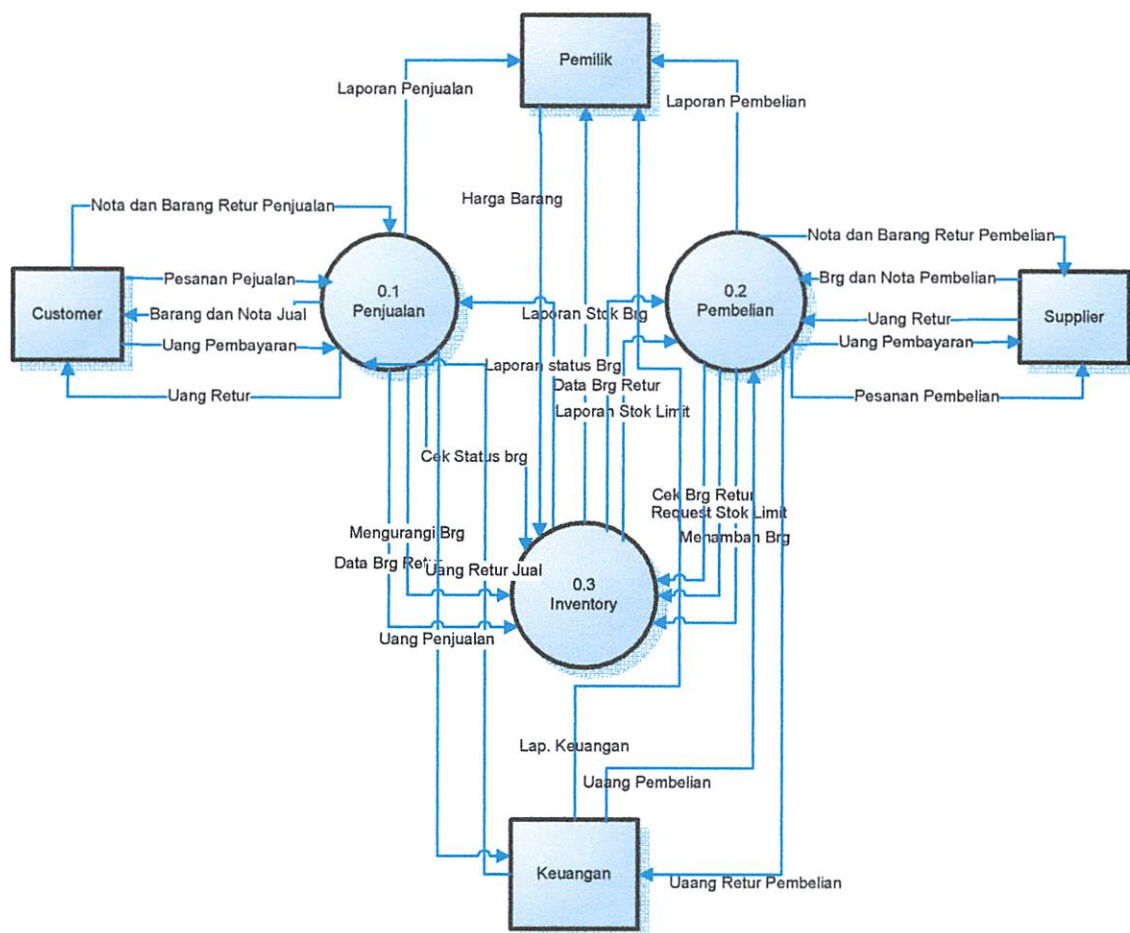


Gambar 3.1 : *Context Diagram*

### 3.2.1.2 *Data Flow Diagram Level (DFD) Level 0*

DFD level 0 adalah *Context diagram* yang dijabarkan menjadi 3 bagian besar proses yaitu proses penjualan, pembelian dan *inventory*. Proses penjualan dan pembelian terhubung dengan proses *inventory* yang mengelola stok barang. Baik proses pembelian, penjualan maupun *inventory* terhubung dengan pemilik untuk memberikan laporan pada masing-masing proses.

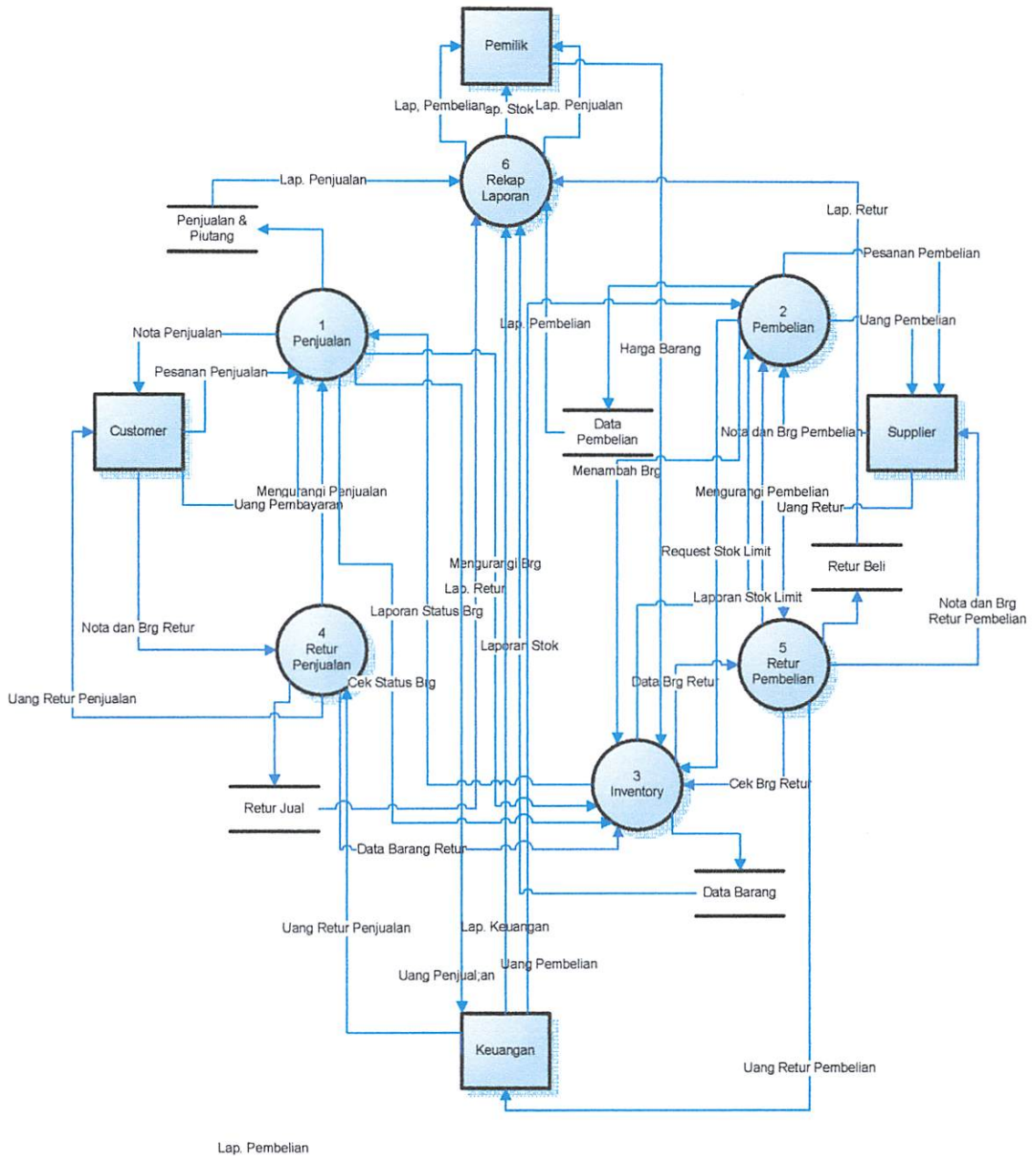




Gambar 3.2 : DFD Level 0 SI Penjualan Sembako

### 3.2.1.3 Data Flow Diagram (DFD) Level 1

DFD *level 1* adalah penjabaran lebih lanjut dari DFD *level 0*, yang terdiri dari proses penjualan, proses retur penjualan, proses pembelian, proses retur pembelian, proses *inventory* dan proses rekap laporan seperti gambar 3.3.



Lap. Pembelian

Gambar 3.3 : DFD Level 1 SI Penjualan Sembako

### 3.2.1.4 Data Flow Diagram (DFD) level 2 Proses Penjualan

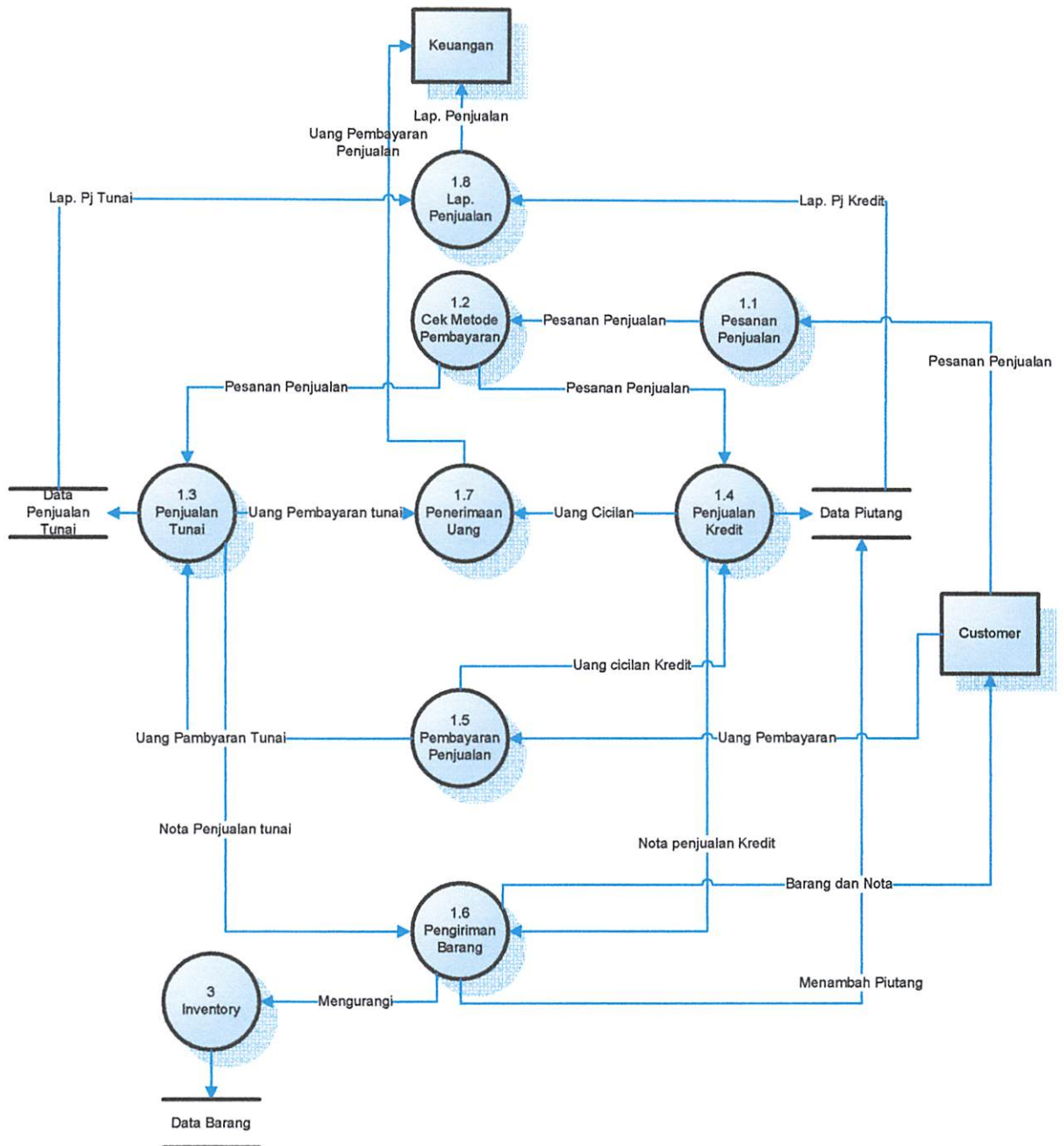
Pada DFD level 2 ini akan dijabarkan lebih lanjut mengenai proses penjualan yang terdiri dari beberapa proses. Pada level 2 terdapat 6 proses yaitu :

- Pesanan Penjualan

Terjadi pada saat *customer* melakukan pemesanan penjualan.

- **Cek metode pembayaran**  
Proses ini dilakukan pada saat menerima pemesanan penjualan. Pegawai yang menerima order menanyakan metode pembayaran yang dilakukan oleh *customer*. Pemesanan penjualan dapat dilakukan dengan cara *customer* datang dan memesan langsung, melalui telepon atau dengan perantaraan tenaga penjualan (*sales*).
  - **Penjualan Tunai**  
Proses ini dilakukan apabila *customer* meminta order penjualan yang dibayar secara tunai.
  - **Penjualan Kredit**  
Proses ini dilakukan apabila *customer* meminta order penjualan yang dibayar secara kredit.
  - **Pembayaran Penjualan**  
Terjadi pada saat *customer* melakukan pembayaran. Pada penjualan tunai, maka proses berlangsung seperti biasa, sedangkan pada proses penjualan kredit maka mengurangi piutang *customer* yang bersangkutan.
  - **Pengiriman barang**  
Proses ini dilakukan setelah *customer* melakukan transaksi dan mendapatkan nota baik nota penjualan tunai maupun kredit.
  - **Penerimaan Uang**  
Proses ini dilakukan setelah *customer* melakukan pembayaran, baik tunai maupun kredit.
  - **Laporan Penjualan**  
Proses ini proses merupakan proses rekap dari data penjualan tunai dan kredit.
  - **Proses *Inventory***  
Proses ini yang melakukan pengelolaan stok barang.
- Data Flow Diagram level 2* proses penjualan dapat dilihat pada gambar 3.4

:



Gambar 3.4 : DFD level 2 Proses Penjualan

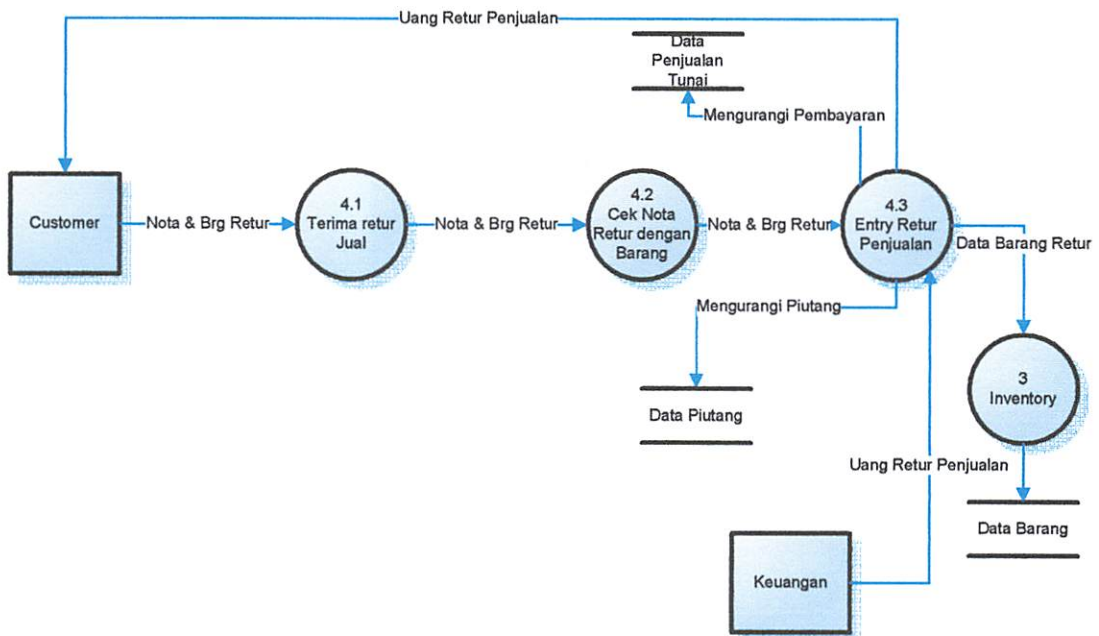
### 3.2.1.5 Data Flow Diagram (DFD) level 2 Retur Penjualan

Retur penjualan terjadi pada saat *customer* mengembalikan barang cacat atau salah kirim. Dibagi dalam 4 proses, yaitu :

- Terima retur penjualan

Retur penjualan diawali dengan masuknya barang retur dari *customer* bersama dengan nota retur penjualan.

- Cek nota retur penjualan dengan barang  
Pegawai toko mencocokkan nota retur penjualan dengan wujud barangnya.
  - Entry retur penjualan  
Apabila barang retur cocok dengan nota retur penjualan, maka petugas memasukkan data retur penjualan. Pada proses ini akan dilakukan pengurangan pembayaran pada penjualan untuk retur penjualan yang kemudian diikuti dengan proses pengembalian uang penjualan yang sudah dibayar. Pada proses memasukkan data ini menambah kuantitas barang retur pada data barang.
  - Memasukkan barang ke gudang (*Inventory*)  
Setelah barang yang ada cocok dengan nota retur penjualan, barang retur, maka barang retur dimasukkan ke gudang pada bagian khusus barang retur, untuk dikembalikan ke *supplier* yang bersangkutan.
- Gambar diagram alir data untuk proses retur penjualan dapat dilihat pada gambar 3.5

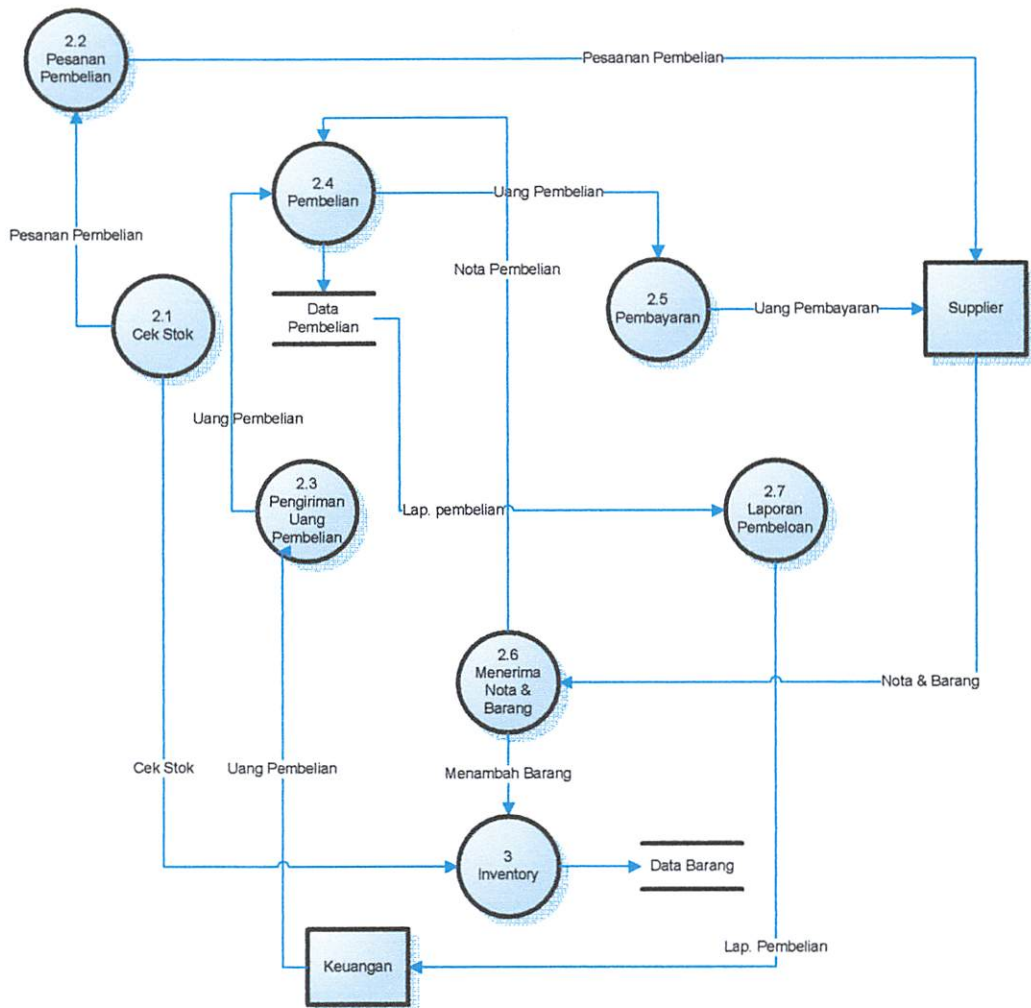


Gambar 3.5 : DFD *level 2* Retur Penjualan

### 3.2.1.6 Sistem Pembelian

Sistem pembelian, seperti yang dapat dilihat pada gambar 3.6, dibagi dalam 7 proses yaitu :

- Cek Stok  
Mengawali proses pembelian
- Pemesanan pembelian  
Proses dimana stok barang menipis
- Pengiriman uang pembelian  
Proses pengiriman uang kepada supplier.
- Menerima barang dan Nota  
Proses dimana barang dan Nota pembelian diterima dari *supplier*.
- Pembayaran pembelian  
Proses pembayaran pembelian dilakukan dengan cara transfer atau melalui tenaga penjualan .
- Laporan Pembelian  
Merupakan Laporan dari data pembelian
- Inventory  
Pada proses *inventory* barang akan bertambah setelah terjadi proses pembelian.



Gambar 3.6 : *Data Flow Diagram (DFD) level 2* Proses Pembelian

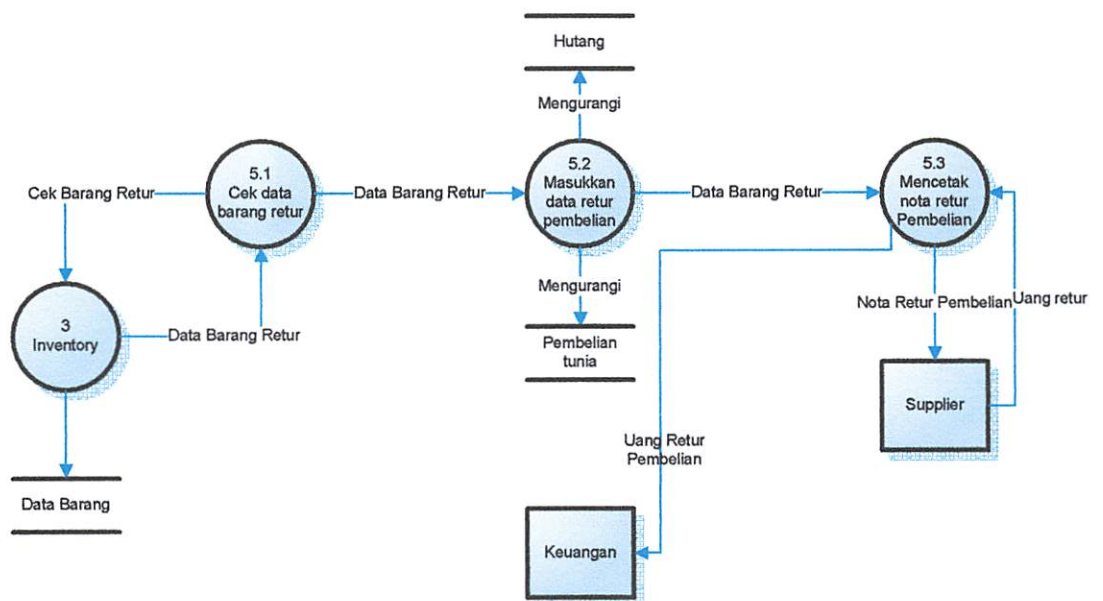
### 3.2.1.7 *Data Flow Diagram (DFD) level 2* Retur Pembelian

Retur pembelian terjadi pada saat *customer* mengembalikan barang cacat. Dibagi dalam 4 proses, yaitu :

- Proses Inventory  
Dimana permintaan pengecekan barang retur dilakukan di proses inventory.
- Cek nota data barang retur  
Pegawai memeriksa data barang retur dari *customer* yang di-*retur* ke *supplier* yang bersangkutan.
- Entry retur pembelian

Data barang-brang yang di-*retur* diinputkan ke dalam data retur, proses ini akan mengurangi hutang kepada *supplier* apabila transaksi dilakukan pada pembelian adalah kredit dan mengurangi pembelian apabila transaksi yang dilakukan adalah tunai. Sebelum proses ini dilakukan bisa menghubungi *supplier* terlebih dahulu.

