SISTEM INFORMASI PENJUALAN DAN PEMBELIAN BAHAN BAKU BANGUNAN DI. PD. PEMBANGUNAN RAYA BERBASIS *CLIENT_SERVER*

SKRIPSI

Diajukan untuk Menempuh Ujian Akhir Sarjana
Program Strata Satu Jurusan Teknik Informatika
Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia

GUGUN SUKMANA
10104036



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

LEMBAR PENGESAHAN

SISTEM INFORMASI PENJUALAN DAN PEMBELIAN BAHAN BAKU BANGUNAN DI. PD. PEMBAGUNAN RAYA BERBASIS *CLIENT_SERVER*

GUGUN SUKMANA
10104036

Pembimbing

Dian Dharmayanti, S.T.
NIP. 41277006005

Menyetujui,

Ketua Jurusan Teknik Informatika

Mira Kania Sabariah.Si., M.T.
NIP. 41277006011

LEMBAR PENGESAHAN

SISTEM INFORMASI PENJUALAN DAN PEMBELIAN BAHAN BAKU BANGUNAN DI. PD. PEMBAGUNAN RAYA BERBASIS *CLIENT_SERVER*

GUGUN SUKMANA
10104036

Penguji I	Penguji II
Dian Dharmayanti, S.T.	**Tati Haruhayati M.,S.T.**
NIP. 41277006005	**NIP. 41277006014**

Penguji III

Mira Kania Sabariah.Si., M.T.
NIP. 41277006011

Bandung, 19 Agustus 2009

Perihal: Plagiat Tugas Akhir

Saya yang bertanda tangan di bawah ini:

Nama : Gugun Sukmana

Nim : 10104036

Judul Tugas Akhir : Sistem Informasi Penjualan dan Pembelian Bahan Baku Bangunan Di. Pd Pembangun Raya

Menyatakan bahwa saya tidak melakukan tindakan meniru, menyalin atau menjiplak tugas akhir/karya ilmiah yang telah ada secara keseluruhan. Apabila saya terbukti melakukan kegiatan tersebut, maka saya bersedia untuk menerima sanksi yang diberikan sesuai dengan ketentuan yang ditetapkan dan berlaku di Jurusan Teknik Informatika Universitas Komputer Indonesia.

Mengetahui,

Yang memberi pernyataan

(Gugun Sukmana)

ABSTRAK

SISTEM INFORMASI PENJUALAN DAN PEMBELIAN BAHAN BAKU BANGUNAN DI. PD. PEMBANGUNAN RAYA BERBASISIS *CLIENT_SERVER*

Oleh

GUGUN SUKMANA
10104036

Dalam kehidupan yang semakin menuntut pada cepatnya pengolahan dan penyampaian informasi, dirasa semakin dibutuhkanya teknologi yang membantu dalam hal informasi. Hal tersebut dirasa dalam berbagai aspek, mulai dari sektor kesehatan, pendidikan hingga menyentuh aspek perdagangan. Untuk mengolah data menjadi informasi yang valid salah satunya kita dapat menggunakan teknologi yang telah terkomputerisasi. Dengan semakin banyaknya data yang harus diolah, pihak PD. Pembangunan Raya merasa bahwa sistem pengolahan yang ada saat ini sudah tidak sesuai lagi dengan perkembangan jaman yang ada. Sistem yang ada saat ini tidak menggunakan komputer sebagai alat pengolahan data, oleh karena itu masih belum terstruktur dengan baik. Terutama untuk mengolah data transaksi penjualan dan pembelian. Dalam sistem ini yang diolah adalah data barang data, pembelian barang, data penjualan barang, data *supplier*, data petugas, data konsumen. Masalah yang terjadi saat ini hasil yang diolah karena belum terstruktur dengan baik adalah data yang diolah terasa lambat, laporan yang dihasilkan harus melewati validasi beberapa kali.

Pada sistem informasi untuk memenuhi tugas ini akan dipaparkan bagaimana rancang bangun dari sistem informasi penjualan dan pembelian . Perancangan sistem ini menggunakan metode *waterfall*, alat yang digunakannya adalah *flowmap*, *Entity Relational Diagram* dan *Contex Diagram*, sedangkan untuk pengumpulan data menggunakan cara observasi, wawancara dan studi pustaka. Sedangkan alat

pengembangan aplikasi database menggunakan MySQL dan bahasa pemrograman menggunakan Borland Delphi 6.0.

Dari aplikasi ini nantinya keuntungan yang didapatkan pada pengolahan data penjualan dan pembelian sudah secara otomatis sehingga mengurangi kemungkinan dalam penginputan data penjualan dan pembelian, dari laporan pun dirasa dapat membantu dengan memperlihatkan hal yang dibutuhkan untuk mengolah data menjadi cepat dan mudah.

Kata Kunci :

Sistem informasi, penjualan dan pembelian, PD. Pembangunan Raya, metodologi *waterfall*.

ABSTRACT

INFORMATION SYSTEM SELLING AND PUCHASING RAW BUILDING IN PD. PEMBANGUNAN RAYA BASED CLIENT-SERVER

By

GUGUN SUKMANA
10104036

In live those assume in speed processing and submission information, its need some technology that helps in information submission. It's feels in every aspect, from healthy, education until trading aspect. To process data to be valid information, its need some computerized technology. When data that needs to be process was increase, PD. Pembangunan Raya feels the system were they belongs now, is not prefer again form this time. Although they have the system information, but for process it's still unstructured. In addiction data process, especially in selling and puchasing system. In this system, processing data are goods data, purchasing data, selling data, officer data n customer data. The problem what faced now are result from process are still unstructured and the data processing are fell slower, for the report from the system yields was needs to be validated repeatedly.

In system information for this final exam will be explained how to design information processing system of employee invoice. Scheme of this system using some method, such as waterfall, for the tools used flow map, ERD and Context Diagram and for data collecting technique using observation, interview and studying from books. For developing database application use Microsoft Access and for assembler use Borland Delphi 6.0

From this application, the benefit are in process selling and purchasing data now using automatic process, its help to prevent some redundancy data and remembering in reduction of receivable, in report what yield helped in shows to process data to be easier and faster.

Key Words:

Information system, selling and purchasing, PD. Pembangunan Raya, methodologies waterfall.

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Dengan memanjatkan rasa puji dan syukur kehadirat Allah SWT, karena atas rahmat dan hidayah-Nya penyususn dapat menyelesaikan Tugas Akhir yang berjudul ”Sistem Informasi Penjualan Dan Pembelian Bahan Baku Bangunan Di. PD. Pembangunan Raya”, yang ditujukan untuk memenuhi salah satu syarat Ujian Akhir Sarjana pada Fakultas Teknik dan Ilmu Komputer Jurusan Teknik Informatika Uniiversitas Komputer Indonesia.

Penulis sangat menyadari kekurangan yang ada pada laporan ini. Kekurangan ini dikarenakan keterbatasan penulis dalam hal ilmu pengetahuan dan pemahaman penulisan laporan. Akan tetapi, penulis berusaha menyusun laporan ini sebaik yang penulis bisa dengan segenap kemampuan dan usaha yang penulis bisa.

Dalam Tugas Akhir ini penyusun sadar banyak bantuan yang telah diberikan dari lingkungan sekitar sehingga tugas akhir ini dapat selesai dengan tepat waktu. Bagai kacang yang tak lupa akan kulitnya penyusun ingin memberikan rasa hormat dan terima kasih yang sangat dalam kepada:

1. Kedua orang tua, Ayahanda Udan suherman dan Ibunda hodjijah beliaulah yang telah memberikan seluruh curahan kehidupanya baik moral maupun moril kepada penulis.
2. Ibu Mira Kania Sabariah , S.T., M.T. selaku Ketua Jurusan Teknik Informatika Fakultas Teknik dan Ilmu Komputer UNIKOM.

3. Ibu Dian Dharmayanti, S.T. selaku pembimbing, dengan segala kesabarannya dalam membimbing penulis menyelesaikan tugas akhir ini.
4. Ibu Mira Kania Sabariah , S.T., M.T. selaku Wali Dosen Kelas IF -1, terima kasih telah menjadi wali yang kami hormati selama ini.
5. kakakku dan adikku yang ku sayangi Nia asyiah, Heni Devira, Yanti Purnama Sari, dan Hani Lesti Lestari terima kasih telah memberikan semua pehatian dan bantuannya saat keadaan senang maupun sulit
6. Dosen – dosen Teknik Informatika UNIKOM atas bimbingannya selama ini.
7. Senu Dwi aussi yang sangat banyak membantu pasti selalu akan ku kenang.
8. Teman-temanku yang selalu memberikanku semangat baik susuh maupun senang anak longhit senu, dian qwey, sonny, bobby yang selalu menemaniku
9. Teman-teman kelas IF – 1 satu perjuangan terutama anak-anak d'lenong jawil, salt, nde, tessa, galih, qnoy,hadi
10. Dan temen-temen ku seperjuangan yang selalu menemani dalam pembuatan skripsi ini satrio, Ryan farera, suherdiana, tessa saya ucapkan terima kasih
11. Serta seluruh pihak yang tidak bisa disebutkan satu persatu.

Semoga kebaikan yang telah diberikan kepada penyusun mendapatkan berkat yang melimpah dari Allah SWT.

Penyusun menyadari bahwasanya dalam penyusunan Tugas Akhir ini masih memiliki banyak kekurangan baik dari segi materi maupun dari segi penyusunannya mengingat terbatasnya pengetahuan dan kemampuan penulis. Untuk itu, dengan kerendahan hati penyusun mohon maaf dan penyusun sangat mengharapkan segala saran dan kritikan yang sekiranya dapat membantu penyusun agar dalam penyusunan selanjutnya bisa lebih baik lagi.

Bandung,19 agustus 2009

Penyusun

DAFTAR ISI

LEMBAR JUDUL
LEMBAR PENGESAHAN
ABSTRAK i
ABSTRACT iii
KATA PENGANTAR iv
DAFTAR ISI vi
DAFTAR TABEL xi
DAFTAR GAMBAR xiv
DAFTAR SIMBOL xviii
DAFTAR LAMPIRAN xx
BAB I PENDAHULUAN 1
1.1 Latar Belakang Masalah 1
1.2 Perumusan Masalah 2
1.3 Maksud dan Tujuan 2
1.3.1 Maksud 2
1.3.2 Tujuan 3
1.4 Batasan Masalah 3
1.5 Metodologi Penelitian 4
1.5.1 Metode Pengumpulan Data 4
1.5.2 Pengembangan Perangkat Lunak 5
1.6 Sistematika Penulisan 7
BAB II TINJAUAN PUSTAKA 8
2.1 Ruang Lingkup Perusahaan 8
2.1.1 Struktur Organisasi 8
2.1.2 Deskripsi Pekerjaan 9
2.2 Landasan Teori 10

2.2.1 Pengertian Sistem 10

2.2.2 Karakteristik Sistem 11
2.2.3 Konsep Dasar Data dan Informasi..... 13

2.2.3.1 Pengertian Data 13

2.2.3.2 Kualitas Informasi 14
2.2.3.3 Nilai Informasi 14
2.2.3.4 Siklus Informasi 15
2.2.4 Konsep Dasar Sistem Informasi..... 16

2.2.4.1 Komponen Sistem Informasi.....16
2.2.4.2 Tujuan Sistem Informasi.....17
2.2.4.3 Manfaat Sistem Informasi.....17

2.2.5 Konsep Pengembangan Sistem.....18
2.2.5.1 Diagram Konteks.....18

2.2,5.2 Diagram Alir Data.....18
2.2.5.3 Kamus Data.....20
2.2.6 Pengertian Basis Data.....21

2.2.6.1 Konsep Dasar Basis Data.....21
2.2.6.2 Basis Data Relasional.....21

2.2.6.3 DDL (*Data Definition Language*).....22

2.2.6.4 DML (*Data Manipulation Language*).....22

2.2.6.5 Pemodelan Basis Data.....23

2.2.6.6 Entity Relationship Diagram (ERD).....23

2.2.6.7 Database Management Sys*tem* (DBMS).....26
2.2.6.8 Client-Server.....27
2.2.7 Perangkat Lunak Pendukung.....28
2.2.7.1 Borland Delphi 6 30
2.2.7.2 MySQL…….……..31

BAB III ANALISIS DAN PERANCANGAN SISTEM 33
3.1 Analisis Sistem…….33
3.1.1 Analisis Prosedur Sistem Berjalan……..33
3.1.1.1 Prosedur Pemesanan……..…34
3.1.1.2 Prosedur Pemerimaan……36
3.1.1.3 Prosedur Penjualan…….38
3.1.1.4 Prosedur Laporan……40
3.1.2 AnalisisKebutuhan Non-Fungsional……42
3.1.2.1 Analisis Pengkodean……..…42
3.1.2.2 Analisis User……..42
3.1.2.3 Analisis Hardware…….43
3.1.2.4 Analisis Software……...44
3.1.2.5 Analisis Jaringan……45
3.1.2.6 Analisis Basis Data……46
3.1.3 Analisis Kebutuhan Fungsional……47
3.1.3.1 Digram Konteks……47
3.1.3.2 DFD……..48

3.1.4 Spesifikasi Proses…….57
3.1.5 Kamus Data……70
3.2 Perancangan Sistem……73
3.2.1 Perancangan Data……...74
3.2.1.1 Skema Relasi……...74
3.2.2 Stuktur Tabel……..78
3.2.3 Perancangan Kode……..84
3.2.4 Perancangan Struktur Menu……88
3.2.4.1 Struktur menu untuk bagian gudang……89
3.2.4.2 Struktur menu untuk bagian counter……90
2.5 Perancangan Antar Muka…….90
3.2.5.1 Spesifikasi Antar Muka……91
3.2.6 Perancangan Pesan……..……..106
3.2.7 Jaringan Semantik…….111
3.2.8 Perancangan Prosedural……112
BAB IV IMPLEMENTASI DAN PENGUJIAN 117
4.1 Implementasi 117
4.1.1 Perangkat Lunak Pendukung 117
4.1.2 Perangkat Keras Pendukung 118
4.1.3 Implementasi Database dan Antar Muka 118
4.1.3.1 Implementasi Database 119
4.1.3.2 Implementasi Antar Muka 128
4.2 Pengujian Alpha 134
4.2.1 Pengujian Sistem 135
4.2.1.1 Pengujian Login 136
4.2.1.2 Pengujian Pengolahan Data pengguna 138
4.2.1.3 Pengujian Pengolahan Data Supplier.....141

4.2.1.4 Pengujian Pengolahan Data Konsumen..... 145
4.2.1.5 Pengujian Pengolahan Data Pembelian 148
4.2.1.6 Pengujian Pengolahan Data Detail pembelian 152
4.2.1.7 Pengujian Pengolahan Data Barang..... 155
4.2.1.8 Pengujian Pengolahan Data penjualan 159
4.2.1.9 Pengujian Pengolahan Data detail penjualan 163
4.2.1.10 Pengujian Pengolahan Data ukuran……..... 166
4.2.1.11 Pengujian Pengolahan Data jenis..... 169
4.2.1.12 Pengujian Pengolahan Data merk..... 172
4.2.1.13 Pengujian Pengolahan Data bahan…….. 175
4.2.1.13 Pengujian Pengolahan Data bentuk……..…….. 178
4.2.1.13 Pengujian Pengolahan Data warna…….. 181
4.3 Pengujian Betha..... 184
BAB V KESIMPULAN DAN SARAN 190
5.1 Kesimpulan 190
5.2 Saran 190
DAFTAR PUSTAKA 192

DAFTAR TABEL

Tabel 3.1 Spesifikasi Proses 57

Tabel 3.2 Kamus Data 70

Tabel 3.3 Struktur file konsumen 78

Tabel 3.4 Struktur File Supplier 78

Tabel 3.5 Struktur File Barang 79

Tabel 3.6 Struktur File Pembelian 79

Tabel 3.7 Struktur File Detail Pembelian..... 80

Tabel 3.8 Struktur File Penjualan 80

Tabel 3.9 Struktur File Detail Penjualan..... 81

Tabel 3.10 Struktur File Merk..... 81

Tabel 3.11 Struktur File Jenis 82

Tabel 3.12 Struktur File Ukuran 82

Tabel 3.13 Struktur File Pengguna 82

Tabel 3.13 Struktur File Bahan 83

Tabel 3.13 Struktur File Bentuk 83

Tabel 3.13 Struktur File Warna ..... …. 84

Tabel 3.14 Struktur File merk jenis..... ….. 84

Tabel 3.15 Struktur File Ukuran jenis 84

Tabel 3.16 Struktur File Bahan jenis 84

Tabel 3.13 Struktur File bentuk jenis 84

Tabel 4.1 Nama tabel beserta file implentasi..... 119

Tabel 4.2 Implementasi antar muka user..... 128

Tabel 4.3 Rencana Pengujian..... 135

Tabel 4.4 Pengujian Login..... 136

Tabel 4.5 Tambah Data pengguna..... 138

Tabel 4. 6 Pengubahan Data pengguna..... 139

Tabel 4.7 Hapus data pengguna..... 140

Tabel 4.8 Tambah Data supplier..... 142

Tabel 4. 9 Pengubahan Data supplier..... 143

Tabel 4.10 Hapus data supplier..... 144

Tabel 4.11 Tambah Data konsumen..... 145

Tabel 4. 12 Pengubahan Data konsumen..... 146

Tabel 4.13 Hapus data konsumen..... 150

Tabel 4.14 Tambah Data pembelian..... 148

Tabel 4. 15 Pengubahan Data pembelian..... 149

Tabel 4.16 Hapus data pembelian..... 151

Tabel 4.17 Tambah Data detail pembelian..... 153

Tabel 4.18 Tambah Data barang..... 155

Tabel 4. 19 Pengubahan Data barang..... 156

Tabel 4.20 Hapus data pengguna..... 158

Tabel 4.21 Tambah Data penjualan..... 159

Tabel 4. 22 Pengubahan Data penjualan..... 161

Tabel 4.23 Hapus data penjualan..... 162

Tabel 4.24 cari Data detail penjualan..... 164

Tabel 4.25 Tambah Data ukuran..... 166

Tabel 4. 26 Pengubahan Data ukuran 167

Tabel 4.27 Hapus data ukuran 168

Tabel 4.28 Tambah Data jenis 169

Tabel 4. 29 Pengubahan Data jenis 170

Tabel 4.30 Hapus data jenis 171

Tabel 4.31 Tambah Data merk 172

Tabel 4. 32 Pengubahan Data merk 173

Tabel 4.33 Hapus data merk 174

Tabel 4.34 Tambah Data bahan 175

Tabel 4. 35 Pengubahan Data bahan ..... .176

Tabel 4.36 Hapus data bahan 177

Tabel 4.37 Tambah Data bentuk 178

Tabel 4. 38 Pengubahan Data bentuk 179

Tabel 4.39 Hapus data bentuk……..... 180

Tabel 4.40 Tambah Data warna 181

Tabel 4.41 Pengubahan Data warna 182

Tabel 4.42 Hapus data warna 183

DAFTAR GAMBAR

Gambar 1.1 Metode Waterfall 6

Gambar 2.1 Struktur Organisasi 9

Gambar 2.2 Siklus informasi 15

Gambar 2.3 *One to One Relationship* 25

Gambar 2.4 *One to Many Relationship* 25

Gambar 2.5 *Many to One Relationship* 25

Gambar 2.6 *Many to Many Relationship* 26

Gambar 3.1 *Flowmap* Pemesanan..... 35

Gambar 3.2 *Flowmap* Penerimaan..... 37

Gambar 3.3 *Flowmap* Penjualan..... 39

Gambar 3.4 *Flowmap* laporan..... 41

Gambar 3.5 Srtuktur jaringan..... 45

Gambar 3.6 *Entity Relationship Diagram* (ERD) 46

Gambar 3.7 *Diagram Konteks*..... 47

Gambar 3.8 DFD Level 1 (Sistem Informasi penjualan dan pembelian**)**..... 49

Gambar 3.9 DFD level 2 proses 1 (pengolahan data master) 50

Gambar 3.10 DFD level 2 proses 2 (transaksi) 50

Gambar 3.11 DFD level 2 proses 3 (pengolahan data pengguna)..... 51

Gambar 3.12 DFD level 2 proses 4 (pembuatan laporan) 51

Gambar 3.13 DFD level 3 proses 2.1 (pengolahan data ukuran) 52

Gambar 3.14 DFD level 3 proses 2.2 (pengolahan data jenis)..... 52

Gambar 3.15 DFD level 3 proses 2.3 ((pengolahan data merk) 53

Gambar 3.16 DFD proses 3 proses 2.4 (pengolahan data barang)..... 53

Gambar 3.17 DFD proses 3 proses 2.5 (pengolahan data supplir)..... 54

Gambar 3.18 DFD proses 3 proses 2.6 (pengolahan data konsumen) 54

Gambar 3.19 DFD proses 3 proses 2.7 (pengolahan data bahan) 55

Gambar 3.20 DFD proses 3 proses 2.8 (pengolahan data bentuk) 55

Gambar 3.21 DFD proses 3 proses 2.9 (pengolahan data warna) 56

Gambar 3.22 DFD proses 3 proses 3.1 (pembelian)..... 56

Gambar 3.23 DFD proses 3 proses 3.2 (penjualan) 57

Gambar 3.24 Skema Relasi……..... 77

Gambar 3.25 Stuktur Menu 88

Gambar 3.26 Stuktur Menu bagian gudang 89

Gambar 3.27 Stuktur Menu bagian counter..... 90

Gambar 3.28 Antar Muka Login..... 91

Gambar 3.29 Antar Muka Menu Utama..... 91

Gambar 3.30 Antar Muka data pengguna..... 92

Gambar 3.31 Antar Muka Data Barang..... 93

Gambar 3.32 Antar Muka Data Konsumen..... 94

Gambar 3.33 Antar Muka Menu data Supplier..... 95

Gambar 3.34 Antar Muka Detail Pembelian..... 96

Gambar 3.35 Antar Muka Jenis Barang..... 97

Gambar 3.36 Antar Muka Menu Data Merk Barang..... 98

Gambar 3.37 Antar Muka Menu data trasaksi pembelian..... 99

Gambar 3.38 Antar Muka Transaksi Penjualan..... 100

Gambar 3.39 Antar Muka Menu Data Ukuran..... 101

Gambar 3.40 Antar Muka Detail Penjualan..... 102

Gambar 3.41 Antar Muka Data Bahan…….……..…… 103

Gambar 3.42 Antar Muka Data Bentuk…… …….……. 104

Gambar 3.43 Antar Muka Data warna……. 105

Gambar 3.44 pesan permintaan user tidak dapat dilakukan..... 106

Gambar 3.45 pesan keluar dari program..... 106

Gambar 3.46 pesan nama dan *password* untuk *login*..... 107

Gambar 3.47 pesan untuk mengisi kata kunci saat melakukan pencarian…….. 107

Gambar 3.48 pesan untuk mengisi semua data……..…… 107

Gambar 3.49 pesan data berhasil disimpan…… 108

Gambar 3.50 pesan permintaan *user* tidak dapat dilakukan…… 108

Gambar 3.51 pesan untuk mengisi data yang akan diubah..... 108

Gambar 3.52 pesan no faktur tidak ditemukan…….. 109

Gambar 3.53 pesan data berhasil diubah…….…… 109

Gambar 3.54 pesan *stock* barang untuk distribusi kurang…….. 109

Gambar 3.55 pesan data berhasil ditambah…… 110

Gambar 3.56 jaringan semantik…….. 111

Gambar 3.57 Prosedur *login*..... 112

Gambar 3.58 Prosedur tambah data……..…….. 113

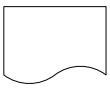
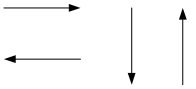
Gambar 3.59 Prosedur ubah data……... 114

Gambar 3.60 Prosedur hapus data……..…… 115

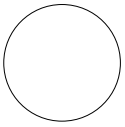
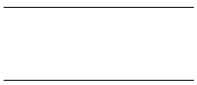
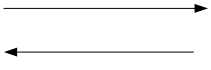
Gambar 3.61 Prosedur cari data……..…… 116

DAFTAR SIMBOL

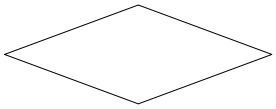
1. Diagram Aliran (*Flowmap*)

No	Simbol	Keterangan
1		Dokumen Menunjukan dokumen sebagai masukan/ keluaran baik secara manual/melaui kompuiter
2		Proses Manual menunjukan proses yang dikerjakan secara manual
3		Penyimpanan Dokumen Digunakan untuk penyimpanan data sebagai arsip secara manual
4		Aliran Data Menunjukan aliran data antar proses

2. *Data Flow Diagram* (DFD)

No	Simbol	Keterangan
1		Proses Menunjukan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran ataupun sebaliknya
2		Terminator Mewakilii entitas luar dimana sistem berkomunikasi
3		Penyimpanan Untuk memodelkan kumpulan data/ paket data
4		Aliran Menggambarkan gerakan paket data atau informasi dari suatu bagian lain dari sistem dimana sistem penyimpanan mewakili lokasi penyimpanan data

3. *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1		Data Entitas Segala sesuatu baik yang nyata maupun abstrak yang datanya akan direkam
2		Relasi Menunjukan adanya hubungan antar sejumlah entitas yang berasal dari himpunan entitas berbeda
3		Aliran Data Menyatakan penghubung antara relasi dengan data entitas dan data entitas dengan atribut

DAFTAR LAMPIRAN

LAMPIRAN A ANTAR MUKA A-1

LAMPIRAN B LISTING PROGRAM B-1

LAMPIRAN C HASIL KUESIONER C-1

LAMPIRAN D SURAT PENELITIAN D-1

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan kehidupan manusia saat ini sangat tinggi, yang ditunjang oleh berbagai aspek diantaranya teknologi, ekonomi, hiburan, dan lain-lain, memaksa setiap orang yang berkecimpung dalam bidangnya untuk terus mengembangkan diri mengikuti perkembangan zaman agar dapat terus bersaing dalam kehidupan. Manusia maupun perusahaan harus melakukan suatu tindakan yang sesuai agar dapat bersaing. Tindakan yang akan diambil oleh manusia maupun perusahaan memerlukan adanya informasi yang mendukung, agar tindakan tersebut sesuai dengan kebutuhan. Informasi dibutuhkan untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Kualitas dari suatu informasi tergantung pada tiga hal pokok yaitu keakuratan, tepat waktu dan relevan.

PD. Pembangunan Raya adalah sebuah perusahaan yang melakukan proses jual beli barang, tentunya melakukan proses pengolahan data agar dapat menghasilkan informasi yang dibutuhkan. Pengolahan data yang berjalan saat ini di PD. Pembangunan Raya bertujuan untuk menghasilkan informasi dan laporan mengenai hal-hal yang berhubungan dengan pembelian dan penjualan barang. Informasi maupun laporan yang dihasilkan dari proses pengolahan data yang saat ini berjalan diantaranya berupa informasi pembelian barang dan laporan penjualan barang. Proses pengolahan data yang dilakukan oleh staf saat ini, masih memiliki

beberapa kekurangan, diantaranya terjadi keterlambatan dalam menghasilkan informasi atau laporan, terjadi kesalahan data yang menyebabkan informasi yang dihasilkan tidak sesuai dengan kenyataan yang ada. Salah satu penyebab terjadinya kesalahan pengolahan data adalah karena untuk setiap proses pengolahan data seperti data datang barang, data barang gudang dan data penjualan barang menggunakan data yang terpisah sehingga sering terjadi perbedaan data.