- Mencetak nota retur pembelian  
Data barang-barang yang telah diinputkan dicetak, lalu disertakan dalam pengembalian barang seperti pada gambar 3.7



Gambar 3.7 : *Data Flow Diagram (DFD) level 2* Retur Pembelian

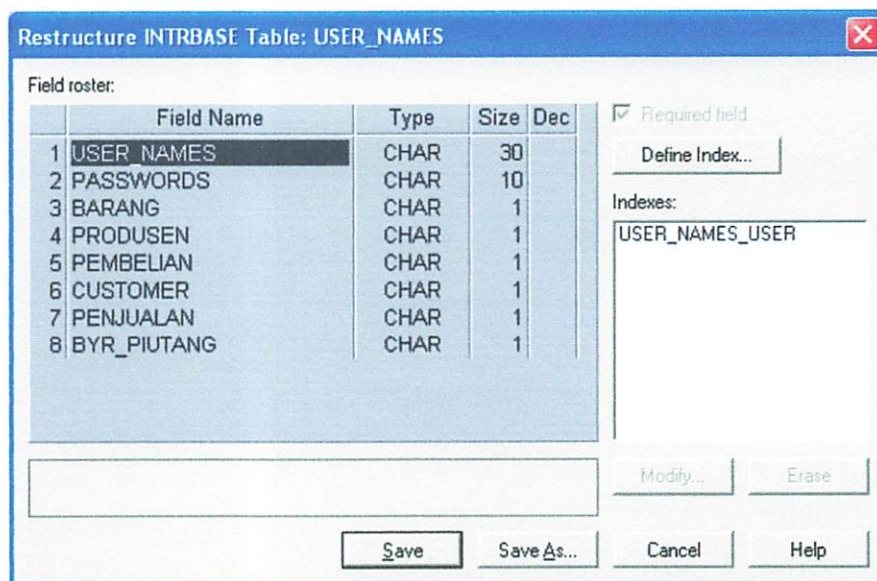
### 3.2.2. Pembuatan Diagram Hubungan Entitas (ERD)

Tahap selanjutnya dalam proses perancangan adalah membuat diagram hubungan entitas. ERD memuat hubungan antar tabel yang terdapat dalam sistem. ERD pada sistem ini menggunakan 13 tabel, antar lain :

1. Tabel *User*



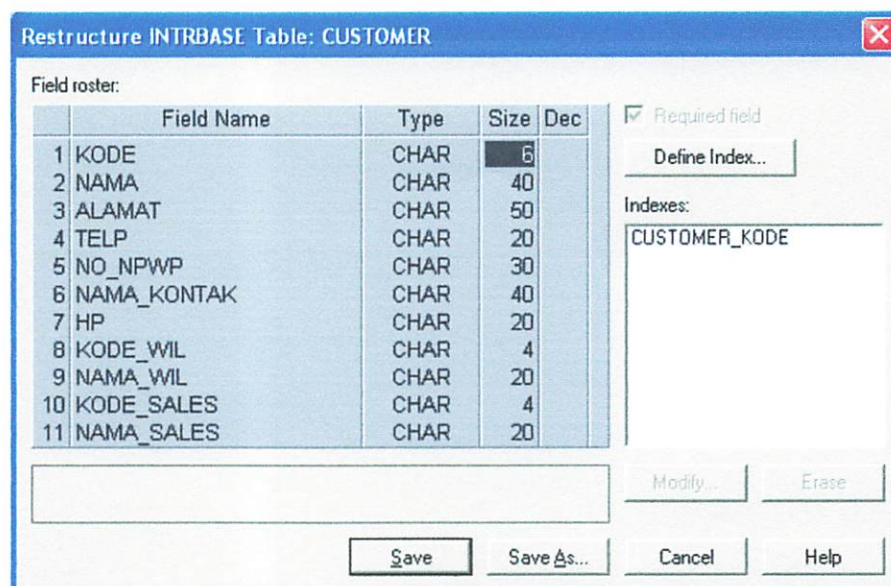
Berisi tentang *user*, *password* dan hak akses masing-masing. Yaitu untuk setiap orang yang mempunyai otoritas dalam dalam pengoperasian sistem informasi ini. Berhubungan dengan tabel *staff*.



Gambar 3.8: Tabel User Name

## 2. Tabel *customer*

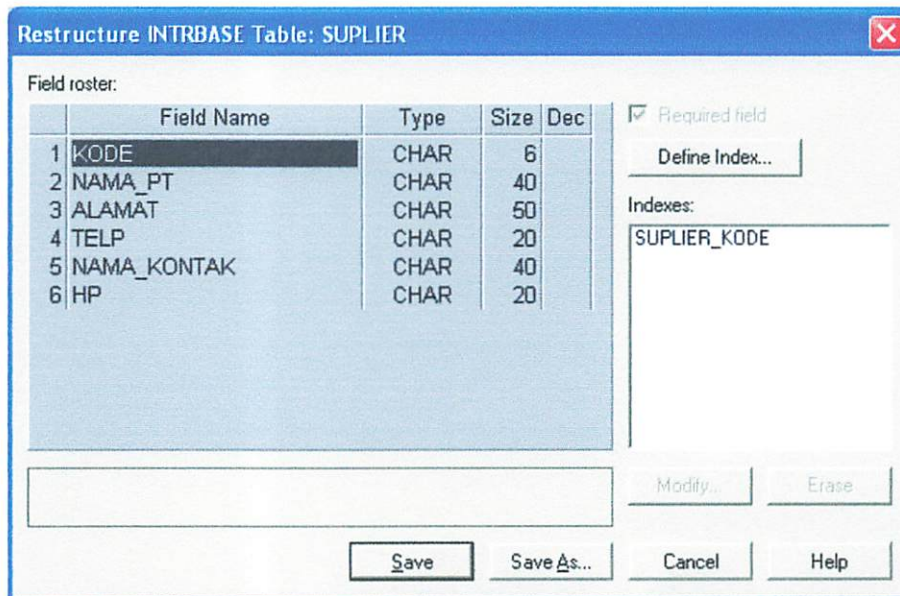
Berisi tentang data lengkap tiap-tiap *customer* untuk dihubungkan dengan tabel-tabel yang berhubungan dengan penjualan yaitu : penjualan piutang dan order penjualan.



Gambar 3.9: Tabel *Customer*

### 3. Tabel *supplier*

Berisi tentang data lengkap tiap-tiap *supplier* untuk dihubungkan dengan tabel-tabel yang berhubungan dengan pembelian yaitu : Pembelian, hutang dan order pembelian.



Gambar 3.10: Tabel *Supplier*

### 4. Tabel Barang

Berisi tentang data lengkap barang. Berhubungan dengan setiap tabel yang membutuhkan data barang, yaitu : semua tabel detail yang meliputi pembelian detail, penjualan detail, hutang detail, piutang detail, order penjualan detail dan order pembelian detail. Tabel retur tidak terhubung secara langsung dengan tabel ini tetapi terlebih dahulu terhubung dengan tabel penjualan, pembelian, hutang maupun piutang master, yang mana setiap tabel master tersebut terhubung dengan tabel detainya yang masing-masing terhubung dengan tabel barang.

Restructure INTRBASE Table: BARANG

Field roster:

	Field Name	Type	Size	Dec
1	KODE	CHAR	6	
2	NAMA	CHAR	40	
3	JENIS	CHAR	30	
4	STOK	DOUBLE		
5	HPP	DOUBLE		
6	NILAI	DOUBLE		
7	HRG_JUAL	DOUBLE		
8	STOK_MIN	DOUBLE		

Required field

Define Index...

Indexes:

BARANG

Modify... Erase

Save Save As... Cancel Help

Gambar 3.11: Tabel Barang

### 5. Tabel Pembelian Master

Berisi semua transaksi pembelian tunai. Berhubungan dengan tabel pembelian detail, *supplier* dan retur pembelian master.

Restructure INTRBASE Table: MBELI

Field roster:

	Field Name	Type	Size	Dec
1	NOMER	LONG		
2	TANGGAL	TIMESTAMF		
3	KODE	CHAR	6	
4	NAMA	CHAR	40	
5	TOTAL_BELI	DOUBLE		

Required field

Define Index...

Indexes:

MBELI\_NOMER

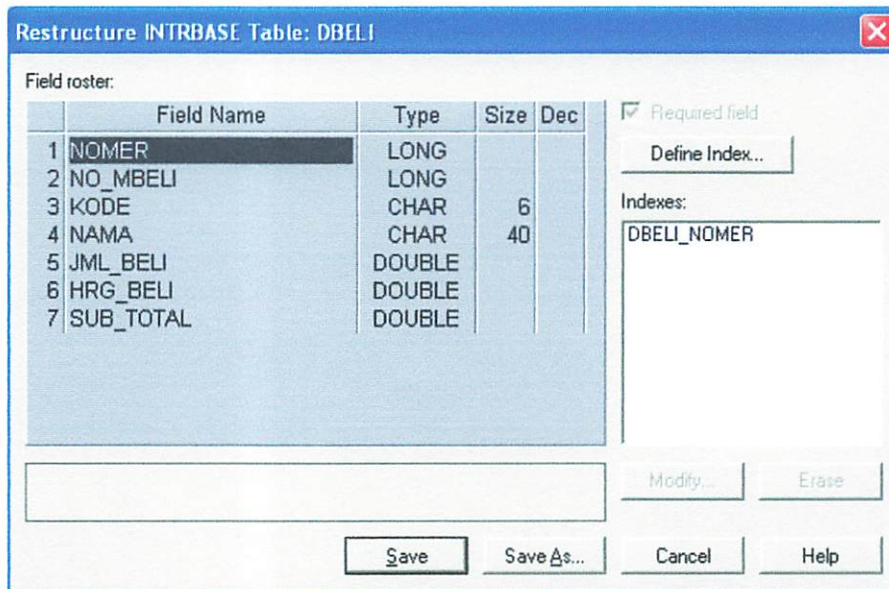
Modify... Erase

Save Save As... Cancel Help

Gambar 3.12: Tabel Pembelian Master

## 6. Tabel Pembelian *Detail*

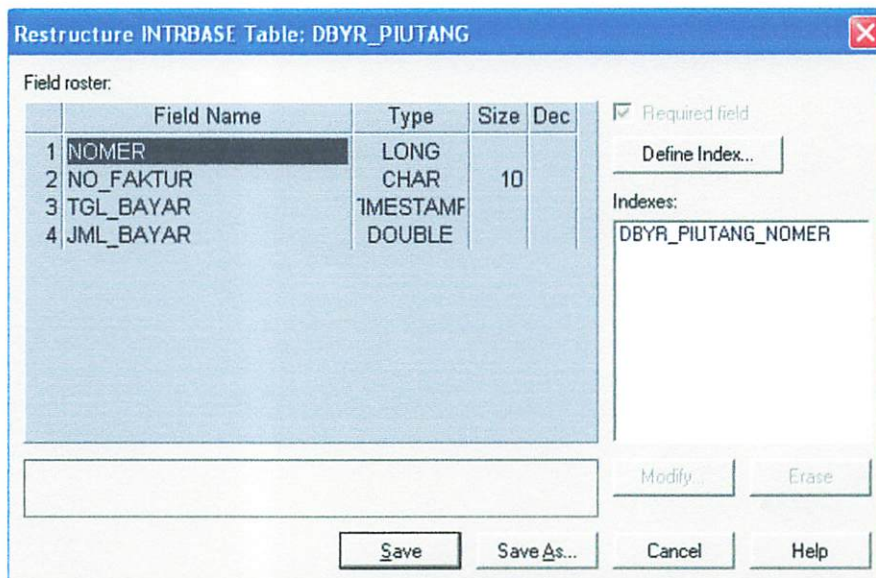
Berisi tentang *detail* transaksi dari master pembelian. Terhubung dengan tabel barang dan pembelian master.



Gambar 3.13: Tabel Detail Pembelian

## 7. Tabel Piutang

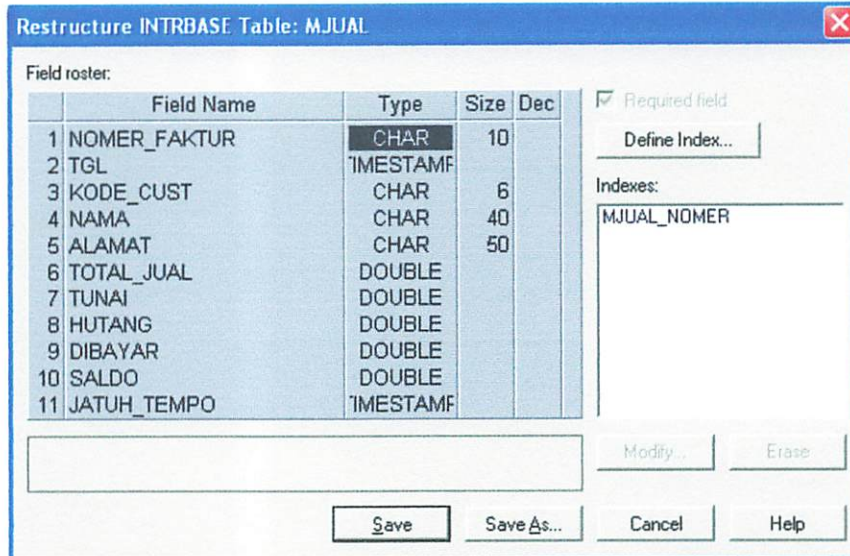
Berisi tentang pembayaran piutang, dan berhubungan dengan tabel penjualan dan *customer*.



Gambar 3.14: Tabel Piutang

## 8. Tabel Penjualan Master

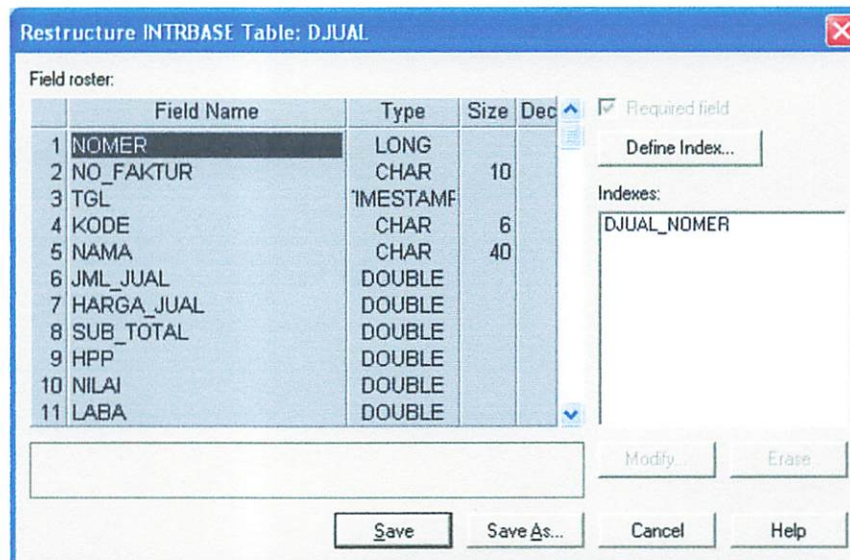
Berisi tentang order penjualan yang akan dilakukan. *Record* pada pemesanan penjualan ini akan di hapus pada saat barang pesanan sudah dikirim. Terhubung dengan tabel pembelian *detail* dan *customer*.



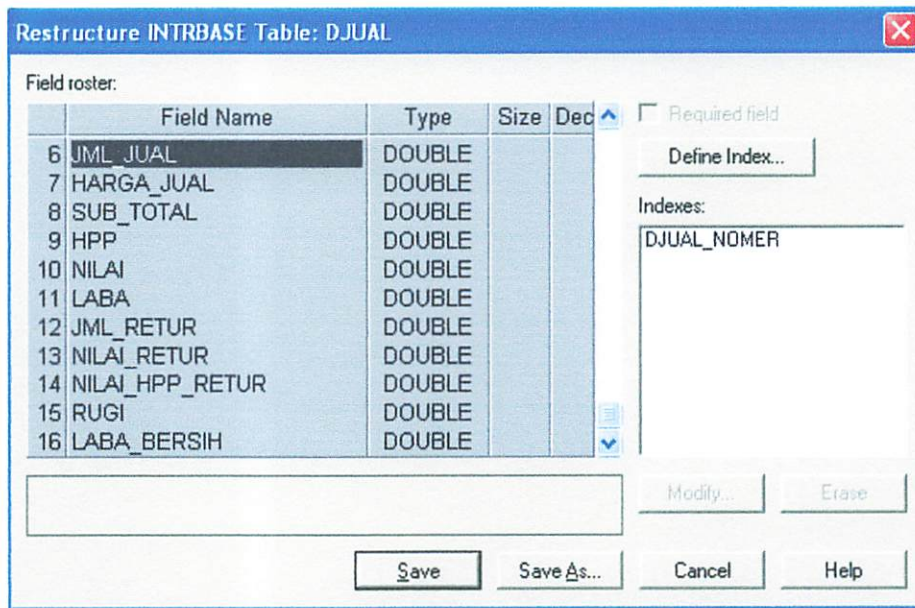
Gambar 3.15: Tabel Penjualan Master

## 9. Tabel Penjualan Detail

Berisi tentang uraian order penjualan order penjualan master, terhubung dengan tabel barang dan tabel penjualan master.



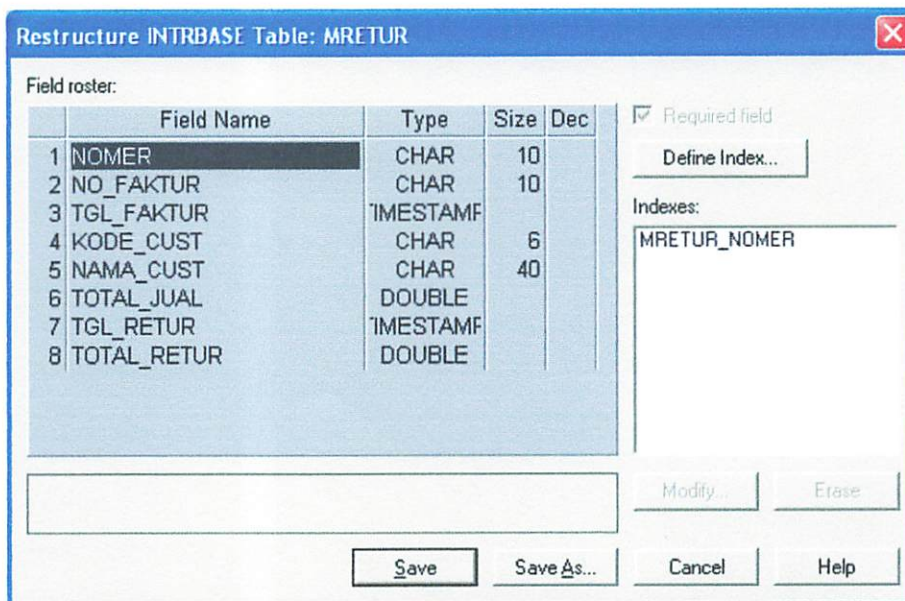
Gambar 3.16: Tabel Penjualan Detail



Gambar 3.17: Lanjutan Tabel Penjualan Detail

#### 10. Tabel Retur Penjualan Master

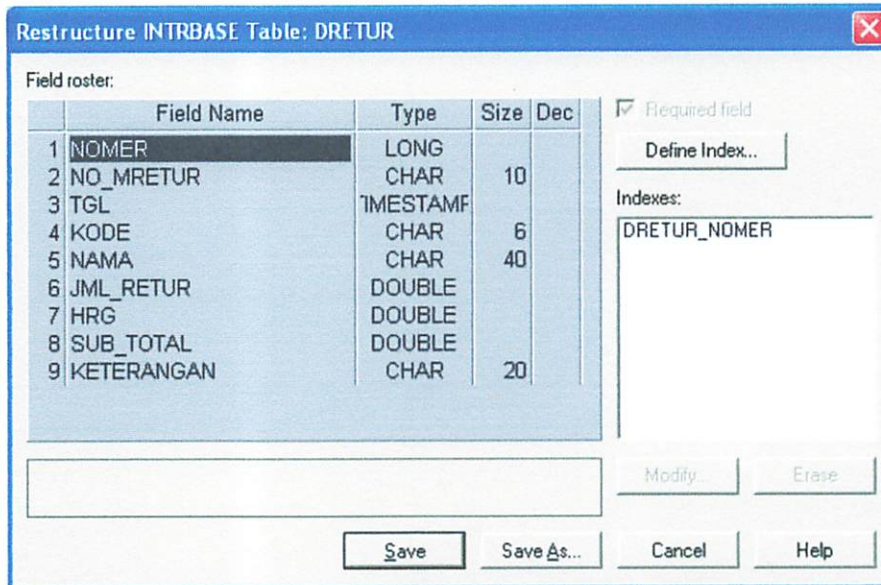
Berisi data retur penjualan yang terhubung dengan tabel retur penjualan detail dan tabel penjualan master.



Gambar 3.18: Tabel retur Penjualan Master

### 11. Tabel Retur Penjualan Detail

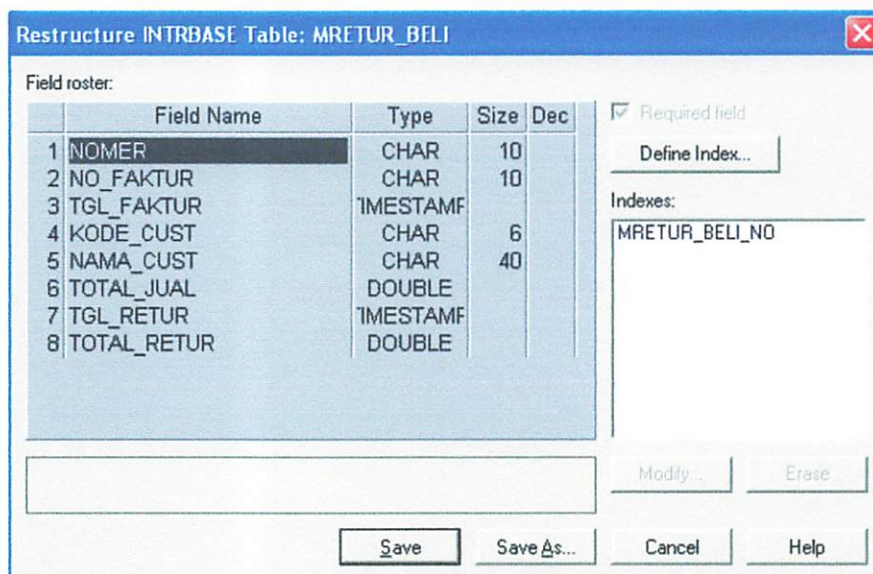
Berisi tentang uraian retur penjualan master. Terhubung hanya dengan retur penjualan master.