Penyampaian informasi dapat dituangkan ke dalam bentuk laporan, sehingga semua pihak dari berbagai kalangan yang berkepentingan dapat memperoleh manfaat dari penyampaian tersebut. Berdasarkan wawancara untuk membangun sebuah aplikasi sistem informasi penjualan dan pembelian bahan baku bangunan di PD. Pembangunan Raya berbasis *client-server*.

1.2 Perumusan Masalah

Dari latar belakang diatas, dapat dirumuskan dalam suatu rumusan masalah yaitu bagaimana membangun aplikasi sistem informasi penjualan dan pembelian bahan baku bangunan di PD. Pembangunan Raya berbasis *client-server*.

1.3 Maksud dan Tujuan

1.3.1 Maksud

Maksud dari penulisan tugas akhir ini adalah untuk membangun Aplikasi Sistem informasi penjualan dan pembelian bahan baku bangunan di PD. Pembangunan Raya Berbasis *Client Server*.

1.3.2 Tujuan

Tujuan dari penulisan peneliatian ini adalah sebagai berikut:

a. Mempercepat dalam menghasilkan informasi agar tidak terjadi keterlambatan dalam memperoleh informasi
b. Mempermudah dan mengurangi kesalahan dalam pengolahan proses penjualan dan pembelian
c. Menyajikan data secara cepat dan akurat serta mendukung penggunaan lebih dari satu komputer.

1.4 Batasan Masalah

Batasan masalah yang dibuat bertujuan untuk membatasi permasalahan yang akan diselesaikan. Adapun batasan masalahnya adalah sebagai berikut :

a. Data yang yang diolah dalam aplikasi ini adalah :
    1. Data barang
    2. Data pembelian barang
    3. Data penjualan barang
    4. Data *supplier*
    5. Data petugas
    6. Data konsumen
b. Proses yang terdapat pada sistem ini berupa :
    1. Pemesanan barang
    2. Penerimaan barang
    3. Penjualan barang

4. Pembuatan laporan

c. Dalam aplikasi ini tidak melakukan proses retur pembelian maupun retur penjualan barang

d. Keluaran dari aplikasi ini berupa dokumen tercetak meliputi:

1. Laporan stok barang
2. Laporan pemesanan barang
3. Laporan penerimaan barang
4. Laporan penjualan barang

e. Sistem ini berbasis *client-server* antara bagian gudang dan bagian counter.

f. Model analisis perangkat lunak yang digunakan adalah pemodelan analisis terstruktur. Alat yang digunakan adalah flowmap dan diagram E-R, dan untuk menggambar proses yang digunakan adalah DFD (Data Flow Diagram)

g. Aplikasi ini dibangun menggunakan Borland Delphi 6 dan mysql sebagai DBMS.

1.5 Metodologi Penelitian

Menurut Pressman [7] metodologi adalah kesatuan metode-metode, prosedur-prosedur, aturan-aturan yang digunakan oleh suatu ilmu pengetahuan, seni atau disiplin yang lain. Metodologi penelitian yang dilakukan adalah :

1.5.1 Tahap pengumpulan data

Tahap pengumpulan data yang digunakan terdiri dari dua macam yaitu :

a. Studi Literatur

Studi literatur adalah pengumpulan data melalui buku-buku, jurnal, halaman web, makalah serta bacaan lain yang sesuai dengan topik yang dibahas.

b. Studi Lapangan

Studi lapangan adalah pengumpulan data dengan dua cara yaitu:

1. *Observasi*

   Tahap observasi adalah pengumpulan data dengan mengadakan penelitian dan peninjauan langsung ke PD. Pembangunan Raya

2. Wawancara

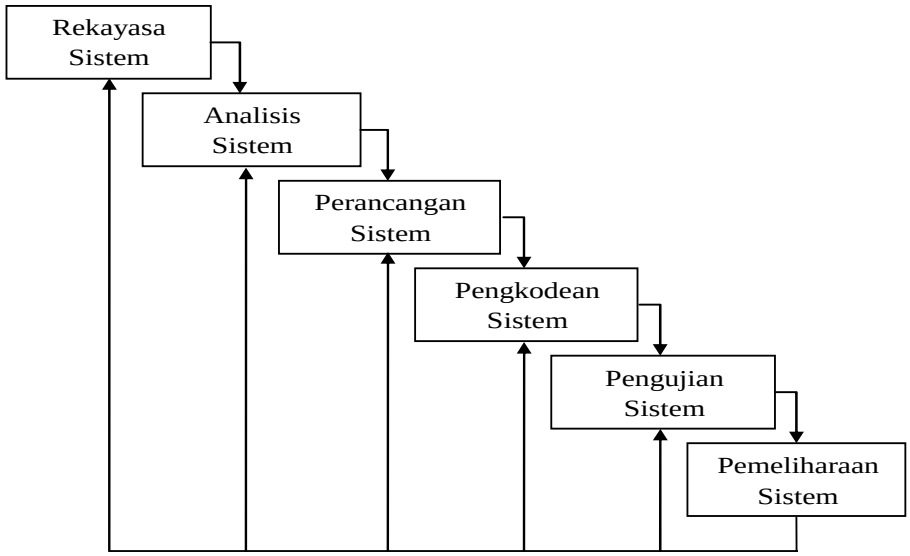
   Tahap wawancara adalah pengumpulan data dengan cara tanya jawab kepada para staff di PD. Pembangunan Raya.

1.5.2 Teknik pengembangan perangkat lunak

Aplikasi ini menggunakan metodologi *Waterfall* adapun tahapannya adalah sebagai berikut:

1. Rekayasa Sistem: tahap ini merupakan kegiatan pengumpulan data sebagai pendukung pembangunan sistem serta menentukan ke arah mana aplikasi ini akan dibangun.
2. Analisis Sistem: Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh aplikasi yang akan dibangun. Tahap ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.

3. Perancangan Sistem: perancangan antarmuka dari hasil analisis kebutuhan yang telah selesai dikumpulkan secara lengkap.
4. Pengkodean Sistem: hasil perancangan sistem diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Aplikasi yang dibangun langsung diuji baik secara unit.
5. Pengujian Sistem: penyatuan unit-unit program kemudian diuji secara keseluruhan.
6. Pemeliharaan Sistem: mengoperasikan aplikasi dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi yang sebenarnya. Dan dapat dilihat pada gambar 1.1



Gambar 1.1 Skema *waterfall*

1.6 Sistematika Penulisan

Sistematika penulisan laporan ini dibagi dalam beberapa bab dengan pokok pembahasan secara umum sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi latar belakang, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Dalam tinjauan pustaka terdiri dari ruang lingkup perusahaan tempat Aplikasi akan dibangun beserta teori yang digunakan dalam membangun aplikasi, mulai dari teori bahasa pembangunnya hingga pada database yang akan digunakan.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis kebutuhan dalam membangun aplikasi ini, analisis sistem yang sedang berjalan pada aplikasi ini sesuai dengan metode pembangunan perangkat lunak yang digunakan. Selain itu terdapat juga perancangan antarmuka untuk aplikasi yang akan dibangun sesuai dengan hasil analisis yang telah dibuat.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini berisi hasil implementasi dari hasil analisis dan perancangan disertai juga dengan hasil pengujian dari aplikasi ini.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan tentang keseluruhan dari pembangunan aplikasi ini dan saran tentang aplikasi ini.

BAB II

TINJAUAN PUSTAKA

2.1 Ruang Lingkup Perusahaan

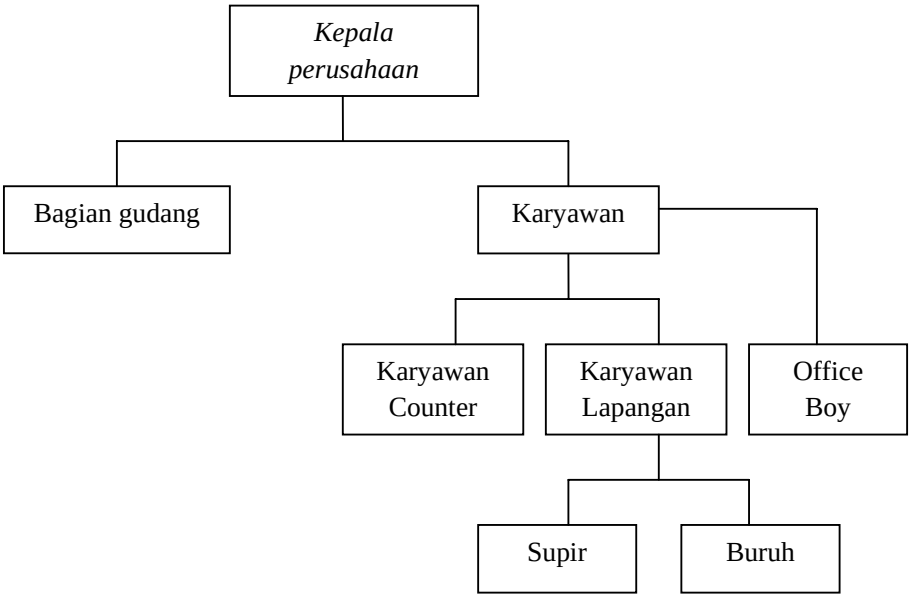
PD. Pembangunan Raya merupakan salah satu PD atau toko milik pribadi yang terletak di Jalan Otto Iskandar Dinata Bandung No.372 Telp (022) – 4206604/4200612. PD. Pembangunan Raya pada awal berdirinya merupakan atas dasar kekekuargaan dan pembukaan diresmikannya PD. Pembangunan Raya pada tanggal 20 September 1976.

PD. Pembangunan Raya menempati tanah seluas 77 m² dengan perincian tanah seluas 77 m² digunakan untuk bangunan gedung dan gudang.

Dilihat dari prospek masa depannya PD. Pembangunan Raya merupakan salah satu untuk memenuhi kebutuhan masyarakat untuk keperluan bahan baku bangunan.

2.1.1 Struktur Organisasi

Untuk lebih jelasnya akan digambarkan mengenai struktur organisasi di PD. Pembangunan Raya dapat dilihat pada gambar 2.1.



Gambar 2.1 Struktur Organisasi PD. Pembangunan Raya

2.1.2 Deskripsi pekerjaan:

1. *Kepala perusahaan*, bertugas mengawasi kinerja seluruh karyawan dan berkewajiban mengambil keputusan serta bertanggung jawab terhadap segala kegiatan di toko kepada *Owner*. Memberikan laporan pembelian dan penjualan setiap bulannya kepada *Owner*.
2. Bagian gudang, bertugas mengolah seluruh data pembelian dan penjualan, serta melakukan hubungan dengan pihak *supplier*. Memberikan laporan mengenai pembelian dan penjualan barang kepada *kepala perusahaan*

3. Karyawan *counter*, bertugas menjaga *counter* dan melayani *customer* yang akan membeli barang. Mengelola data barang yang ada di *counter* masing-masing dan melaporkannya kepada kordinator
4. Karyawan lapangan, bertugas dilapangan atau diluar kantor
5. Supir, bertugas mengirimkan dan membawa barang kepada *customer*
6. Buruh, bertugas mengangkut dan membawa barang kepada *customer*
7. *Office Boy*, bertugas menjaga kebersihan dan kondisi toko.

2.2 Landasan Teori

Dalam sub bab ini akan dibahas tentang teori-teori yang berhubungan dengan sistem informasi penjualan dan pembelian bahan baku bangunan Di PD. Pembangunan Raya.

2.2.1 Pengertian Sistem

Sistem dapat didefinisikan menurut dua kelompok pendekatan sistem, yaitu yang melakukan pada prosedurnya dan yang menekankan pada komponen atau elemen. Menurut JOG [1] *Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.* Dari pengertian diatas dapat diambil suatu kesimpulan bahwa suatu sistem merupakan elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan tertentu.

2.2.2 Karakteristik Sistem

Menurut JOG [1] suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu memiliki komponen-komponen *(components)*, batas sistem *(boundary)*, lingkungan luar sistem *(environment)*, penghubung *(interprest)*, masukan *(input)*, keluaran *(output)*, pengolah *(process)* dan sasaran *(objective)* dan tujuan *(goal)*.

1. Komponen Sistem (*System Components*)

Komponen sistem atau elemen-elemen sistem dapat berupa suatu sub sistem atau bagian-bagian dari sistem. Setiap sistem baik besar maupun kecil, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu yang lebih besar yang disebut *supra system*.

2. Batas Sistem (*System Boundary*)

Batas sistem merupakan daerah-daerah yang membatasi antara satu sistem dengan sistem lainnya dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

3. Lingkungan Luar Sistem (*System Environment*)

Lingkungan luar sistem dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan yang

merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem *(System Interprest)*

Penghubung sistem merupakan media penghubung antara satu sub sistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung akan terjadi interaksi antar subsistem, sehingga membentuk satu kesatuan.

5. Masukan Sistem *(System Input)*

Masukan adalah suatu energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Contoh *maintenance input* di dalam sistem komputer adalah program, yang digunakan untuk mengoperasikan komputer. Sedangkan *signal input* adalah energi yang diproses untuk mendapatkan keluaran. Contoh *signal input* di dalam sistem komputer adalah data, yang dapat diolah menjadi Informasi.

6. Keluaran Sistem *(System Output)*

Keluaran (*Output*) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah Sistem (*System Process*)

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya, yang bertugas untuk merubah masukan menjadi keluaran.

8. Sasaran Sistem (*System Objective*)

Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Suatu operasi sistem akan berguna dan berhasil apabila mencapai sasaran atau tujuannya. Sasaran sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.2.3 Konsep Dasar Data dan Informasi

2.2.3.1 Pengertian Data

Data adalah kumpulan dari fakta-fakta, kejadian-kejadian yang dapat berupa simbol, angka, huruf, dan lain-lain yang berguna bagi suatu pengolahan data *(process*) atau sebagai masukan (*input*) bagi suatu proses(Drs. Moekijat) [2]. .

Pengertian Informasi

Informasi merupakan hasil pengolahan dari sebuah model, formasi, organisasi, ataupun suatu perubahan bentuk dari data yang memiliki nilai tertentu, dan bisa digunakan untuk menambah pengetahuan bagi yang menerimanya. Dalam hal ini, data bisa dianggap sebagai obyek dan informasi adalah suatu subyek yang bermanfaat bagi penerimanya. Informasi juga bisa disebut sebagai hasil pengolahan atau pemrosesan data.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata, tetapi data merupakan bentuk yang masih mentah dan belum dapat memberikan arti banyak bagi pemakai, sehingga perlu diolah lebih lanjut untuk menghasilkan informasi yang dibutuhkan. Dengan kata lain informasi adalah hasil dari pengolahan data.

2.2.3.2 Kualitas Informasi

Kualitas dari suatu informasi tergantung pada lima hal pokok yaitu:

a. Akurat (*accurate*)

Informasi harus bebas dari kesalahan-kesalahan dan tidak menyesatkan, dalam hal ini informasi harus jelas mencerminkan maksudnya.

b. Tepat waktu (*time lines*)

Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan suatu landasan dalam mengambil sebuah keputusan dimana bila pengambilan keputusan terlambat maka akan berakibat fatal untuk organisasi.

c. Relevan (*relevance*)

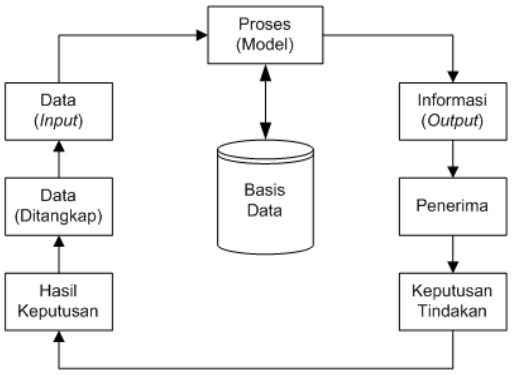
Informasi harus mempunyai manfaat untuk pemakainya, dimana relevansi informasi untuk tiap-tiap individu berbeda tergantung pada yang menerima dan yang membutuhkan.

2.2.3.3 Nilai Informasi

Nilai dari Informasi *(value of information)* ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu Informasi dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan Informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan.

2.2.3.4 Siklus Informasi

Data agar menjadi lebih berarti dan berguna dalam bentuk Informasi, maka perlu diolah melalui suatu model tertentu. Data yang telah diolah tersebut kemudian diterima oleh penerima, lalu penerima membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat sejumlah data kembali. Data tersebut akan ditangkap sebagai *input*, dan diproses kembali lewat suatu model dan seterusnya sehingga membentuk suatu siklus. Siklus ini disebut dengan siklus Informasi (*information cycle*) atau disebut pula siklus pengolahan data (*processing cycles*).



Gambar 2.2 Siklus informasi

2.2.4 Konsep Dasar Sistem Informasi

Sistem informasi dalam sebuah sistem meliputi pemasukan data (*input*) kemudian diolah melalui suatu model dalam pemrosesan data, dan hasil informasi akan ditangkap kembali sebagai suatu input dan seterusnya sehingga membentuk siklus informasi yang dapat diperoleh dari sistem informasi sebagai sistem khusus dalam organisasi untuk mengolah informasi tersebut

Sistem informasi adalah suatu sistem yang dibuat oleh manusia yang terdiri dari komponen-komponen dalam organisasi untuk mencapai suatu tujuan yaitu menyajikan informasi. Pengertian lain dari sistem informasi adalah sekumpulan prosedur organisasi yang pada saat dilaksanakan dan memberikan informasi bagi pengambil keputusan dan/atau untuk mengendalikan informasi.

2.2.4.1 Komponen Sistem Informasi

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building blok*) (Leman)[3] yaitu:

a. *Hardware* yaitu suatu perangkat keras dalam komputer yang kita bisa sentuh dan rasakan.
b. *Software* yaitu suatu perangkat lunak di dalam komputer yang berfungsi untuk mengoperasikan suatu aplikasi di dalam sistem komputer.
c. Data yaitu sekumpulan karakter yang diterima sebagai masukan (*input*) untuk sistem informasi dan disimpan serta diolah.
d. Prosedur yaitu suatu urutan pekerjaan tata usaha yang biasanya melibatkan beberapa orang dalam satu bagian atau lebih, dan disusun untuk menjamin

adanya perlakuan yang seragam terhadap transaksi-transaksi perusahaan yang terjadi.

e. *User* yaitu orang yang terlibat dalam sistem informasi seperti operator, pemimpin sistem informasi, dan sebagainya.

2.2.4.2 Tujuan Sistem Informasi

1. Integrasi sistem

   Menghubungkan sistem individu atau kelompok

2. Efisiensi pengelolaan sistem

   Pengelolaan data berkaitan dengan sistem informasi

3. Dukungan keputusan untuk manajemen

   Melengkapi informasi guna kebutuhan proses pengambilan kebutuhan

2.2.4.3 Manfaat Sistem Informasi

Sistem Informasi memiliki beberapa manfaat, yaitu:

1. Menghemat tenaga kerja
2. Peningkatan efisiensi
3. Mempercepat proses
4. Perbaikan dokumentasi
5. Pencapaian standar
6. Perbaikan keputusan

2.2.5 Konsep Pengembangan Sistem

Perancangan sistem secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen sistem Informasi yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau *user* mengenai sistem yang baru. Sedangkan desain sistem secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainnya yang akan mengimplementasikan sistem.

Penggambaran dan rancangan model sistem Informasi secara logika dapat dibuat dalam bentuk Diagram Konteks dan Diagram Alir Data (DAD) atau Data *Flow Diagram* (DFD).

2.2.5.1 Diagram Konteks

Diagram konteks merupakan arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antar sistem dengan bagian luar (kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.

2.2.5.2 Diagram Alir Data

Diagram Alir Data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.
3. Tidak semua bagian dari sistem harus ditunjukkan dengan jumlah level yang sama.

Simbol-simbol yang digunakan pada diagram alir data atau *data flow diagram* antara lain:

a. Entitas luar (*external entity*)

Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem, disimbolkan dengan suatu kotak notasi. External entity tidak termasuk bagian dari sistem bila sistem informasi dirancang untuk satu bagian maka bagian lain yang masih terkait menjadi external entity.

b. Arus data (*data flow*)

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukan dengan arah panah dan garis diberi nama atau arus data yang mengalir. Arus data ini mengalir diantara proses, data store dan menunjukan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

c. Proses (*proccess*)

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data keluar. Proses berfungsi

mentransformasikan satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa data keluaran. Proses sering juga disebut *bubble*.

d. Simpanan data (*data store*)

Simpanan data merupakan tempat penyimpanan data yang ada dalam sistem. Data *store* dapat disimbolkan dengan dua garis sejajar atau dua garis dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke *database*.

2.2.5.3 Kamus Data

Kamus data merupakan kumpulan data-data. Kamus data adalah katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Kamus data merupakan tempat penyimpanan definisi dari aliran-aliran data, file-file dan proses-proses dalam sebuah sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat memberikan informasi mengenai definisi struktur pemakaian masing-masing elemen, dapat mendefinisikan data yang mengalir di sistem dengan lengkap, dapat menghindari duplikasi elemen-elemen dan menghindari konflik antara elemen-elemen. Kamus data berfungsi untuk membantu pelaku sistem untuk mengartikan alokasi secara detail dan mengorganisasikan semua elemen data yang digunakan dalam sistem secara persis sehingga pemakai dan penganalisis sistem mempunyai dasar pengertian yang sama tentang masukan, keluaran, penyimpanan dan proses.

2.2.6 Pengertian Basis Data

Basis data terdiri dari kata basis dan data. Basis dapat diartikan gudang atau tempat bersarang dan data yang berarti representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, hewan, peristiwa, konsep dan sebaginya yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya.

Jadi dapat disimpulkan bahwa basis data merupakan kumpulan data (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Atau bisa diartikan sebagai kumpulan file, tabel, arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.2.6.1 Konsep Dasar Basis Data

Basis data (*database*) dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis data menunjukan suatu kumpulan data yang dipakai dalam sistem informasi disebut sistem basis data (*database system*).

2.2.6.2 Basis Data Relasional

Konsep sebuah *database* adalah terdiri atas tabel-tabel yang terorganisasi. Tabel-tabel tersebut dapat saling berelasi untuk menghasilkan suatu informasi, untuk mengakses data yang ada dalam tabel-tabel tersebut digunakan sebuah perintah SQL (*Structured Query Language*)

2.2.6.3 DDL (*Data Definition Language*)

Merupakan kelompok perintah yang digunakan untuk melakukan pendefinisian database dan pendefinisian tabel. Dengan kelompok perintah dalam DDL ini maka kita dapat membuat tabel, mengubah srukturnya, menghapus tabel, membuat indeks untuk tabel, dan lain-lain yang bermuara pada pembentukan struktur database. DDL atau *Data Definition Language* adalah bagian dari SQL yang digunakan untuk mendefinisikan data dan objek database, dimana terdapat perintah-perintah sebagai berikut :

1. *CREATE*

   Untuk mendefinisikan database, maupun tabel sebagai data yang akan disimpan maupun diakses

2. *ALTER*

   Untuk memodifikasi tabel, baik itu menambah, menghapus, maupun mengganti kolom/field pada tabel

3. *DROP*

   Untuk menghapus tabel dan *database*

2.2.6.4 DML (*Data Manipulation Language*)

Perintah (statement) SQL digunakan untuk melakukan manipulasi data dalam database, menambahkan (*insert*), Mengubah (*update*), menghapus (*delete*), mengambil dan mencari data (*query*). DML atau *Data Manipulation Language*

adalah bagian dari SQL yang digunakan untuk memulihkan dan memanipulasi data. terdapat perintah-perintah yang digunakan dalam DML adalah sebagai berikut:

4. *SELECT*

   Untuk mengambil atau menampilkan data dari tabel pada database

5. *INSERT*

   Untuk menyisipkan data pada tabel

6. *UPDATE*

   Untuk memperbaharui nilai suatu data dalam database

7. *DELETE*

   Untuk menghapus record pada tabel

2.2.6.5 Pemodelan Basis Data

Pemodelan basis data dibutuhkan agar diperoleh suatu sistem yang sesuai dengan apa yang diinginkan, dimana dapat melalui tahapan berikut :

2.2.6.6 *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram atau biasa dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah *database*. Diagram ini memiliki dua komponen utama yaitu *entity* dan relasi. Untuk melambangkan fungsi diatas maka digunakan simbol-simbol yang bisa dilihat pada daftar simbol.

Elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut:

1. *Entity* (Entitas)

Pada E-R diagram, entity digambarkan dengan sebuah bentuk persegi panjang. Entity adalah sesuatu apa saja yang ada didalam sistem, nyata maupun abstrak dimana data tersimpan. Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama, yaitu : orang, benda, lokasi kejadian (terdapat unsur waktu didalamnya).

2. *Relationship* (Relasi)

Pada E-R diagram, relationship dapat digambarkan dengan sebuah bentuk belah ketupat. Relationship adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya relationship diberi nama dengan kata kerja dasar, sehinga memudahkan untuk melakukan pembacaan relasinya.

3. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relationship. Maksudnya adalah sesutau yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relationship, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan relationship.

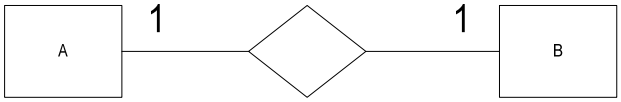
4. Kardinalitas

Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lainnya. Dari sejumlah kemungkinan banyaknya hubungan yang terjadi dari entitas, kardinalitas relasi merujuk kepada hubungan maksimum

yang terjadi dari entitas yang satu ke entitas yang lainnya dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu :

a. One to one Relationship

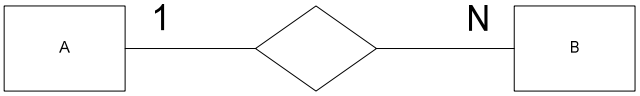
Tingkat hubungann satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Gambar 2.3 *One to One Relationship*

b. One to many Relationship

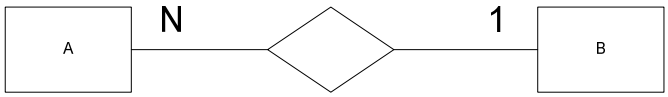
Tingkat hubungan satu ke banyak adalah untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua.



Gambar 2.4 *One to Many Relationship*

c. Many To One Relationship

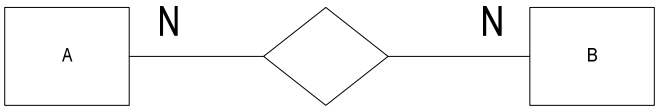
Untuk banyak kejadian pada entitas yang pertama hanya dapat mempunyai satu hubungan dengan kejadian pada entitas yang kedua.



Gambar 2.5 *Many to One Relationship*

d. Many to many Relationship

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua.



Gambar 2.6 *Many to Many Relationship*

5. *Key* (Kunci)

Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam suatu entitas. *Key* memiliki beberapa jenis sesuai dengan kegunaannya masing-masing, yaitu *primary key* (kunci utama), *foreign key* (kunci tamu).

2.2.6.7 *Database Management System* (DBMS)

Database adalah kumpulan data yang saling berkaitan, berhubungan yang disimpan secara bersama-sama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi berbagai kebutuhan. Data-data ini harus mengandung semua Informasi untuk mendukung semua kebutuhan sistem.

Proses dasar yang dimiliki oleh *database* ada 4, yaitu :

1. Pembuatan data-data baru (*create database*)
2. Penambahan data (*insert*)
3. Mengubah data (*edit*)
4. Menghapus data (*delete*)

Sistem manajemen *database* (*Database Management System*) merupakan sistem pengoperasian dan sejumlah data pada komputer. Dengan sistem ini dapat merubah data, memperbaiki data yang salah dan menghapus data yang tidak dapat dipakai. Sistem manajemen database merupakan suatu perluasan *software* sebelumnya mengenai *software* pada generasi komputer yang pertama.

Salah satu tujuan DBMS adalah untuk menyediakan fasilitas atau antarmuka (*interface*) dalam melihat atau menikmati data kepada pemakai. Untuk itu, sistem tersebut seringkali akan menyembunyikan detail tentang bagaimana data disimpan, dipakai atau dipelihara. Karena itu, seringkali data yang dilihat oleh pemakai sebelumnya berbeda dengan yang tersimpan secara fisik. Perangkat lunak yang termasuk DBMS adalah : dBase III+, dBaseIV, Fox Base, Interbase, Ms.Access dan Borland Paradox atau Borland-Interbase.

2.2.6.8 Client-Server

Konsep dari client server adalah sebagai Sebuah aplikasi yang dapat dianggap sebagai requestor (*client*) atau dapat juga dianggap sebagai provider (*server*). [4]

Biasanya jumlah *client* jauh lebih banyak daripada jumlah *server* dan mampu memberikan layanan kepada banyak *client* dengan kemampuan yang sama sebagaimana ketika hanya melayani sebuah *client* dari sisi suatu arsitektur

client server, bahwa *client* adalah sebuah aplikasi yang berjalan pada komputer pribadi dan bergantung pada *server* untuk mengerjakan oprerasi. Sedangkan *server* adalah node yang memungkinkan node lain pada LAN untuk mengakses sumbernya. *Server* ini bersifat terdedikasi yang artinya node tersebut dapat dipakai dengan cara lain.

2.2.7 Perangkat Lunak Pendukung

Perangkat lunak merupakan suatu kebutuhan yang harus dipenuhi oleh suatu sistem. Kebutuhan ini diperlukan untuk mencapai suatu tujuan.

2.2.7.1 Borland Delphi 6

Borland delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrograman yang terstruktur. Keunggulan lain delphi adalah dapat dipergunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows (Husni) [5].

1. Komponen Delphi

Component palette terdiri dari beberapa komponen yang dapat dipilih yang digunakan untuk menangani beberapa tugas pemrograman. Komponen-komponen yang terletak pada bagian *component palette* sudah ditata dalam beberapa tab yang masing-masing menunjukan maksud dan fungsi. Masing-masing tab

ditampilkan dalam konfigurasi default yang semua juga tergantung pada versi delphi yang digunakan.

2. Fitur Pada Delphi 6

Fitur baru dan perbaikan yang ada pada Borland Delphi 6 ini adalah :

a. IDE (Interface Development Environtment)

Lingkungan pengembangan aplikasi (IDE) Borland Delphi 6 telah mengalami perubahan dari versi sebelumnya. Diantaranya, terdapat Compiler Message, perubahan pada Component Pallete, Code Insight dan Debugger

b. Web

Borland Delphi 6 menyediakan Intraweb buatan AtoZed Software, yang dapat digunakan untuk membuat aplikasi *web server* dengan sarana standar. Borland Delphi 6 juga mendukung pada Apache 2. Borland juga menghilangkan Win-CGI sebagai target aplikasi Web server dan web service. Fasilitas untuk *Web server* juga mengalami perbaikan - perbaikan

c. COM

Sekarang ini dengan Delphi 6.0, dapat membuat CoClass wrapper bagi pengembangan – pengembangan .NET dengan cara menggunakan kotak dialog *Import Type Library*. Dengan adanya *resulting wrapper*, maka fitur interoperabilitas dari Microsoft’s .NET Framework dapat digunakan.

d. Database

Pada Delphi 6.0 driver dbExpress telah diupdate bagi Informix SE, Oracle 9i, DB2 7.2, InterBase 6.5, dan MySQL 3.23.49. Driver baru bagi

MSSQL 2000 juga tersedia. Disamping itu, beberapa hal baru dan perubahan juga dilakukan pada komponen *database*. Borland juga telah membuang SQL Links. Borland merekomendasikan pemakaian dbExpress bagi database SQL Server yang diakses di Delphi.

e. Component Library

Jika ditelusuri komponen librari Delphi 6.0, maka akan ditemukan komponen baru, unit baru, komponen yang berubah, komponen yang hilang dan komponen yang mendukung bagi tema Windows XP.

f. Runtime Library

Beberapa perubahan di Runtime Library antara lain ialah perubahan pada unit Classes, Math, StdConv,StrUtils, SysUtils, VarCmplx, dan Variants.

g. Compiler

Kompiler Delphi dcc32 sekarang ini support terhadap tiga warning kompiler tambahan, yaitu *Unsafe_Type*, *Unsafe_Code*, dan *Unsafe_Cast*. Warnings tersebut defaultnya adalah *disabled*, tetapi dapat di-*enabled*. Fitur ini sangat membantu kita ketika akan mem-port kode ke lingkungan eksekusi terkendali di *platform* Microsoft's .NET.

h. Model Maker

Sarana baru yang disebut ModelMaker dapat membantu memudahkan proses desain, konstruksi, dan pengelolaan *class* dan *interface*. *Model Maker* juga memiliki sarana untuk pembuatan diagram UML-style, yang dapat dipakai untuk membuat dan memodifikasi *source code project*.

2.2.7.2 MySQL

MySQL adalah suatu sistem manajemen database. Suatu database adalah sebuah kumpulan data yang terstruktur. Untuk menambahkan, mengakses, dan memproses data yang tersimpan pada suatu database komputer anda memerlukan sistem manajemen database seperti MySQL. Karena komputer sangat unggul dalam menangani sejumlah besar data, sistem manajemen database memainkan suatu peranan yang penting dalam komputasi, baik sebagai utility stand-alone maupun bagian dari aplikasi lainnya.

Suatu database relasional menyimpan data dalam tabel-tabel terpisah. Hal ini memungkinkan kecepatan dan fleksibilitas. Tabel-tabel yang dihubungkan dengan relasi yang ditentukan membuatnya bisa mengkombinasikan data dari beberapa tabel pada suatu permintaan. Bagian SQL dari kata MYSQL berasal dari "Structured Query Language" bahasa paling umum yang dipergunakna untuk mengakses database.

Beberapa perintah dasar SQL yang sering dipergunakan pada MySQL adalah sebagai berikut :

a. Create Database

Yaitu perintah yang digunakan untuk membuat database baru.

Sintaks : Create database database_nama database

b. Drop Database

Yaitu perintah yang digunakan untuk menghapus database.

Sintaks : Drop Tabel Tabel_name

c. Create Tabel

Yaitu perintah yang digunakan untuk membuat tabel baru.

Sintaks Create Tabel tabel_name (create_definition)

d. Describe

Yaitu perintah yang digunakan untuk mendeskripsikan tabel atau logam

Sintaks Describe (Desc) tabel [colum]

e. Alter Tabel

Yaitu perintah yang digunakan untuk memodifikasi tabel

Sintaks Alter [Ignor] Tabel table_name

f. Drop Tabel

Yaitu perintah yang digunakan untuk menghapus tabel

Sintaks Drop Tabel tabel_name [tabel_name..]

g. Delete

Yaitu perintah yang digunakan untuk menghapus record dri tabel

Sintaks Delete From tabel_name Where Where_definiition

h. Select

Yaitu perintah yang digunakan untuk query ke database

Sintaks select*from tabel_name

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Sebelum memasuki tahap perancangan program, tahap analisis dilakukan agar nantinya dalam merancang program tidak terjadi kesalahan. Dikarenakan hal tersebut analisis adalah tahap penting sebelum memasuki tahap selanjutnya, apabila pada tahap ini terjadi kesalahan maka tahap selanjutnya sudah dipastikan akan terjadi kesalahan.

Analisis sistem bertujuan untuk mempelajari prosedur yang sedang berjalan sekarang dan kebutuhan atau keinginan dari orang yang akan menggunakan aplikasi ini.

Tujuan dari perancangan system ini secara garis besar adalah untuk menghasilkan bentuk perancangan yang dapat memenuhi kebutuhan akan penyelesaian masalah secara tepat dan benar .

3.1.1 Analisis Prosedur Sistem Berjalan

Tujuan analisis prosedur adalah untuk mengetahui suatu proses yang ada dalam sistem dan siapa pelakunya dan mengecek peran dari pihak yang terkait, serta untuk mengetahui apakah informasi yang dibutuhkan atau dihasilkan oleh suatu pihak sudah sesuai dengan kebutuhan.

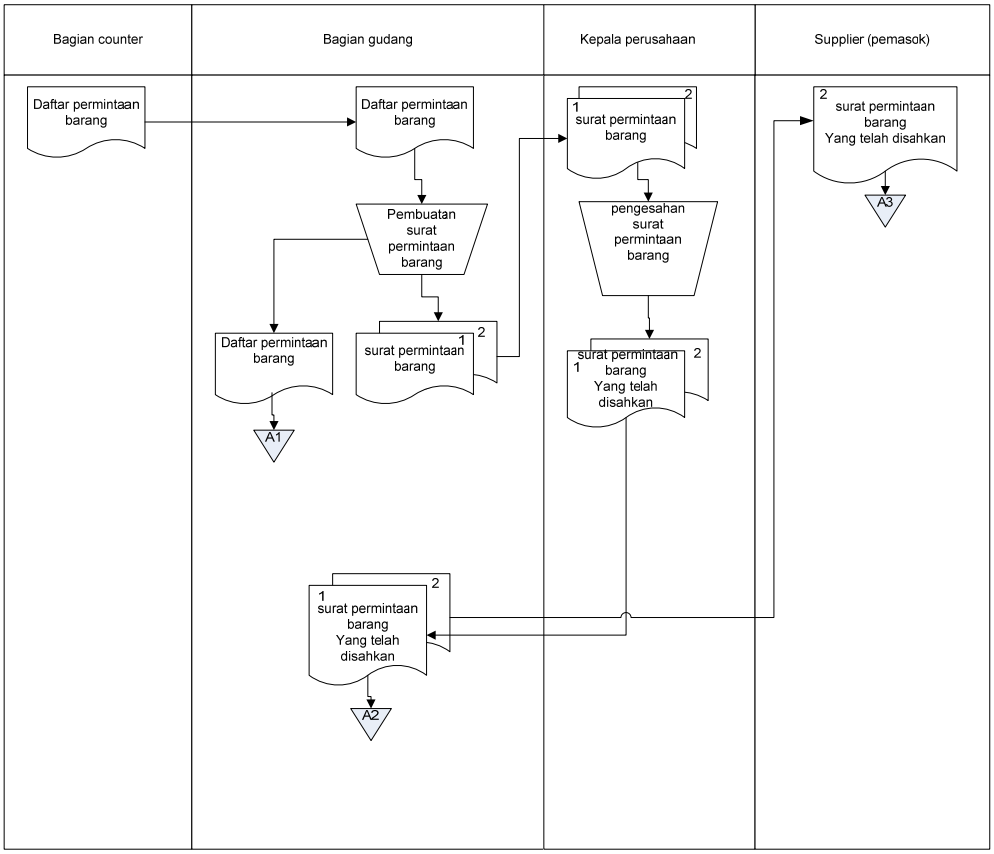
Adapun dalam prosedur sistem yang sedang berjalan saat ini terbagi menjadi empat bagian, yaitu: proses prosedur pemesanan, prosedur penerimaan

prosedur penjualan dan prosedur pelaporan. Analisis terhadap sistem yang berjalan tersebut bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut, dan masalah-masalah yang dihadapi, yaitu:

3.1.1.1 Prosedur Pemesanan

Prosedur ini berisi mengenai proses prosedur barang pemesanan. Adapun prosedurnya adalah sebagai berikut :

1. Bagian counter memberikan daftar permintaan kebagian gudang
2. kemudian bagian gudang membuat surat pemesanan barang dijadikan rangkap dua dan daftar permintaan barang disimpan sebagai arsip
3. selelah itu bagian gudang memberikan surat pemesanan barang yang rangkap dua ke kepala perusahaan untuk disahkan
4. setelah disahkan oleh kepala perusahaan kemudian di berikan lagi ke bagian gudang dan surat pemesanan barang yang telah disahkan tersebut satu dibuat arsip untuk bagian gudang dan satunya lagi diberikan kepada supplier untuk dijadikan arsip.



Gambar 3.1 *Flowmap* Pemesanan

Keterangan :

A1 : Arsip daftar permintaan barang

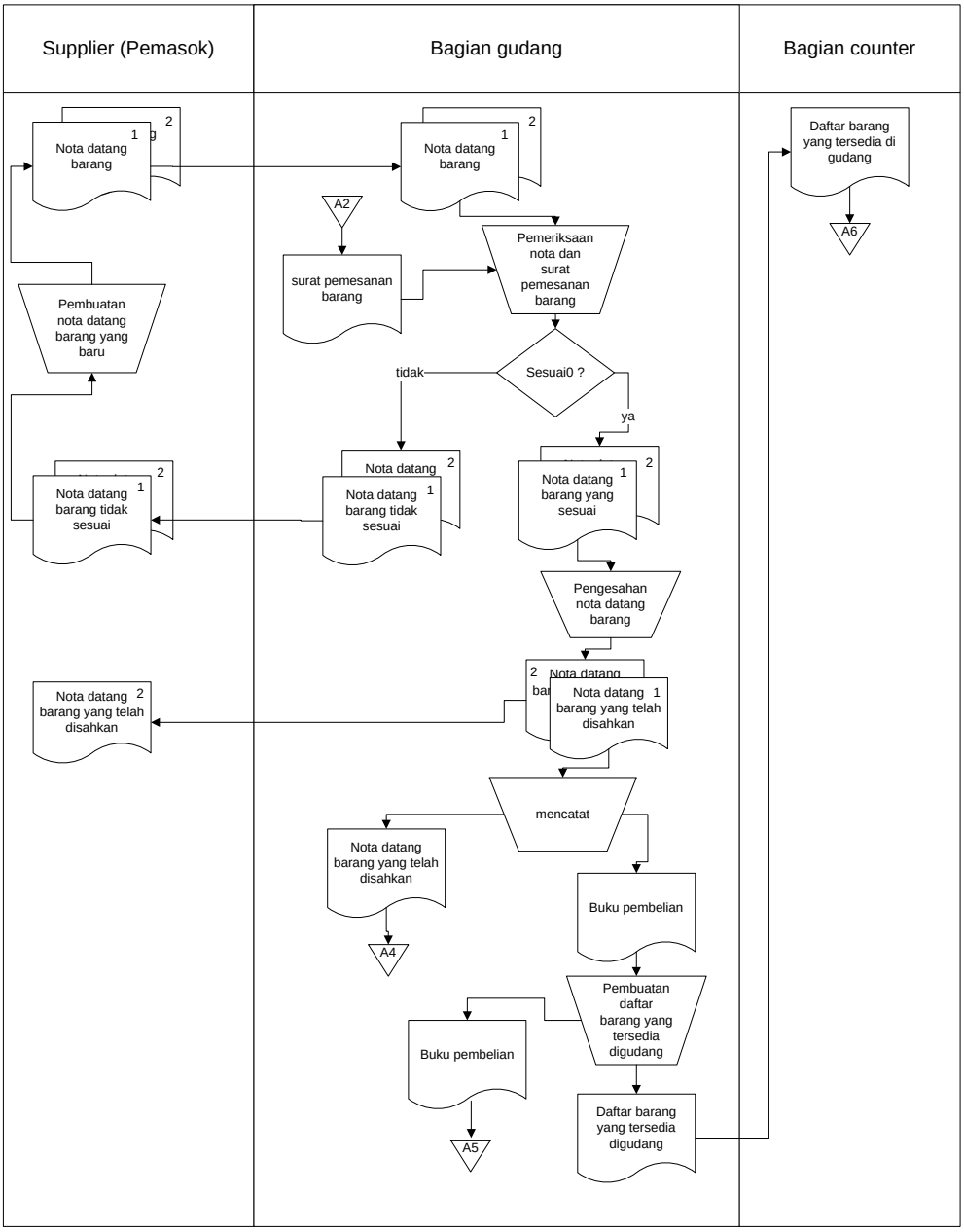
A2 : Arsip surat pemesanan barang yang telah disahkan di bagian gudang

A3 : Arsip surat pemesanan barang yang telah disahkan di bagian supplier.

3.1.1.2 Prosedur Pemerimaan

Prosedur ini berisi mengenai proses prosedur barang penerimaan. Adapun prosedurnya adalah sebagai berikut :

1. supplier meberikan nota deatang barang ke bagian gudang sebanyak dua rangkap
2. bagian gudang menerima nata datang barang dan memerikdsa nota dan pemesanan barang tersebut
3. sesudah diperiksa jika tidak sesuai nota datang barang tersebut diberikan lagi ke supplier untuk membuat nota datang barang yang baru yang sesuai dengan pemesanan dan jika sudah sesuai kemudian nota datang barang yang baru tersebut diberikan lagi ke bagian gudang dengan nota datang yang sudah sesuai dengan pesanan
4. jika diperiksa oleh bagian gudang nota datang barang sudah sesuai maka oleh bagian gudang disahkan dan dua rangkap nota barang tersebut yang satu diberikan kebagian supplier dan yang satu lagi akan dilakukan pengupdatean di bagian gudang
5. setelah itu nota datang barng tersebut di up date oleh bagian gudang yang satu untuk dijadikan arsip di bagian gudang dan yang satu lagi di buat daftar barng yang ter sedia di gudang untuk diberikan ke counter untuk dijadikan arsip.oleh bagian counter .



Gambar 3.2 *Flowmap* Penerimaan

Keterangan :

A2 : Arsip surat pemesanan barang

A4 : Arsip nota datang barang yang telah disahkan di bagian gudang

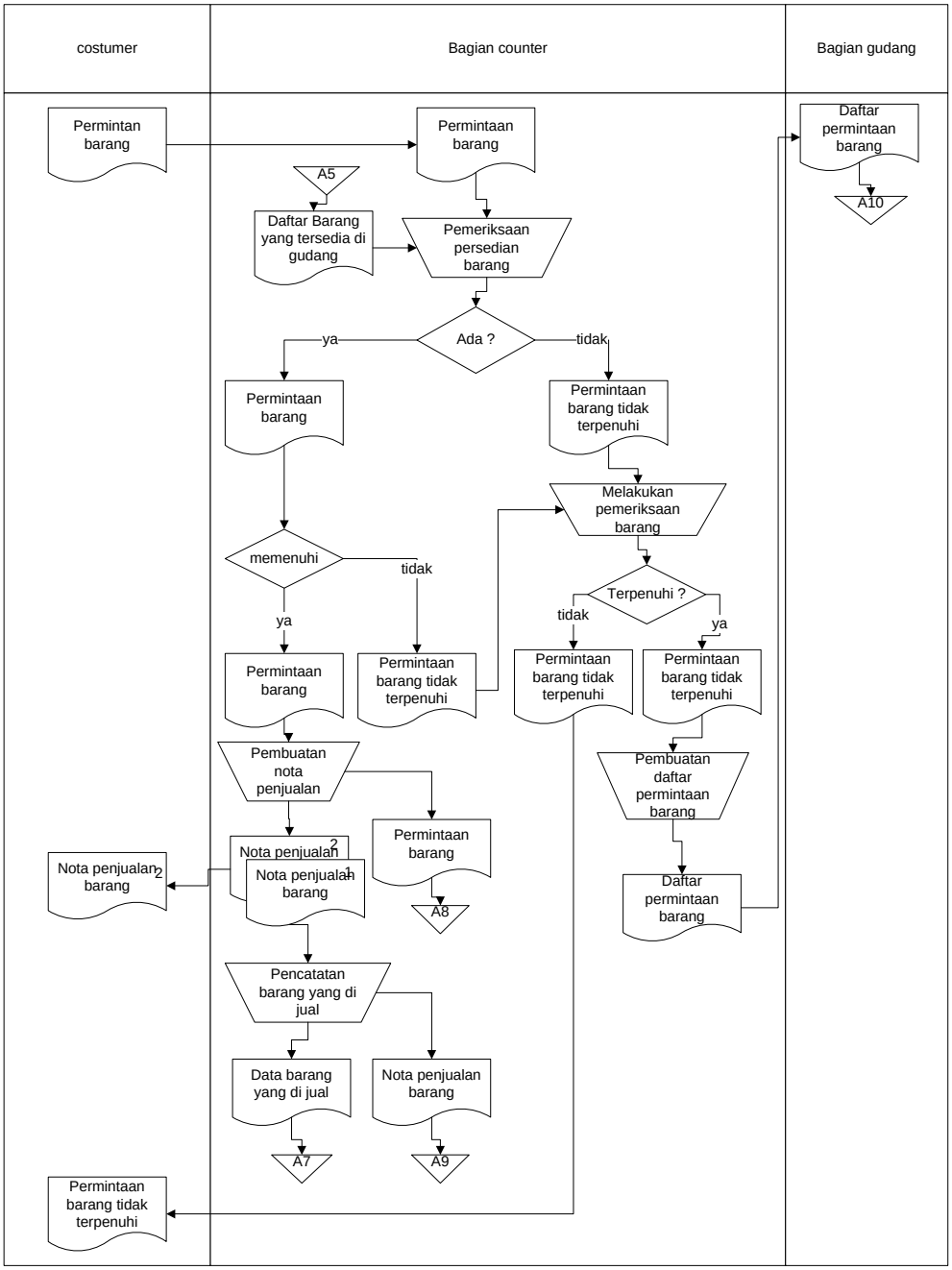
A5 : Arsip buku pembelian

A6 : Arsip daftar datang barang yang tersedia di gudang.

3.1.1.3 Prosedur Penjualan

Prosedur ini berisi mengenai proses prosedur barang penjualan. Adapun prosedurnya adalah sebagai berikut :

1. konsumen melakukan permintaan barang kebagian counter
2. kemudian bagian counter memeriksa persediaan barang yang tersedia digudang dengan mengecek atau melihat daftar barang yang tersedia digudang
3. setelah bagian counter memeriksa jika ada periksa jumlah barang jika ya memenuhi bagian counter membuat nota penjualan barang rangkap dua yang satu diberikan ke konsumen dan yang satu lagi sebelum diarsipkan di catat barang apa saja yang dijual kemudian permintaan dijadikan arsip dan data barang yang dijual dan nota penjualan barang dijadikan arsip oleh bagian counter
4. kemudian jika diperiksa jumlah barang tidak memenuhi dan konsumen tidak jadi membeli maka permintaan barang tersebut diberikan lagi ke konsumen
5. dan jika barang tidak memenuhi dan knsumen melakukn pemesanan bagian counter membuat permintaan barang dan diberikan kebagian gudang untuk diperiksa dan diberikan ke supplier.



Gambar 3.3 *Flowmap* Penjualan

Keterangan :

A5 : Arsip daftar barang yang tersedia digudang

A7 : Arsip data barang yang dijual

A8 : Arsip permintaan barang

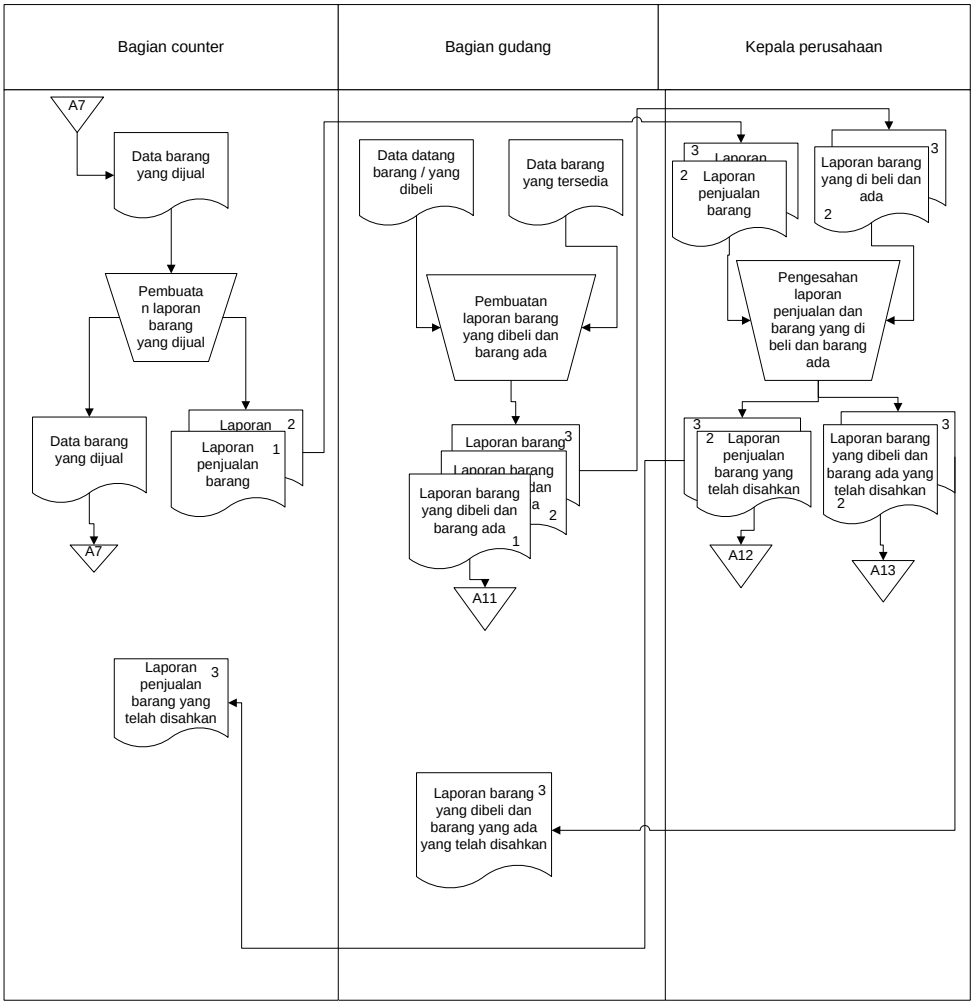
A9 : Arsip nota penjualan barang

A10 : Arsip daftar permintaan barang

3.1.1.4 Prosedur Laporan

Prosedur ini berisi mengenai proses prosedur laporan barang. Adapun prosedurnya adalah sebagai berikut :

1. bagian counter memeriksa data barang yang dijual
2. kemudian membuat laporan yang dijual sebanyak dua rangkap untuk diberikan ke kepala perusahaan untuk disahkan oleh kepala perusahaan setelah laporan barang yang dibeli barang yang ada disahkan yang satu di buat arsip oleh kepala perusahaan dan satu lagi diberikan ke bagian gudang
3. bagian counter membuat laporan yang dibeli dan barang yang ada sebanyak tiga rangkap yang satu dibuat arsip untuk bagian gudang dan yang dua diberikan ke bagian kepala perusahaan untuk disahkan setelah disahkan yang satu dibuat arsip oleh kepala perusahaan dan yang satu lagi diberikan kebagian counter.