Gambar 3.19: Tabel Retur Penjualan detail

### 12. Tabel Retur Pembelian Master

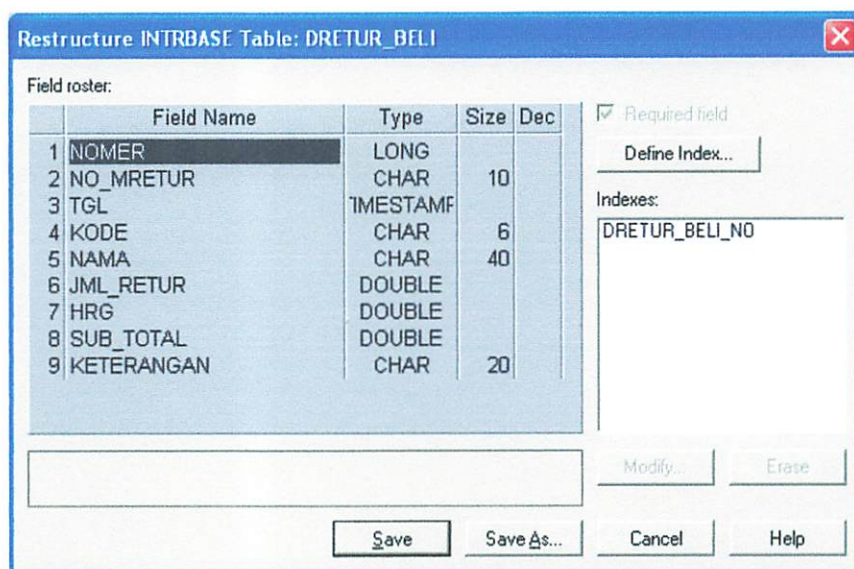
Berisi data retur pembelian yang terhubung dengan tabel pembelian master dan tabel retur pembelian detail.



Gambar 3.20: Tabel Retur Pembelian Master

### 13. Tabel Retur Pembelian Detail

Berisi uraian retur pembelian master. Terhubung dengan tabel retur pembelian master.

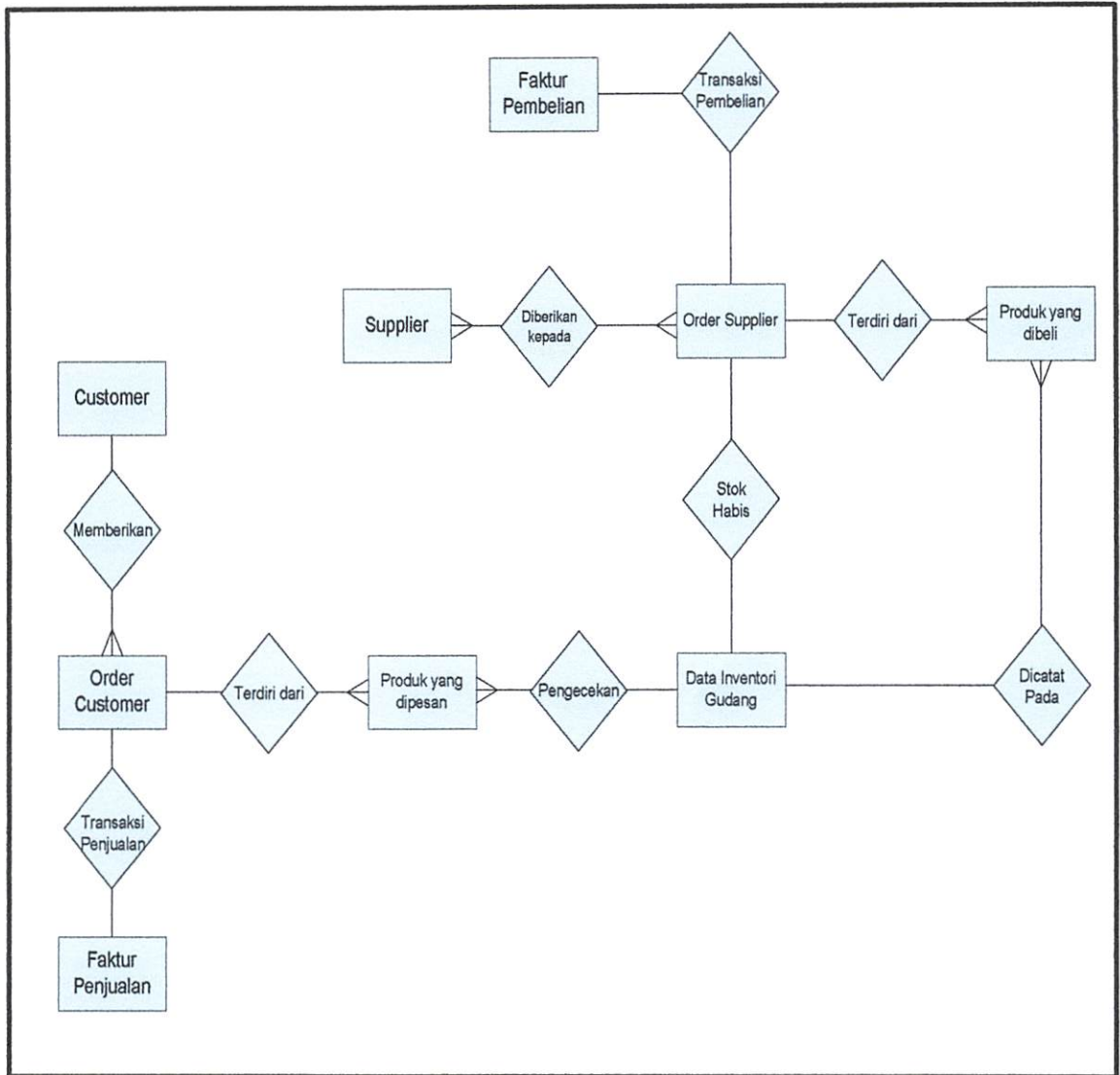


Gambar 3.21: Tabel Retur pembelian Detail



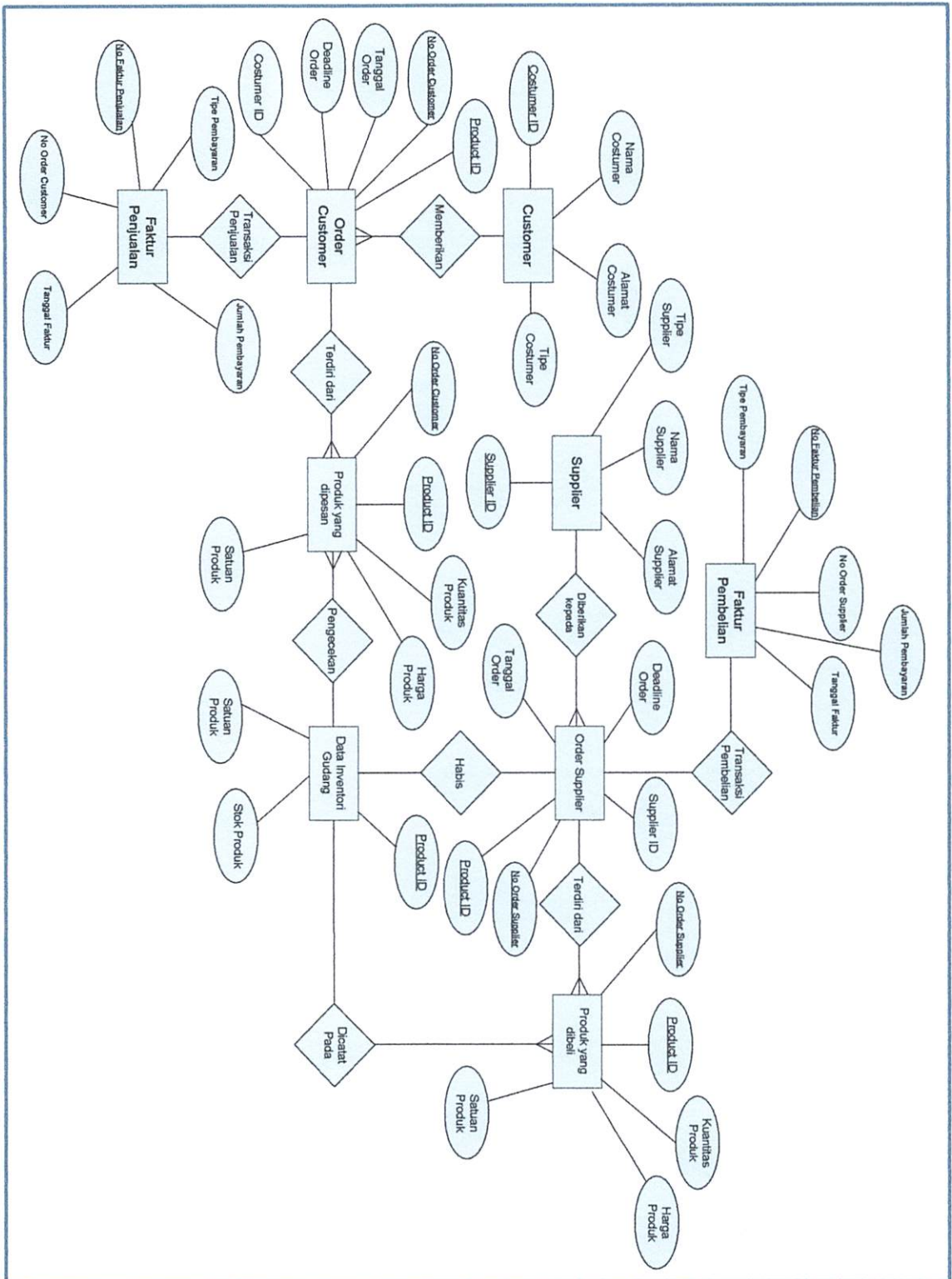
### 3.2.2.1 Gambar Diagram Hubungan Entitas (ERD)

#### 3.2.2.1.1 *Conceptual ERD*



Gambar 3.22: *Conceptual ERD*

3.2.2.1.2 Detailed ERD

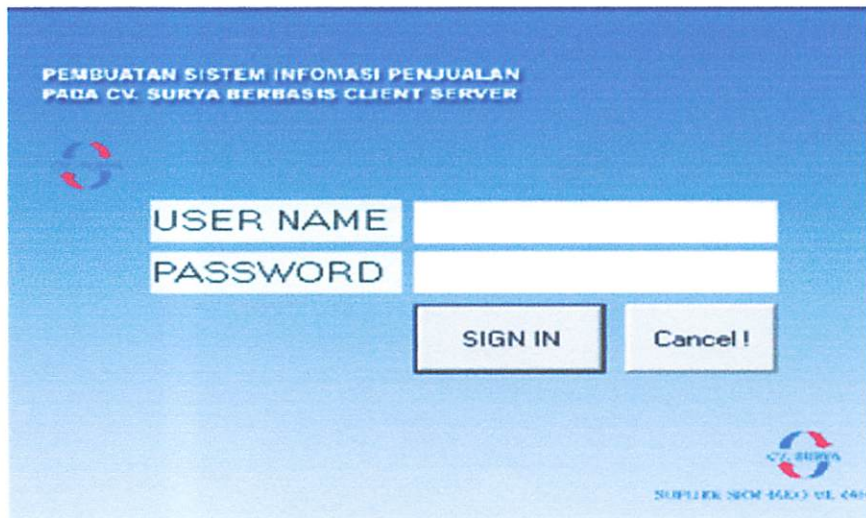


Gambar 3.23: Detailed ERD

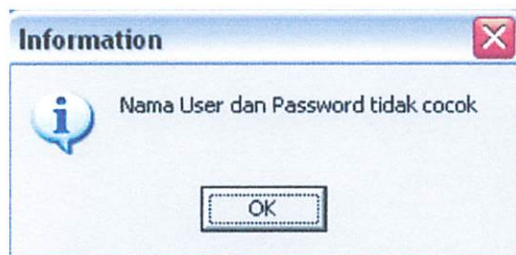
## BAB IV IMPLEMENTASI DAN PENGUJIAN

### 4.1 Form Login

Saat pertama kali menjalankan program akan muncul *form login* untuk tiap-tiap user yang telah di tentukan hak aksesnya oleh Administrator. *User* akan diminta untuk memasukkan nama dan *password*-nya. Apabila data user tidak ada atau tidak sama dengan data yang terdapat pada *database* maka akan muncul informasi yang berisi ” Nama User Dan Password tidak Cocok”, sehingga tidak dapat masuk ke dalam program. Untuk lebih jelasnya tampak pada gambar 4.1 dan 4.2 di bawah ini :



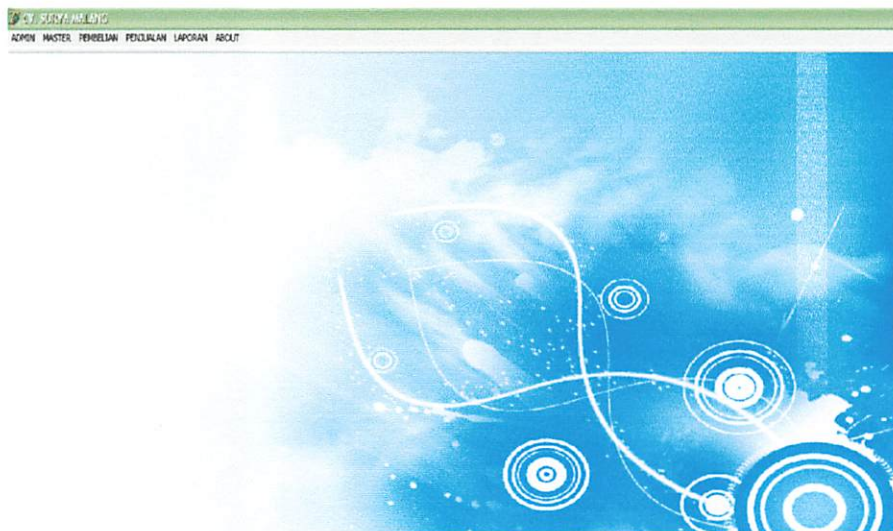
Gambar 4.1 Tampilan Form Login User



Gambar 4.2 Tampilan Form Nama User dan Password Salah

## 4.2 *Form Utama*

Setelah akses login disetujui, maka akan mengaktifkan tombol-tombol yang diperlukan untuk transaksi sesuai dengan level akses user yang telah ditentukan oleh Administrator.



Gambar 4.3 Tampilan *Form Utama Administrator*

## 4.3 *Form Level User*

*Form Level User* mempunyai fasilitas untuk menambah *user* baru, hapus *user* dan menentukan hak akses dari *user* tersebut, dan ini hanya bisa dilakukan oleh administrator. Terlihat pada gambar 4.4, untuk menjalankan fungsi-fungsi tombol-tombol di dalamnya.

Gambar 4.4 Tampilan Form Setlogin

#### 4.4 Menu Master

Menu master merupakan menu yang bisa di akses oleh semua user. Menu ini terdiri dari *form* data barang, *form* data *supplier*, dan *form* data *custome*. Menu ini hanya berupa laporan yang digunakan untuk memudahkan user dalam menjalankan tugasnya masing-masing.

##### 4.4.1 *Form* data Barang

*Form* ini berisi tentang laporan data barang dan stok barang, seperti pada gambar 4.5.

KODE	NAMA	STOK	HPP	NILAI	HARG. JUAL
20011	BERAS TAWDN 25KG	8	132.000	1.056.000	135.000
20012	BERAS MENTARI 25 KG	9	128.000	1.152.000	130.000
20021	GULA PASIR 50 KG	10	290.000	2.900.000	295.000
20022	GULA PASIR 25 KG	10	150.000	1.500.000	155.000
20031	MINYAK GORENG BIMOLI 5 L	25	34.000	850.000	35.000
20032	MINYAK GORENG BIMOLI 1 L	50	18.000	900.000	19.000
20033	MINYAK GORENG TROPICAL 5 L	25	38.000	950.000	40.000
20034	MINYAK GORENG TROPICAL 1 L	50	20.000	1.000.000	21.000
20041	TELUR 1 KG	50	7.000	350.000	8.000
20051	LPG 3 KG	30	12.000	360.000	12.500
20052	LPG 12 KG	15	72.000	1.080.000	75.000

Gambar 4.5 Tampilan *Form* laporan data barang

#### 4.4.2 Form data Supplier

Form ini berisi tentang laporan data *supplier*, seperti pada gambar 4.5

KODE	NAMA_PT	ALAMAT
102	MTR KEDIRI	Kediri
103	SAKTI PERSADA RAYA	Blitar
104	PT. MASON NARAJI	Malang
105	PT. MULTIMAS NABATI ASAHAN	Surabaya
106	CV. CIPTA MANDIRI	Wonokerto Kab. Malang
107	CV. MEGA KENCANA	Jl. Tirta gangga 31 Malang
101	UD. SARI TANI MOJOKERTO	

Gambar 4.6 Tampilan Form laporan data supplier

#### 4.4.3 Form data Customer

Form ini berisi tentang laporan data *customer*, seperti pada gambar 4.5

KODE	NAMA	ALAMAT	TELP
1001	PAK YUDI	JL. BUNGA KUMIS KUCING 36 F	
1002	YUS	SUMBERSARI	
1003	BU MEGARIA FIRLANASARI	JL. TIRTA GANGGA 31	0341 333280

Gambar 4.7 Tampilan Form laporan data customer

## 4.5 Menu Pembelian

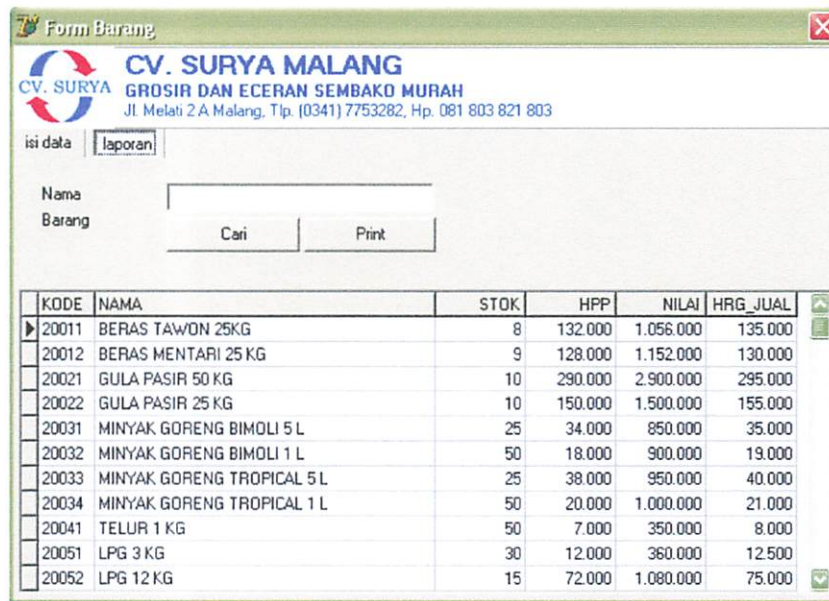
Isi dari menu pembelian ini adalah hal-hal yang menyangkut proses pembelian. Pada menu ini terdapat beberapa macam *form* antara lain, *form* barang, *form* supplier, *form* pembelian, *form* retur pembelian.

### 4.5.1 Form Barang

Pada *form* barang ini berisi menu isi data dan laporan. Dimana pada menu isi data hanya bisa digunakan untuk mengganti harga jual dari barang yang telah di beli dari supplier seperti pada gambar 4.8. Sedangkan pada menu laporan berisi tentang laporan data barang yaitu: stok, Hpp(harga pokok pembelian), nilai, dan harga jual seperti pada gambar 4.9

The image shows a screenshot of a software application window titled "Form Barang". The window has a header section with a logo for "CV. SURYA" and the text "CV. SURYA MALANG GROSIR DAN ECERAN SEMBAKO MURAH" along with contact information: "Jl. Melati 2 A Malang, Tlp (0341) 7753282, Hp 081 803 821 803". Below the header, there are two tabs: "isi data" (which is selected) and "laporan". The "isi data" tab contains a form with several input fields: "KODE", "NAMA", "STOK" (with a value of 0), "HPP" (with a value of 0), "NILAI" (with a value of 0), "HARGA JUAL" (with a value of 0), and "STOK MINIMAL" (with a value of 0). At the bottom of the form, there are two buttons: "Save" and "Clear".

Gambar 4.8 *Form* barang pada menu Isi data



KODE	NAMA	STOK	HPP	NILAI	HRG_JUAL
20011	BERAS TAWDN 25KG	8	132.000	1.056.000	135.000
20012	BERAS MENTARI 25 KG	9	128.000	1.152.000	130.000
20021	GULA PASIR 50 KG	10	290.000	2.900.000	295.000
20022	GULA PASIR 25 KG	10	150.000	1.500.000	155.000
20031	MINYAK GORENG BIMOLI 5 L	25	34.000	850.000	35.000
20032	MINYAK GORENG BIMOLI 1 L	50	18.000	900.000	19.000
20033	MINYAK GORENG TROPICAL 5 L	25	38.000	950.000	40.000
20034	MINYAK GORENG TROPICAL 1 L	50	20.000	1.000.000	21.000
20041	TELUR 1 KG	50	7.000	350.000	8.000
20051	LPG 3 KG	30	12.000	360.000	12.500
20052	LPG 12 KG	15	72.000	1.080.000	75.000

Gambar 4.9 *Form* barang pada menu Laporan

#### 4.5.2 *Form Supplier*

Pada *form supplier* yang dapat dilihat pada gambar 4.10 dan 4.11. Dimana pada *form supplier* ini terdapat dua menu utama yaitu menu isi data dan menu laporan. Dimana pada menu isi data digunakan untuk menginputkan data dari masing-masing *supplier*. Dan pada menu laporan data berisi laporan data dari *supplier* yang telah di inputkan pada menu isi data dan terdapat menu *search* dan *print* untuk memudahkan dalam pencarian dan print laporan.



**Form Suplier**

**CV. SURYA MALANG**  
GROSIR DAN ECERAN SEMBAKO MURAH  
Jl. Melati 2 A Malang, Tlp. (0341) 7753282, Hp. 081 803 821 803

isi data | laporan data

Kode Suplier

Nama Suplier

Alamat

Telepon

Nama Kontak

Handphone

Save Clear

Gambar 4.10 *Form Suplier* pada menu Isi data

**Form Suplier**

**CV. SURYA MALANG**  
GROSIR DAN ECERAN SEMBAKO MURAH  
Jl. Melati 2 A Malang, Tlp. (0341) 7753282, Hp. 081 803 821 803

isi data | laporan data

Nama Suplier  Cari Print

KODE	NAMA_PT	ALAMAT
102	MTR KEDIRI	Kediri
103	SAKTI PERSADA RAYA	Blitar
104	PT. MASON NARAJI	Malang
105	PT. MULTIMAS NABATI ASAHAN	Surabaya
106	CV. CIPTA MANDIRI	Wonokerto Kab. Malang
107	CV. MEGA KENCANA	Jl. Tirta gangga 31 Malang
101	UD. SARI TANI MOJOKERTO	

Gambar 4.11 *Form Suplier* pada menu Laporan data

#### 4.5.3 Form Pembelian

Pada *form* pembelian merupakan perjanjian antara perusahaan dengan *supplier* tentang nama, jumlah, dan harga yang dibeli oleh perusahaan. Pada *form* pembelian ini terdapat dua menu yaitu isi data dan laporan data. Dimana pada menu isi data yang ditunjukkan pada gambar 4.12 terdapat menu input tanggal, kode *supplier*, nama *supplier*, kode barang, nama barang, jumlah, harga dan subtotal. Sedangkan pada menu laporan data yang ditunjukkan pada gambar 4.13, hanya berisi laporan-laporan yang telah di inputkan pada menu isi data, dan laporan pembelian ini secara otomatis langsung terkoneksi dengan form barang.

Form Pembelian

**CV. SURYA MALANG**  
Grosir DAN ECERAN SEMBAKO MURAH  
Jl. Melati 2 A Malang. Tlp. (0341) 7753282. Hp. 081 803 821 803

isi data | laporan data

Tanggal: 1/20/2011 No. Pembelian: 1

Kode Suplier:

Nama Suplier:

No.	Kode	Nama Barang	Jumlah	Harga	Sub Total
1					0
2					0
3					0
4					0
5					0
6					0
7					0
8					0
9					0
10					0
11					0

Total Beli 0

Clear Save

Gambar 4.12 Form pembelian pada menu Isi data

**Form Pembelian**

**CV. SURYA MALANG**  
Grosir DAN ECERAN SEMBAKO MURAH  
Jl. Melati 2 A Malang. Tlp. (0341) 7753282. Hp. 081 803 821 803

isi data | laporan data

Mulai Tanggal: 11/02/2011
  Sampai Tanggal: 11/02/2011

Cari Print

NOMER	TANGGAL	KODE	NAMA	TOTAL_BELI
1	09/02/2011	101	UD. SARITANI MOJOKERTO	1.320.000
2	09/02/2011	102	MTR KEDIRI	1.280.000
3	09/02/2011	103	SAKTI PERSADA RAYA	4.400.000
4	09/02/2011	104	PT. MASON NARAJI	1.440.000
5	09/02/2011	105	PT. MULTIMAS NABATI ASAHAN	3.700.000
6	09/02/2011	107	CV. MEGA KENCANA	810.000
7	09/02/2011	106	CV. CIPTA MANDIRI	350.000

KODE	NAMA	JML_BELI	HRG_BELI	SUB_TOTAL
20011	BERAS TAWON 25KG	10	132.000	1.320.000

Gambar 4.13 Form pembelian pada menu Laporan data

**LAPORAN PEMBELIAN**

**CV. SURYA MALANG**  
GROSIR DAN ECERAN SEMBAKO MURAH  
Jl. Melati 2 A Malang. Tlp. (0341) 7753282. Hp. 081 803 821 803

No Beli	: 5
Tanggal	: 09/02/2011
Kode Produsen	: 105
Nama Produsen	: PT. MULTIMAS NABATI ASAHAN

Kode	Nama Barang	Jml Jual	Harga	Sub Total
20032	MINYAK GORENG BIMOLI 1 L	50	18.000	900.000
20031	MINYAK GORENG BIMOLI 5 L	25	34.000	850.000
20034	MINYAK GORENG TROPICAL 1 L	50	20.000	1.000.000
20033	MINYAK GORENG TROPICAL 5 L	25	38.000	950.000

Total Pembelian : 3.700.000

Gambar 4.14 Tampilan Cetak struk Pembelian

#### 4.5.4 Form Retur Pembelian

Berisi tentang daftar retur yang telah dilayani berdasarkan faktur pembelian yang telah diproses. Barang-barang yang diretur akan dikembalikan ke pihak *supplier* untuk diproses.

Gambar 4.15 Form retur pembelian pada menu *Update*

No Retur	Tgl Retur	No Beli	Tgl Beli	Kode	Nama Produsen	Total Beli	Total Retur
1	10/11/2010	1	10/11/2010	001	BULOG MALANG	55,000,000	1,100,000

Kode	Nama Barang	Jml Retur	Harga Jual	Sub Total Retur	Keterangan
001	LPG 12 KG	2	50,000	100,000	Salah Kirim
002	BERAS MENTARI 100KG	2	500,000	1,000,000	Salah Kirim

Gambar 4.16 Form retur pembelian pada menu *read*

## 4.6 Menu Penjualan

### 4.6.1 Form Penjualan

Isi dari menu penjualan ini adalah hal-hal yang menyangkut proses penjualan. Setiap kali terdapat penjualan barang ke *customer* harus membuat penjualan. *Customer* membayar berdasarkan nota penjualan.

Gambar 4.17 Form penjualan

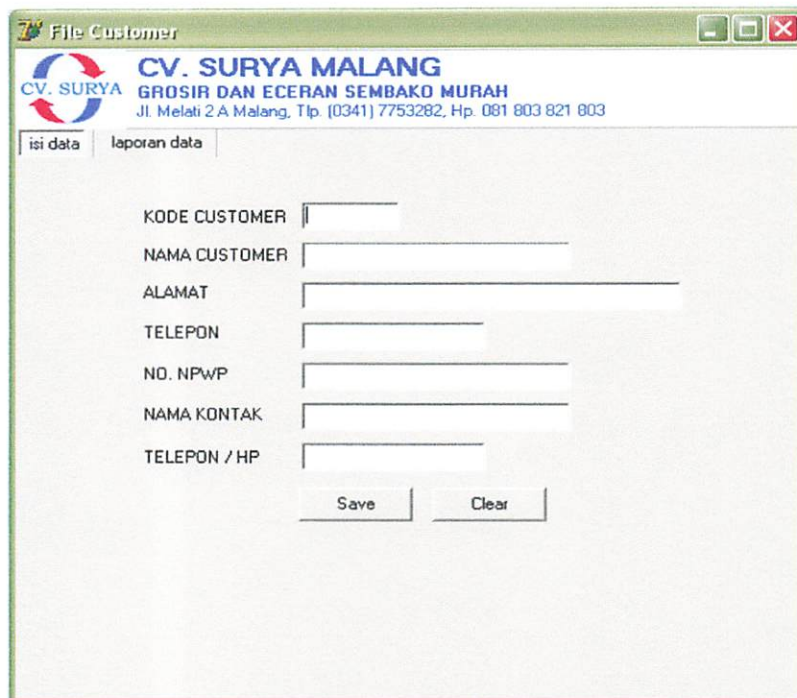
Kode	Nama Barang	Jml Jual	Harga	Sub Total
20012	BERAS MENTARI 25 KG	1	130000	130000

Total Penjualan : 130000  
Bayar Tunai : 130000

Gambar 4.18 Cetak Struk Penjualan

#### 4.6.2 Form Customer

*Form Customer* adalah form yang digunakan untuk menginputkan data *customer* untuk mengetahui apakah *customer* tersebut menjadi member atau bukan. sehingga harga jual untuk member dan bukan member bisa dibedakan dan untuk *customer member* bisa melakukan pembelian secara kredit untuk pembayarannya.



The screenshot shows a software window titled "File Customer". At the top left is the CV. SURYA logo, and to its right is the text "CV. SURYA MALANG" followed by "GROSIR DAN ECERAN SEMBAKO MURAH" and the address "Jl. Melati 2 A Malang, Tlp. (0341) 7753282, Hp. 081 803 821 803". Below this is a tabbed interface with "isi data" selected. The form contains the following fields:

- KODE CUSTOMER
- NAMA CUSTOMER
- ALAMAT
- TELEPON
- NO. NPWP
- NAMA KONTAK
- TELEPON / HP

At the bottom of the form are two buttons: "Save" and "Clear".

Gambar 4.19 *Form Customer* pada menu Isi Data

File Customer

CV. SURYA  
**CV. SURYA MALANG**  
 GROSIR DAN ECERAN SEMBAKO MURAH  
 Jl. Melati 2 A Malang, Tlp. (0341) 7753282, Hp. 081 803 821 803

isi data | laporan data

NAMA CUSTOMER

Cari Print

KODE	NAMA	ALAMAT
▶ 1001	PAK YUDI	JL. BUNGA KUMIS KUCING 36 F
1002	YUS	SUMBERSARI
1003	BU MEGARIA FIRLANASARI	JL. TIRTA GANGGA 31

Gambar 4.20 *Form Customer* pada menu Laporan Data

#### 4.6.3 *Form* Laporan Rugi laba

Pada form Laporan rugi laba ini digunakan untuk memantau atau mengecek laporan rugi laba pertanggal yang sudah terakumulasi.

Laporan Rugi Laba

Mulai Tanggal 04/11/2010

Sampai Tanggal 14/11/2010

Batal Print

Gambar 4.21 *Form* Laporan Rugi Laba

#### 4.6.4 Form Bayar Piutang

Pada *form* ini berisi tentang daftar piutang toko/ customer yang harus dilunasi berdasarkan nomer faktur penjualan. Laporan ini didasarkan pada tanggal bayar *customer*-nya. Pada *form* bayar piutang ini terdiri dari tiga menu yaitu menu daftar piutang pada gambar 4.17, menu bayar piutang pada gambar 4.18, menu jatuh tempo pada gambar 4.19.

Gambar 4.22 *Form* daftar piutang

Pada menu bayar piutang ini berisi tentang pelunasan piutang yang telah dilakukan oleh *customer* yang sudah tercatat pada waktu transaksi penjualan.



Gambar 4.23 Form bayar piutang

Pada menu jatuh tempo ini berisi tentang jatuh tempo *customer* yang melakukan pembelian secara kredit dan jatuh temponya ditentukan berdasarkan pada tanggal waktu *customer* melakukan pembelian.

Gambar 4.24 Form Jatuh tempo

## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Setelah melakukan analisa, merancang dan mengimplementasikan Sistem Informasi penjualan sembako berbasis client server, maka diperoleh kesimpulan sebagai berikut:

1. Dengan dikembangkannya sistem informasi ini dapat mempermudah dan mempercepat aktifitas kerja pada CV. Surya, baik dalam menangani transaksi penjualan, pembelian dan pembuatan laporan-laporan.
2. Dengan adanya sistem informasi ini akan dapat mempersingkat waktu dalam pencarian data dan perhitungan laba rugi yang lebih akurat.
3. Output dari Sistem Informasi penjualan Sembako berupa laporan-laporan yang akan mempermudah pimpinan dalam pengontrolan CV itu sendiri.

#### **5.2 Saran**

Dalam perencanaan dan pembuatan aplikasi ini terdapat beberapa hal yang perlu diperhatikan dalam pengembangan lebih lanjut, yaitu :

1. Aplikasi ini masih dapat dikembangkan lebih jauh lagi ke dalam bentuk web yang terkoneksi dengan internet untuk memudahkan pelayanan kepada pelanggan dan bisa memberikan informasi lebih luas kepada masyarakat.
2. Diharapkan pada pengembangan aplikasi program sistem informasi penjualan berikutnya, dapat di tambahkan *notification limit stok* pada *form inventory* untuk mengantisipasi terjadinya kekososngan stok (*empty stok* ).

## DAFTAR PUSTAKA

1. A, James, O'Brien, 2004, *Management Information System-Managing Information Technology in the Business Enterprise*, McGraw-Hill, Inc.
2. Abdia, Away, Gunaidi, 2008, *The Shourtcut Of Delphi for Accounting*, Informatika , Bandung.
3. C, Kenneth, Laudon dan P, 2005, Jane, Laudon, *Sistem Informasi Manajemen, Mengelola Perusahaan Digital*, Andi, Yogyakarta.
4. Dewa, Hasta, Putranta, ST, 2004, MM, *Pengantar Sistem dan Teknologi Informasi*, Amus, Yogyakarta.
5. K, Brian, Williams dan C, 2005, Stacey, Sawyer, *Using Information Technology, A Practical Introduction to Computers & Communications*, McGraw-Hill.
6. L, Jeffrey, Whitten, D, Lonnie, Bently dan C, Kevin, Dittman, 2004, *System Analysis and Design Methods Sixth Edition*, McGraw-Hill, Inc.
7. Mark Whitehorn dan Bill Marklyn, 2003, *Seluk Beluk Database Relasional*, Erlangga, Jakarta.
8. M., Thomas, Connolly dan E, Carolyn, Begg, 2001, *Database System A Practical Approach to Design, Implementation, and Management*, University of Paisley.
9. McFadden, Fred R. dan Hoffer, Jeffrey A, 1994, *Modern Database Management*, The Benjamin/Cummings Publishing Company, Inc., United States of America.
10. Sutedjo, Budi dan Oetomo, Dharma, S.Kom, MM, 2002, *Perancangan dan Pembangunan Sistem Informasi*, Andi, Yogyakarta.



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG

## INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

---

---

### BERITA ACARA UJIAN SKRIPSI FAKULTAS TEKNOLOGI INDUSTRI

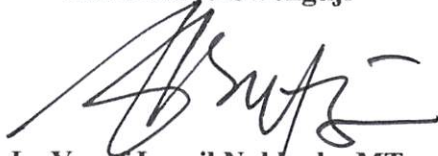
Nama : Gunawan Sasmito Adi Wibowo  
NIM : 06.12.574  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika  
Judul Skripsi : Pengembangan Sistem Informasi Penjualan Sembako Pada CV. Surya  
Malang Berbasis Client Server.

Dipertahankan di hadapan tim penguji ujian skripsi jenjang Strata Satu ( S-1 ) pada :


Hari : Senin  
Tanggal : 14 Februari 2011  
Dengan Nilai : 78,7 (B+) *o*

#### Panitia Ujian Skripsi,

**Ketua Majelis Penguji**


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

**Sekretaris Majelis Penguji**


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

#### Anggota Penguji,

**Dosen Penguji I**

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

**Dosen Penguji II**

  
Sandy Nataly M, S.Kom  
NIP.P.1030800418



PERKUMPULAN PENGELOLA PENDIDIKAN UMUM DAM TEKNOLOGI NASIONAL MALANG

# INSTITUT TEKNOLOGI NASIONAL MALANG

FAKULTAS TEKNOLOGI INDUSTRI  
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN  
PROGRAM PASCASARJANA MAGISTER TEKNIK

PT. BNI (PERSERO) MALANG  
BANK NIAGA MALANG

Kampus I : Jl. Bendungan Sigura-gura No. 2 Telp. (0341) 551431 (Hunting), Fax. (0341) 553015 Malang 65145  
Kampus II : Jl. Raya Karanglo, Km 2 Telp. (0341) 417636 Fax. (0341) 417634 Malang

## FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan Ujian Skripsi Jenjang Program Strata Satu (S-1) Jurusan Teknik Elektro Konsentrasi Teknik Komputer & Informatika, maka perlu adanya perbaikan Skripsi untuk mahasiswa :

Nama : Gunawan Sasmito Adi Wibowo  
NIM : 06.12.574  
Jurusan : Teknik Elektro S-1  
Konsentrasi : Teknik Komputer & Informatika  
Masa Bimbingan : 24 November 2010 s/d 24 April 2011  
Judul Skripsi : Pengembangan Sistem Informasi Penjualan Sembako Pada CV. Surya Malang Berbasis Client Server.

Tanggal	Uraian	Paraf
Penguji II 14 Februari 2011	<ol style="list-style-type: none"><li>1. Penulisan-penulisan di laporan dari abstraksi sampai akhir(ada coretan-coretan perbaikan)</li><li>2. DFD :<ul style="list-style-type: none"><li>- Hilangkan kata-kata laporan dari sistem ke keuangan.</li><li>- Dari pemilik ada penentuan harga barang.</li></ul></li><li>3. Kesimpulan kurang Detail.</li><li>4. Hapus Saran No.2 dan cari saran yang berupa ide ide yang mengembangkan program.</li><li>5. Daftar pustaka menggunakan number. Tidak ada membahas tentang Delphi (ditambah referensi Delphi).</li></ol>	

Mengetahui,

Dosen Pembimbing I

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

Dosen Pembimbing II

A. Faisol, ST  
NIP.P.1031000431

Dosen Penguji,

Dosen Penguji I

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

Dosen Penguji II

Sandy Nataly M, S.Kom  
NIP.P.1030800418



**CV. SURYA MALANG**  
**GROSIR DAN ECERAN SEMBAKO MURAH**  
Jl. Melati 2 A Malang, Flexi (0341) 7753282, Hp 081803821803

Malang, 10 Agustus 2010

Sifat : Biasa  
Lampiran : -  
Perihal : Survey

Kepada :  
Yth. Dekan Fakultas Teknologi Industri  
Institut Teknologi Nasional Malang  
Jl. Bendungan Sigura-gura No. 2  
MALANG

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Alhamdulillah, puji syukur kami panjatkan kehadiran Allah SWT, dengan curahan rahmat-Nya semoga kita senantiasa sehat walafiat sehingga dapat melaksanakan aktifitas sehari-hari dengan baik.

Dengan ini diberitahukan bahwa kami dapat menerima mahasiswa anda :

Yang bernama : **Gunawan Sasmito Adi W**  
NIM : **06.12.574**  
Fakultas : **Teknologi Industri**  
Jurusan : **Teknik Elektro S-1**  
Konsentrasi : **Teknik Komputer dan Informatika S-1**

untuk mencari informasi dan bahan/data untuk penyusunan Skripsi dengan ketentuan sebagai berikut :

1. Bahan/data yang didapat dari CV. Surya Hanya Untuk Keperluan Skripsi
2. Selama melaksanakan Survey agar mahasiswa mematuhi segala peraturan yang berlaku di CV. Surya Malang

Demikian untuk diketahui dan dilaksanakan sebagaimana mestinya, terima kasih.  
*Wassalamualikum Warahmatullahi Wabarakatuh*

Malang 10 Agustus 2010

Manager/Pemilik

CV. SURYA

SETIAWAN



## FORMULIR BIMBINGAN SKRIPSI

Nama : Gunawan Sasmito Adi Wibowo  
Nim : 06.12.574  
Masa Bimbingan : 24 November 2010 s/d 24 Mei 2011  
Judul Skripsi : Pengembangan Sistem Informasi Penjualan Sembako Pada CV.  
Surya Malang Berbasis Client Server.

No	Tanggal	Uraian	Paraf Pembimbing
1	26-11-2010	Acc BAB I	
2	29-11-2010	Acc BAB II	
3	30-11-2010	Acc BAB III	
4	06-12-2010	Revisi BAB IV; Ganti Tampilan Password	
5	07-12-2010	Acc BAB IV	
6	07-12-2010	Revisi Bab V; Penjelasan Kesimpulan dan Saran	
7	8-12-2010	Acc BAB V	
8			
9			
10			

Malang, Februari 2011  
Dosen Pembimbing I

**(M. Ibrahim Ashari ST, MT)**

NIP.P.1030100358



## FORMULIR BIMBINGAN SKRIPSI

Nama : Gunawan Sasmito Adi Wibowo  
Nim : 06.12.574  
Masa Bimbingan : 24 November 2010 s/d 24 Mei 2011  
Judul Skripsi : Pengembangan Sistem Informasi Penjualan Sembako Pada CV.  
Surya Malang Berbasis Client Server.

No	Tanggal	Uraian	Paraf Pembimbing
1	29-11-2010	Acc BAB I dan II	
2	30-11-2010	Revisi DFD BAB III	
3	02-12-2010	Revisi DFD BAB III	
4	14-12-2010	Acc BAB III	
5	17-12-2010	Revisi BAB IV; Ganti Tampilan Password	
6	19-12-2010	Acc BAB IV	
7	19-12-2010	Revisi Bab V; Penjelasan Kesimpulan dan Saran	
8	20-12-2010	Acc BAB V	
9			
10			