Gambar 3.4 *Flowmap* laporan

Keterangan :

A7 : Arsip data barang yang dijual

A11 : Arsip laporan barang yang dibeli dan barang ada

A12 : Arsip laporan penjualan barang yang telah disahkan

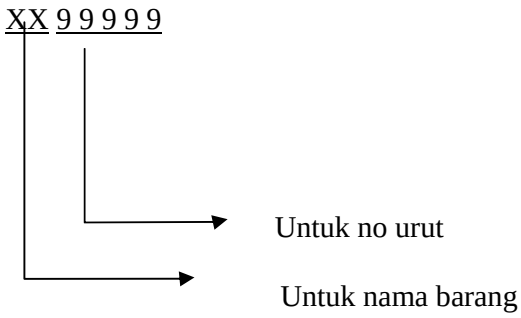
A13 : Arsip barang yang dibeli dan barang ada yang telah disahkan.

3.1.2 Analisis Kebutuhan Non-Fungsional

3.1.2.1 Analisis Pengkodean

Analisis pengkodean ini digunakan untuk kode yang digunakan dalam mengklasifikasikan data yang terlibat dalam Sistem Informasi Penjualan dan pembelian Di PD Pembangunan Raya. Setelah dilakukan analisis maka pengkodean di PD.Pembangunan Raya tersebut.

1. Kode barang



Contoh : KB 9999, berarti Kran Bak dengan nomber urut 9999

3.1.2.2 Analisis User

Bagian yang terlibat disistem sedang berjalan, yaitu Bagian gudang, analisis user tersebut adalah:

1. Nama : Jhono Kartasasmita

   Jabatan : Karyawan Bag. gudang

   Pendidikan : Strata I. Sarjana Ekonomi

   Kemampuan Komputer : Menggunakan Word, Excel dan program pengolahan data penjualan dan pembelian yang telah ada. Sistem Operasi yang dikuasai berbasis Windows.

Bagian yang terlibat disistem sedang berjalan, yaitu Bagian counter, analisis user tersebut adalah:

2. Nama : Epon

Jabatan : Karyawan Bag. counter

Pendidikan : Strata I. Sarjana Ekonomi

Kemampuan Komputer : Menggunakan Word, Excel dan program pengolahan data penjualan dan pembelian yang telah ada. Sistem Operasi yang dikuasai berbasis Windows.

Berdasarkan hasil analisis tersebut dengan latar belakang kemampuan menggunakan komputer yang terbiasa menggunakan office, maka disimpulkan bahwa *user* dapat menggunakan Sistem Informasi Penjualan dan pembelian.

3.1.2.3 Analisis Hardware

Analisis yang telah dilakukan dalam kebutuhan hardware untuk pengembangan dan kebutuhan minimum yang disarankan agar aplikasi ini dapat berjalan dengan baik, adalah:

1. Hardware yang ada sekarang di PD.Pembangunan Raya sebanyak dua PC, yaitu :
    a. Processor 2,66 Mhz.
    b. Hardisk 80 Gb.
    c. Ram 512 Mb.
    d. VGA 32 Mb.
    e. Printer

2. Kebutuhan minimum yang disarankan untuk program ini:
    a. Processor 933Mhz.
    b. Hardisk 40 Gb.
    c. Ram 512 Mb.
    d. CD – RW.
    e. VGA 32 Mb.
    f. Printer

Jadi setelah dilakukan analisis terhadap hardware, saat ini mencukupi untuk menjalankan aplikasi ini nantinya. Tetapi akan lebih baik apabila ditambahkan CD – RW sebagai sarana *back up data* yang nantinya akan dipindahkan dalam bentuk CD.

3.1.2.4 Analisis Software

Analisis yang telah dilakukan dalam kebutuhan Software untuk pengembangan dan menjalankan yang disarankan agar aplikasi ini dapat berjalan dengan baik, adalah:

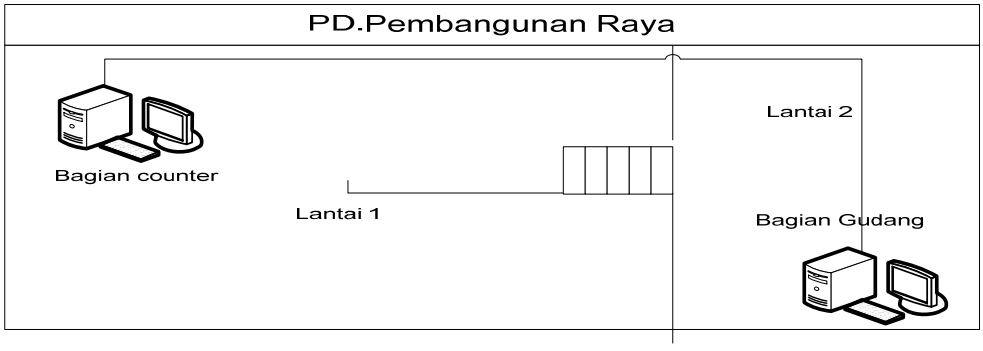
1. Software yang ada saat ini adalah:
    a. Windows XP SP I
    b. Microsoft Office
2. Software yang dibutuhkan untuk menjalankan aplikasi ini adalah:
    a. Windows XP SP I
    b. DBMS menggunakan Microsoft Access
    c. Borland Delphi versi 6.0

Jadi setelah dilakukan analisis terhadap software, saat ini masih dirasa kurang untuk menjalankan aplikasi ini nantinya. Software yang harus ditambahkan adalah Microsoft Access yang nantinya akan digunakan sebagai DBMS serta Borland Delphi versi 6.

3.1.2.5 Analisis Jaringan

Komputer yang digunakan saat ini Di PD. Pembangunan Raya bersifat *Stand Alone* jadi tidak menggunakan jaringan. Aplikasi sistem informasi yang akan dibangun bersifat *client-server* sehingga memerlukan adanya jaringan antar komputer yang akan menggunakan aplikasi ini.

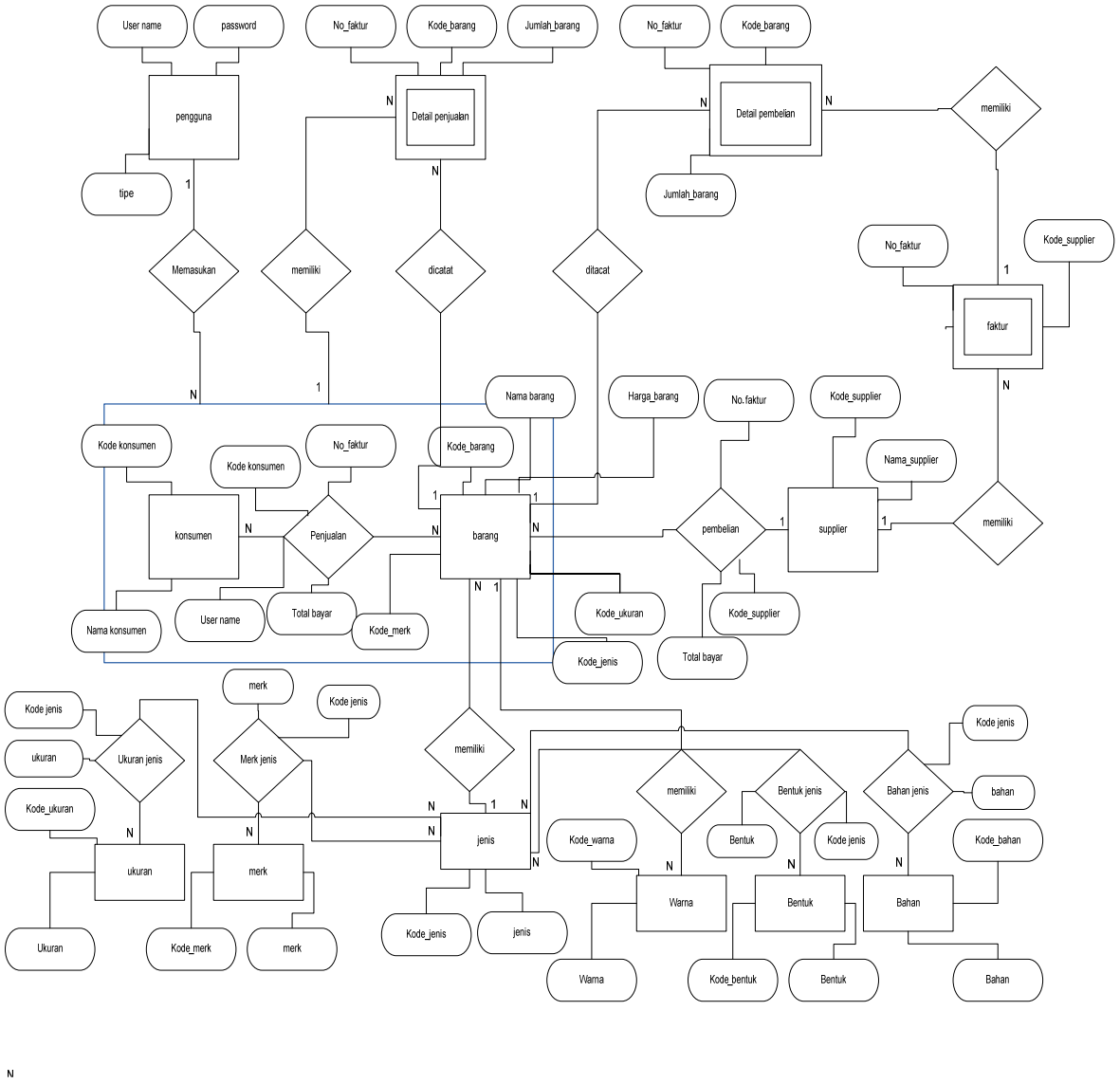
Komputer yang akan digunakan dalam aplikasi ini berada disatu bangunan, oleh karena itu jenis jaringan yang akan digunakan adalah LAN (Local Area Network). Komputer yang akan terlibat dalam jaringan adalah komputer dibagian *EDP* dan komputer dibagian kasir. Komputer dibagian *EDP* dalam aplikasi ini akan berfungsi sebagai *Server* dan komputer dibagian kasir menjadi *Client*. Media transmisi yang akan digunakan adalah kabel UTP, masing-masing komputer harus memiliki LAN Card dan berikut gambar 3.5 stuktur jaringan yang akan dibangun di PD. Pembangunan Raya berbasis *client-server*.



Gambar 3.5 Srtuktur jaringan

3.1.2.6 Analisis Basis Data

Struktur logika dari suatu data base dapat diekpresikan dengan menggunakan ERD. Diagram relasi entitas ini dibutuhkan untuk memodelkan file basis data Penjualan dan pembelian Di PD. Pembangunan Raya Adapun ERD digambarkan sebagai berikut:



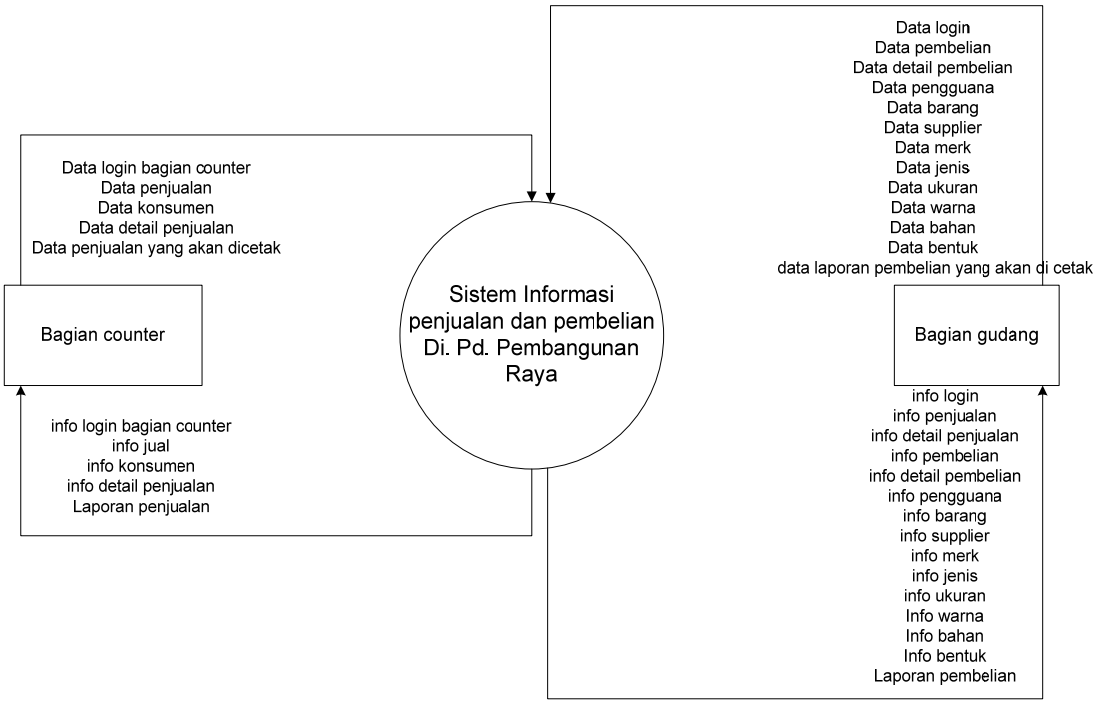
Gambar 3.6 *Entity Relationship Diagram* (ERD)

3.1.3 Analisis Kebutuhan Fungsional

3.1.3.1 Diagram Konteks

Diagram Konteks adalah diagram yang menggambarkan hubungan antar *Entitas Eksternal* dengan sistem yang akan dibangun. Dimana data yang dimasukan oleh bagian komponen eksternal akan diproses di dalam sistem dan akan menghasilkan laporan yang diinginkan oleh komponen eksternal tersebut sesuai dengan data yang dimasukan.

Diagram konteks dari aplikasi pengolahan penjualan dan pembelian yang akan dibangun dapat dilihat dibawah ini.



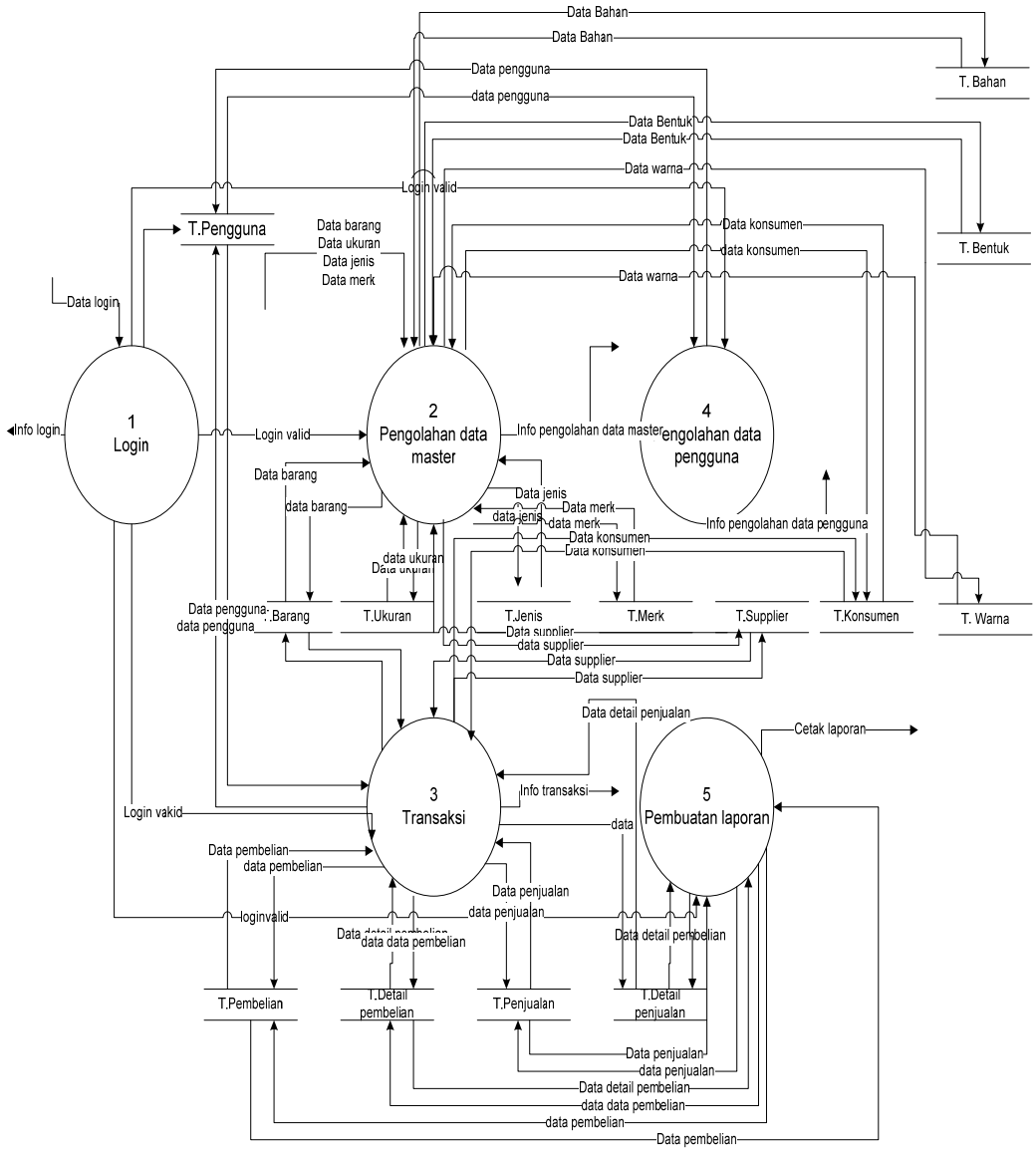
Gambar 3.7 *Diagram Konteks*

3.1.3.2 DFD

Data Flow Diagram (DFD) dibuat untuk menggambarkan arus data yang terstruktur dan jelas dari mulai pengisian data sampai dengan keluarannya. Arus data pada *Data Flow Diagram (DFD)* ini berupa masukan untuk sistem oleh *entitas eksternal* atau keluaran dari sistem. Adapun gambar *Data Flow Diagram (DFD)* ini dapat dilihat pada gambar dibawah ini.

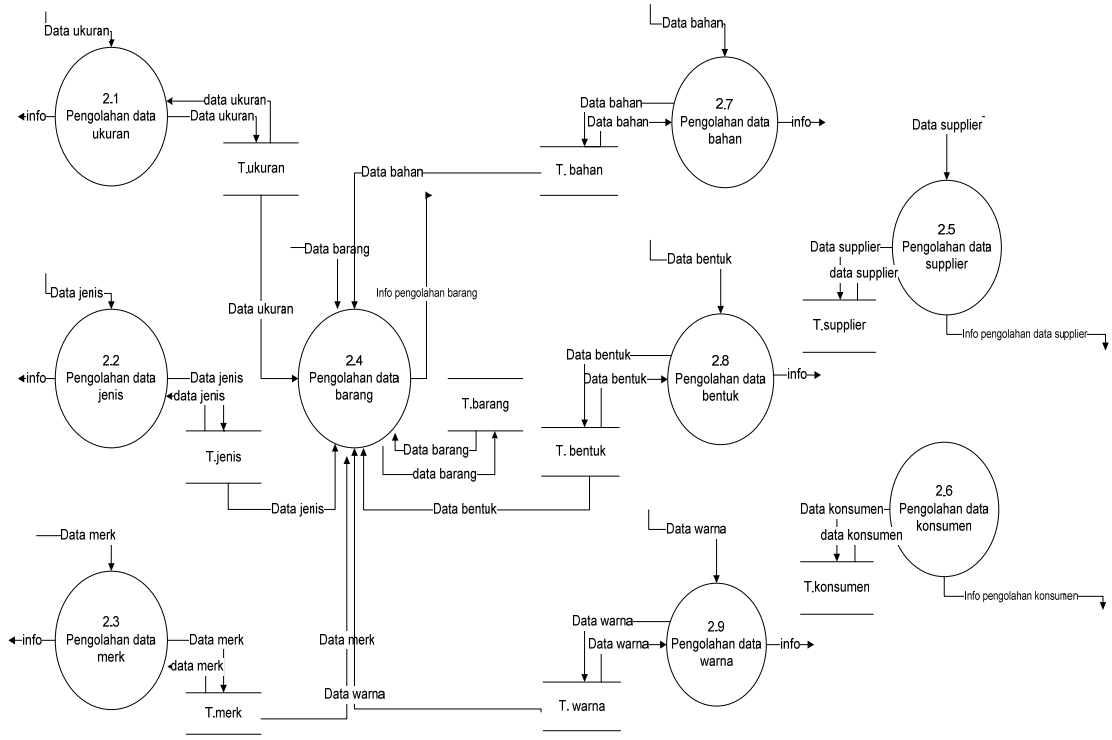
DFD level 1

Diagram alir data merupakan sebuah representasi dari suatu sistem yang menggambarkan bagian-bagian dari sistem tersebut beserta keterkaitan antara bagian-bagian yang ada. Dari diagram alir data ini seseorang bisa mengetahui sumber dari informasi di dalam sistem maupun tujuan dari masukan yang berasal dari entitas eksternal. Adapun diagram alir data level satu dari sistem informasi penjualan dan pembelianadalah sebagai berikut:



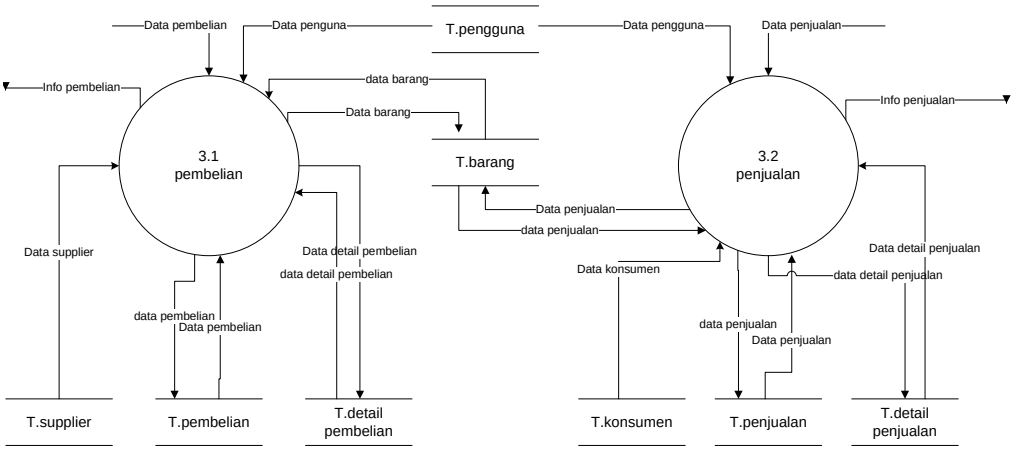
Gambar 3.8 DFD Level 1 (Sistem Informasi penjualan dan pembelian)

DFD level 2 proses 1 (pengolahan data master)



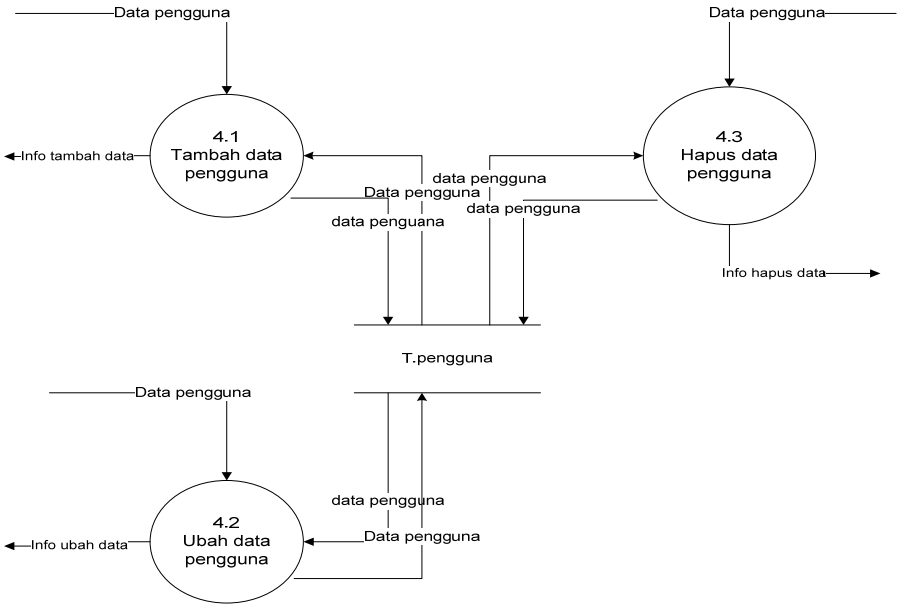
Gambar 3.9 DFD level 2 proses 1 (pengolahan data master)

DFD level 2 proses 2 (transaksi)



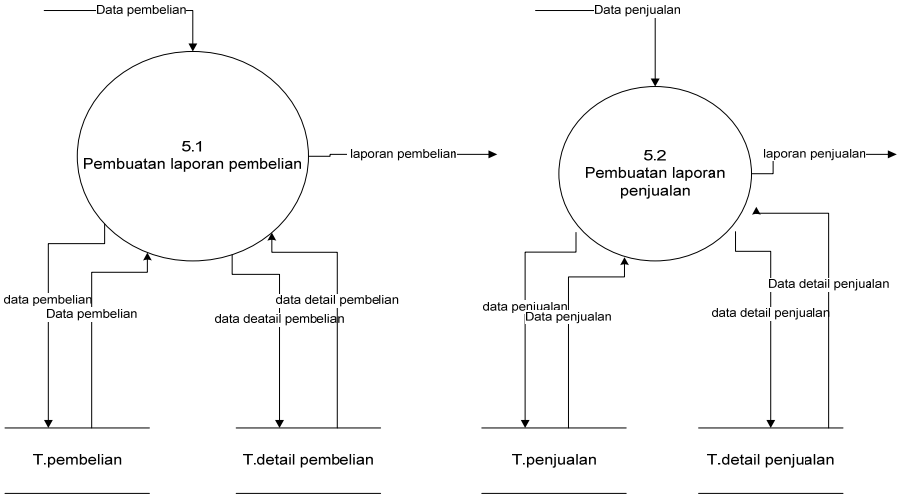
Gambar 3.10 DFD level 2 proses 2 (transaksi)

DFD level 2 proses 3 (pengolahan data pengguna)