Malang, Februari 2011  
Dosen Pembimbing II

(A. Fajsel, ST)  
NIP.P.1031000431



```
unit UJual;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ComCtrls, StdCtrls, Grids, ExtCtrls, Db, DBTables, DBGrids, Menus,  
Buttons, jpeg;
```

```
type
```

```
TFJual = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Panel1: TPanel;
  Edit1: TEdit;
  Panel2: TPanel;
  S: TStringGrid;
  Panel3: TPanel;
  Label5: TLabel;
  Button1: TButton;
  DBGrid1: TDBGrid;
  Panel4: TPanel;
  Button2: TButton;
  Q1: TQuery;
  DataSource1: TDataSource;
  DateTimePicker2: TDateTimePicker;
  DateTimePicker3: TDateTimePicker;
  CheckBox1: TCheckBox;
  CheckBox2: TCheckBox;
  Q2: TQuery;
  DataSource2: TDataSource;
  DataSource3: TDataSource;
  PopupMenu1: TPopupMenu;
  Hapus1: TMenuItem;
  Label12: TLabel;
  Panel5: TPanel;
  Edit2: TEdit;
  Panel6: TPanel;
  DateTimePicker4: TDateTimePicker;
  Label13: TLabel;
  Label14: TLabel;
  Label15: TLabel;
  DateTimePicker1: TDateTimePicker;
  Splitter1: TSplitter;
  DBGrid2: TDBGrid;
  Q1NOMER_FAKTUR: TStringField;
  Q1TGL: TDateTimeField;
  Q1KODE_CUST: TStringField;
  Q1NAMA: TStringField;
  Q1ALAMAT: TStringField;
  Q1TOTAL_JUAL: TFloatField;
  Q1TUNAI: TFloatField;
  Q1HUTANG: TFloatField;
  Q1DIBAYAR: TFloatField;
  Q1SALDO: TFloatField;
  Q1JATUH_TEMPO: TDateTimeField;
  Q2NOMER: TIntegerField;
  Q2NO_FAKTUR: TStringField;
  Q2TGL: TDateTimeField;
  Q2KODE: TStringField;
  Q2NAMA: TStringField;
  Q2JML_JUAL: TFloatField;
  Q2HARGA_JUAL: TFloatField;
  Q2SUB_TOTAL: TFloatField;
  Q2HPP: TFloatField;
  Q2NILAI: TFloatField;
```

jual

```
Q2LABA: TFloatField;
Q2JML_RETUR: TFloatField;
Q2NILAI_RETUR: TFloatField;
Q2NILAI_HPP_RETUR: TFloatField;
Q2RUGI: TFloatField;
Q2LABA_BERSIH: TFloatField;
Button3: TButton;
Panel7: TPanel;
Image2: TImage;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
procedure FormShow(Sender: TObject);
procedure Edit1Exit(Sender: TObject);
procedure SKeyPress(Sender: TObject; var Key: Char);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure Button1Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Hapus1Click(Sender: TObject);
procedure Edit2Exit(Sender: TObject);
procedure DateTimePicker1KeyPress(Sender: TObject; var Key: Char);
procedure PageControl1Change(Sender: TObject);
procedure Edit2KeyPress(Sender: TObject; var Key: Char);
procedure SExit(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure DateTimePicker4Exit(Sender: TObject);
procedure DateTimePicker4Change(Sender: TObject);
procedure DateTimePicker1Change(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
  procedure SubTotal ;
public
  { Public declarations }
end;

var
  FJual: TFJual;

implementation

uses UDM, UMain, UCustomer, UPenjualan_R, Ubarang;

{$R *.DFM}

procedure TFJual.FormShow(Sender: TObject);
var i : Integer ;
begin
  // PageControl1.ActivePageIndex := 0 ;
  // Edit1.Setfocus ;
  Panel1.Caption := IntToStr(DM.MJual.RecordCount+1) ;
  DateTimePicker1.DateTime := Date ;
  DateTimePicker4.DateTime := Date + 30 ;

  S.Cells[0,0] := 'No.' ;
  S.Cells[1,0] := 'Kode' ;
  S.Cells[2,0] := 'Nama Barang' ;
  S.Cells[3,0] := 'Stok' ;
  S.Cells[4,0] := 'Jumlah' ;
  S.Cells[5,0] := 'Harga' ;
  S.Cells[6,0] := 'Sub Total' ;

  S.Colwidths[0] := 30 ;
  S.Colwidths[1] := 60 ;
  S.Colwidths[2] := 200 ;
  S.Colwidths[3] := 65 ;
  S.Colwidths[4] := 65 ;
  S.Colwidths[5] := 65 ;
  S.Colwidths[6] := 65 ;
```

jual

```
for i := 1 to S.RowCount -1 do
begin
  S.Cells[0,i]:= IntToStr(i) ;
end ;

DateTimePicker1.Date := Now ;
DateTimePicker1.DateTime := Date;
DateTimePicker2.Date := Now ;
DateTimePicker2.DateTime := Date;
DateTimePicker3.Date := Now ;
DateTimePicker3.DateTime := Date;
```

d;

```
procedure TFJual.Edit1Exit(Sender: TObject);
```

```
begin
```

```
  panel2.caption := '' ;
  if edit1.Text <> '' then
  begin
    if dm.Customer.FindKey([edit1.text]) then
    begin
      panel2.Caption := dm.Customer.fieldbyname('Nama').AsString ;
      panel5.Caption := dm.Customer.fieldbyname('Alamat').AsString ;
      subtotal ;
    end
    else
    begin
      subtotal ;
      messageDlg('Kode customer tidak ditemukan', mtInformation, [mbOK],0) ;
      FCustomer.ShowModal ;
      edit1.setfocus ;
      exit ;
    end ;
  end ;
```

d;

```
procedure TFJual.SKeyPress(Sender: TObject; var Key: Char);
```

```
var j, baris : Integer ;
```

```
begin
```

```
  if S.Col in [1,4] then
  begin
    if not (key in ['0'..'9', #8, #13]) then key := #0 ;
  end ;

  if ((S.col = 1) and (key = #13)) then
  begin
    baris := S.Row ;
    for j := 1 to S.RowCount-1 do
    begin
      if j <> baris then
      begin
        if S.Cells[1,j] = S.Cells[1,baris] then
        begin
          S.Cells[1,baris] := '' ;
        end ;
      end ;
    end ;
    Dm.Barang.Refresh;
    if Dm.Barang.FindKey([S.Cells[1,S.Row]]) then
    begin
      S.Cells[2,S.Row] := Dm.Barang.fieldbyname('Nama').AsString ;
      S.Cells[3,S.Row] := FormatFloat('#,0',Dm.Barang.fieldbyname('Stok').AsFloat) ;
      S.Cells[5,S.Row] := FormatFloat('#,0',Dm.Barang.fieldbyname('Hrg_Jual').AsFloat) ;
    end
    else
    begin
      messageDlg('Kode Barang tidak ditemukan atau sudah ada', mtInformation, [mbOK], 0);
      FBarang.ShowModal ;
      S.Col := 1 ;
      exit ;
    end ;
  end ;
```

2.

jual

end ;

end ;

```
if S.Col in [2,3,5,6] then key := #0 ;
if ((S.Col in [4]) and (key = #13)) then
begin
  if FMain.NR(S.Cells[4,S.Row]) > FMain.NR(S.Cells[3,S.Row]) then
    S.Cells[4,S.Row] := S.Cells[3,S.Row] ;

  S.Cells[4,S.Row] := formatFloat('#,0', FMain.NR(S.Cells[4,S.Row])) ;
end ;
```

```
if key = #13 then
  Begin
    subtotal ;
    If S.Col=1 Then S.Col :=4 Else
      Begin
        If S.Col=4 Then
          Begin
            S.Col :=1;
            S.Row:=S.Row+1;
          End;
        End;
      End;
  End;
End;
```

nd;

procedure TFJual.SubTotal ;

```
var i : Integer ;
total : Double ;
cek : boolean ;
begin
  total := 0 ;
  cek := true ;

  for i := 1 to S.RowCount-1 do
  begin
    S.Cells[6,i] := formatfloat('#,0', FMain.NR(S.Cells[4,i])*FMain.NR(S.Cells[5,i])) ;
    total := total + FMain.NR(S.Cells[6,i]) ;

    If ((fmain.NR(s.cells[6,i]) <=0) And (s.cells[2,i] <> '')) Then
      cek := false ;
    end ;
    panel3.caption := formatfloat('#,0', total) ;
    panel6.caption := formatfloat('#,0', total) ;
    if ((panel2.caption <> '') and (cek = true) and (total > 0)) then
      button1.enabled := true
    else button1.enabled := false ;
  }
end ;
```

nd ;

procedure TFJual.Edit1KeyPress(Sender: TObject; var Key: Char);

```
begin
  if not(key in ['0'..'9', #13, #8]) then key := #0 ;
  If (Key=Chr(VK_return)) Then SendMessage(Handle, WM_NEXTDLGCTL, 0, 0);
end;
```

procedure TFJual.Button1Click(Sender: TObject);

```
var i, Nomer : Integer ;
ada_user_Lain_mengurangi: Boolean;
begin
  //Mengecek Client Server
  Ada_user_Lain_mengurangi:=False;
  for i:=1 to s.rowcount-1 do
  begin
    Dm.Barang.REFRESH;
    if Dm.Barang.FindKey([S.Cells[1,i]]) then
      begin
```

```

                                jual
s.Cells[2,i] := Dm.Barang.fieldbyname('Nama').AsString ;
s.Cells[3,i] := FormatFloat('#,0',Dm.Barang.fieldbyname('Stok').AsFloat) ;
s.Cells[5,i] := FormatFloat('#,0',Dm.Barang.fieldbyname('Hrg_Jual').AsFloat) ;

If fmain.NR(s.cells[4,i]) > fmain.NR(s.cells[3,i]) Then
  Ada_user_Lain_mengurangi:=True;
End;
end;

If Ada_user_Lain_mengurangi=True Then
  Begin
    MessageDlg('Stok tidak cukup !'+#13+
      'Karena ada user lain yang mengurangi'+#13+
      'Cek lagi stoknya', mtInformation, [mbOK],0);
    //SpeedButton1.Enabled:=False;
    s.SetFocus;
    Exit;
  End;

DM.MJual.Insert ;
DM.MJual.FieldByName('Nomer_Faktur').AsInteger := DM.MJual.RecordCount + 1 ;
Nomer := DM.MJual.FieldByName('Nomer_Faktur').AsInteger ;
DM.MJual.FieldByName('Tgl').AsDateTime:= DateTimePicker1.Date ;
DM.MJual.FieldByName('Kode_Cust').AsString := Edit1.Text ;
DM.MJual.FieldByName('Nama').AsString := Panel2.Caption ;
DM.MJual.FieldByName('Alamat').AsString := Panel5.Caption ;

DM.MJual.FieldByName('Total_Jual').AsFloat := FMain.NR(Panel3.Caption) ;
DM.MJual.FieldByName('Tunai').AsFloat := FMain.NR(Edit2.Text) ;
DM.MJual.FieldByName('Hutang').AsFloat := FMain.NR(Panel6.Caption) ;
DM.MJual.FieldByName('Saldo').AsFloat := FMain.NR(Panel6.Caption) ;
If DateTimePicker4.Enabled=True Then
DM.MJual.FieldByName('Jatuh_Tempo').AsDateTime := DateTimePicker4.Date ;

DM.MJual.Post ;
DM.MJual.ApplyUpdates ;
DM.MJual.CommitUpdates ;
DM.MJual.Refresh ;

for i := 1 to S.RowCount - 1 do
begin
  if ((S.Cells[1,i] <> '') and (S.Cells[2,i] <> '') and (FMain.NR(S.Cells[6,i]) > 0))
  begin
    DM.DJual.Insert ;

    DM.DJual.FieldByName('Nomer').AsInteger := DM.DJual.RecordCount + 1 ;
    DM.DJual.FieldByName('No_Faktur').AsInteger := Nomer ;
    DM.DJual.FieldByName('Tgl').AsDateTime :=
      Todate(FormatDateTime('dd/mm/yyyy',DateTimePicker1.Date)) ;
    DM.DJual.FieldByName('Kode').AsString := S.Cells[1,i] ;
    DM.DJual.FieldByName('Nama').AsString := (S.Cells[2,i]) ;
    DM.DJual.FieldByName('Jml_Jual').AsFloat := FMain.NR(S.Cells[4,i]) ;
    DM.DJual.FieldByName('Harga_Jual').AsFloat := FMain.NR(S.Cells[5,i]) ;
    DM.DJual.FieldByName('Sub_Total').AsFloat := FMain.NR(S.Cells[6,i]) ;

    if Dm.Barang.FindKey([S.Cells[1,i]]) then
      DM.DJual.FieldByName('HPP').AsFloat := Dm.Barang.fieldbyname('HPP').AsFloat ;

    DM.DJual.FieldByName('Nilai').AsFloat := FMain.NR(S.Cells[4,i]) *
      l.DJual.FieldByName('HPP').AsFloat ;
    DM.DJual.FieldByName('Laba').AsFloat := DM.DJual.FieldByName('Sub_Total').AsFloat -
      l.DJual.FieldByName('Nilai').AsFloat ;
    DM.DJual.FieldByName('Jml_retur').AsFloat := 0 ;

    DM.DJual.FieldByName('Nilai_Retur').AsFloat := 0;
    DM.DJual.FieldByName('Nilai_HPP_Retur').AsFloat := 0 ;
    DM.DJual.FieldByName('Rugi').AsFloat := 0 ;
    DM.DJual.FieldByName('Laba_Bersih').AsFloat := DM.DJual.FieldByName('Laba').AsFloat

```

jual

```
DM.DJual.FieldByName('Rugi').AsFloat ;

DM.DJual.Post ;
DM.DJual.ApplyUpdates ;
DM.DJual.CommitUpdates ;
DM.DJual.Refresh ;

if Dm.Barang.FindKey([S.Cells[1,i]]) then
begin
DM.Barang.edit ;

DM.Barang.FieldByName('Stok').AsFloat := Dm.Barang.FieldByName('Stok').AsFloat -
Main.NR(S.Cells[4,i]) ;
DM.Barang.FieldByName('Nilai').AsFloat := Dm.Barang.FieldByName('Nilai').AsFloat -
FMain.NR(S.Cells[4,i]) * Dm.Barang.FieldByName('hpp').AsFloat) ;

if ((Dm.Barang.FieldByName('Stok').AsFloat > 0) and
Dm.Barang.FieldByName('Nilai').AsFloat > 0)) then
begin
DM.Barang.FieldByName('HPP').AsFloat := Dm.Barang.FieldByName('Nilai').AsFloat /
Dm.Barang.FieldByName('Stok').AsFloat ;
end
else
begin
DM.Barang.FieldByName('Stok').AsFloat := 0 ;
DM.Barang.FieldByName('HPP').AsFloat := 0 ;
DM.Barang.FieldByName('Nilai').AsFloat := 0 ;
end ;

DM.Barang.Post ;
DM.Barang.ApplyUpdates ;
DM.Barang.CommitUpdates ;
DM.Barang.Refresh ;
end ;
end ;
end ;
Edit1.Text := '' ;
Panel2.Caption := '' ;
Panel5.Caption := '' ;
Panel3.Caption := '0' ;
Panel6.Caption := '0' ;
Edit2.Text := '0' ;

for i := 1 to S.RowCount -1 do
begin
S.Cells[0,i] := IntToStr(i) ;
S.Cells[1,i] := '' ;
S.Cells[2,i] := '' ;
S.Cells[3,i] := '' ;
S.Cells[4,i] := '' ;
S.Cells[5,i] := '' ;
S.Cells[6,i] := '0' ;
end ;
Button1.Enabled:=False;
DateTimePicker1.DateTime := Date ;
DateTimePicker4.DateTime := Date + 30 ;
Panel1.Caption := InttoStr(DM.MJual.RecordCount+1) ;
end;

procedure TFJual.CheckBox1Click(Sender: TObject);
begin
if checkbox1.Checked then
begin
DateTimePicker2.Enabled := True ;
DateTimePicker2.Color := clWhite ;
end
else
begin
DateTimePicker2.Enabled := False ;
DateTimePicker2.Color := clMenu ;
```

```
end ;
end;
```

```
procedure TFJual.CheckBox2Click(Sender: TObject);
begin
  if checkbox2.Checked then
    begin
      DateTimePicker3.Enabled := True ;
      DateTimePicker3.Color := clWhite ;
    end
  else
    begin
      DateTimePicker3.Enabled := False ;
      DateTimePicker3.Color := clMenu ;
    end ;
end;
```

```
procedure TFJual.Button2Click(Sender: TObject);
begin
  Q1.Close ;
  Q1.SQL.Clear ;
  Q1.SQL.Add('select * from MJual where Nomer_Faktur > 0') ;

  if CheckBox1.Checked then
    begin
      Q1.SQL.Add('and Tgl >=:xx ') ;
      Q1.ParamByName('xx').AsDateTime := DateTimePicker2.Date ;
    end;

  if CheckBox2.Checked then
    begin
      Q1.SQL.Add('and Tgl <=:yy ') ;
      Q1.ParamByName('yy').AsDateTime := DateTimePicker3.Date ;
    end;

  Q1.prepare ;
  Q1.Open ;

  Q2.Close ;
  Q2.Open ;
end;
```

```
procedure TFJual.Hapus1Click(Sender: TObject);
Var
  I: Integer;
begin
  with S Do
    Begin
      Rows[Row].Clear;
      For i:= Row To RowCount - 1 Do
        Begin
          IF (i <> RowCount - 1) Then
            Begin
              Cells[1,i]:= Cells[1,i+1];
              Cells[2,i]:= Cells[2,i+1];
              Cells[3,i]:= Cells[3,i+1];
              Cells[4,i]:= Cells[4,i+1];
              Cells[5,i]:= Cells[5,i+1];
              Cells[6,i]:= Cells[6,i+1];
            End;
          End;
        SubTotal;
      End;
    end;
```

```
procedure TFJual.Edit2Exit(Sender: TObject);
begin
  If (Fmain.NR(Edit2.Text)>=Fmain.NR(Pane13.Caption)) Then
    Begin
      Edit2.Text:=Pane13.Caption;
      DateTimePicker4.Enabled:=False;
```

jua1

```
DateTimePicker4.Color:=C1Menu;
Panel6.Caption:='0';
End Else
Begin
    Panel6.Caption:=FormatFloat('#,0',Fmain.NR(Panel3.Caption)-Fmain.NR(Edit2.Text));
    DateTimePicker4.Enabled:=True;
    DateTimePicker4.Color:=C1White;
End;

end;

procedure TFJua1.DateTimePicker1KeyPress(Sender: TObject; var Key: Char);
begin
    If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;

procedure TFJua1.PageControl1Change(Sender: TObject);
begin
    If PageControl1.ActivePageIndex=1 Then
        Begin
            Button2Click(Sender) ;
        End;
end;

procedure TFJua1.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
    if not(key in['0'..'9',#13,#8]) then key := #0 ;
    If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;

procedure TFJua1.SExit(Sender: TObject);
begin
    SubTotal;
end;

procedure TFJua1.FormClose(Sender: TObject; var Action: TCloseAction);
var i : integer ;
begin
    Edit1.Text := '' ;
    Panel2.Caption := '' ;
    Panel5.Caption := '' ;
    Panel3.Caption := '0' ;
    Panel6.Caption := '0' ;
    Edit2.Text := '0';

    for i := 1 to S.RowCount -1 do
        begin
            S.Cells[0,i]:= IntToStr(i) ;
            S.Cells[1,i]:= '' ;
            S.Cells[2,i]:= '' ;
            S.Cells[3,i]:= '' ;
            S.Cells[4,i]:= '' ;
            S.Cells[5,i]:= '' ;
            S.Cells[6,i]:= '0' ;
        end ;
end;

procedure TFJua1.DateTimePicker4Exit(Sender: TObject);
begin
    If DateTimePicker4.date <= DateTimePicker1.Date Then
        DateTimePicker4.date:=DateTimePicker1.date;
end;

procedure TFJua1.DateTimePicker4Change(Sender: TObject);
begin
    If DateTimePicker4.date <= DateTimePicker1.Date Then
        DateTimePicker4.date:=DateTimePicker1.date+1;
end;
```



```

                                jual
procedure TFJual.DateTimePicker1Change(Sender: TObject);
begin
  If DateTimePicker4.date <= DateTimePicker1.Date Then
    DateTimePicker4.date:=DateTimePicker1.date+1;
end;

procedure TFJual.Button3Click(Sender: TObject);
begin
  If ((PageControl1.ActivePageIndex=1) And (Dbgrid1.Fields[0].AsString<>'')) Then
    Begin
      FPenjualan_R :=TFPenjualan_R.Create(Self);
      FPenjualan_R.Q1.Close;
      FPenjualan_R.Q1.Sql.Clear;

      FPenjualan_R.q1.sql.add('select * from mJual where Nomer_faktur=:XX');
      FPenjualan_R.q1.ParamByName('xx').asstring:=DbGrid1.Fields[0].AsString;
      FPenjualan_R.Q1.Prepare;
      FPenjualan_R.Q1.Open;
      FPenjualan_R.Q2.Open;
      FPenjualan_R.Preview;
      FPenjualan_R.Q2.Close;
      FPenjualan_R.Q1.Close;
      FPenjualan_R.Free;
    End;
  end;
end.

```

## dialog jual

```

unit UDialog_jual;

interface

uses
  Windows, Messages, Sysutils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ComCtrls;

type
  TFDialog_jual = class(TForm)
    D1: TDateTimePicker;
    D2: TDateTimePicker;
    Button1: TButton;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    procedure FormShow(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FDialog_jual: TFDialog_jual;

implementation

uses URugi_Laba;

{$R *.DFM}

procedure TFDialog_jual.FormShow(Sender: TObject);
begin
  D1.Date:=Date;
  D2.Date:=Date;
end;

procedure TFDialog_jual.Button1Click(Sender: TObject);
begin
  FRugi_Laba :=TFRugi_Laba.Create(Self);
  FRugi_Laba.Q11.Close;
  FRugi_Laba.Q11.Sql.Clear;
  FRugi_Laba.Q11.SQL.Add('Select * from DJual where NOmer > 0') ;
  if CheckBox1.Checked then
  begin
    FRugi_Laba.Q11.SQL.Add('and Tgl>=:xx ') ;
    FRugi_Laba.Q11.parambyname('xx').AsDateTime := D1.Date ;
  end ;

  if CheckBox2.Checked then
  begin
    FRugi_Laba.Q11.SQL.Add('and Tgl<=:yy ') ;
    FRugi_Laba.Q11.parambyname('yy').AsDateTime := D2.Date ;
  end ;

  FRugi_Laba.Q11.Prepare;
  FRugi_Laba.Q11.Open;

  If ((CheckBox1.Checked) And (CheckBox2.Checked)) Then
  Begin
    FRugi_Laba.QrLabel1.Caption:= 'Periode : '+DateToStr(D1.date)+' sampai
'+DateToStr(D2.date) ;
  End;
  If ((CheckBox1.Checked) And (CheckBox2.Checked=False)) Then
  Begin
    FRugi_Laba.QrLabel1.Caption:= 'Periode mulai : '+DateToStr(D1.date)+' sampai

```

## dialog jual

```
ekarang';  
End;  
  
If ((CheckBox1.Checked=false) And (CheckBox2.Checked)) Then  
Begin  
FRugi_Laba.Qrlabel1.Caption:= 'Periode mulai awal sampai '+DateToStr(D2.date);  
End;  
FRugi_Laba.Preview;  
FRugi_Laba.Q11.Close;  
FRugi_Laba.Free;  
  
end;  
  
procedure TFDiallog_jual.CheckBox1Click(Sender: TObject);  
begin  
if checkbox1.Checked then  
begin  
D1.Enabled := True ;  
D1.Color := clWhite ;  
end  
else  
begin  
D1.Enabled := False ;  
D1.Color := clMenu ;  
end ;  
  
end;  
  
procedure TFDiallog_jual.CheckBox2Click(Sender: TObject);  
begin  
if checkbox2.Checked then  
begin  
D2.Enabled := True ;  
D2.Color := clWhite ;  
end  
else  
begin  
D2.Enabled := False ;  
D2.Color := clMenu ;  
end ;  
  
end;  
  
procedure TFDiallog_jual.FormCreate(Sender: TObject);  
begin  
  
end;  
  
end.
```

```
unit Ulevel_user;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls;
```

```
type
```

```
TFLevel_User = class(TForm)
  ComboBox1: TComboBox;
  Label1: TLabel;
  Edit1: TEdit;
  Button1: TButton;
  Button2: TButton;
  Label2: TLabel;
  GroupBox1: TGroupBox;
  CheckBox0: TCheckBox;
  CheckBox1: TCheckBox;
  CheckBox2: TCheckBox;
  CheckBox5: TCheckBox;
  CheckBox6: TCheckBox;
  CheckBox9: TCheckBox;
  procedure FormShow(Sender: TObject);
  procedure ComboBox1Change(Sender: TObject);
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure ComboBox1KeyPress(Sender: TObject; var Key: Char);
  procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
```

```
FLevel_User: TFLevel_User;
```

```
implementation
```

```
uses Udm, UUser_Names;
```

```
{ $R *.DFM }
```

```
procedure TFLevel_User.FormShow(Sender: TObject);
begin
  Combobox1.Clear;
  Dm.User_Names.First;
  Repeat
    Combobox1.Items.Add(Dm.User_Names.FieldName('User_Names').AsString);
    Dm.User_Names.Next;
  Until Dm.User_Names.Eof;
  Combobox1.ItemIndex:=0;
  Combobox1Change(Sender);
end;
```

```
procedure TFLevel_User.ComboBox1Change(Sender: TObject);
begin
  If Dm.User_Names.FindKey([Combobox1.Text]) Then
  Begin
    Edit1.text:=Dm.User_Names.FieldName('PASSWORDS').AsString;

    If Dm.User_Names.FieldName('Barang').AsString='1' Then
      Checkbox0.Checked:=True Else Checkbox0.Checked:=False;

    If Dm.User_Names.FieldName('Produsen').AsString='1' Then
      Checkbox1.Checked:=True Else Checkbox1.Checked:=False;

    If Dm.User_Names.FieldName('Pembelian').AsString='1' Then
      Checkbox2.Checked:=True Else Checkbox2.Checked:=False;
```