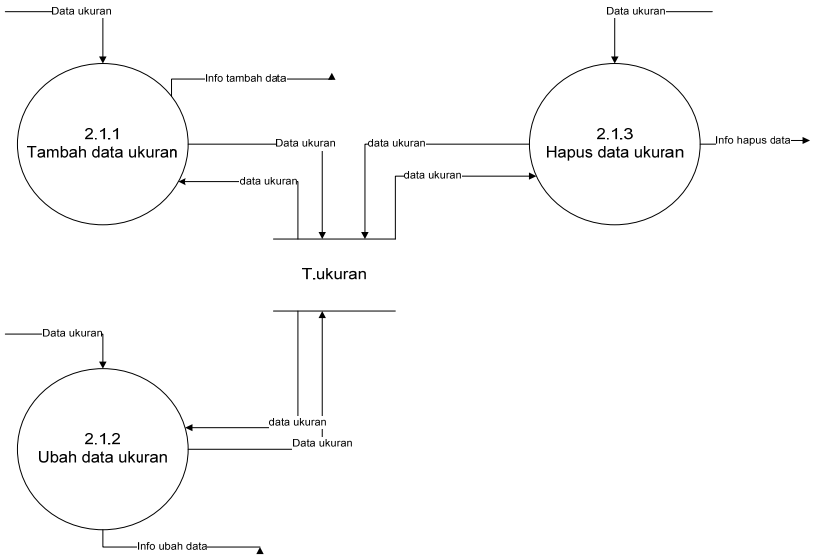
Gambar 3.11 DFD level 2 proses 3 (pengolahan data pengguna)

DFD level 2 proses 4 (pembuatan laporan)



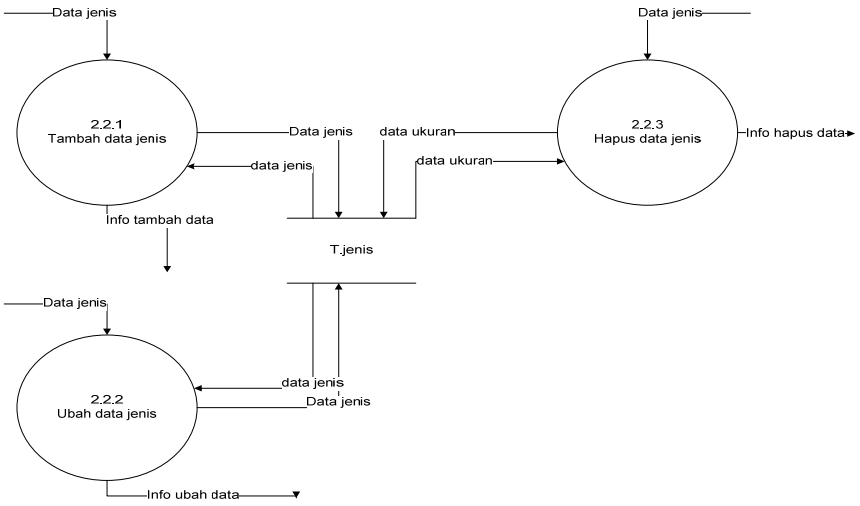
Gambar 3.12 DFD level 2 proses 4 (pembuatan laporan)

DFD level 3 proses 2.1 (pengolahan data ukuran)



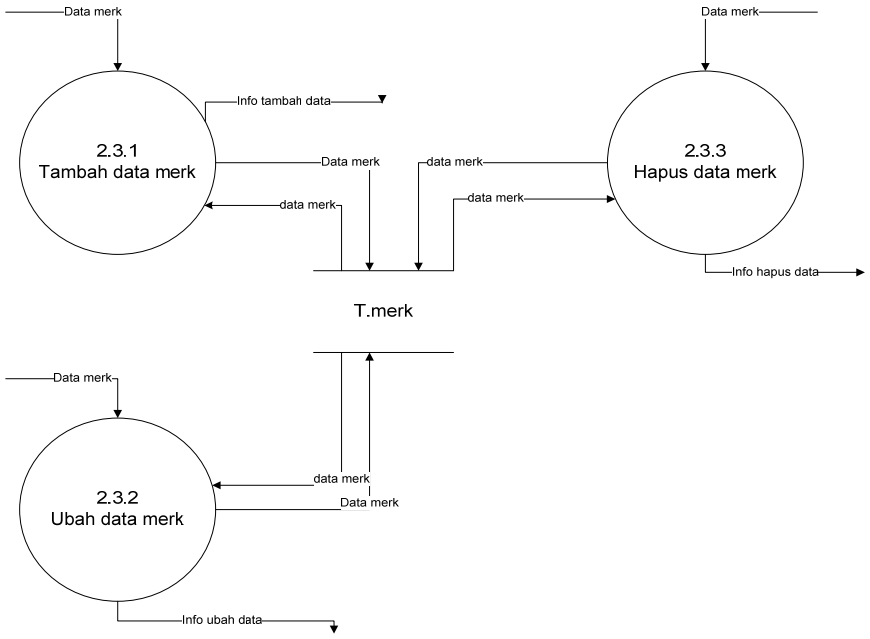
Gambar 3.13 DFD level 3 proses 2.1 (pengolahan data ukuran)

DFD level 3 proses 2.2 (pengolahan data jenis)



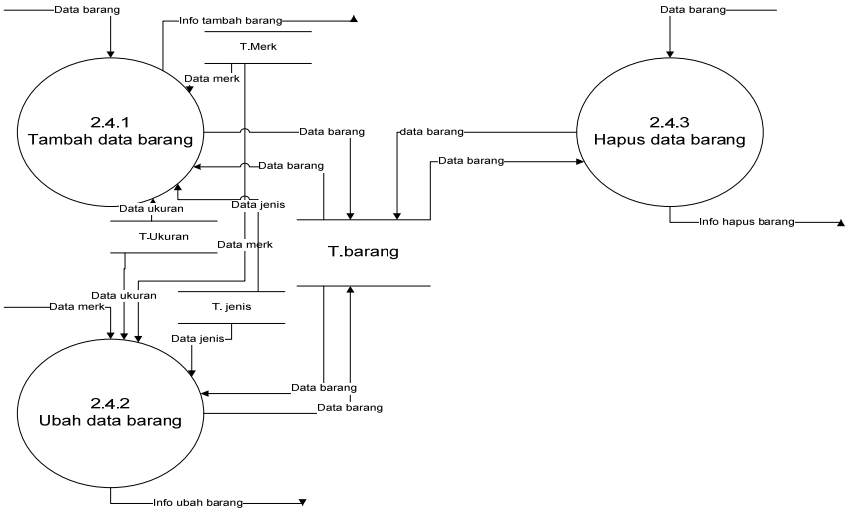
Gambar 3.14 DFD level 3 proses 2.2 (pengolahan data jenis)

DFD level 3 proses 2.3 (pengolahan data merk)



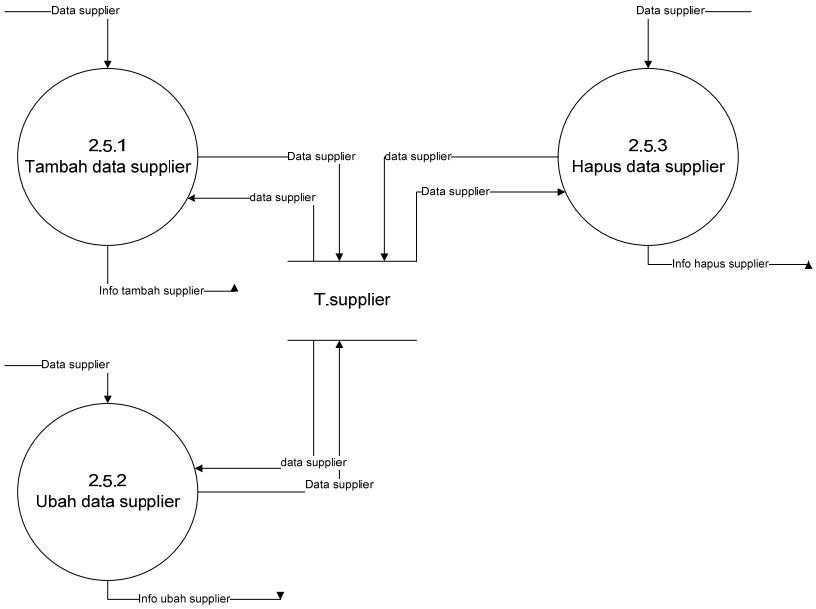
Gambar 3.15 DFD level 3 proses 2.3 ((pengolahan data merk)

DFD proses 3 proses 2.4 (pengolahan data barang)



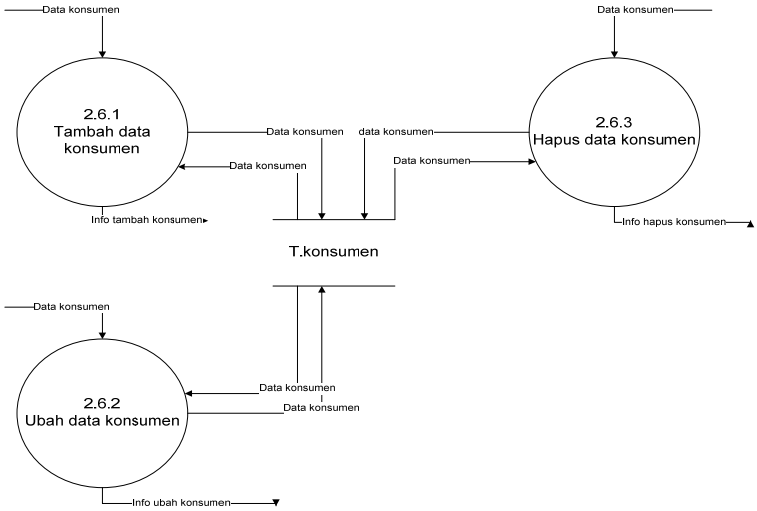
Gambar 3.16 DFD proses 3 proses 2.4 (pengolahan data barang)

DFD proses 3 proses 2.5 (pengolahan data supplier)



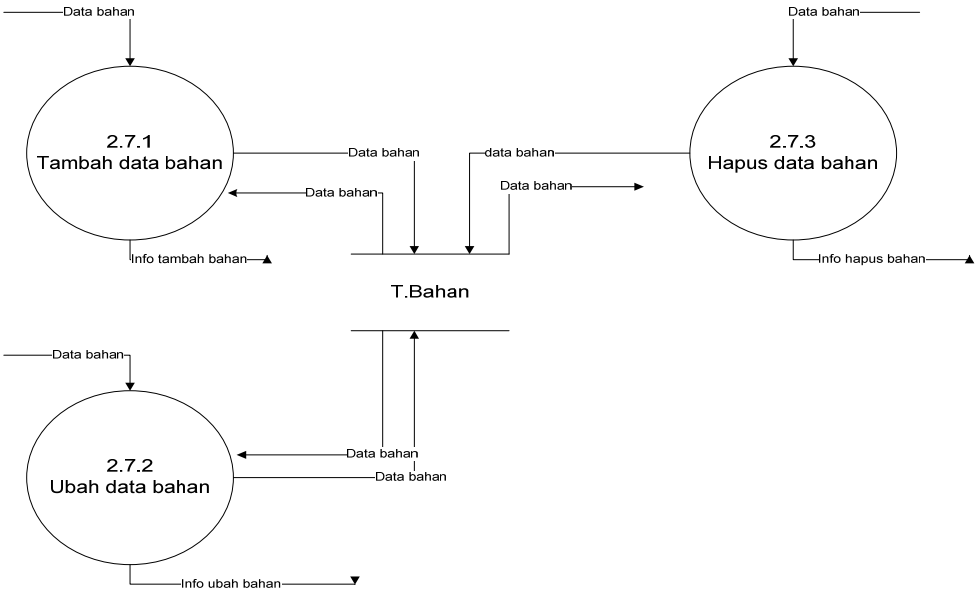
Gambar 3.17 DFD proses 3 proses 2.5 (pengolahan data supplier)

DFD proses 3 proses 2.6 (pengolahan data konsumen)



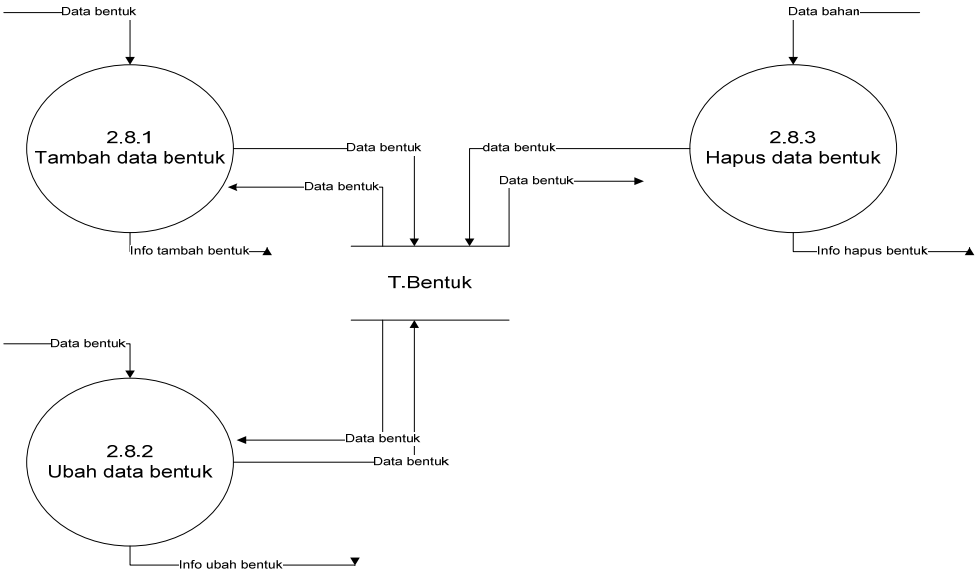
Gambar 3.18 DFD proses 3 proses 2.6 (pengolahan data konsumen)

DFD proses 3 proses 2.7 (pengolahan data bahan)



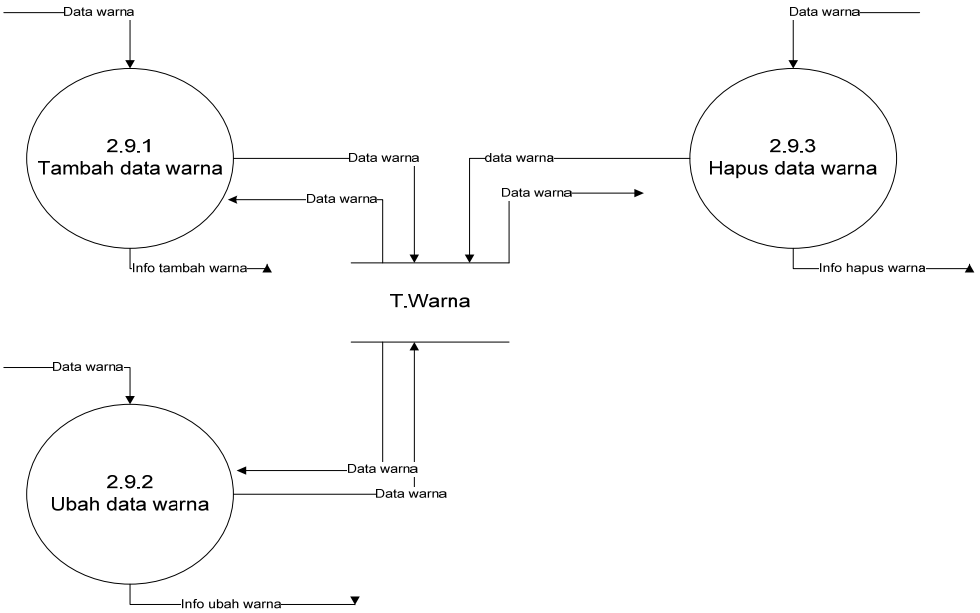
Gambar 3.19 DFD proses 3 proses 2.7 (pengolahan data bahan)

DFD proses 3 proses 2.8 (pengolahan data bentuk)



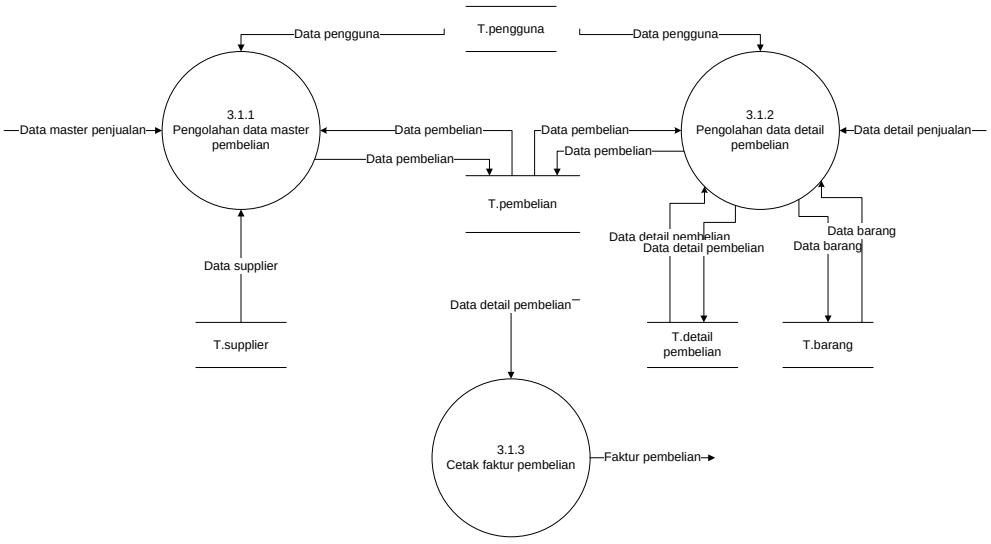
Gambar 3.20 DFD proses 3 proses 2.8 (pengolahan data bentuk)

DFD proses 3 proses 2.9 (pengolahan data warna)



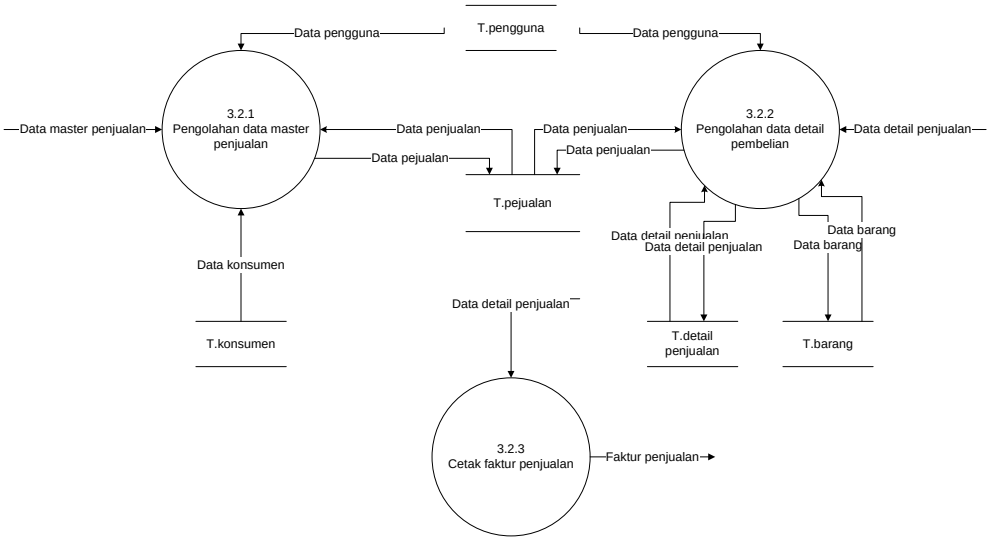
Gambar 3.21 DFD proses 3 proses 2.9 (pengolahan data warna)

DFD proses 3 proses 3.1 (pembelian)



Gamabar 3.22 DFD proses 3 proses 3.1 (pembelian)

DFD proses 3 proses 3.2 (penjualan)



Gambar 3.23 DFD proses 3 proses 3.2 (penjualan)

3.1.4 Spesifikasi Proses

Spesifikasi proses adalah penjelasan dari aliran data yang mengalir dalam Data *flow* diagram. Spesifikasi proses dari DFD yang dibangun dapat dilihat pada tabel 3.1

Tabel 3.1 Spesifikasi proses Sistem Informasi Penjualan dan pembelian Di.PD Pembangunan Raya Berbasis *Client Server*

No.	Proses	Keterangan
1	No. Proses	1
	Nama Proses	Login
	Source	Bagian gudang,bagian counter
	Input	Data login
	Output	Info login
	Destination	T.pengguna

	Logika Proses	Begin
		{
		pengguna memasukan data *login sesuai hak akses*
		sistem memanggil data pengguna sesuai data login
		If nama pengguna and password ada
		Then
		Tampilkan antar muka sesuai hak akses

No.	Proses	Keterangan

		Else
		Muncul antar muka pesan dan data user dikosongkan
		}
		End

2	No. Proses	2.1.1
	Nama Proses	Tambah data ukuran
	Source	Bagian gudang
	Input	Data data ukuran
	Output	Info ukuran
	Destination	*T.ukuran*

	Logika Proses	Begin
		{
		Masukkan data ukuran yang akan ditambah ke tabel ukuran
		}
		End

3	No. Proses	2.1.2
	Nama Proses	Ubah data ukuran
	Source	*Bagian gudang*
	Input	Data ukuran
	Output	Info ukuran
	Destination	*T.ukuran*

	Logika Proses	Begin
		{
		Pengguna mengubah data ukuran yang ada dalam tabel ukuran
		Masukan data ukuran yang akan di ubah
		If Data ditemukan
		Then
		Ubah data ukuran
		Else
		Tampil pesan
		}
		End

4	No. Proses	2.1.3
	Nama Proses	Hapus data ukuran
	Source	*Bagian gudang*
	Input	Data ukuran
	Output	Info ukuran
	Destination	*T.ukuran*

	Logika Proses	Begin
		{
		Pengguna menghapus data ukuran yang ada dalam tabel ukuran
		Masukan data ukuran yang akan di hapus
		If Data ditemukan
		Then
		hapus data ukuran
		Else
		Tampil pesan
		}
		End

5	No. Proses	2.2.1
	Nama Proses	Tambah data jenis
	Source	Bagian gudang
	Input	Data data jenis

No.	Proses	Keterangan
	Output	Info jenis
	Destination	*T.jenis*

	Logika Proses	Begin
		{
		Masukkan data jenis yang akan ditambah ke tabel jenis
		}
		End

6	No. Proses	2.2.2
	Nama Proses	Ubah data jenis
	Source	Bagian gudang
	Input	Data jenis
	Output	Info jenis
	Destination	*T.jenis*

	Logika Proses	Begin
		{
		Pengguna mengubah data jenis yang ada dalam tabel jenis
		Masukan data jenis yang akan di ubah
		If Data ditemukan
		Then
		Ubah data jenis
		Else
		Tampil pesan
		}
		End

7	No. Proses	2.2.3
	Nama Proses	Hapus data jenis
	Source	Bagian gudang
	Input	Data jenis
	Output	Info jenis
	Destination	*T.jenis*

	Logika Proses	Begin
		{
		Pengguna menghapus data jenis yang ada dalam tabel jenis
		Masukan data jenis yang akan di hapus
		If Data ditemukan
		Then
		hapus data jenis
		Else
		Tampil pesan
		}
		End

8	No. Proses	2.3.1
	Nama Proses	Tambah data merk
	Source	Bagian gudang
	Input	Data data merk
	Output	Info merk
	Destination	*T.merk*

	Logika Proses	Begin
		{
		Masukkan data merk yang akan ditambah ke tabel merk
		}
		End

9	No. Proses	2.3.2

No.	Proses	Keterangan
	Nama Proses	Ubah data merk
	Source	Bagian gudang
	Input	Data merk
	Output	Info merk
	Destination	*T.merk*

	Logika Proses	Begin
		{
		Pengguna mengubah data merk yang ada dalam tabel merk
		Masukan data merk yang akan di ubah
		If Data ditemukan
		Then
		Ubah data merk
		Else
		Tampil pesan
		}
		End

10	No. Proses	2.3.3
	Nama Proses	Hapus data merk
	Source	Bagian gudang
	Input	Data merk
	Output	Info merk
	Destination	*T.merk*

	Logika Proses	Begin
		{
		Pengguna menghapus data merk yang ada dalam tabel merk
		Masukan data merk yang akan di hapus
		If Data ditemukan
		Then
		hapus data merk
		Else
		Tampil pesan
		}
		End

11	No. Proses	2.4.1
	Nama Proses	Tambah data barang
	Source	Bagian gudang
	Input	Data data barang
	Output	Info barang
	Destination	*T.barang*

	Logika Proses	Begin
		{
		Masukkan data barang yang akan ditambah ke tabel barang
		}
		End

12	No. Proses	2.4.2
	Nama Proses	Ubah data barang
	Source	Bagian gudang
	Input	Data barang
	Output	Info barang
	Destination	*T.barang*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data barang yang ada dalam tabel barang
		Masukan data barang yang akan di ubah
		If Data ditemukan
		Then
		Ubah data barang
		Else
		Tampil pesan
		}
		End

13	No. Proses	2.4.3
	Nama Proses	Hapus data barang
	Source	Bagian gudang
	Input	Data barang
	Output	Info barang
	Destination	*T.barang*

	Logika Proses	Begin
		{
		Pengguna menghapus data barang yang ada dalam tabel barang
		Masukan data barang yang akan di hapus
		If Data ditemukan
		Then
		hapus data barang
		Else
		Tampil pesan
		}
		End

14	No. Proses	2.5.1
	Nama Proses	Tambah data supplier
	Source	Bagian gudang
	Input	Data data supplier
	Output	Info supplier
	Destination	*T.supplier*

	Logika Proses	Begin
		{
		Masukkan data supplier yang akan ditambah ke tabel supplier
		}
		End

15	No. Proses	2.5.2
	Nama Proses	Ubah data supplier
	Source	Bagian gudang
	Input	Data supplier
	Output	Info supplier
	Destination	*T.supplier*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data supplier yang ada dalam tabel supplier
		Masukan data barang yang akan di ubah
		If Data ditemukan
		Then
		Ubah data supplier
		Else
		Tampil pesan
		}
		End

16	No. Proses	2.5.3
	Nama Proses	Hapus data supplier
	Source	Bagian gudang
	Input	Data supplier
	Output	Info supplier
	Destination	*T.supplier*

	Logika Proses	Begin
		{
		Pengguna menghapus data supplier yang ada dalam tabel supplier
		Masukan data barang yang akan di hapus
		If Data ditemukan
		Then
		hapus data supplier
		Else
		Tampil pesan
		}
		End

17	No. Proses	2.6.1
	Nama Proses	Tambah data konsumen
	Source	bagian counter
	Input	Data data konsumen
	Output	Info konsuen
	Destination	*T.merk*

	Logika Proses	Begin
		{
		Masukkan data konsumen yang akan ditambah ke tabel konsumen
		}
		End

18	No. Proses	2.6.2
	Nama Proses	Ubah data konsumen
	Source	bagian counter
	Input	Data konsumen
	Output	Info konsumen
	Destination	*T.konsumen*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data konsumen yang ada dalam tabel konsumen
		Masukan data konsumen yang akan di ubah
		If Data ditemukan
		Then
		Ubah data konsumen
		Else
		Tampil pesan
		}
		End

19	No. Proses	2.6.3
	Nama Proses	Hapus data konsumen
	Source	bagian counter
	Input	Data konsumen
	Output	Info konsumen
	Destination	*T.konsumen*

	Logika Proses	Begin
		{
		Pengguna menghapus data konsumen yang ada dalam tabel konsumen
		Masukan data konsumen yang akan di hapus
		If Data ditemukan
		Then
		hapus data konsumen
		Else
		Tampil pesan
		}
		End

20	No. Proses	2.7.1
	Nama Proses	Tambah data bahan
	Source	Bagian gudang
	Input	Data bahan
	Output	Info bahan
	Destination	*T.bahan*

	Logika Proses	Begin
		{
		Masukkan data bahan yang akan ditambah ke tabel bahan
		}
		
		End

21	No. Proses	2.7.2
	Nama Proses	Ubah data bahan
	Source	Bagian gudang
	Input	Data bahan
	Output	Info bahan
	Destination	*T.bahan*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data bahan yang ada dalam tabel bahan
		Masukan data bahan yang akan di ubah
		If Data ditemukan
		Then
		Ubah data bahan
		Else
		Tampil pesan
		}
		
		End

22	No. Proses	2.7.3
	Nama Proses	Hapus data bahan
	Source	Bagian gudang
	Input	Data bahan
	Output	Info bahan
	Destination	*T.bahan*

	Logika Proses	Begin
		{
		Pengguna menghapus data bahan yang ada dalam tabel bahan
		Masukan data bahan yang akan di hapus
		If Data ditemukan
		Then
		hapus data bahan
		Else
		Tampil pesan
		}
		End

23	No. Proses	2.8.1
	Nama Proses	Tambah data bentuk
	Source	Bagian gudang
	Input	Data bentuk
	Output	Info bentuk
	Destination	*T.bentuk*

	Logika Proses	Begin
		{
		Masukkan data bentuk yang akan ditambah ke tabel bahan
		}
		
		End

24	No. Proses	2.8.2
	Nama Proses	Ubah data bentuk
	Source	Bagian gudang
	Input	Data bentuk
	Output	Info bentuk
	Destination	*T.bentuk*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data bentuk yang ada dalam tabel bentuk
		Masukan data bentuk yang akan di ubah
		If Data ditemukan
		Then
		Ubah data bentuk
		Else
		Tampil pesan
		}
		
		End

25	No. Proses	2.8.3
	Nama Proses	Hapus data bentuk
	Source	Bagian gudang
	Input	Data bentuk
	Output	Info bentuk
	Destination	*T.bentuk*

	Logika Proses	Begin
		{
		Pengguna menghapus data bentuk yang ada dalam tabel bentuk
		Masukan data bentuk yang akan di hapus
		If Data ditemukan
		Then
		hapus data bentuk
		Else
		Tampil pesan
		}
		End

26	No. Proses	2.9.1
	Nama Proses	Tambah data warna
	Source	Bagian gudang
	Input	Data warna
	Output	Info warna
	Destination	*T.warna*

	Logika Proses	Begin
		{
		Masukkan data warna yang akan ditambah ke tabel warna
		}
		
		End

27	No. Proses	2.9.2
	Nama Proses	Ubah data warna
	Source	Bagian gudang
	Input	Data warna
	Output	Info warna
	Destination	*T.warna*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna mengubah data warna yang ada dalam tabel warna
		Masukan data warna yang akan di ubah
		If Data ditemukan
		Then
		Ubah data warna
		Else
		Tampil pesan
		}
		
		End

28	No. Proses	2.9.3
	Nama Proses	Hapus data warna
	Source	Bagian gudang
	Input	Data warna
	Output	Info warna
	Destination	*T.warna*

	Logika Proses	Begin
		{
		Pengguna menghapus data warna yang ada dalam tabel warna
		Masukan data warna yang akan di hapus
		If Data ditemukan
		Then
		hapus data warna
		Else
		Tampil pesan
		}
		End

29	No. Proses	3.1.1
	Nama Proses	Pengolahan data master pembelian
	Source	*Bagian gudang, bagian counter*
	Input	Data master pembelian, data supplier, data pengguna, data pembelian
	Output	Info pembelian
	Destination	*T.pengguna,T. Pembelian*

	Logika Proses	Begin
		{
		Masukkan data pengolahan data master pembelian yang akan dicari atau disaring. Bila data yang disaring atau dicari ada maka akan tampil
		}
		End

30	No. Proses	3.1.2
	Nama Proses	Pengolahan data detail pembelian
	Source	*Bagian gudang, bagian counter*
	Input	Data barang , data detail pembelian, data pengguna
	Output	Info barang info detail pembelian
	Destination	*T.pengguna T.detai pembelian, T.barang*

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Masukkan data detail pembelian yang akan dicari atau disaring. Bila data yang disaring atau dicari ada maka akan tampil
		}
		End

31	No. Proses	3.1.3
	Nama Proses	Cetak faktur
	Source	Pembelian
	Input	Data detail pembelian
	Output	Info faktur pembelian
	Destination	*Bagian gudang, bagian counter*

	Logika Proses	Begin
		{
		pengguna memasukan data *cetak faktur*
		sistem memanggil data yang akan dicetak
		If laporan ada
		Then
		Lakukan pencetakan
		Else
		Tidak cetak faktur
		}
		End

32	No. Proses	3.2.1
	Nama Proses	Pengolahan data Master penjualan
	Source	Bagian gudang, bagian counter
	Input	Data master penjualan,data konsumen,data pengguna
	Output	Info master penjualan data konsumen
	Destination	*T.penjualan,T.konsumen*

	Logika Proses	Begin
		{
		Masukkan pengolahan data master penjualan langsung*r* yang akan dicari atau disaring. Bila data yang disaring atau dicari ada maka akan tampil
		}
		End

33	No. Proses	3.2.2
	Nama Proses	Pengolahan data Detail penjualan
	Source	Bagian gudang, bagian counter
	Input	Data data detail penjualan, data barang,data pengguna
	Output	Info detail penjualan
	Destination	*T.barang, T.detail penjualan,T.penjualan*

	Logika Proses	Begin
		{
		Masukkan pengolahan data master penjualan langsung*r* yang akan dicari atau disaring. Bila data yang disaring atau dicari ada maka akan tampil
		}
		End

34	No. Proses	3.2.3
	Nama Proses	Cetak faktur penjualan
	Source	Bagian gudang, bagian counter
	Input	Data detail penjualan

No.	Proses	Keterangan
	Output	Faktur penjualan
	Destination	

	Logika Proses	Begin
		{
		Masukkan pengolahan data master pembelian langsung*r* yang akan dicari atau disaring. Bila data yang disaring atau dicari ada maka akan tampil
		}
		End

35	No. Proses	4.1
	Nama Proses	Tambah data pengguna
	Source	Bagian gudang, bagian counter
	Input	Data pengguna
	Output	info pengguna
	Destination	*T.pengguna*

	Logika Proses	Begin
		{
		Masukkan data pengguna yang akan ditambah ke tabel pengguna
		}
		End

36	No. Proses	4.2
	Nama Proses	Ubah data pengguna
	Source	Bagian gudang, bagian counter
	Input	Data pengguna
	Output	Info pengguna
	Destination	*T.pengguna*

	Logika Proses	Begin
		{
		Pengguna mengubah data guru yang ada dalam tabel pengguna
		Masukan data pengguna yang akan di ubah
		If Data ditemukan
		Then
		Ubah data pengguna
		Else
		Tampil pesan
		}
		End

37	No. Proses	4.3
	Nama Proses	Hapus data pengguna
	Source	Bagian gudang, bagian counter
	Input	Data pengguna
	Output	Info lpengguna
	Destination	T.pengguna

No.	Proses	Keterangan

	Logika Proses	Begin
		{
		Pengguna menghapus data guru yang ada dalam tabel pengguna
		Masukan data pengguna yang akan di hapus
		If Data ditemukan
		Then
		hapus data pengguna
		Else
		Tampil pesan
		}
		End

38	No. Proses	5.1
	Nama Proses	Pembuatan laporan pembelian
	Source	Bagian gudang, bagian counter
	Input	Data pembelian, data pembelian
	Output	Info pembelian info detail pembelian
	Destination	*T.pembelian, T.detail pembelian*

	Logika Proses	Begin
		{
		pengguna memasukan data *laporan*
		sistem memanggil data pengguna sesuai nama
		If laporan ada
		Then
		Lakukan pencetakan laporan pembelian
		Else
		Tidak ada laporan tidak ada cetak laporan
		}
		End

39	No. Proses	5.2
	Nama Proses	Pembuatan laporan penjualan
	Source	Bagian gudang, bagian counter
	Input	Data penjualan, data detail penjualan
	Output	Info penjualan, info detail penjualan
	Destination	Tpenjual, T.detail pembelian

	Logika Proses	Begin
		{
		pengguna memasukan data *laporan penjualan*
		sistem memanggil data pengguna sesuai laporan
		If laporan ada
		Then
		Lakukan pencetakan laporan penjualan
		Else
		Tidak ada laporan tidak ada cetak laporan
		}
		End

3.1.5 Kamus data

Kamus data digunakan untuk merancang input, output, laporan dan *database* yang berfungsi untuk menjelaskan semua data yang digunakan didalam aplikasi. Kamus data untuk sistem yang akan dibangun dapat dilihat pada tabel 3.2.

1	Nama	Data *login*
	Where used / how used	1. proses 1 login (masukan) 2. proses 4.1 tambah data user (masukan) 3. proses 4.2 ubah data user (masukan) 4. proses 4.3 ubah status data user (masukan)
	Deskripsi	Berisi data *user*
	Struktur Data	User name + *password+ jenis pengguna*
	User name *Password* *Jenis pengguna*	[A..Z] [a..z | A..Z | 0..9] [A,,Z | a,,z]
2	Nama	Data barang
	Where used / how used	Proses 2.4.1 tambah barang (masukan) Proses 2.4.2 ubah barang (masukan) Proses 2.4.3 hapus barang (masukan)
	Deskripsi	Berisi data barang
	Struktur Data	Kode barang + nama barang + warna + harga + stok + bentuk + kode ukuran + kode jenis + kode merk
	Kode barang Nama barang Warna Harga Stok Bentuk Kode ukuran Kode jenis Kode merk	[0..9] [A..Z] [a..z | A..Z] [0..9] [a..z | A..Z | 0..9] [a..z | A..Z] [a..z | A..Z | 0..9] [a..z | A..Z | 0..9] [a..z | A..Z | 0..9]
3	Nama	Data pembelian
	Where used / how used	*1. proses 3.1.1 pengolahan data master pembelian (masukan)* 2. proses 3.1.2 pengolahan data pembelian (masukan) 3. proses 3.1.3 cetak faktur pembelian (keluaran)
	Deskripsi	Berisi data pembelian yang akan ditambah, diubah, dicari
	Struktur Data	No faktur + Tgl pembelian + kode *supplier* + total bayar + user name
	No faktur Tanggal pembelian Kode supplier Total bayar User name	[0..9] [0..9] [0..9] [0..9] [a..z | A..Z]
4	Nama	*Data Detail pembelian*

	Where used / how used	cari *detail pembelian* (masukan) tambah *detail pembelian* (masukan) ubah *detail pembelian* (masukan) *detail pembelian* (masukan)
	Deskripsi	Berisi data *detail pembelian* yang akan ditambah, diubah, dicari
	Struktur Data	No faktur + kode barang + jumlah + harga + total bayar + user name
	No faktur Kode barang Jumlah Harga Total bayar	[0..9] [0..9] [0..9] [0..9] [0..9]
5	Nama	Data penjualan
	Where used / how used	1. proses 3.2.1 pengolahan data master penjualan (masukan) 2. proses 3.2.2 pengolahan data detail penjualan (masukan) 3. proses 3.2.3 cetak faktur penjualan (masukan)
	Deskripsi	Berisi data distribusi gudang yang akan ditambah, diubah, dicari
	Struktur Data	No faktur + Tgl penjualan + total bayar + user name + nama konsumen + kode konsumen
	No faktur Tanggal penjualan Nama konsumen Kode konsumen Total bayar User name	[0..9] [0..9] [a..z | A..*Z*] [0..9] [a..z | A..*Z*]
6	Nama	*Data Detail penjualan*
	Where used / how used	cari *detail penjualan* (masukan) tambah *detail penjualan* (masukan) ubah *detail penjualan* (masukan) *detail penjualan* (masukan)
	Deskripsi	Berisi data *detail penjualan* yang akan ditambah, diubah, dicari
	Struktur Data	No faktur + kode barang + jumlah + nama barang + harga
	No faktur Kode barang Jumlah Nama barang Harga jual	[0..9] [0..9] [0..9] [0..9] [0..9]
7	Nama	Data supplier
	Where used / how used	pengubahan data barang supplier (masukan)
	Deskripsi	Berisi data supplier yang ada pada kenyataannya di*counter dan gudang*
	Struktur Data	Kode supplier + nama supplier + alamat + no.telp + kontak person

	Kode supplier Nama supplier Alamat No. Telp Kontak person	[0..9] [a..z | A..Z] [a..z | A..Z][0..9] [0..9] [a..z | A..Z]
8	Nama	Data konsumen
	Where used / how used	pengubahan data konsumen (masukan)
	Deskripsi	Berisi data konsumen yang ada pada kenyataannya di counter dan gudang
	Struktur Data	Nama konsumen + kode konsumen + Alamat + no. telp
	Nama konsumen Kode konsumen Alamat No. Tlp	[a..z | A..Z] [0..9] [a..z | A..Z][0..9] [0..9]
9	Nama	Data ukuran
	Where used / how used	pengubahan data ukuran (masukan)
	Deskripsi	Berisi data ukuran yang ada pada kenyataannya di counter dan gudang
	Struktur Data	Kode ukuran + ukuran
	Kode ukuran Jenis	[0..9] [a..z | A..Z]
10	Nama	Data jenis
	Where used / how used	pengubahan data jenis (masukan)
	Deskripsi	Berisi data jenis yang ada pada kenyataannya di counter dan gudang
	Struktur Data	Kode ukuran + jenis
	Kode ukuran Jenis	[0..9] [a..z | A..Z]
11	Nama	Data merk
	Where used / how used	pengubahan data merk (masukan)
	Deskripsi	Berisi data merk yang ada pada kenyataannya di counter dan gudang
	Struktur Data	Kode merk + merk
	Kode merk Merk	[0..9] [a..z | A..Z]
12	Nama	Data pengguna
	Where used / how used	Tambah perngguna (masukan)
	Deskripsi	Berisi data pengguana yang akan ditambah,diubah,dihapus
	Struktur Data	User name + Password + tipe
	User name password Jenis pengguna	[a..z | A..Z] [0..9] [A..Z]
10	Nama	Data bahan
	Where used / how used	pengubahan data bahan (masukan)
	Deskripsi	Berisi data bahan yang ada pada kenyataannya di gudang
	Struktur Data	Kode bahan + bahan
	Kode bahan bahan	[0..9] [a..z | A..Z]

11	Nama	Data bentuk
	Where used / how used	pengubahan data bentuk (masukan)
	Deskripsi	Berisi data bentuk yang ada pada kenyataannya di gudang
	Struktur Data	Kode bentuk + bentuk
	Kode bentuk Bentuk	[0..9] [a..z | A..Z]
12	Nama	Data warna
	Where used / how used	Tambah warna (masukan)
	Deskripsi	Berisi data warna yang akan ditambah,diubah,dihapus
	Struktur Data	Kode warna + warna
	Kode warna Warna	[0..9] [a..z | A..Z]
13	Nama	Data bentuk jenis
	Where used / how used	Tambah bentuk jenis (masukan)
	Deskripsi	Berisi data bentuk jenis yang akan ditambah,diubah,dihapus
	Struktur Data	Kode jenis + kode bentuk
	Kode warna Warna	[0..9] [a..z | A..Z]
14	Nama	Data bahan jenis
	Where used / how used	Tambah bahan jenis (masukan)
	Deskripsi	Berisi data bahan jenis yang akan ditambah,diubah,dihapus
	Struktur Data	Kode jenis + kode bahan
	Kode warna Warna	[0..9] [a..z | A..Z]
15	Nama	Data ukuran jenis
	Where used / how used	Tambah ukuran jenis (masukan)
	Deskripsi	Berisi data ukuran jenis yang akan ditambah,diubah,dihapus
	Struktur Data	Kode jenis + kode ukuran
	Kode warna Warna	[0..9] [a..z | A..Z]
16	Nama	Data merk jenis
	Where used / how used	Tambah merk jenis (masukan)
	Deskripsi	Berisi data merk jenis yang akan ditambah,diubah,dihapus
	Struktur Data	Kode jenis + kode ukuran
	Kode warna Warna	[0..9] [a..z | A..Z]

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan untuk menggambarkan perbedaan antara sistem yang sedang berjalan dengan sistem

yang diusulkan. Perancangan sistem ini menggunakan pendekatan sistem terkomputerisasi.

Langkah-langkah yang dilakukan dalam tahapan perancangan sistem ini adalah sebagai berikut:

1. Perancangan Data
    a. Skema relasi
    b. Struktur *File*
2. Arsitektur
3. Perancangan antar muka
    a. Perancangan *Form*
    b. Perancangan Pesan
4. Perancangan procedural

3.2.1 Perancangan Data

Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan atau merancang data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah skema relasi dan struktur *file*. Dari dua hasil tersebut, implementasi basis data akan bisa dikerjakan.

3.2.1.1 Skema relasi

Skema relasi antar *file* merupakan gabungan antar *file* yang mempunyai kunci utama yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci tersebut. Pada proses ini elemen-elemen data dikelompokan menjadi satu *file database* beserta *entitas* dan hubungannya. Tabel-

tabel yang digunakan dalam Rancang Bangun Sistem Informasi Penjualan dan pembelianDI.PD. Pembangunan Raya Berbasis *Client Server* yaitu:

1. T.Pengguna

   {*User_name, Password, Jenis_pengguna}

2. T.Supplier

   {*Kode_supplier, Nama_supplier, Alamat, No.Telp, Kontak_person}

3. T.Konsumen

   {*Kode_konsumen, nama_konsumen, Alamat, No_Telp,}

4. T.Pembelian

   {*No_faktur, Tanggal_pembelian, Kode_supplier, Total_bayar, user_name}

5. T.Detail_pembelian

   {*No_faktur, Kode_barang, Jumlah, Harga, Total_bayar}

6. T.Barang

   {*Kode_barang, Nama_barang, Warna, Harga, Stok, Bentuk, Kode_ukuran, Kode_jenis, Kode_merk}

7. T.Penjualan

   {*No.faktur, Tanggal_penjualan, nama_konsumen, kode_konsumen, Total_Bayar, user_name}

8. T.Detail_penjualan

   {*No_faktur, Kode_barang, Jumlah, Nama_barang, Harga}

9. T.Ukuran

   {*Kode_ukuran, Ukuran}

10. T.Jenis

{*Kode_jenis, Jenis}

11. T.Merk

    {*Kode_merk, Merk}

12. T.Bahan

    {*Kode_bahan, Bahan}

13. T.Bentuk

    {*Kode_Bentuk, Bentuk}

14. T.Warna

    {*Kode_Warna, Warna}

15. T.Merk_jenis

    {*Kode_jenis, Kode_merk}

16. T.Bahan_jenis

    {*Kode_jenis, Kode_bahan}

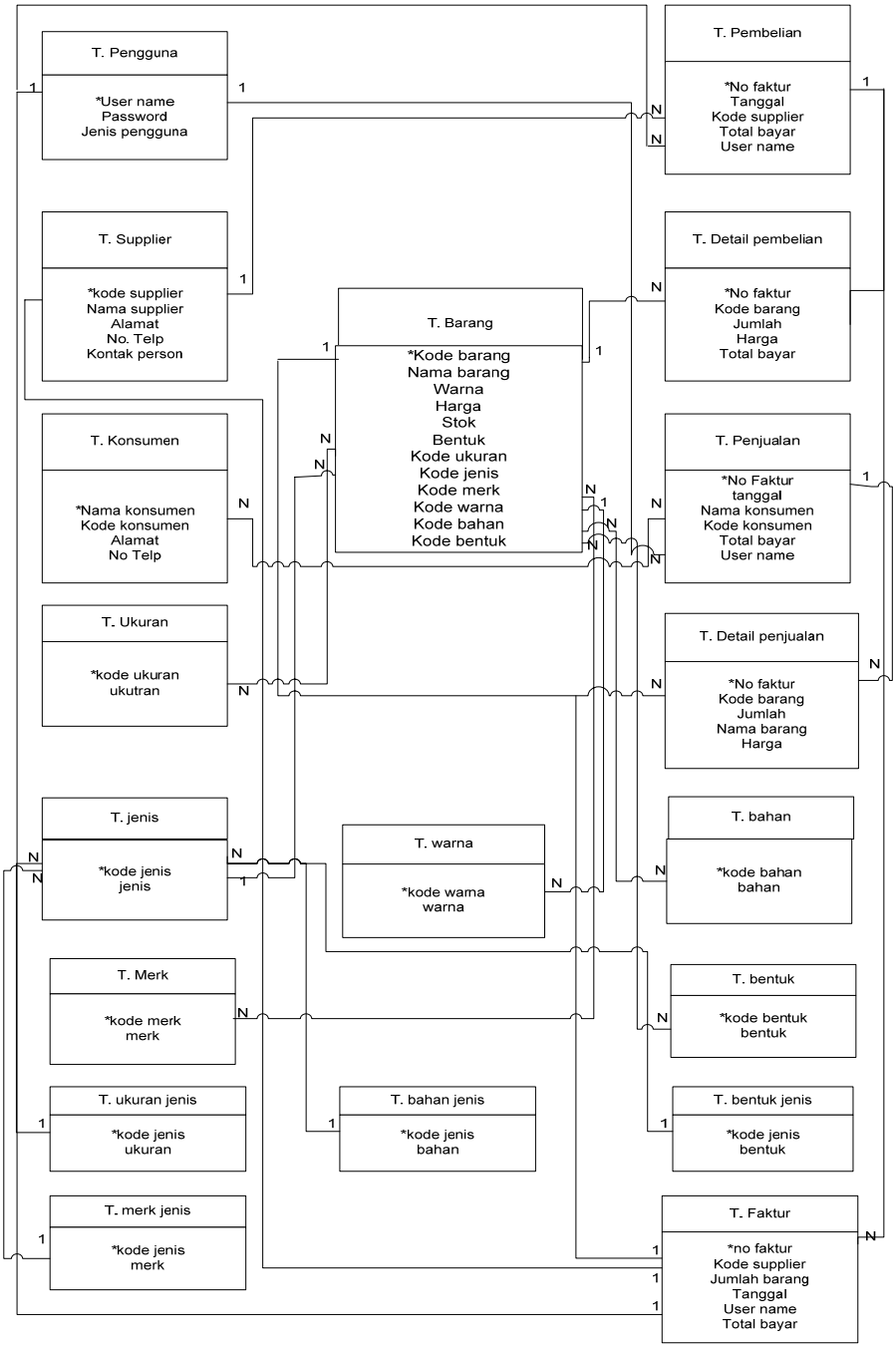
17. T.Bentuk_jenis

    {*Kode_jenis, Kode_Bentuk}

18. T.Ukuran_jenis

    {*Kode_ukuran, Kode_ukuran}

Untuk menggambarkan secara jelas skema relasi digunakan diagram skema supaya lebih terlihat hubungan antar tabel. Adapun diagram skema dari sistem informasi Penjualan Dan Pembelian Bahan Baku Bangunan Di PD.Pembangunan Raya dapat dilihat pada gambar 3.24 dibawah ini :



Gambar 3.24 Skema Relasi

3.2.2 Struktur Tabel

Struktur tabel adalah spesifikasi *file* dari tiap tabel yang digunakan dalam sistem yang akan dibangun. Struktur tabel dari sistem yang dibangun adalah sebagai berikut:

a. Tabel Konsumen

Nama Tabel : t_konsumen

Primary Key : nama konsumen

Fungsi : Untuk menyimpan data konsumen

Tabel 3.3 Struktur *File* konsumen

Nama *Field*	Tipe Data	Ukuran	Keterangan
Nama konsumen	Varchar	30	*Primary key*
Kode konsumen	Varchar	6	
No. Telp	Varchar	15	
Alamat	Varchar	30	

b. Tabel *supplier*

Nama Tabel : t_supplier

Primary Key : kode supplier

Fungsi : Untuk menyimpan data *supplier*

Tabel 3.4 Struktur *File supplier*

Nama *Field*	Tipe Data	Ukuran	Keterangan
Kode supplier	Varchar	6	*Primary key*
Nama supplier	Varchar	30	
Alamat	Varchar	30	
No.telp	Varchar	15	
Kontak person	Varchar	30	

c. Tabel barang

Nama Tabel : t_barang

Primary Key : kode barang

Fungsi : Untuk menyimpan data barang

Tabel 3.5 Struktur *File* barang

Nama *field*	Tipe data	Ukuran	Keterangan
Kode barang	Varchar	8	*Primary key*
Nama barang	Varchar	30	
Kode merk	Float	6	Foreign key dari tabel merk, ke field kode merk
Kode ukuran	Varchar	6	Foreign key dari tabel ukuran, ke field kode ukuran
Kode jenis	Varchar	6	Foreign key dari tabel jenis, ke field kode jenis
Bentuk	Varchar	20	
Stok	Varchar	30	
Harga	Varchar	20	
Warna	Char	20	

d. Tabel pembelian

Nama Tabel : t_pembelian

Primary Key : no faktur

Fungsi : Untuk menyimpan data pembelian barang

Tabel 3.6 Struktur *File pembelian*

Nama *field*	Tipe data	Ukuran	Keterangan
No faktur	Varchar	10	*Primary key*
Tgl pembelian	Date		
User name	Varchar	30	Foreign key dari tabel pengguna, ke field user name
Total bayar	Integer	20	
Kode supplier	Varchar	10	Foreign key dari tabel supplier, ke field kode supplier

e. Tabel detail pembelian

Nama Tabel : t_detailpembelian

Primary Key : no faktur

Fungsi : Untuk menyimpan data detail pembelian barang

Tabel 3.7 Struktur *File* detail *pembelian*

Nama *field*	Tipe data	Ukuran	Keterangan
No faktur	Varchar	10	*Primary key*
Kode barang	Varchar	8	Foreign key dari tabel barang, ke field kode barang
Jumlah	Int	20	
Harga	Int	20	
Total bayar	Int	20	

f. Tabel penjualan

Nama Tabel : t_penjualan

Primary Key : no faktur

Fungsi : Untuk menyimpan data penjualan

Tabel 3.8 Struktur *File* penjualan

Nama *field*	Tipe data	Ukuran	Keterangan
No faktur	Varchar	10	*Primary key*
Tgl penjualan	Date		
Nama konsumen	Varchar	30	Foreign key dari tabel konsumen, ke field nama pelanggan
Total bayar	Int	20	
User name	Varchar	30	Foreign key dari tabel pengguna, ke field user name
Kode konsumen	Varchar	6	

g. Tabel detail penjualan

Nama Tabel : t_detail penjualan

Primary Key : no faktur

Fungsi : Untuk menyimpan data detail penjualan barang

Tabel 3.9 Struktur *File* detail *penjualan*

Nama *field*	Tipe data	Ukuran	Keterangan
No faktur	Varchar	10	*Primary key*
Kode barang	Varchar	8	Foreign key dari tabel barang, ke field kode barang
Nama barang	Varchar	30	
Jumlah	Int	20	
Harga	Int	20	

h. Tabel merk

Nama Tabel : t_merk

Primary Key : kode merk

Fungsi : Untuk menyimpan data merk

Tabel 3.10 Struktur *File* merk

Nama *field*	Tipe data	Ukuran	Keterangan
kode merk	Varchar	6	*Primary key*
Merk	Varchar	30	

i. Tabel jenis

Nama Tabel : t_jenis

Primary Key : kode jenis

Fungsi : Untuk menyimpan data jenis barang

Tabel 3.11 Struktur *File* jenis

Nama *field*	Tipe data	Ukuran	Keterangan
kode jenis	Varchar	6	*Primary key*
Jenis	Varchar	30	

j. Tabel ukuran

Nama Tabel : t_ukuran

Primary Key : kode ukuran

Fungsi : Untuk menyimpan data ukuran barang

Tabel 3.12 Struktur *File* ukuran

Nama *field*	Tipe data	Ukuran	Keterangan
kode ukuran	Varchar	6	*Primary key*
Ukuran	Varchar	30	

k. Tabel pengguna

Nama Tabel : t_pengguna

Primary Key : user name

Fungsi : Untuk menyimpan data pengguna

Tabel 3.13 Struktur *File* pengguna

Nama *field*	Tipe data	Ukuran	Keterangan
User name	Varchar	30	*Primary key*
Password	Varchar	10	
Jenis pengguna	Varchar	20	

l. Tabel Bahan

Nama Tabel : t_bahan

Primary Key : kode bahan

Fungsi : Untuk menyimpan data bahan barang

Tabel 3.14 Struktur *File* bahan

Nama *field*	Tipe data	Ukuran	Keterangan
kode bahan	Varchar	6	*Primary key*
bahan	Varchar	30	

m. Tabel Bentuk

Nama Tabel : t_bentuk

Primary Key : kode bentuk

Fungsi : Untuk menyimpan data bentuk barang

Tabel 3.15 Struktur *File* bentuk

Nama *field*	Tipe data	Ukuran	Keterangan
kode bentuk	Varchar	6	*Primary key*
Bentuk	Varchar	30	

n. Tabel Warna

Nama Tabel : t_warna

Primary Key : kode warna

Fungsi : Untuk menyimpan data warna barang

Tabel 3.16 Struktur *File* warna

Nama *field*	Tipe data	Ukuran	Keterangan
kode warna	Varchar	6	*Primary key*
warna	Varchar	30	