## leveluser

```
If Dm.User_Names.FieldByName('Customer').AsString='1' Then  
    Checkbox5.Checked:=True Else Checkbox5.Checked:=False;  
  
If Dm.User_Names.FieldByName('Penjualan').AsString='1' Then  
    Checkbox6.Checked:=True Else Checkbox6.Checked:=False;
```

```
End;
```

```
end;
```

```
procedure TFLevel_User.Button1Click(Sender: TObject);  
begin  
    If Dm.User_Names.FIndKey([Combobox1.Text]) Then  
        Begin  
            Dm.User_Names.Edit;  
  
            Dm.User_Names.FieldByName('PASSWORDS').AsString:=Edit1.Text;  
  
            If Checkbox0.Checked=True Then  
                Dm.User_Names.FieldByName('barang').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('barang').AsString:='0';  
  
            If Checkbox1.Checked=True Then  
                Dm.User_Names.FieldByName('Produsen').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('Produsen').AsString:='0';  
  
            If Checkbox2.Checked=True Then  
                Dm.User_Names.FieldByName('Pembelian').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('Pembelian').AsString:='0';  
  
            If Checkbox5.Checked=True Then  
                Dm.User_Names.FieldByName('Customer').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('Customer').AsString:='0';  
  
            If Checkbox6.Checked=True Then  
                Dm.User_Names.FieldByName('Penjualan').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('Penjualan').AsString:='0';  
  
            If Checkbox9.Checked=True Then  
                Dm.User_Names.FieldByName('Byr_Piutang').AsString:='1'  
            Else  
                Dm.User_Names.FieldByName('Byr_Piutang').AsString:='0';  
  
            Dm.User_Names.Post;  
            Dm.User_Names.ApplyUpdates;  
            Dm.User_Names.CommitUpdates;  
            Dm.User_Names.Refresh;  
        End;  
end;
```

```
procedure TFLevel_User.Button2Click(Sender: TObject);  
begin  
    FUser_names.ShowModal;  
    Combobox1Change(Sender);  
end;
```

```
procedure TFLevel_User.ComboBox1KeyPress(Sender: TObject; var Key: Char);  
begin  
    Key:=#0;  
end;
```

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

```

unit UCustomer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, Grids, DBGrids, ExtCtrls, Db, DBTables, Buttons,
  Menus, jpeg;

type
  TFCustomer = class(TForm)
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    Button1: TButton;
    Panel1: TPanel;
    Label10: TLabel;
    Edit9: TEdit;
    Button2: TButton;
    DBGrid1: TDBGrid;
    DataSource1: TDataSource;
    Q1: TQuery;
    Q1KODE: TStringField;
    Q1NAMA: TStringField;
    Q1ALAMAT: TStringField;
    Q1TELP: TStringField;
    Button3: TButton;
    Button4: TButton;
    Panel2: TPanel;
    Image2: TImage;
    Label13: TLabel;
    Label14: TLabel;
    Label15: TLabel;
    procedure FormShow(Sender: TObject);
    procedure ComboBox1KeyPress(Sender: TObject; var Key: Char);
    procedure Edit2KeyPress(Sender: TObject; var Key: Char);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
    procedure Button1Click(Sender: TObject);
    procedure PageControl1Change(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Edit1Exit(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Panel1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FCustomer: TFCustomer;

implementation

uses UDM, UMain, UCustomer_R;

```

```
[$R *.DFM}
```

```
procedure TFCustomer.FormShow(Sender: TObject);
begin
```

```
    edit5.text := '';
    Button2click(sender);
```

```
    // PageControl1.ActivePageIndex := 0;
    // Edit1.SetFocus;
end;
```

```
procedure TFCustomer.ComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
If (Key=Chr(VK_return)) Then  SendMessage(Handle,WM_NEXTDLGCTL,0,0) Else Key:=#0;
end;
```

```
procedure TFCustomer.Edit2KeyPress(Sender: TObject; var Key: Char);
begin
If (Key=Chr(VK_return)) Then  SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;
```

```
procedure TFCustomer.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if not(key in['0'..'9',#13,#8]) then key := #0 ;
    If (Key=Chr(VK_return)) Then  SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;
```

```
procedure TFCustomer.Button1Click(Sender: TObject);
begin
```

```
    if ((Edit1.text <> '') and (edit2.text <> '')) then
    begin
    if DM.Customer.FindKey([Edit1.text]) then
    DM.Customer.edit else DM.Customer.Insert ;
    DM.Customer.FieldName('Kode').AsString := Edit1.text ;
    DM.Customer.FieldName('Nama').AsString := Edit2.text ;
    DM.Customer.FieldName('Alamat').AsString := Edit3.text ;
    DM.Customer.FieldName('Telp').AsString := Edit4.text ;
    DM.Customer.FieldName('No_NPWP').AsString := Edit5.text ;
    DM.Customer.FieldName('Nama_Kontak').AsString := Edit6.text ;
    DM.Customer.FieldName('HP').AsString := Edit7.text ;
```

```
    DM.Customer.Post ;
    DM.Customer.ApplyUpdates ;
    DM.Customer.CommitUpdates ;
    edit1.text := '';
    edit2.text := '';
    edit3.text := '';
    edit4.text := '';
    edit5.Text := '';
    edit6.Text := '';
    edit7.Text := '';
```

```
    edit1.setfocus ;
    end
    else
    Begin
    MessageDlg('Kode dan nama customer harus diisi !', mtInformation, [mbOK], 0) ;
    edit1.setfocus ;
    End;
    end;
```

```
procedure TFCustomer.PageControl1Change(Sender: TObject);
begin
    Button2click(sender);
end;
```

```
procedure TFCustomer.Button2Click(Sender: TObject);
```

customer

```

n
Q1.Close ;
Q1.SQL.Clear ;
Q1.SQL.Add('select * from Customer') ;
if edit9.text <> '' then
begin
    Q1.SQL.Add('where Nama like:xx ') ;
    Q1.parambyname('xx').AsString := '%' + edit9.text + '%' ;
end;
Q1.prepare ;
Q1.Open ;

edure TFCustomer.Edit1Exit(Sender: TObject);
n
edit2.Text := '' ;
edit3.Text := '' ;
edit4.Text := '' ;
edit5.Text := '' ;
edit6.Text := '' ;
edit7.Text := '' ;

if edit1.Text <> '' then
begin
if DM.Customer.FindKey([Edit1.text]) then
begin
edit1.Text := DM.Customer.FieldName('Kode').AsString ;
edit2.Text := DM.Customer.FieldName('Nama').AsString ;
edit3.Text := DM.Customer.FieldName('Alamat').AsString ;
edit4.Text := DM.Customer.FieldName('Telp').AsString ;
edit5.Text := DM.Customer.FieldName('No_NPWP').AsString ;
edit6.Text := DM.Customer.FieldName('Nama_Kontak').AsString ;
edit7.Text := DM.Customer.FieldName('HP').AsString ;

Button1.enabled := True ;
end
else if messagedlg('Kode customer tidak ditemukan, isi baru?', mtConfirmation, [mbyes,
], 0) = mrYes then
begin
edit2.SetFocus ;
Button1.enabled := True ;
end
else
begin
edit1.Text := '' ;
Button1.enabled := False ;
end ;
end ;

edure TFCustomer.Button3Click(Sender: TObject);
n
edit1.text := '' ;
edit2.text := '' ;
edit3.text := '' ;
edit4.text := '' ;
edit5.Text := '' ;
edit6.Text := '' ;
edit7.Text := '' ;

edit1.setfocus ;

edure TFCustomer.Button4Click(Sender: TObject);
n
FCustomer_R :=TFCustomer_R.Create(Self);
FCustomer_R.Q1.Close;
FCustomer_R.Q1.Sql.Clear;

FCustomer_R.Q1.SQL.Add('select * from Customer') ;
if edit9.text <> '' then

```



```
customer
begin
    FCustomer_R.Q1.SQL.Add('where Nama like:xx ');
    FCustomer_R.Q1.parambyname('xx').AsString := '%' + edit9.text + '%';
end;
FCustomer_R.Q1.prepare ;
FCustomer_R.Q1.Open ;
FCustomer_R.Preview;
FCustomer_R.Q1.Close;
FCustomer_R.Free;
end;

procedure TFCustomer.Panel1Click(Sender: TObject);
begin
end;

end.
```

```
unit UBeli;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ComCtrls, StdCtrls, Grids, ExtCtrls, Db, DBTables, DBGrids, Buttons,  
Menus, jpeg;
```

```
type
```

```
TFBeli = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Panel1: TPanel;
  DateTimePicker1: TDateTimePicker;
  Edit1: TEdit;
  Panel2: TPanel;
  S: TStringGrid;
  Panel3: TPanel;
  Label5: TLabel;
  Button1: TButton;
  DBGrid1: TDBGrid;
  Panel4: TPanel;
  Button2: TButton;
  Q1: TQuery;
  DataSource1: TDataSource;
  DateTimePicker2: TDateTimePicker;
  DateTimePicker3: TDateTimePicker;
  CheckBox1: TCheckBox;
  CheckBox2: TCheckBox;
  Q1NOMER: TIntegerField;
  Q1TANGGAL: TDateTimeField;
  Q1KODE: TStringField;
  Q1NAMA: TStringField;
  Q1TOTAL_BELI: TFloatField;
  Q2: TQuery;
  DataSource2: TDataSource;
  DataSource3: TDataSource;
  Q2NOMER: TIntegerField;
  Q2NO_MBELI: TIntegerField;
  Q2KODE: TStringField;
  Q2NAMA: TStringField;
  Q2JML_BELI: TFloatField;
  Q2HRG_BELI: TFloatField;
  Q2SUB_TOTAL: TFloatField;
  PopupMenu1: TPopupMenu;
  HapusItem1: TMenuItem;
  Splitter1: TSplitter;
  DBGrid2: TDBGrid;
  Button3: TButton;
  Button4: TButton;
  Panel5: TPanel;
  Image2: TImage;
  Label10: TLabel;
  Label11: TLabel;
  Label12: TLabel;
  procedure FormShow(Sender: TObject);
  procedure Edit1Exit(Sender: TObject);
  procedure SKeyPress(Sender: TObject; var Key: Char);
  procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  procedure Button1Click(Sender: TObject);
  procedure CheckBox1Click(Sender: TObject);
  procedure CheckBox2Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure SExit(Sender: TObject);
  procedure PageControl1Change(Sender: TObject);
```

```

                                beli
procedure DateTimePicker2Change(Sender: TObject);
procedure DateTimePicker3Change(Sender: TObject);
procedure DateTimePicker1KeyPress(Sender: TObject; var Key: Char);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure HapusItemDel1Click(Sender: TObject);
procedure HapusItem1Click(Sender: TObject);
procedure DateTimePicker3Exit(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
{ Private declarations }
procedure SubTotal ;
public
{ Public declarations }
end;

```

```

var
  FBel: TFBeli;

```

```

implementation

```

```

uses UDM, UMain, Ubeli_R, USupplier, Ubarang;

```

```

{$R *.DFM}

```

```

procedure TFBeli.FormShow(Sender: TObject);
  var i : Integer ;
begin

```

```

  Panel1.Caption := IntToStr(DM.Mbeli.RecordCount+1) ;

```

```

  DateTimePicker1.DateTime := Date;
  DateTimePicker2.DateTime := Date;
  DateTimePicker3.DateTime := Date;

```

```

  S.Cells[0,0] := 'No.' ;
  S.Cells[1,0] := 'Kode' ;
  S.Cells[2,0] := 'Nama Barang' ;
  S.Cells[3,0] := 'Jumlah' ;
  S.Cells[4,0] := 'Harga' ;
  S.Cells[5,0] := 'Sub Total' ;
  S.Cells[6,0] := 'HPP' ;

```

```

  S.Colwidths[0] := 30 ;
  S.Colwidths[1] := 60 ;
  S.Colwidths[2] := 165 ;
  S.Colwidths[3] := 80 ;
  S.Colwidths[4] := 80 ;
  S.Colwidths[5] := 80 ;
  S.Colwidths[6] := 0 ;

```

```

  for i := 1 to S.RowCount -1 do
  begin
    S.Cells[0,i]:= IntToStr(i) ;
  end ;

```

```

  Edit1.Text := '' ;
  Panel2.Caption := '' ;
  Panel3.Caption := '0' ;
  Panel1.Caption := IntToStr(DM.Mbeli.RecordCount+1) ;

```

```

  for i := 1 to S.RowCount -1 do
  begin
    S.Cells[0,i]:= IntToStr(i) ;
    S.Cells[1,i]:= '' ;
    S.Cells[2,i]:= '' ;
    S.Cells[3,i]:= '' ;
    S.Cells[4,i]:= '' ;
    S.Cells[5,i]:= '0' ;

```

beli

```
end ;

DateTimePicker1.DateTime := Date ;
DateTimePicker2.DateTime := Date ;
DateTimePicker3.DateTime := Date ;

/ PageControl1.ActivePageIndex := 0 ;
/ DateTimePicker1.Setfocus ;
;

cedure TFBeli.Edit1Exit(Sender: TObject);
in
panel2.caption := '' ;
if edit1.Text <> '' then
begin
if DM.Supplier.FindKey([edit1.text]) then
begin
panel2.Caption := DM.Supplier.fieldbyname('Nama_PT').AsString ;
subtotal ;
end
else
begin
subtotal ;
messageDlg('Kode produsen tidak ditemukan', mtInformation, [mbOK],0) ;
FSupplier.ShowModal ;
Edit1.text:=FSupplier.DBGrid1.Fields[0].AsString; } untuk searching
Edit1Exit(Sender); } dan menginput kode supplier

edit1.setfocus ;
exit ;
end ;
end ;
;

cedure TFBeli.SKeyPress(Sender: TObject; var Key: Char);
j, baris : Integer ;
in
if S.Col in [1,3,4] then
begin
if not (key in ['0'..'9', #8, #13]) then key := #0 ;
end ;

if ((S.col = 1) and (key = #13)) then
begin
baris := S.Row ;
for j := 1 to S.RowCount-1 do
begin
if j <> baris then
begin
if S.Cells[1,j] = S.Cells[1,baris] then
begin
S.Cells[1,baris] := '' ;
end ;
end ;
end ;
end ;

if Dm.Barang.FindKey([S.Cells[1,S.Row]]) then
begin
S.Cells[2,S.Row] := Dm.Barang.fieldbyname('Nama').AsString ;
S.Cells[6,S.Row] := FormatFloat('#,0',Dm.Barang.fieldbyname('HPP').AsFloat);
end
else
begin
messageDlg('Kode Barang tidak ditemukan atau sudah ada', mtInformation, [mbOK], 0) ;
FBarang.ShowModal ;
S.Cells[1, S.Row]:=FBarang.DBGrid1.Fields[0].AsString; / untuk searching kode.

S.Col := 1 ;
exit ;
end ;
```

```

                                beli
end ;

if s.Col in [2,5] then key := #0 ;
if ((S.Col in [3,4]) and (key = #13)) then
begin
    S.Cells[3,S.ROW] := formatFloat('#,0', FMain.NR(S.Cells[3,S.ROW])) ;
    S.Cells[4,S.ROW] := formatFloat('#,0', FMain.NR(S.Cells[4,S.ROW])) ;
end ;
if key = #13 then
    Begin
        subtotal ;
        If S.Col=1 Then S.Col :=3 Else
            Begin
                If S.Col=3 Then S.Col :=4 Else
                    Begin
                        If S.Col=4 Then
                            Begin
                                S.Col :=1;
                                S.ROW:=S.ROW+1;
                            End;
                        End;
                    End;
                End;
            End;
        End;
    End;
End;

nd;

procedure TFBeli.SubTotal ;
var i : Integer ;
    total : Double ;
    cek : boolean ;
begin
    total := 0 ;
    cek := true ;

    for i := 1 to S.RowCount-1 do
        begin
            S.Cells[5,i] := formatfloat('#,0', FMain.NR(S.Cells[3,i])*FMain.NR(S.Cells[4,i])) ;
            total := total + FMain.NR(S.Cells[5,i]) ;

            If ((fmain.NR(s.cells[5,i]) <=0) And (s.cells[2,i] <> '')) Then
                cek := false ;
            end ;
            panel3.caption := formatfloat('#,0', total) ;
            if ((panel2.caption <> '') and (cek = true) and (total > 0)) then
                button1.enabled := true
            else button1.enabled := false ;
        end ;
    end ;

procedure TFBeli.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
    if not(key in['0'..'9',#13,#8]) then key := #0 ;
    If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;

procedure TFBeli.Button1Click(Sender: TObject);
var i, Nomer : Integer ;
begin
    DM.MBeli.Insert ;

    DM.Mbeli.FieldName('Nomer').AsInteger := DM.MBeli.RecordCount + 1 ;
    Nomer := DM.Mbeli.FieldName('Nomer').AsInteger ;
    DM.Mbeli.FieldName('Tanggal').AsDateTime := DateTimePicker1.Date ;
    DM.Mbeli.FieldName('Kode').AsString := Edit1.Text ;
    DM.Mbeli.FieldName('Nama').AsString := Panel2.Caption ;
    DM.Mbeli.FieldName('Total_Beli').AsFloat := FMain.NR(Panel3.Caption) ;

    DM.Mbeli.Post ;
    DM.Mbeli.ApplyUpdates ;
    DM.Mbeli.CommitUpdates ;
    DM.Mbeli.Refresh ;

```

beli

```

for i := 1 to S.RowCount - 1 do
begin
    if ((S.Cells[1,i] <> '') and (S.Cells[2,i] <> '') and (FMain.NR(S.Cells[5,i]) > 0))
    then
        begin
            DM.DBeli.Insert ;

            DM.Dbeli.FieldName('Nomer').AsInteger := DM.DBeli.RecordCount + 1 ;
            DM.Dbeli.FieldName('No_Mbeli').AsInteger := Nomer ;
            DM.Dbeli.FieldName('Kode').AsString := S.Cells[1,i] ;
            DM.Dbeli.FieldName('Nama').AsString := S.Cells[2,i] ;
            DM.Dbeli.FieldName('Jml_Beli').AsFloat := FMain.NR(S.Cells[3,i]) ;
            DM.Dbeli.FieldName('Hrg_Beli').AsFloat := FMain.NR(S.Cells[4,i]) ;
            DM.Dbeli.FieldName('Sub_Total').AsFloat := FMain.NR(S.Cells[5,i]) ;

            DM.Dbeli.Post ;
            DM.Dbeli.ApplyUpdates ;
            DM.Dbeli.CommitUpdates ;
            DM.Dbeli.Refresh ;

            if Dm.Barang.FindKey([S.Cells[1,i]]) then
                begin

                    Dm.Barang.edit ;

                    Dm.Barang.FieldName('Stok').AsFloat := Dm.Barang.FieldName('Stok').AsFloat +
                    FMain.NR(S.Cells[3,i]) ;
                    Dm.Barang.FieldName('Nilai').AsFloat := Dm.Barang.FieldName('Nilai').AsFloat +
                    FMain.NR(S.Cells[5,i]) ;

                    if ((Dm.Barang.FieldName('Stok').AsFloat > 0) and
                    Dm.Barang.FieldName('Nilai').AsFloat > 0)) then
                        begin
                            Dm.Barang.FieldName('HPP').AsFloat := Dm.Barang.FieldName('Nilai').AsFloat /
                            Dm.Barang.FieldName('Stok').AsFloat ;
                        end
                    else
                        begin
                            Dm.Barang.FieldName('Stok').AsFloat := 0 ;
                            Dm.Barang.FieldName('HPP').AsFloat := 0 ;
                            Dm.Barang.FieldName('Nilai').AsFloat := 0 ;
                        end ;

                    Dm.Barang.Post ;
                    Dm.Barang.ApplyUpdates ;
                    Dm.Barang.CommitUpdates ;
                    Dm.Barang.Refresh ;
                end ;
            end ;
            Edit1.Text := '' ;
            Panel2.Caption := '' ;
            Panel3.Caption := '0' ;
            Panel1.Caption := IntToStr(DM.Mbeli.RecordCount+1) ;

            for i := 1 to S.RowCount -1 do
            begin
                S.Cells[0,i]:= IntToStr(i) ;
                S.Cells[1,i]:= '' ;
                S.Cells[2,i]:= '' ;
                S.Cells[3,i]:= '' ;
                S.Cells[4,i]:= '' ;
                S.Cells[5,i]:= '' ;
                S.Cells[6,i]:= '' ;
            end ;
            Button1.Enabled:=False;
        end;
    procedure TFBeli.CheckBox1Click(Sender: TObject);
begin