Tabel 3.17 Struktur *File* merk jenis

Nama *field*	Tipe data	Ukuran	Keterangan
kode jenis	Varchar	6	*Primary key*
Kode merk	Varchar	30	

Tabel 3.18 Struktur *File* ukuran jenis

Nama *field*	Tipe data	Ukuran	Keterangan
kode jenis	Varchar	6	*Primary key*
Kode ukuran	Varchar	30	

Tabel 3.19 Struktur *File* bahan jenis

Nama *field*	Tipe data	Ukuran	Keterangan
kode jenis	Varchar	6	*Primary key*
Kode bahan	Varchar	30	

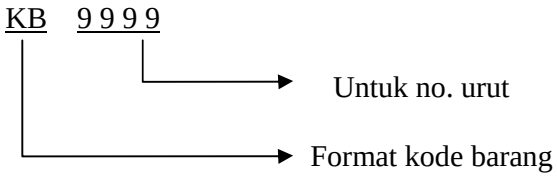
Tabel 3.20 Struktur *File* bentuk jenis

Nama *field*	Tipe data	Ukuran	Keterangan
kode jenis	Varchar	6	*Primary key*
Kode bentuk	Varchar	30	

3.2.3 Perancangan Kode

Kode yang akan digunakan dalam sistem yang akan dibangun adalah kode barang yang sudah ada, ditambah kode faktur penjualan dan faktur pembelian, kode jenis, kode merk dan kode ukuran Perancangan kode dari nota yang akan digunakan adalah sebagai berikut:

1. Kode barang



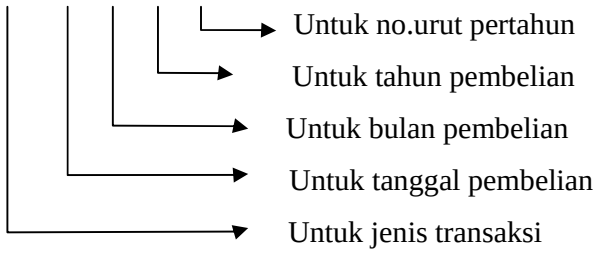
Contoh :

KB 0001, berarti Kran Bak pengelompokan dengan nomber urut 0001

2. Kode Faktur jual dan beli

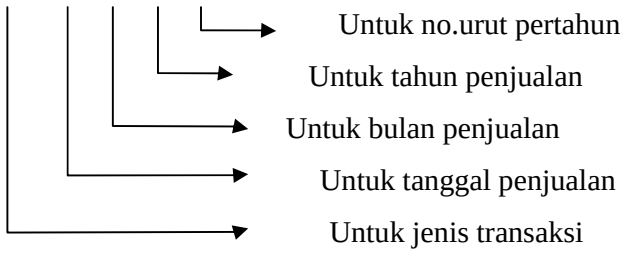
XX 99 99 99 999

FB 99 99 99 999



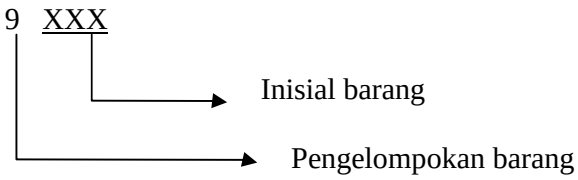
XX 99 99 99 999

FB 99 99 99 999



Contoh: FJ 01 08 09 001, berarti jenis transaksi penjualan bulan agustus tahun 2009 dengan no urut 0001

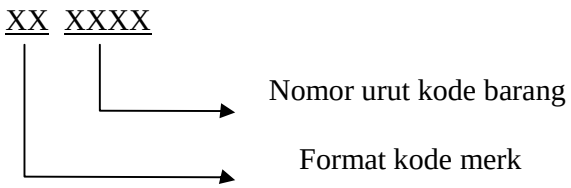
3. Jenis Barang



Contoh:

1 KTM, berarti Pengelompokan Kamar mandi, Jenis nya Kran, Inisial barangnya Tembok

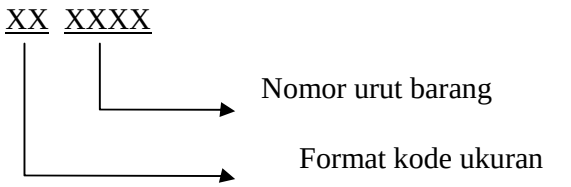
4. Merk Barang



Contoh:

KK 0001,berarti merk barangnya yang bernomor urut 0001

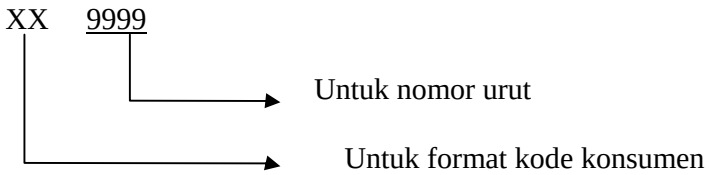
5. Ukuran Barang



Contoh:

KU0001,berarti ukuran barangnya yang bernomor urut 0001 berarti 10 cm

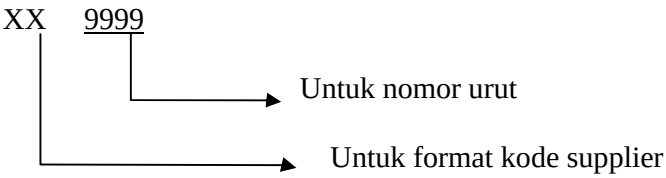
6. Kode konsumen



Contoh:

KK0001, berarti konsumen dengan nomor urut 0001

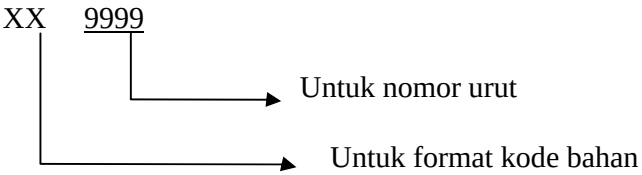
7. Kode supplier



Contoh:

KS0001, berarti supplier dengan nomor urut 0001

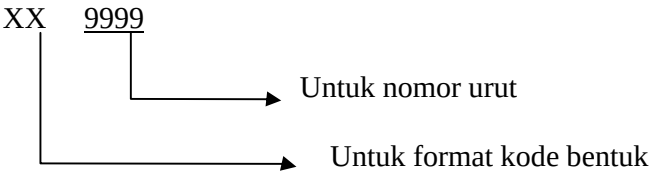
8. Kode bahan



Contoh:

BH0001, berarti bahan dengan nomor urut 0001

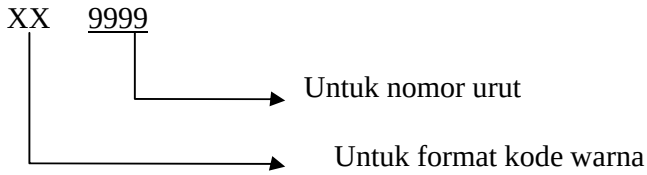
9. Kode bentuk



Contoh:

BN0001, berarti bentuk dengan nomor urut 0001

10. Kode supplier

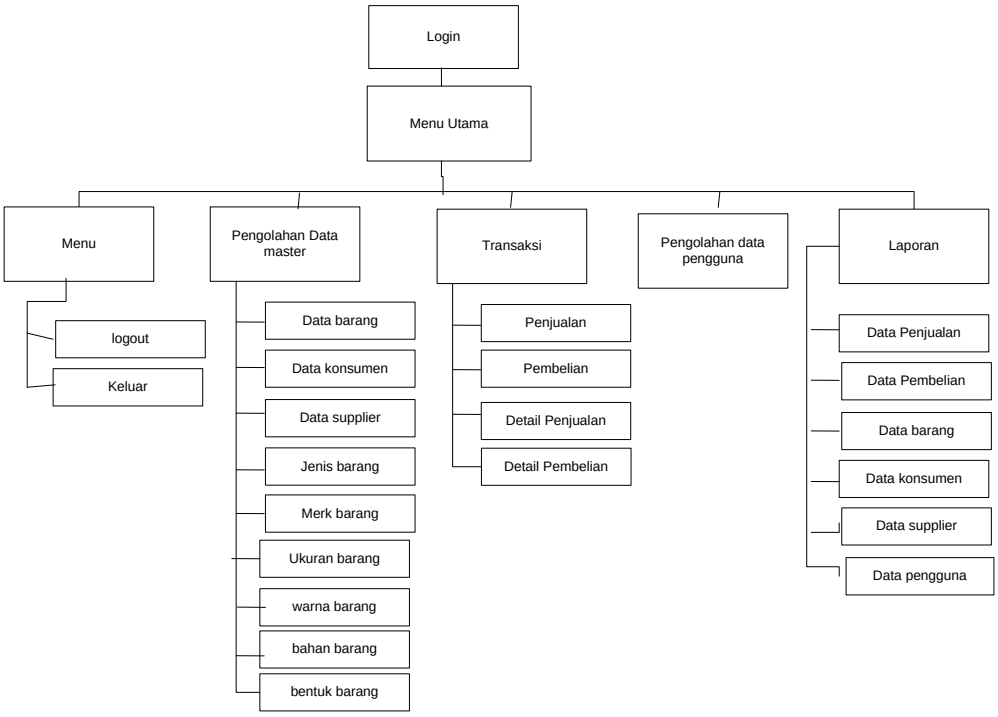


Contoh:

KW0001, berarti warna dengan nomor urut 0001

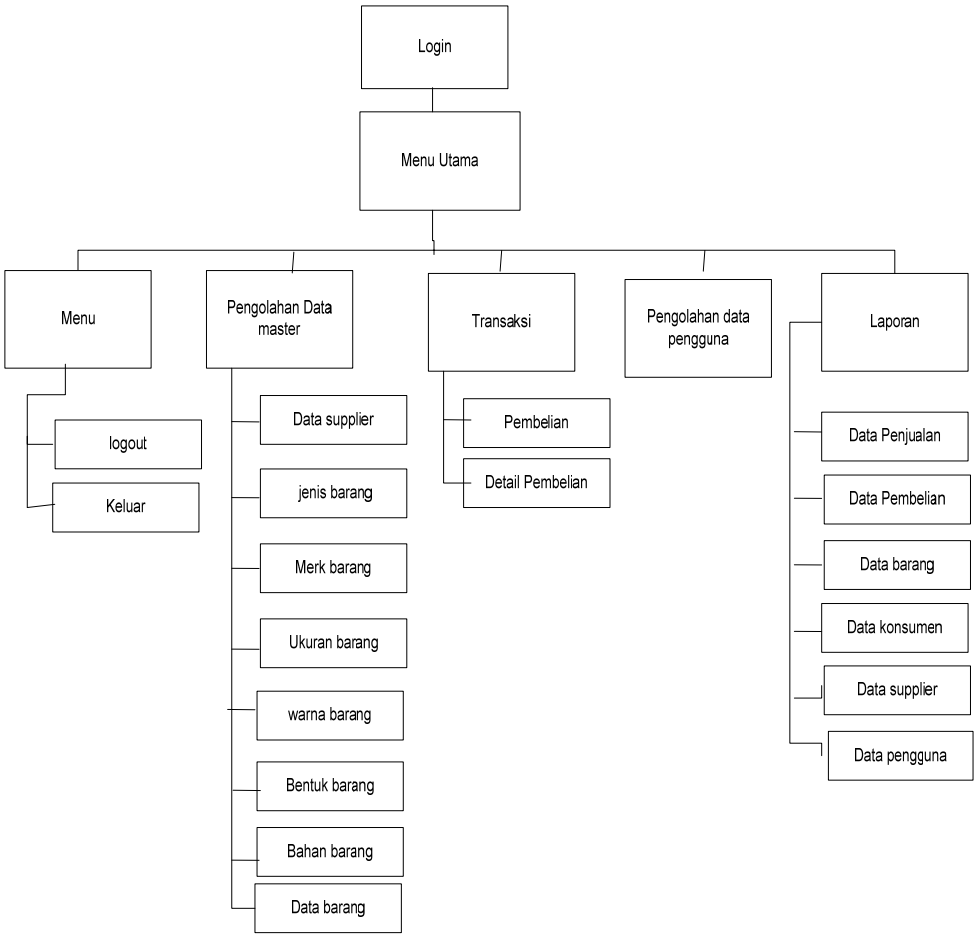
3.2.4 Perancangan Struktur Menu

Perancangan antarmuka sistem informasi Penjualan dan pembelian pada PD.Pembangunan Raya, yaitu:



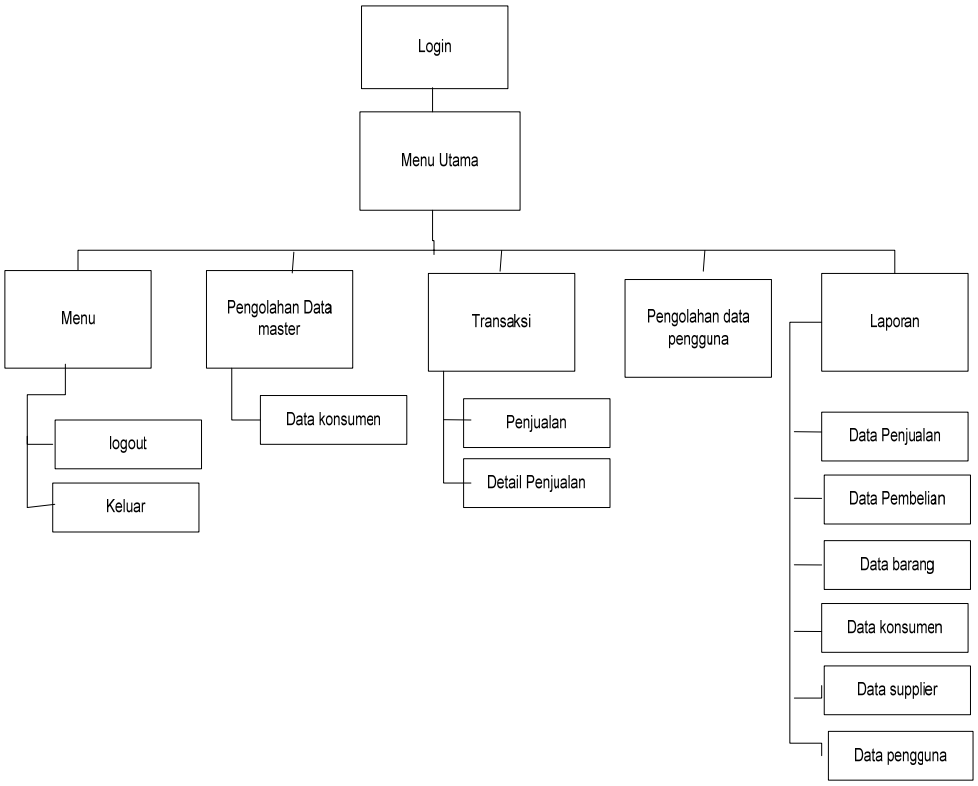
Gambar 3.25 Stuktur Menu

3.2.4.1 struktur menu untuk bagian gudang



Gambar 3.26 Stuktur Menu bagian gudang

3.2.4.2 struktur menu untuk bagian counter



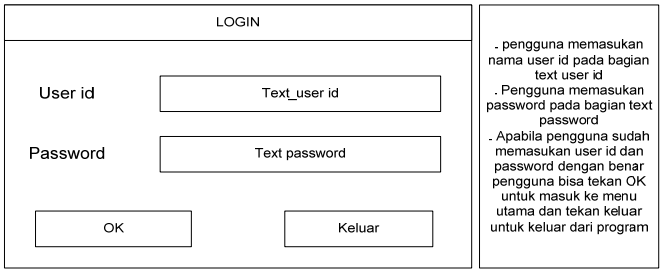
Gambar 3.27 Stuktur Menu bagian counter

3.2.5 Perancangan Antar Muka

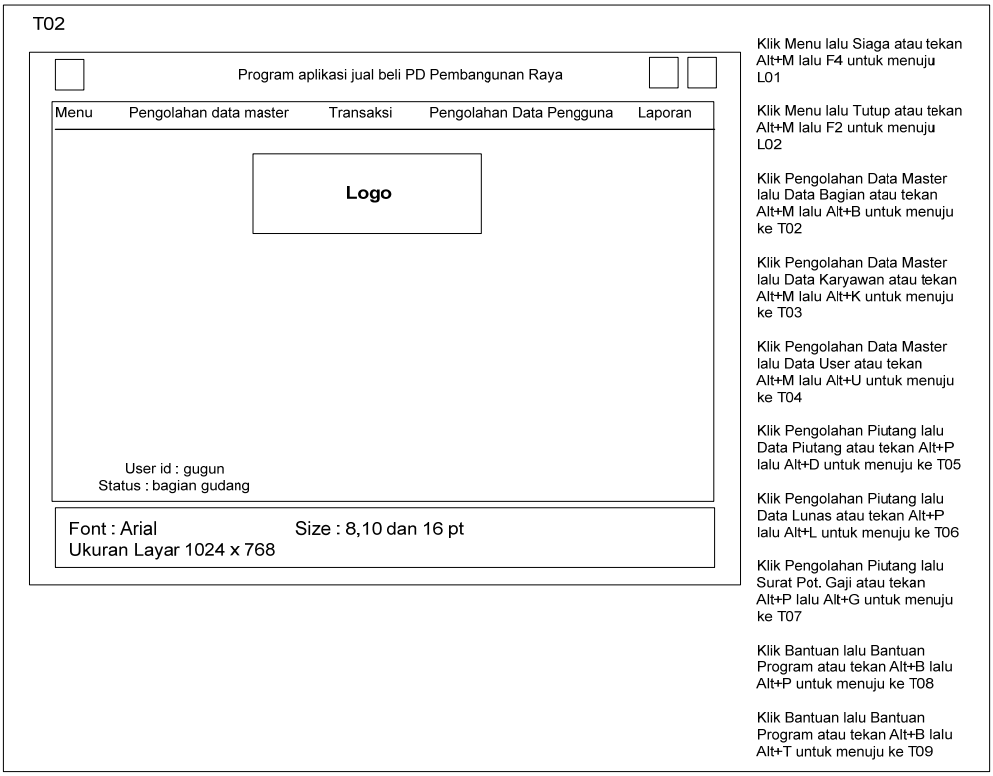
Perancangan antarmuka dibuat untuk menggambarkan tampilan program yang akan digunakan oleh pengguna untuk berinteraksi dengan aplikasi yang dibuat. Perancangan dibuat berdasarkan tampilan antarmuka baik *input* maupun *output* yang akan dihasilkan aplikasi saat diimplementasikan.

3.2.5.1 Spesifikasi Antar Muka

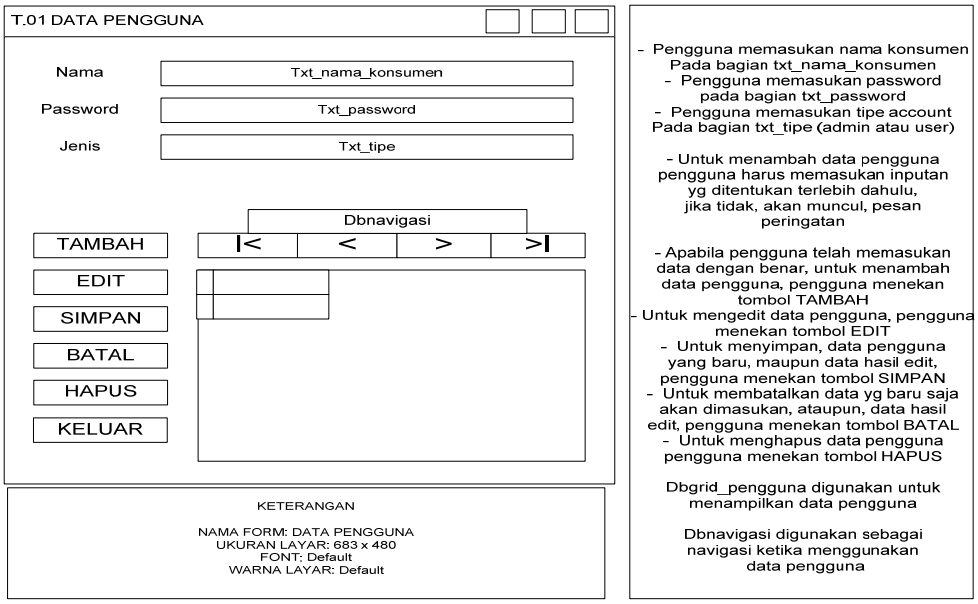
Spesifikasi antar muka merupakan suatu bentuk tampilan dari program yang akan dibuat utuk kebutuhan *interface* dengan *user*. Dibawah ini adalah Spesifikasi Antar Muka Sistem Informasi Iformasi Penjualan dan pembelian DI.PD Pembangunan Raya Berbasisi *Client_ Server.*