```

beli

```
if checkbox1.Checked then
begin
    DateTimePicker2.Enabled := True ;
    DateTimePicker2.Color := clWhite ;
end
else
begin
    DateTimePicker2.Enabled := False ;
    DateTimePicker2.Color := clMenu ;
end ;
end ;
```

end;

```
procedure TFBeli.CheckBox2Click(Sender: TObject);
begin
```

```
if checkbox2.Checked then
begin
    DateTimePicker3.Enabled := True ;
    DateTimePicker3.Color := clWhite ;
end
else
begin
    DateTimePicker3.Enabled := False ;
    DateTimePicker3.Color := clMenu ;
end ;
end ;
```

end;

```
procedure TFBeli.Button2Click(Sender: TObject);
begin
```

```
Q1.Close ;
Q1.SQL.Clear ;
Q1.SQL.Add('select * from MBeli where Nomer > 0') ;
if CheckBox1.Checked then
begin
    Q1.SQL.Add('and Tanggal >=:xx ') ;
    Q1.parambyname('xx').AsDateTime := DateTimePicker2.Date ;
end;

if CheckBox2.Checked then
begin
    Q1.SQL.Add('and Tanggal <=:yy ') ;
    Q1.parambyname('yy').AsDateTime := DateTimePicker3.Date ;
end;
```

```
Q1.prepare ;
Q1.Open ;
```

```
Q2.Close;
Q2.Open ;
```

end;

```
procedure TFBeli.SExit(Sender: TObject);
```

```
begin
    subtotal ;
end;
```

```
procedure TFBeli.PageControl1Change(Sender: TObject);
```

```
begin
    If PageControl1.ActivePageIndex=1 Then
        Button2Click(Sender);
end;
```

```
procedure TFBeli.DateTimePicker2Change(Sender: TObject);
```

```
begin
    Button2Click(Sender);
end;
```

```
procedure TFBeli.DateTimePicker3Change(Sender: TObject);
```

beli

```
begin
Button2Click(Sender);
end;

procedure TFBeli.DateTimePicker1KeyPress(Sender: TObject; var Key: Char);
begin
If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);
end;

procedure TFBeli.FormClose(Sender: TObject; var Action: TCloseAction);
var i : integer ;
begin
Edit1.Text := '' ;
Panel2.Caption := '' ;
Panel3.Caption := '0' ;
Panel1.Caption := IntToStr(DM.Mbeli.RecordCount+1) ;

for i := 1 to S.RowCount -1 do
begin
S.Cells[0,i]:= IntToStr(i) ;
S.Cells[1,i]:= '' ;
S.Cells[2,i]:= '' ;
S.Cells[3,i]:= '' ;
S.Cells[4,i]:= '' ;
S.Cells[5,i]:= '0' ;
end ;

button1.Enabled := false ;
end;

procedure TFBeli.HapusItemDel1Click(Sender: TObject);
Var
I: Integer;
begin
With S Do
Begin
Rows[Row].Clear;
Cells[0,Row]:= IntToStr(Row)+'.';
For i:= Row To RowCount - 1 Do
Begin
IF (i <> RowCount - 1) Then
Begin
Cells[1,i]:= Cells[1,i+1];
Cells[2,i]:= Cells[2,i+1];
Cells[3,i]:= Cells[3,i+1];
Cells[4,i]:= Cells[4,i+1];
Cells[5,i]:= Cells[5,i+1];
End;
End;
SubTotal;
End;
end;

procedure TFBeli.HapusItem1Click(Sender: TObject);
Var
I: Integer;
begin
With S Do
Begin
Rows[Row].Clear;
Cells[0,Row]:= IntToStr(Row)+'.';
For i:= Row To RowCount - 1 Do
Begin
IF (i <> RowCount - 1) Then
Begin
Cells[1,i]:= Cells[1,i+1];
Cells[2,i]:= Cells[2,i+1];
Cells[3,i]:= Cells[3,i+1];
Cells[4,i]:= Cells[4,i+1];
Cells[5,i]:= Cells[5,i+1];
End;
End;
end;
```



beli

```
End;
SubTotal;
End;
nd;

procedure TFBeli.DateTimePicker3Exit(Sender: TObject);
begin
If DateTimePicker3.Date<DateTimePicker2.Date Then DateTimePicker3.Date:=DateTimePicker2.Date
nd;

procedure TFBeli.Button3Click(Sender: TObject);
begin
f ((PageControl1.ActivePageIndex=1) And (Dbgrid1.Fields[0].AsString<>'')) Then
Begin
    FBel_i_R :=TFBel_i_R.Create(Self);
    FBel_i_R.Q1.Close;
    FBel_i_R.Q1.Sql.Clear;
    FBel_i_R.q1.sql.add('select * from mBeli where Nomer=:XX');
    FBel_i_R.q1.ParamByName('xx').asString:=DbGrid1.Fields[0].AsString;
    FBel_i_R.Q1.Prepare;
    FBel_i_R.Q1.Open;
    FBel_i_R.Q2.Open;
    FBel_i_R.Preview;
    FBel_i_R.Q2.Close;
    FBel_i_R.Q1.Close;
    FBel_i_R.Free;
End;
nd;

procedure TFBeli.Button4Click(Sender: TObject);
var I : Integer;
begin
Edit1.Text := '' ;
Panel2.Caption := '' ;
Panel3.Caption := '0' ;
Panel1.Caption := IntToStr(DM.Mbeli.RecordCount+1) ;

for i := 1 to S.RowCount -1 do
begin
    S.Cells[0,i]:= IntToStr(i) ;
    S.Cells[1,i]:= '' ;
    S.Cells[2,i]:= '' ;
    S.Cells[3,i]:= '' ;
    S.Cells[4,i]:= '' ;
    S.Cells[5,i]:= '' ;
    S.Cells[6,i]:= '' ;
end ;
Button1.Enabled:=False;
nd;
nd.
```

```
unit URetur;
```

```
interface
```

```
uses
```

```
windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
Grids, ComCtrls, ExtCtrls, StdCtrls, Db, DBTables, DBGrids, Buttons,
Menus;
```

```
type
```

```
TFRetur = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  GroupBox1: TGroupBox;
  GroupBox2: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Label12: TLabel;
  Edit1: TEdit;
  Panel2: TPanel;
  Panel5: TPanel;
  Panel3: TPanel;
  Panel4: TPanel;
  Label5: TLabel;
  Label7: TLabel;
  Panel1: TPanel;
  D1: TDateTimePicker;
  S: TStringGrid;
  Label9: TLabel;
  N1: TPanel;
  Label13: TLabel;
  N2: TPanel;
  ComboBox5: TComboBox;
  N3: TPanel;
  Label15: TLabel;
  N4: TPanel;
  Q1: TQuery;
  Button1: TButton;
  Panel6: TPanel;
  Button2: TButton;
  DateTimePicker2: TDateTimePicker;
  DateTimePicker3: TDateTimePicker;
  CheckBox1: TCheckBox;
  CheckBox2: TCheckBox;
  DBGrid1: TDBGrid;
  DataSource1: TDataSource;
  DataSource2: TDataSource;
  Q2: TQuery;
  DataSource3: TDataSource;
  Q11: TQuery;
  PN1: TPanel;
  Label18: TLabel;
  Label14: TLabel;
  Label17: TLabel;
  PN2: TPanel;
  Q11NOMER: TStringField;
  Q11NO_FAKTUR: TStringField;
  Q11TGL_FAKTUR: TDateTimeField;
  Q11KODE_CUST: TStringField;
  Q11NAMA_CUST: TStringField;
  Q11TOTAL_JUAL: TFloatField;
  Q11TGL_RETUR: TDateTimeField;
  Q11TOTAL_RETUR: TFloatField;
  Q2NOMER: TIntegerField;
  Q2NO_MRETUR: TStringField;
  Q2TGL: TDateTimeField;
  Q2KODE: TStringField;
  Q2NAMA: TStringField;
```

returjual

```
Q2JML_RETUR: TFloatField;
Q2HRG: TFloatField;
Q2SUB_TOTAL: TFloatField;
Q2KETERANGAN: TStringField;
Button3: TButton;
PopupMenu1: TPopupMenu;
HapusItem1: TMenuItem;
Splitter1: TSplitter;
DBGrid2: TDBGrid;
MainMenu1: TMainMenu;
Print1: TMenuItem;
Panel7: TPanel;
Image2: TImage;
Label10: TLabel;
Label11: TLabel;
Label6: TLabel;
procedure FormShow(Sender: TObject);
procedure ComboBox5Exit(Sender: TObject);
procedure SClick(Sender: TObject);
procedure Edit1Exit(Sender: TObject);
procedure SKeyPress(Sender: TObject; var Key: Char);
procedure Button1Click(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var Key: Char);
procedure D1KeyPress(Sender: TObject; var Key: Char);
procedure Button2Click(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure CheckBox2Click(Sender: TObject);
procedure PageControl1Change(Sender: TObject);
procedure ComboBox5KeyPress(Sender: TObject; var Key: Char);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button3Click(Sender: TObject);
procedure Print1Click(Sender: TObject);
private
  { Private declarations }
  procedure KaliJes;
public
  { Public declarations }
end;

var
  Fretur: TFretur;

implementation

uses UDM, UMain, URetur_R;

{$R *.DFM}

procedure TFretur.FormShow(Sender: TObject);

  var i : Integer ;
begin
  // PageControl1.ActivePageIndex := 0 ;
  // D1.Setfocus ;
  Panel1.Caption := Inttostr(DM.Mretur.RecordCount+1) ;
  D1.DateTime := Date ;
  //DateTimePicker4.DateTime := Date + 30 ;

  S.Cells[0,0] := 'No.' ;
  S.Cells[1,0] := 'Kode' ;
  S.Cells[2,0] := 'Nama Barang' ;
  S.Cells[3,0] := 'Jml Jual' ;
  S.Cells[4,0] := 'Jml Retur' ;
  S.Cells[5,0] := 'Hrg Jual' ;
  S.Cells[6,0] := 'Sub Ttl Jual' ;
  S.Cells[7,0] := 'Sub Ttl Retur' ;
  S.Cells[8,0] := 'Alasan Return' ; //Rusak, salah kirim
  S.Cells[9,0] := 'HPP Jual' ;
  S.Cells[10,0] := 'No Djual' ;
```

returjual

```
S.Colwidths[0] := 30 ;  
S.Colwidths[1] := 45 ;  
S.Colwidths[2] := 150 ;  
S.Colwidths[3] := 50 ;  
S.Colwidths[4] := 50 ;  
S.Colwidths[5] := 50 ;  
S.Colwidths[6] := 70 ;  
S.Colwidths[7] := 70 ;  
S.Colwidths[8] := 80 ;  
S.Colwidths[9] := 0 ;  
S.Colwidths[10] := 0 ;
```

```
for i := 1 to S.RowCount -1 do  
begin  
S.Cells[0,i]:= IntToStr(i) ;  
end ;
```

```
D1.Date := Date;  
D1.Date := NOW ;
```

```
DateTimePicker2.Date := NOW ;  
DateTimePicker2.DateTime := Date;  
DateTimePicker3.Date := NOW ;  
DateTimePicker3.DateTime := Date;
```

```
// DateTimePicker2.DateTime := Date;  
// DateTimePicker3.DateTime := Date;  
end;
```

```
procedure TFretur.ComboBox5Exit(Sender: TObject);  
begin  
S.Cells[8,S.Row]:=Combobox5.Text;  
Combobox5.Visible:=False;  
end;
```

```
procedure TFretur.SClick(Sender: TObject);  
var  
r:TRect;  
indexRow,indexCol:Integer;  
begin
```

```
indexCol:=S.Col;  
indexRow:=S.Row;  
r:=S.CellRect(S.Col,S.Row);  
case S.Col of  
8: begin  
If S.Row <> S.RowCount Then  
Begin  
Combobox5.Left:=S.Left+r.Left+1;  
Combobox5.Top:=S.Top+r.Top;  
Combobox5.Visible:=True;  
if S.Cells[indexCol,indexRow]<>' ' then  
Combobox5.Text:=S.Cells[indexCol,indexRow];  
Combobox5.ItemIndex:=0;  
Combobox5.SetFocus;  
Exit;  
End;  
end;
```

```
end;  
end;
```

```
procedure TFretur.Edit1Exit(Sender: TObject);  
var I : Integer;  
Sudah_Retur:Boolean;  
begin  
Sudah_Retur:=False;
```

```
if edit1.Text <> ' ' then  
begin  
If Dm.Mjual.FIndkey([Fmain.NR(Edit1.Text)]) Then
```

returjual

Begin

```
Panel3.Caption := DM.MJual.FieldByName('Nomer_Faktur').AsString;  
Panel3.Caption := DM.MJual.FieldByName('Tgl').AsString;  
Panel4.Caption:=DM.MJual.FieldByName('Kode_Cust').AsString;  
Panel2.Caption:=DM.MJual.FieldByName('Nama').AsString;  
Panel5.Caption:=DM.MJual.FieldByName('Alamat').AsString;
```

```
N1.Caption:=FormatFloat('#,0',DM.MJual.FieldByName('Total_Jual').AsFloat);
```

```
N2.Caption:=FormatFloat('#,0',DM.MJual.FieldByName('Tunai').AsFloat+DM.MJual.FieldByName('Dibayar').AsFloat);
```

```
N3.Caption:=FormatFloat('#,0',DM.MJual.FieldByName('Saldo').AsFloat);
```

Q1.Close;

Q1.Sql.Clear;

Q1.Sql.Add('Select \* from Djual where No\_Faktur=:XX');

Q1.ParamByName('XX').AsInteger:=StrToInt(Edit1.Text);

Q1.Sql.Add('Order By Kode');

Q1.Prepare;

Q1.Open;

If Q1.RecordCount > 0 Then

Begin

S.RowCount:=Q1.RecordCount +1 ;

Q1.First;

I:=1;

repeat

S.Cells[1,I]:=Q1.FieldByName('Kode').AsString;

S.Cells[2,I]:=Q1.FieldByName('Nama').AsString;

S.Cells[3,I]:=FormatFloat('#,0',Q1.FieldByName('Jml\_Jual').AsFloat);

S.Cells[5,I]:=FormatFloat('#,0',Q1.FieldByName('Harga\_jual').AsFloat);

S.Cells[6,I]:=FormatFloat('#,0',Q1.FieldByName('Sub\_Total').AsFloat);

S.Cells[9,I]:=FormatFloat('#,0',Q1.FieldByName('HPP').AsFloat);

S.Cells[10,I]:=FormatFloat('#,0',Q1.FieldByName('Nomer').AsInteger);

S.Cells[8,I]='Rusak';

If Q1.FieldByName('Jml\_Retur').AsFloat > 0 Then Sudah\_Retur:=True;

I:=I+1;

Q1.Next;

Until Q1.Eof;

End;

//Cek Retur

If ((StrToDate(Panel3.Caption)+7 < D1.Date) Or (Sudah\_Retur=True))Then

Begin

Panel3.Caption:='0';

Panel4.Caption:='';

Panel2.Caption:='';

Panel5.Caption:='';

Panel1.Caption:='';

N1.Caption:='0';

N2.Caption:='0';

N3.Caption:='0';

N4.Caption:='0';

Edit1.text:='';

For I:=1 To S.RowCount-1 Do

Begin

S.Cells[1,I]:='';

S.Cells[2,I]:='';

S.Cells[3,I]:='';

S.Cells[4,I]:='';

S.Cells[5,I]:='';

S.Cells[6,I]:='';

S.Cells[7,I]:='';

S.Cells[8,I]:='';

S.Cells[9,I]:='';

S.Cells[10,I]:='';

End;

MessageDlg('Faktur Lebih dari 7 Hari atau'+#13+

Sudah pernah diretur, tidak boleh diretur

```

                                returjual
gi',MTInformation,[MBOk],0);
    Button1.Enabled:=False;

    Edit1.SetFocus;
End Else Button1.Enabled:=True;

End
else
begin
if messagedlg('No. faktur tidak ditemukan, periksa lagi', mtInformation, [mbOK],0) =
OK then
begin
edit1.SetFocus ;
end ;
end ;

end ;
d;

procedure TFretur.Kalijes;
var
i: Integer;
total:Double;
begin
Total:=0;
For I:=1 To S.RowCount-1 DO
Begin
If Trim(S.Cells[3,i])='' Then S.Cells[3,i]:='0';
S.Cells[7,i]:=FormatFloat('#,0',Fmain.NR(S.Cells[4,i]) * Fmain.NR(S.Cells[5,i]));
Total:=Total+ Fmain.NR(S.Cells[7,i]);
End;
N4.Caption:=FormatFloat('#,0',Total);
If Fmain.NR(N4.Caption) > Fmain.NR(N3.Caption) Then
Begin
PN1.Caption:=FormatFloat('#,0',Fmain.NR(N4.Caption)-Fmain.NR(N3.Caption) );
PN2.Caption:=FormatFloat('#,0',Fmain.NR(N3.Caption) );
End Else
Begin
PN1.Caption:='0';
PN2.Caption:=FormatFloat('#,0',Fmain.NR(N4.Caption) );
End;
If ((Fmain.NR(N4.Caption)>0) AND (Panel4.Caption<>'')) Then
Button1.Enabled:=True Else Button1.Enabled:=False;
d;

procedure TFretur.SKeyPress(Sender: TObject; var Key: Char);
begin
if S.Col=4 then
begin
if not (key in ['0'..'9', #8, #13]) then key := #0 ;
If Fmain.NR(S.Cells[4,S.ROW])>=Fmain.NR(S.Cells[3,S.ROW]) Then
S.Cells[4,S.ROW]:=S.Cells[3,S.ROW];
end Else Key:=#0;
Kalijes;
d;

procedure TFretur.Button1Click(Sender: TObject);
var
i, Nomer, Cari : Integer ;
begin
Q1.Close ;
Q1.SQL.Clear ;
Q1.SQL.Add('select * from Mretur where No_Faktur =:XX') ;
Q1.ParamByName('XX').AsString:=Edit1.text;
Q1.prepare ;
Q1.Open ;
Q1.First;

```

returjual

```
f Q1.RecordCount = 0 Then
Cari:=0 Else
Cari:=StrToInt(Q1.FieldName('Nomer').AsString);
begin
f DM.MRetur.Findkey([Cari]) Then
Begin
MessageDlg('Sudah diretur User lain'+#13+ 'Tidak Boleh Diretur
agi',MTInformation,[MBOK],0);
Exit;
End Else
Begin
DM.MRetur.Insert ;
DM.MRetur.FieldName('Nomer').AsInteger := DM.MRetur.RecordCount + 1 ;
Nomer := DM.MRetur.FieldName('Nomer').AsInteger ;

DM.MRetur.FieldName('No_faktur').AsInteger :=StrToInt(Edit1.Text);
DM.MRetur.FieldName('Tgl_Faktur').AsDateTime := Strtodate(Panel3.caption);

DM.MRetur.FieldName('Tgl_retur').AsDateTime := D1.Date ;
DM.MRetur.FieldName('Kode_Cust').AsString := Panel4.Caption ;
DM.MRetur.FieldName('Nama_Cust').AsString := Panel2.Caption ;
DM.MRetur.FieldName('Total_Jual').AsFloat := FMain.NR(N1.Caption) ;
DM.MRetur.FieldName('Total_Retur').AsFloat := FMain.NR(N4.Caption) ;

DM.MRetur.Post ;
DM.MRetur.ApplyUpdates ;
DM.MRetur.CommitUpdates ;
DM.MRetur.Refresh ;

for i := 1 to S.RowCount - 1 do
begin
if FMain.NR(S.Cells[4,i]) > 0 then
begin
DM.DRetur.Insert ;

DM.DRetur.FieldName('Nomer').AsInteger := DM.DRetur.RecordCount + 1 ;
DM.DRetur.FieldName('No_mretur').AsInteger := Nomer ;
DM.DRetur.FieldName('Tgl').AsDateTime := D1.Date ;
DM.DRetur.FieldName('Kode').AsString := S.Cells[1,i] ;
DM.DRetur.FieldName('Nama').AsString := (S.Cells[2,i]) ;
DM.DRetur.FieldName('Jml_retur').AsFloat := FMain.NR(S.Cells[4,i]) ;
DM.DRetur.FieldName('Hrg').AsFloat := FMain.NR(S.Cells[5,i]) ;
DM.DRetur.FieldName('Sub_Total').AsFloat := FMain.NR(S.Cells[7,i]) ;
DM.DRetur.FieldName('Keterangan').AsString := S.Cells[8,i] ;
DM.DRetur.Post ;
DM.DRetur.ApplyUpdates ;
DM.DRetur.CommitUpdates ;
DM.DRetur.Refresh ;

If DM.Djual.FindKey([S.Cells[10,i]]) then
begin
DM.Djual.edit;
DM.Djual.FieldName('Jml_Retur').AsFloat :=FMain.NR(S.Cells[4,i]) ;
DM.Djual.FieldName('Nilai_retur').AsFloat := FMain.NR(S.Cells[7,i]) ;
DM.Djual.FieldName('Nilai_Hpp_Retur').AsFloat :=
DM.Djual.FieldName('HPP').AsFloat * FMain.NR(S.Cells[4,i]);
DM.Djual.FieldName('Rugi').AsFloat :=
DM.Djual.FieldName('Nilai_retur').AsFloat-DM.Djual.FieldName('Nilai_Hpp_Retur').AsFloat;
DM.Djual.FieldName('Laba_Bersih').AsFloat :=
DM.Djual.FieldName('Laba').AsFloat - DM.Djual.FieldName('Rugi').AsFloat;
DM.Djual.Post ;
DM.Djual.ApplyUpdates ;
DM.Djual.CommitUpdates ;
DM.Djual.Refresh ;
End;
end;
end;
end;
```

```

                returjual
If S.Cells[8,i]='Salah Kirim' Then
Begin
    //Mencari Di Kode Barang, Mengembalikan stok dan nilai
    if Dm.Barang.FindKey([S.Cells[1,i]]) then
        begin
            Dm.Barang.edit ;
            Dm.Barang.FieldByName('Stok').AsFloat :=
Dm.Barang.FieldByName('Stok').AsFloat + FMain.NR(S.Cells[4,i]) ;
            Dm.Barang.FieldByName('Nilai').AsFloat :=
Dm.Barang.FieldByName('Nilai').AsFloat + ( FMain.NR(S.Cells[4,i]) * FMain.NR(S.Cells[9,i]) )
            if ((Dm.Barang.FieldByName('Stok').AsFloat > 0) and
(Dm.Barang.FieldByName('Nilai').AsFloat > 0)) then
                begin
                    Dm.Barang.FieldByName('HPP').AsFloat :=
Dm.Barang.FieldByName('Nilai').AsFloat / Dm.Barang.FieldByName('Stok').AsFloat ;
                    end
                else
                    begin
                        Dm.Barang.FieldByName('Stok').AsFloat := 0 ;
                        Dm.Barang.FieldByName('HPP').AsFloat := 0 ;
                        Dm.Barang.FieldByName('Nilai').AsFloat := 0 ;
                    end ;

                    Dm.Barang.Post ;
                    Dm.Barang.ApplyUpdates ;
                    Dm.Barang.CommitUpdates ;
                    Dm.Barang.Refresh ;
                end ;
            End;

        end ;
    end ;

If (Fmain.NR(Pn2.Caption) > 0 ) Then
Begin
    dm.Dbyr_piutang.Insert;
    dm.Dbyr_piutang.Fieldbyname('Nomer').asinteger:=dm.Dbyr_piutang.recordcount+1;
    dm.Dbyr_piutang.Fieldbyname('No_Faktur').asinteger:=StrToInt(Edit1.text);
dm.Dbyr_piutang.Fieldbyname('Tgl_bayar').asdatetime:=StrToDate(FormatDateTime('dd/mm/yyyy',D1
ate));
    dm.Dbyr_piutang.Fieldbyname('Jml_Bayar').asfloat:=fmain.nr(PN2.Caption);
    dm.Dbyr_piutang.post;
    dm.Dbyr_piutang.ApplyUpdates;
    dm.Dbyr_piutang.CommitUpdates;
    dm.Dbyr_piutang.refresh;
End;

If dm.Mjual.Findkey([StrToInt(Edit1.Text)]) Then
Begin
    dm.Mjual.Edit;
    dm.Mjual.Fieldbyname('Dibayar').asfloat:=
    dm.Mjual.Fieldbyname('Dibayar').asfloat+ fmain.nr(PN2.Caption);

    dm.Mjual.Fieldbyname('Saldo').asfloat:=
    dm.Mjual.Fieldbyname('hutang').asfloat-
    dm.Mjual.Fieldbyname('Dibayar').asfloat;

    dm.Mjual.post;
    dm.Mjual.ApplyUpdates;
    dm.Mjual.CommitUpdates;
    dm.Mjual.refresh;
End;

Panel3.Caption:='0' ;
Panel4.Caption:='';
Panel2.Caption:='';

```



returjual

```
Panel5.Caption:='';  
Panel1.Caption:='';
```

```
N1.Caption:='0';  
N2.Caption:='0';  
N3.Caption:='0';  
N4.Caption:='0';
```

```
PN1.Caption:='0';  
PN2.Caption:='0';
```

```
Edit1.text:='';  
Button1.Enabled:=False;  
For I:=1 To S.RowCount-1 Do
```

```
  Begin  
    S.Cells[1,I]:='';  
    S.Cells[2,I]:='';  
    S.Cells[3,I]:='';  
    S.Cells[4,I]:='';  
    S.Cells[5,I]:='';  
    S.Cells[6,I]:='';  
    S.Cells[7,I]:='';  
    S.Cells[8,I]:='';  
    S.Cells[9,I]:='';  
    S.Cells[10,I]:='';
```

```
  End;
```

```
  Edit1.SetFocus;
```

```
End;  
End;
```

```
end;
```

```
procedure TFretur.Edit1KeyPress(Sender: TObject; var Key: Char);
```

```
begin  
  if not(key in['0'..'9',#13,#8]) then key := #0 ;  
  If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);  
end;
```

```
procedure TFretur.D1KeyPress(Sender: TObject; var Key: Char);
```

```
begin  
  If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);  
end;
```

```
procedure TFretur.Button2Click(Sender: TObject);
```

```
begin  
  Q11.Close ;  
  Q11.SQL.Clear ;  
  Q11.SQL.Add('select * from MRETUR where Nomer > 0') ;  
  if CheckBox1.Checked then  
  begin  
    Q11.SQL.Add('and Tg]_RETUR >=:xx ') ;  
    Q11.parambyname('xx').ASDateTime := DateTimePicker2.Date ;  
  end;
```

```
  if CheckBox2.Checked then
```

```
  begin  
    Q11.SQL.Add('and Tg]_RETUR <=:yy ') ;  
    Q11.parambyname('yy').ASDateTime := DateTimePicker3.Date ;  
  end;
```

```
  Q11.prepare ;  
  Q11.Open ;
```

```
  Q2.Close ;  
  Q2.Open ;
```

```
end;
```

```
procedure TFretur.CheckBox1Click(Sender: TObject);
```

```
begin  
  if checkbox1.Checked then  
  begin  
    DateTimePicker2.Enabled := True ;
```

```

        DateTimePicker2.Color := clWhite ; returjua1
    end
else
begin
    DateTimePicker2.Enabled := False ;
    DateTimePicker2.Color := clMenu ;
end ;
end;

procedure TFretur.CheckBox2Click(Sender: TObject);
begin
    if checkbox2.Checked then
    begin
        DateTimePicker3.Enabled := True ;
        DateTimePicker3.Color := clWhite ;
    end
    else
    begin
        DateTimePicker3.Enabled := False ;
        DateTimePicker3.Color := clMenu ;
    end ;
end;

procedure TFretur.PageControl1Change(Sender: TObject);
begin
    If PageControl1.ActivePageIndex=1 Then
    Begin
        Button2Click(Sender);
    End;
end;

procedure TFretur.ComboBox5KeyPress(Sender: TObject; var Key: Char);
begin
    Key:=#0;
end;

procedure TFretur.FormClose(Sender: TObject; var Action: TCloseAction);
var i : integer ;
begin
    Panel3.Caption:='0';
    Panel4.Caption:='';
    Panel2.Caption:='';
    Panel5.Caption:='';
    Panel1.Caption:='';

    N1.Caption:='0';
    N2.Caption:='0';
    N3.Caption:='0';
    N4.Caption:='0';
    Edit1.text:='';

    For i:=1 To S.RowCount-1 Do
    Begin
        S.Cells[1,i]:='';
        S.Cells[2,i]:='';
        S.Cells[3,i]:='';
        S.Cells[4,i]:='';
        S.Cells[5,i]:='';
        S.Cells[6,i]:='';
        S.Cells[7,i]:='';
        S.Cells[8,i]:='';
        S.Cells[9,i]:='';
        S.Cells[10,i]:='';
    End;
end;

procedure TFretur.Button3Click(Sender: TObject);
var i : integer ;
begin
    Panel3.Caption:='0';

```

returjual

```
Panel4.Caption:='';  
Panel2.Caption:='';  
Panel5.Caption:='';  
Panel1.Caption:='';
```

```
N1.Caption:='0';  
N2.Caption:='0';  
N3.Caption:='0';  
N4.Caption:='0';  
Edit1.text:='';
```

```
For i:=1 To S.RowCount-1 Do  
  Begin  
    S.Cells[1,i]:='';  
    S.Cells[2,i]:='';  
    S.Cells[3,i]:='';  
    S.Cells[4,i]:='';  
    S.Cells[5,i]:='';  
    S.Cells[6,i]:='';  
    S.Cells[7,i]:='';  
    S.Cells[8,i]:='';  
    S.Cells[9,i]:='';  
    S.Cells[10,i]:='';  
  End;
```

```
  button1.Enabled := false ;
```

```
end;
```

```
procedure TFretur.Print1Click(Sender: TObject);  
begin
```

```
  FRetur_R :=TFRetur_R.Create(Self);  
  FRetur_R.Q1.Close;  
  FRetur_R.Q1.Sql.Clear;
```

```
  FRetur_R.q1.sql.add('select * from mretur where Nomer=:XX');  
  FRetur_R.q1.ParamByName('xx').asstring:=DbGrid1.Fields[0].AsString;  
  FRetur_R.Q1.Prepare;  
  FRetur_R.Q1.Open;  
  FRetur_R.Q2.Open;  
  FRetur_R.Preview;  
  FRetur_R.Q2.Close;  
  FRetur_R.Q1.Close;  
  FRetur_R.Free;
```

```
end;
```

```
end.
```

## dialogbeli

```

unit ULap_Beli;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ComCtrls, Buttons;

type
  TFLap_Beli = class(TForm)
    SpeedButton1: TSpeedButton;
    DateTimePicker2: TDateTimePicker;
    DateTimePicker3: TDateTimePicker;
    CheckBox1: TCheckBox;
    CheckBox2: TCheckBox;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure SpeedButton1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLap_Beli: TFLap_Beli;

implementation

uses ULap_Pembelian, ULap_Penjualan;

{$R *.DFM}

procedure TFLap_Beli.Button1Click(Sender: TObject);
begin
  close ;
end;

procedure TFLap_Beli.CheckBox1Click(Sender: TObject);
begin
  if checkbox1.Checked then
  begin
    DateTimePicker2.Enabled := True ;
    DateTimePicker2.Color := clWhite ;
  end
  else
  begin
    DateTimePicker2.Enabled := False ;
    DateTimePicker2.Color := clMenu ;
  end ;
end;

procedure TFLap_Beli.CheckBox2Click(Sender: TObject);
begin
  if checkbox2.Checked then
  begin
    DateTimePicker3.Enabled := True ;
    DateTimePicker3.Color := clWhite ;
  end
  else
  begin
    DateTimePicker3.Enabled := False ;
    DateTimePicker3.Color := clMenu ;
  end ;
end;

procedure TFLap_Beli.FormShow(Sender: TObject);

```

## dialogbeli

begin

```
DateTimePicker2.Date := Date ;  
DateTimePicker3.Date := Date ;
```

```
checkbox1.Checked := false ;  
checkbox2.Checked := false ;
```

end;

```
procedure TFLap_Beli.SpeedButton1Click(Sender: TObject);
```

begin

```
FLap_Pembelian :=TFLap_Pembelian.Create(Self);
```

```
FLap_Pembelian.Q11.Close;
```

```
FLap_Pembelian.Q11.Sql.Clear;
```

```
FLap_Pembelian.Q11.SQL.Add('Select Nomer, Tanggal, Kode, Nama, Total_Beli From MBeli  
where Nomer > 0');
```

```
if CheckBox1.Checked then
```

begin

```
FLap_Pembelian.Q11.SQL.Add('and Tgl >=:xx ');
```

```
FLap_Pembelian.Q11.parambyname('xx').AsDateTime := DateTimePicker2.Date ;
```

end;

```
if CheckBox2.Checked then
```

begin

```
FLap_Pembelian.Q11.SQL.Add('and Tgl <=:yy ');
```

```
FLap_Pembelian.Q11.parambyname('yy').AsDateTime := DateTimePicker3.Date ;
```

end;

```
FLap_Pembelian.Q11.Prepare;
```

```
FLap_Pembelian.Q11.Open;
```

```
if ((checkbox1.Checked = false) and (checkbox2.Checked = false)) then
```

begin

```
FLap_Pembelian.TM.Caption:= 'Seluruhnya' ;
```

```
FLap_Pembelian.TS.Enabled := false ;
```

```
FLap_Pembelian.Q1Label21.Enabled := false ;
```

end

else

begin

```
FLap_Pembelian.TM.Caption:= DateToStr(DateTimePicker2.date) ;
```

```
FLap_Pembelian.TS.Caption:= DateToStr(DateTimePicker3.date) ;
```

end ;

```
FLap_Pembelian.Preview ;
```

```
FLap_Pembelian.Q11.Close;
```

```
FLap_Pembelian.Free;
```

end;

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

begin

end;

end.

```
unit Ubayar_piutang;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ComCtrls, Grids, DBGrids, ExtCtrls, Db, DBTables, StdCtrls, jpeg, Buttons;
```

```
type
```

```
TFbayar_piutang = class(TForm)
  PageControl1: TPageControl;
  TabSheet1: TTabSheet;
  TabSheet3: TTabSheet;
  Query1: TQuery;
  DataSource1: TDataSource;
  Panel1: TPanel;
  DBGrid1: TDBGrid;
  DataSource2: TDataSource;
  Query2: TQuery;
  DataSource3: TDataSource;
  Query1KODE_CUST: TStringField;
  Query1NAMA: TStringField;
  Query1SUM: TFloatField;
  Query1SUM_1: TFloatField;
  Query1SUM_2: TFloatField;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  E: TEdit;
  Q3: TQuery;
  DataSource4: TDataSource;
  SpeedButton1: TSpeedButton;
  Panel4: TPanel;
  GroupBox2: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  date1: TLabel;
  Label4: TLabel;
  Label5: TLabel;
  Label3: TLabel;
  Label6: TLabel;
  Label11: TLabel;
  P2: TPanel;
  P1: TPanel;
  P4: TPanel;
  P5: TPanel;
  P6: TPanel;
  P7: TPanel;
  P8: TPanel;
  P3: TPanel;
  Q3NOMER: TIntegerField;
  Q3NO_FAKTUR: TStringField;
  Q3TGL_BAYAR: TDateTimeField;
  Q3JML_BAYAR: TFloatField;
  TabSheet4: TTabSheet;
  Panel5: TPanel;
  DBGrid3: TDBGrid;
  Y1: TDateTimePicker;
  Y2: TDateTimePicker;
  Label7: TLabel;
  Label12: TLabel;
  Button1: TButton;
  Q4: TQuery;
  DataSource5: TDataSource;
  DT: TDateTimePicker;
  Query2NOMER_FAKTUR: TStringField;
  Query2TGL: TDateTimeField;
  Query2KODE_CUST: TStringField;
  Query2NAMA: TStringField;
  Query2ALAMAT: TStringField;
  Query2TOTAL_JUAL: TFloatField;
```

*form*

piutang

```

Query2TUNAI: TFloatField;
Query2HUTANG: TFloatField;
Query2DIBAYAR: TFloatField;
Query2SALDO: TFloatField;
Query2JATUH_TEMPO: TDateTimeField;
Q4NOMER_FAKTUR: TStringField;
Q4TGL: TDateTimeField;
Q4KODE_CUST: TStringField;
Q4NAMA: TStringField;
Q4ALAMAT: TStringField;
Q4TOTAL_JUAL: TFloatField;
Q4TUNAI: TFloatField;
Q4HUTANG: TFloatField;
Q4DIBAYAR: TFloatField;
Q4SALDO: TFloatField;
Q4JATUH_TEMPO: TDateTimeField;
DBGrid2: TDBGrid;
Panel20: TPanel;
Button2: TButton;
Button3: TButton;
GroupBox3: TGroupBox;
D: TDBGrid;
Panel2: TPanel;
Image2: TImage;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
procedure PageControl1Change(Sender: TObject);
procedure EExit(Sender: TObject);
procedure EKeyPress(Sender: TObject; var Key: Char);
procedure FormShow(Sender: TObject);
procedure SpeedButton1Click(Sender: TObject);
procedure DtKeyPress(Sender: TObject; var Key: Char);
procedure Button1Click(Sender: TObject);
procedure Panel5Exit(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure DTChange(Sender: TObject);
procedure DTEExit(Sender: TObject);

```

```

private
{ Private declarations }
public
{ Public declarations }
end;

```

```

var
Fbayar_piutang: TFbayar_piutang;

```

implementation

```

uses UDM, UMain, UBayar_Piutang_R, UJatuh_Tempo_R;

```

```

{$R *.DFM}

```

```

procedure TFbayar_piutang.PageControl1Change(Sender: TObject);
begin
//Panel2.Caption:='Daftar Putang dari : '+DbGrid1.Fields[0].AsString+
'+DbGrid1.Fields[1].AsString;

If Dm.MJual.FIndkey([DbGrid2.Fields[0].AsString]) Then
Begin
P3.Caption:=DM.MJual.FieldName('NOMER_FAKTUR').AsString;
P4.Caption:=DM.MJual.FieldName('Tgl').AsString;

IF DT.Date<StrToDate(P4.Caption) Then
Dt.Date:=StrToDate(P4.Caption);

P1.Caption:=DM.MJual.FieldName('Kode_Cust').AsString;
P2.Caption:=DM.MJual.FieldName('Nama').AsString;

P5.Caption:=FormatFloat('#,0',Dm.MJual.FieldName('Hutang').AsFloat);

```

```

                                piutang
P8.Caption:=FormatFloat('#,0',Dm.MJual.FieldByName('Dibayar').AsFloat);
P7.Caption:=FormatFloat('#,0',Dm.MJual.FieldByName('Saldo').AsFloat);
P6.Caption:=Dm.MJual.FieldByName('Jatuh_tempo').AsString;

```

```

Q3.Close;
Q3.Sql.Clear;
Q3.Sql.Add('Select * from DByr_Piutang where NO_FAKTUR=:XX');
Q3.ParamByName('Xx').AsInteger:=StrToInt(P3.Caption);
Q3.Prepare;
Q3.Open;

```

```

If Fmain.NR(P7.Caption) >0 Then
Begin
    E.Enabled:=True;
    E.Color:=CLWhite;
    DT.Enabled:=True;
    Dt.Color:=CLWhite;
    SpeedButton1.Enabled:=True;
End Else
begin
    E.Enabled:=False;
    E.Color:=CLMenu;
    DT.Enabled:=False;
    Dt.Color:=CLMenu;
    SpeedButton1.Enabled:=False;
End;

```

```

End;
If DBGrid1.Fields[0].AsString<>' ' Then
    DTChange(Sender);
If PageControl1.ActivePageIndex=3 Then
    Button1Click(Sender);
end;

```

```

procedure Tfbayar_piutang.EExit(Sender: TObject);
begin
    If Fmain.NR(E.Text) > Fmain.NR(P7.Caption) Then
        Begin
            E.Text:=P7.Caption;
        End;

```

```

    E.Text:=FormatFloat('#,0',Fmain.NR(E.Text));
end;

```

```

procedure Tfbayar_piutang.EKeyPress(Sender: TObject; var Key: Char);
begin
    If Not(Key In ['0'..'9', #8, #13]) Then Key:=#0;
    If (Key=Chr(VK_return)) Then SendMessage(Handle,WL_NEXTDLGCTL,0,0);
end;

```

```

procedure Tfbayar_piutang.FormShow(Sender: TObject);
begin

```

```

    DT.DateTime := Date;
    Y1.date := Date ;
    Y2.date := Date ;
    Y1.Checked:=False;
    Y2.Checked:=False;

```

```

    Query1.Close;
    Query2.Close;
    Query1.Open;
    Query2.Open;
    PageControl1.ActivePageIndex:=0;

```

```

end;

```

```

procedure Tfbayar_piutang.SpeedButton1Click(Sender: TObject);
begin

```



piutang

```
If Fmain.NR(E.Text) <=0 Then  
Begin  
Exit;  
End;
```

```
//Mengecek Client Server  
DM.Mjual.Refresh;  
If DM.Mjual.FindKey([StrToInt(P3.Caption)]) Then  
Begin  
If (DM.Mjual.FieldByName('Saldo').AsFloat<>Fmain.NR(P7.Caption)) Then  
Begin  
MessageDlg('Ada User lain membayar piutang !'+#13+  
'Jadi.....'+#13+  
'Cek Lagi Total Pembayaran', mtInformation, [mbOK],0);  
PageControl1.ActivePageIndex:=0;  
Query1.Close;  
Query2.Close;  
Query1.Open;  
Query2.Open;  
Exit;  
End;  
End;
```

```
dm.Dbyr_piutang.Insert;  
dm.Dbyr_piutang.Fieldbyname('Nomer').asinteger:=dm.Dbyr_piutang.recordcount+1;  
dm.Dbyr_piutang.Fieldbyname('No_Faktur').asinteger:=StrToInt(P3.Caption);  
dm.Dbyr_piutang.Fieldbyname('Tgl_bayar').asdatetime:=StrToDate(FormatDateTime('dd/mm/yyyy',DT.  
te));  
dm.Dbyr_piutang.Fieldbyname('Jml_Bayar').asfloat:=fmain.nr(E.Text);  
dm.Dbyr_piutang.post;  
dm.Dbyr_piutang.ApplyUpdates;  
dm.Dbyr_piutang.CommitUpdates;  
dm.Dbyr_piutang.refresh;
```

```
If dm.Mjual.Findkey([StrToInt(P3.Caption)]) Then  
Begin  
dm.Mjual.Edit;  
dm.Mjual.Fieldbyname('Dibayar').asfloat:=  
dm.Mjual.Fieldbyname('Dibayar').asfloat+ Fmain.NR(E.Text);  
  
dm.Mjual.Fieldbyname('Saldo').asfloat:=  
dm.Mjual.Fieldbyname('hutang').asfloat-  
dm.Mjual.Fieldbyname('Dibayar').asfloat;  
  
dm.Mjual.post;  
dm.Mjual.ApplyUpdates;  
dm.Mjual.CommitUpdates;  
dm.Mjual.refresh;  
End;  
PageControl1.ActivePageIndex:=0;  
Query1.Close;  
Query2.Close;  
Query1.Open;  
Query2.Open;  
E.Text:='0';  
Exit;
```

```
procedure TFbayar_piutang.DtKeyPress(Sender: TObject; var Key: Char);  
begin  
If (Key=Chr(VK_return)) Then SendMessage(Handle,WM_NEXTDLGCTL,0,0);  
end;
```

```
procedure TFbayar_piutang.Button1Click(Sender: TObject);  
begin  
Q4.close;  
Q4.sql.clear;  
Q4.sql.add('select * from Mjual where Saldo >0');
```

piutang

```
If Y1.Checked Then
  Begin
    Q4.Sql.Add('And Jatuh_Tempo>=:XX');
    Q4.ParambyName('Xx').AsdateTime:=Y1.DateTime;
  End;

  If Y2.Checked Then
    Begin
      Q4.Sql.Add('And Jatuh_Tempo<=:YY');
      Q4.ParambyName('YY').AsdateTime:=Y2.DateTime;
    End;

  Q4.Prepare;
  Q4.Open;
```

end;

```
procedure Tfbayar_piutang.Panel5Exit(Sender: TObject);
```

```
begin
{
  FBayar_Piutang_R :=TFBayar_Piutang_R.Create(Self);
  FBayar_Piutang_R.A1.Caption:=P1.Caption;
  FBayar_Piutang_R.A2.Caption:=P2.Caption;
  FBayar_Piutang_R.A3.Caption:=P3.Caption;
  FBayar_Piutang_R.A4.Caption:=P4.Caption;
  FBayar_Piutang_R.A5.Caption:=P5.Caption;
  FBayar_Piutang_R.A6.Caption:=P6.Caption;
  FBayar_Piutang_R.A7.Caption:=P7.Caption;
  FBayar_Piutang_R.A8.Caption:=P8.Caption;

  FBayar_Piutang_R.Q3.Close;
  FBayar_Piutang_R.Q3.Sql.Clear;
  FBayar_Piutang_R.Q3.Sql.Add('Select * from DByr_Piutang where No_Faktur=:XX');
  FBayar_Piutang_R.Q3.ParambyName('Xx').AsInteger:=StrToInt(P3.Caption);
  FBayar_Piutang_R.Q3.Prepare;
  FBayar_Piutang_R.Q3.Open;

  FBayar_Piutang_R.Preview;
  FBayar_Piutang_R.Q3.Close;
  FBayar_Piutang_R.Free;
}
{
  FJatuh_Tempo_R :=TFJatuh_Tempo_R.Create(Self);

  FJatuh_Tempo_R.Q1.Close;
  FJatuh_Tempo_R.Q1.Sql.Clear;
  FJatuh_Tempo_R.Q1.Sql.Add('Select * from Mjual where saldo >0');
  If Y1.Checked Then
    Begin
      FJatuh_Tempo_R.Q1.Sql.Add('And Jatuh_Tempo>=:XX');
      FJatuh_Tempo_R.Q1.ParambyName('Xx').AsdateTime:=Y1.DateTime;
    End;

    If Y2.Checked Then
      Begin
        FJatuh_Tempo_R.Q1.Sql.Add('And Jatuh_Tempo<=:YY');
        FJatuh_Tempo_R.Q1.ParambyName('YY').AsdateTime:=Y2.DateTime;
      End;

  FJatuh_Tempo_R.Q1.Prepare;
  FJatuh_Tempo_R.Q1.Open;

  FJatuh_Tempo_R.Preview;
  FJatuh_Tempo_R.Q1.Close;
  FJatuh_Tempo_R.Free;
}
end;
```

```
procedure Tfbayar_piutang.Button2Click(Sender: TObject);
```

begin

piutang

```
FJatuh_Tempo_R :=TFJatuh_Tempo_R.Create(Self);
FJatuh_Tempo_R.Q1.Close;
FJatuh_Tempo_R.Q1.Sql.Clear;
FJatuh_Tempo_R.Q1.Sql.Add('Select * from Mjual where saldo >0');
If Y1.Checked Then
  Begin
    FJatuh_Tempo_R.Q1.Sql.Add('And Jatuh_Tempo>=:XX');
    FJatuh_Tempo_R.Q1.ParamByName('Xx').AsDateTime:=Y1.DateTime;
  End;

  If Y2.Checked Then
    Begin
      FJatuh_Tempo_R.Q1.Sql.Add('And Jatuh_Tempo<=:YY');
      FJatuh_Tempo_R.Q1.ParamByName('YY').AsDateTime:=Y2.DateTime;
    End;

  FJatuh_Tempo_R.Q1.Prepare;
  FJatuh_Tempo_R.Q1.Open;

  FJatuh_Tempo_R.Preview;
  FJatuh_Tempo_R.Q1.Close;
  FJatuh_Tempo_R.Free;
```

end;

```
procedure TFBayar_piutang.Button3Click(Sender: TObject);
begin
```

```
  FBayar_Piutang_R :=TFBayar_Piutang_R.Create(Self);
  FBayar_Piutang_R.A1.Caption:=P1.Caption;
  FBayar_Piutang_R.A2.Caption:=P2.Caption;
  FBayar_Piutang_R.A3.Caption:=P3.Caption;
  FBayar_Piutang_R.A4.Caption:=P4.Caption;
  FBayar_Piutang_R.A5.Caption:=P5.Caption;
  FBayar_Piutang_R.A6.Caption:=P6.Caption;
  FBayar_Piutang_R.A7.Caption:=P7.Caption;
  FBayar_Piutang_R.A8.Caption:=P8.Caption;

  FBayar_Piutang_R.Q3.Close;
  FBayar_Piutang_R.Q3.Sql.Clear;
  FBayar_Piutang_R.Q3.Sql.Add('Select * from DByr_Piutang where No_Faktur=:XX');
  FBayar_Piutang_R.Q3.ParamByName('Xx').AsInteger:=StrToInt(P3.Caption);
  FBayar_Piutang_R.Q3.Prepare;
  FBayar_Piutang_R.Q3.Open;

  FBayar_Piutang_R.Preview;
  FBayar_Piutang_R.Q3.Close;
  FBayar_Piutang_R.Free;
```

end;

```
procedure TFBayar_piutang.DTChange(Sender: TObject);
begin
```

```
  IF DT.Date>StrToDate(P6.Caption) Then
    Dt.Date:=StrToDate(P6.Caption);
end;
```

```
procedure TFBayar_piutang.DTExit(Sender: TObject);
begin
```

```
  IF DT.Date>StrToDate(P6.Caption) Then
    Dt.Date:=StrToDate(P6.Caption);
end;
```

end.