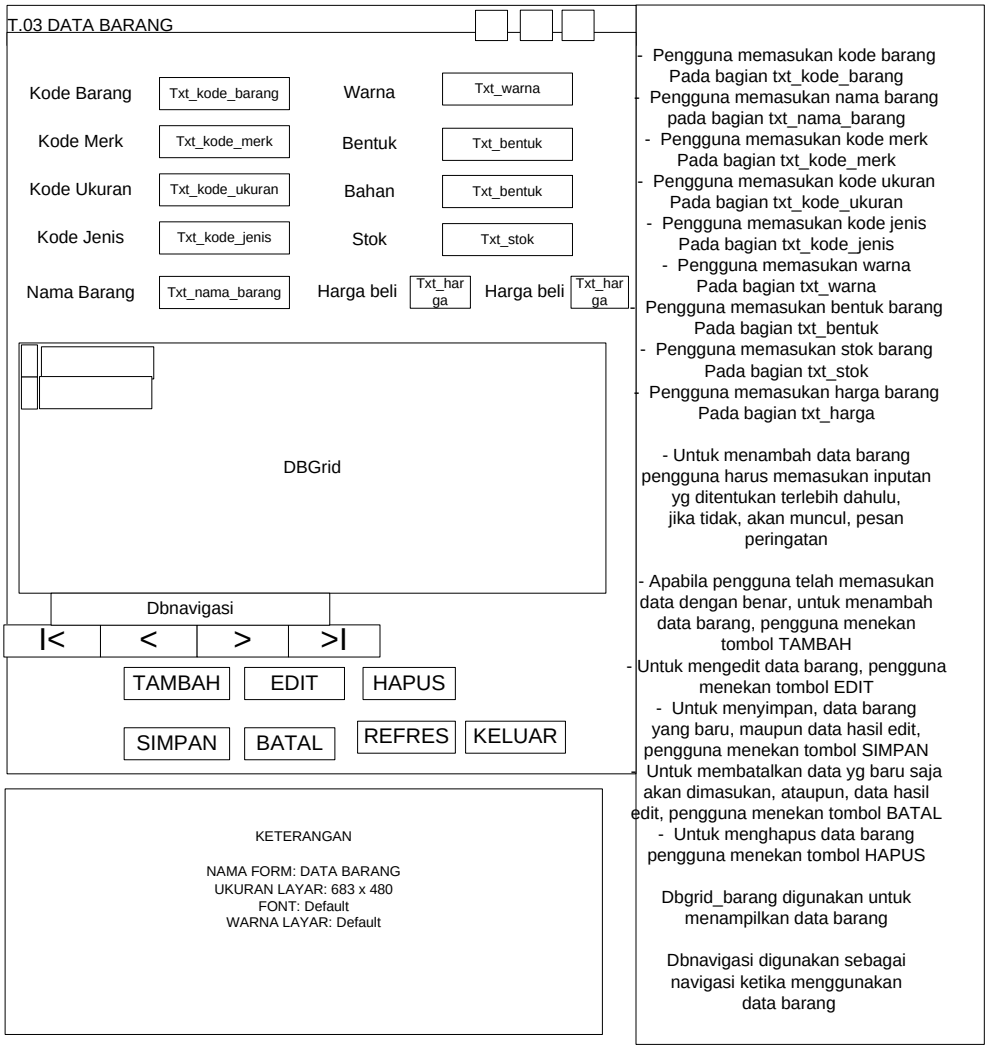
Gambar 3.28 Antar Muka Login



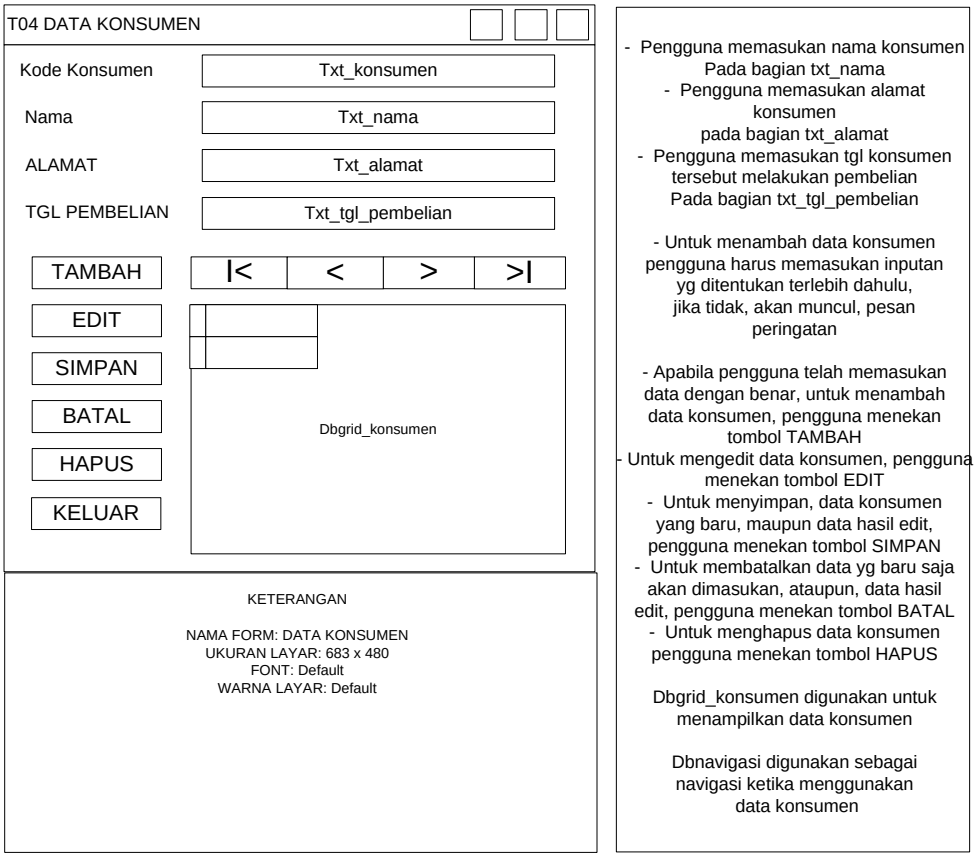
Gambar 3.29 Antar Muka Menu Utama



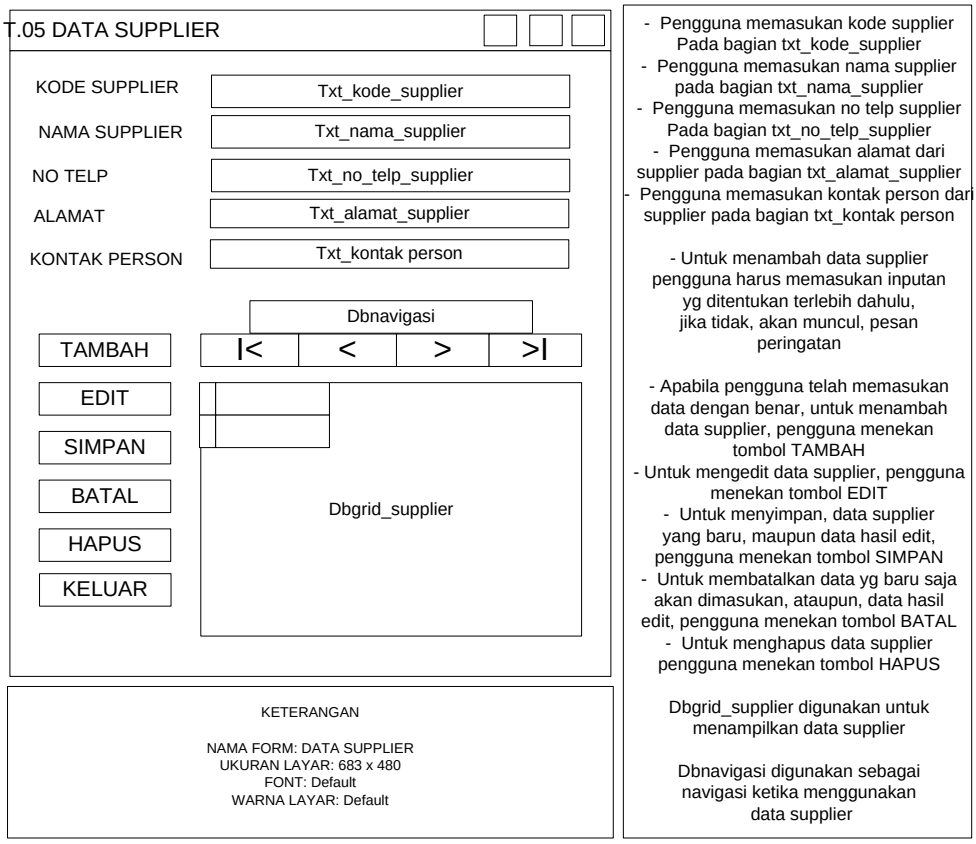
Gambar 3.30 Antar Muka data pengguna



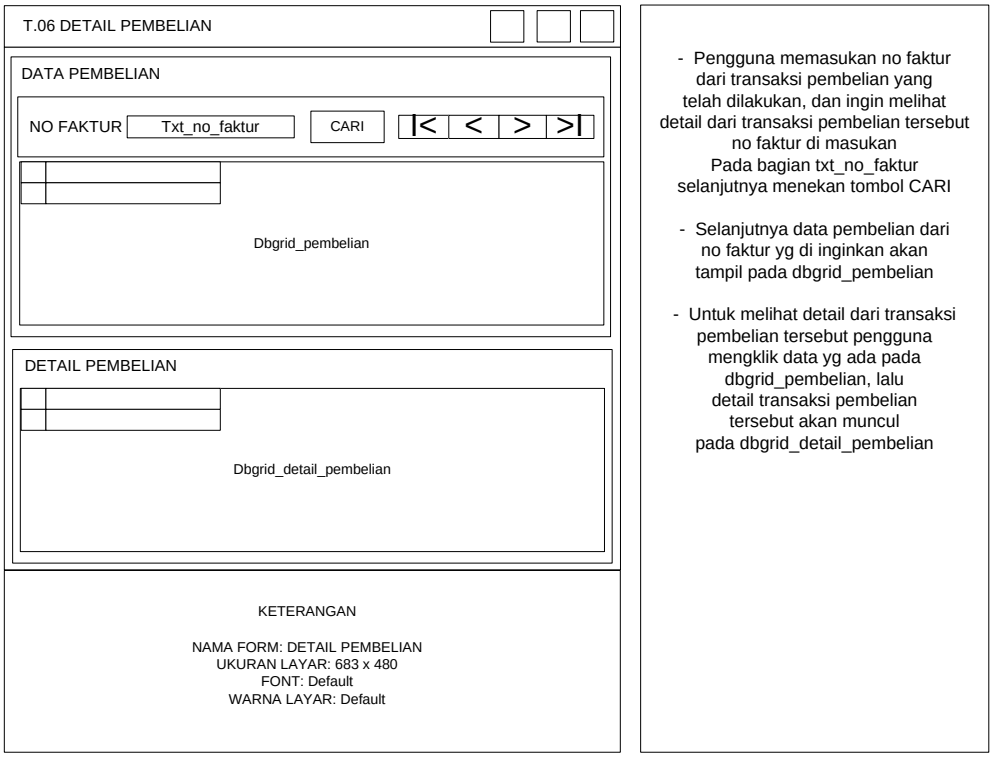
Gambar 3.31 Antar Muka Data Barang



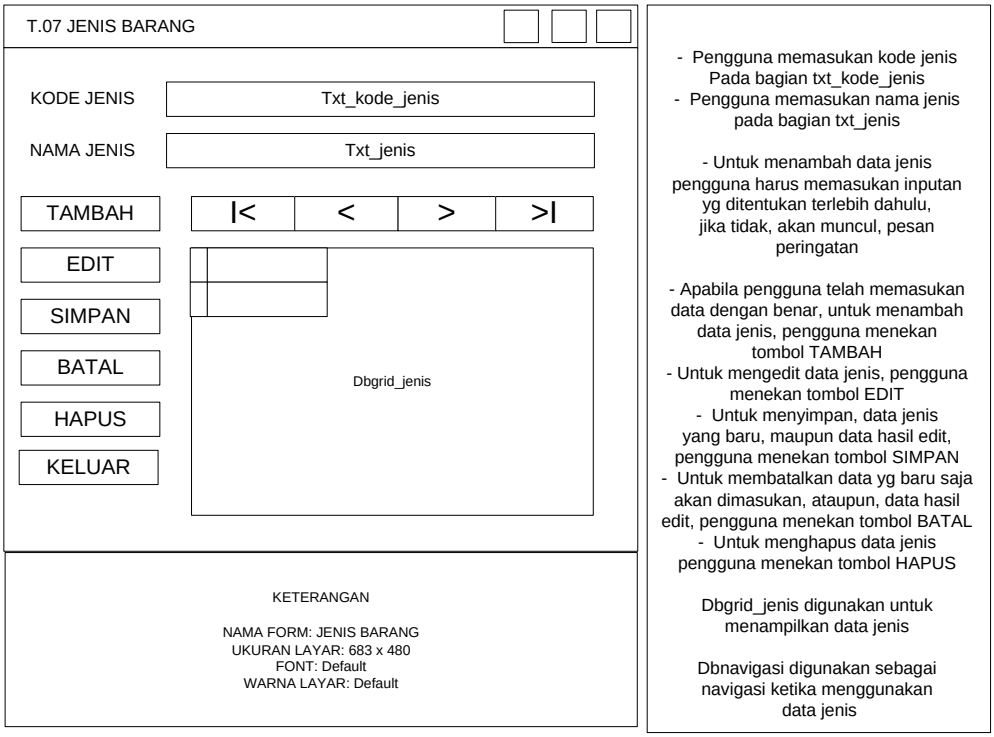
Gambar 3.32 Antar Muka Data Konsumen



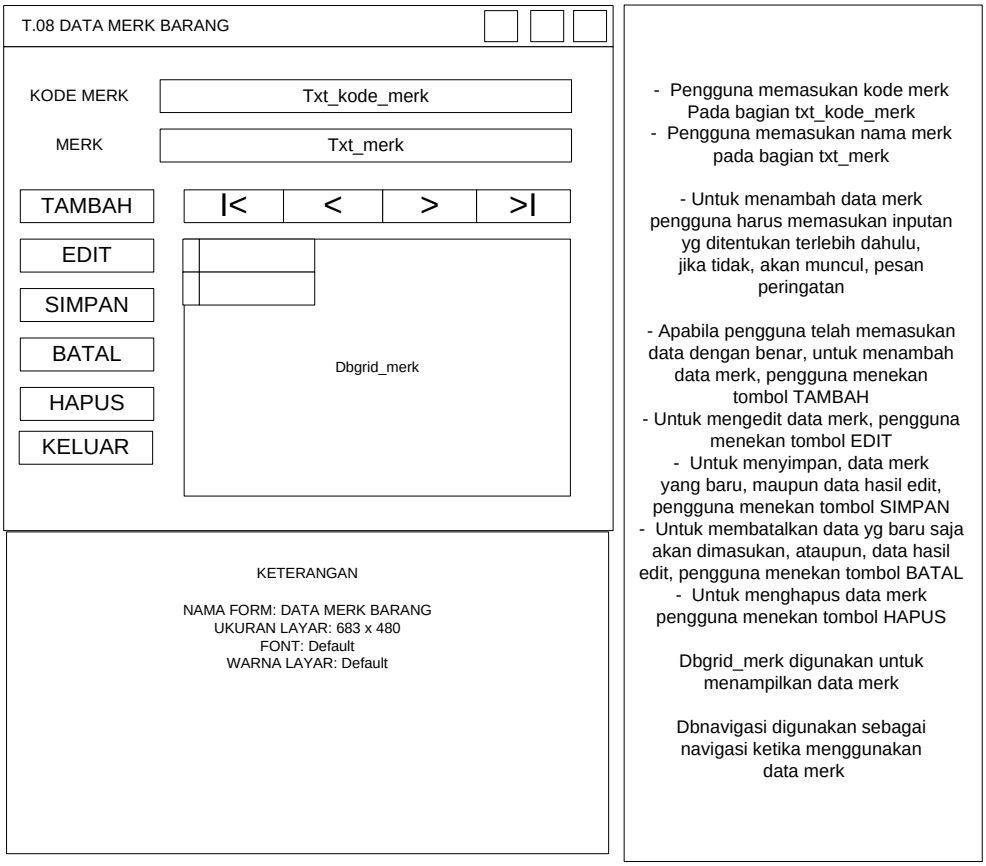
Gambar 3.33 Antar Muka Menu data Supplier



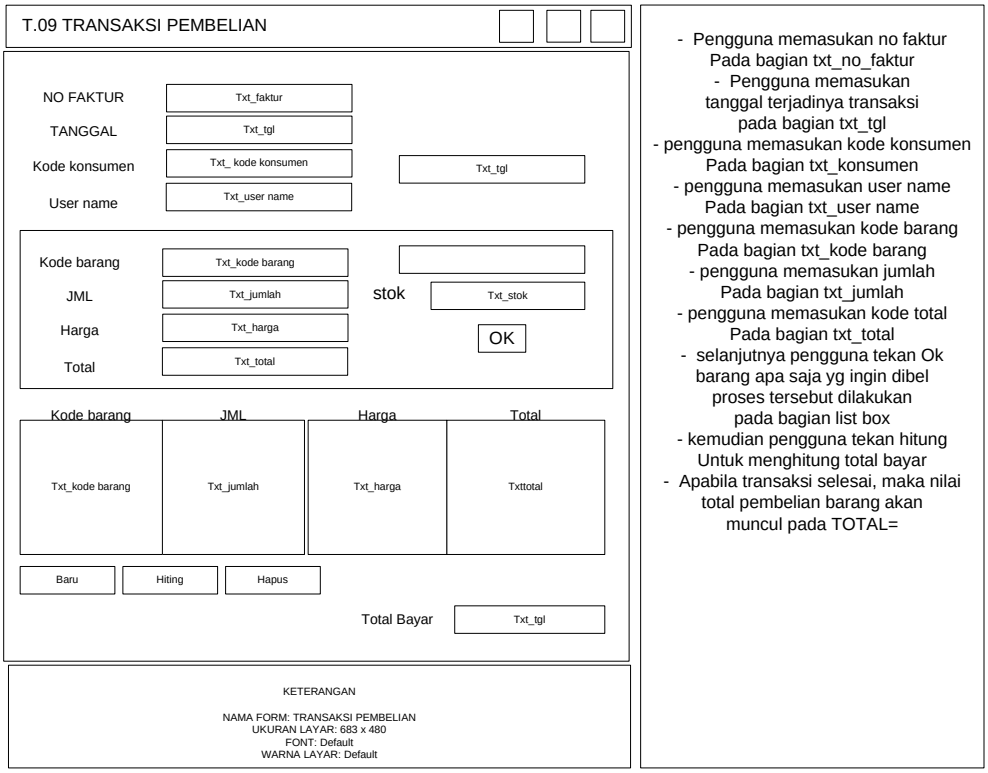
Gambar 3.34 Antar Muka Detail Pembelian



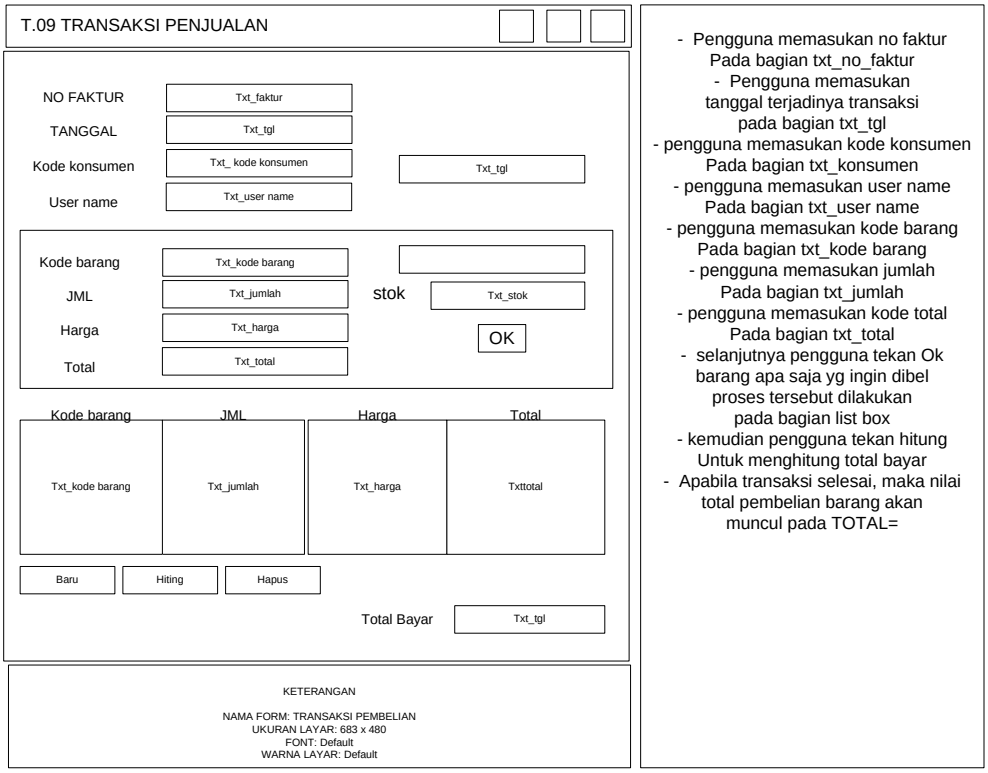
Gambar 3.35 Antar Muka Jenis Barang



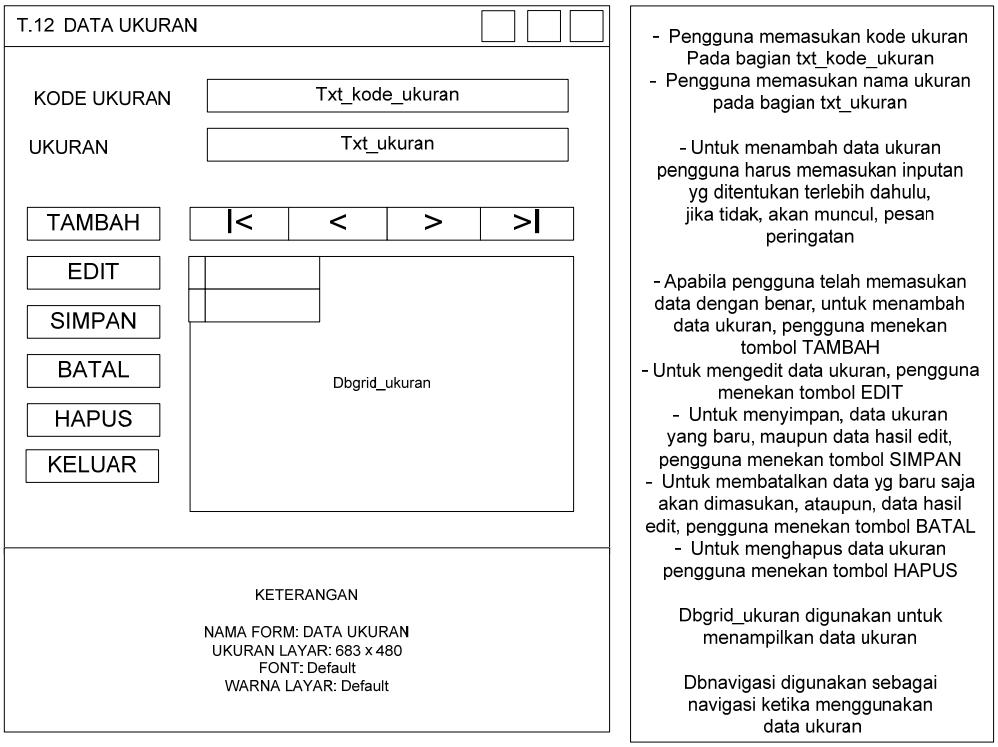
Gambar 3.36 Antar Muka Menu Data Merk Barang



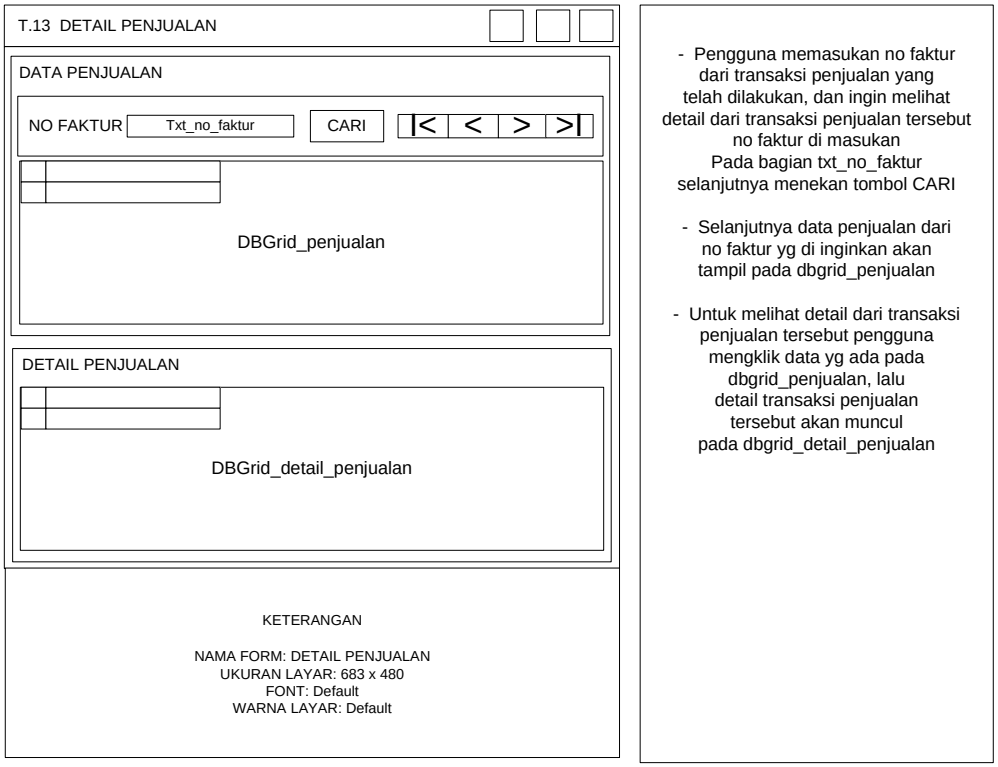
Gambar 3.37 Antar Muka Menu data trasaksi pembelian



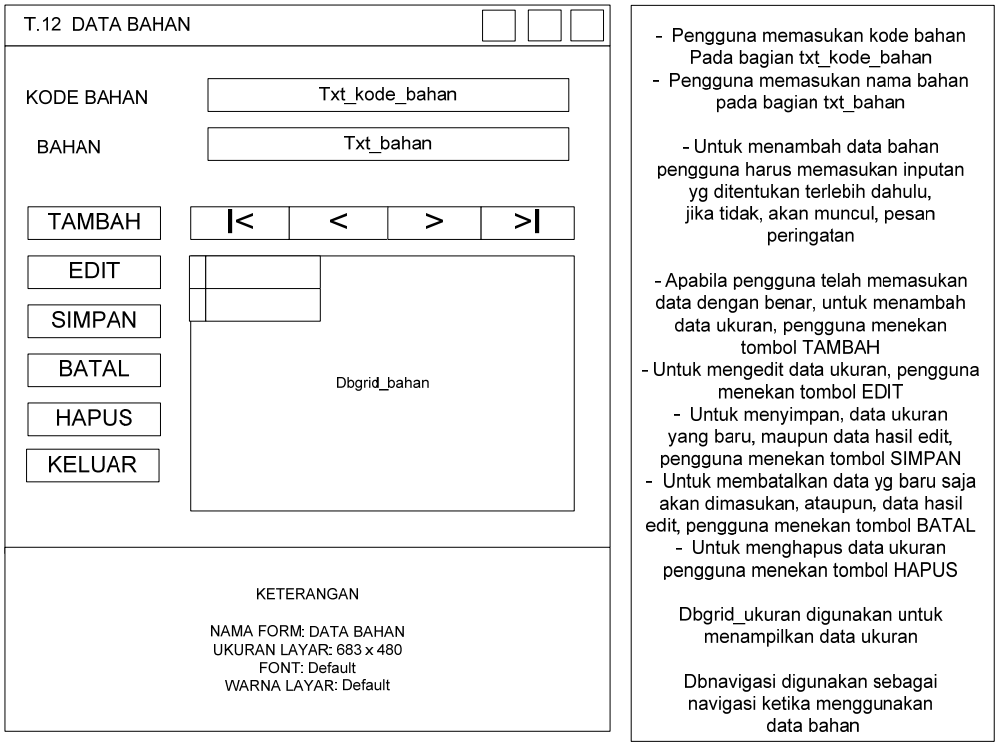
Gambar 3.38 Antar Muka Transaksi Penjualan



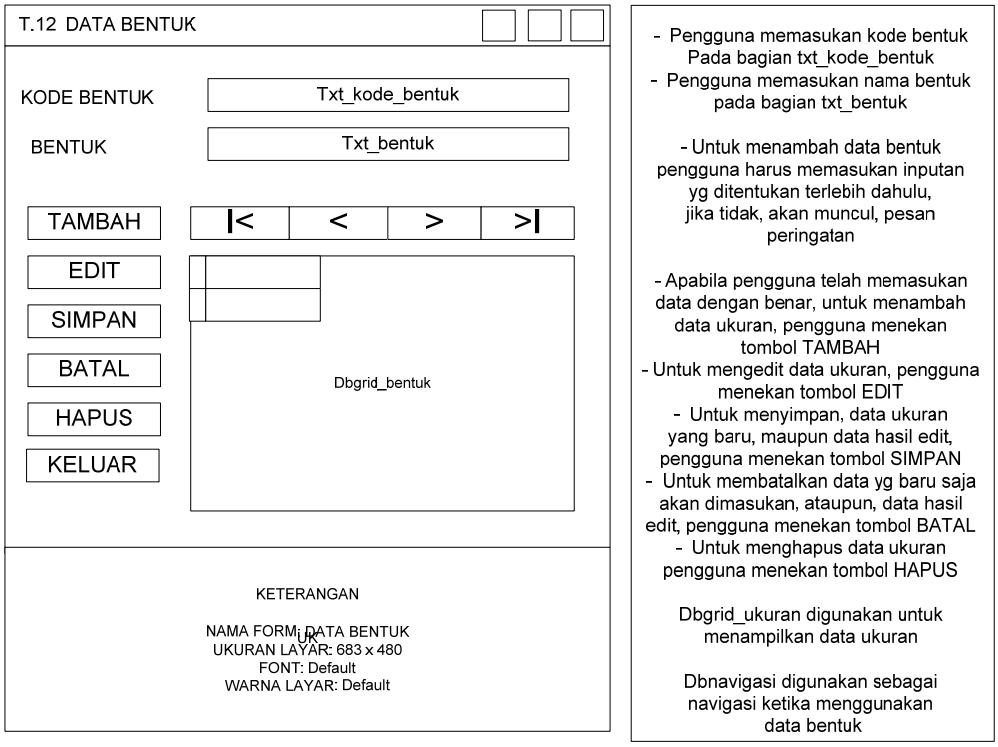
Gambar 3.39 Antar Muka Menu Data Ukuran



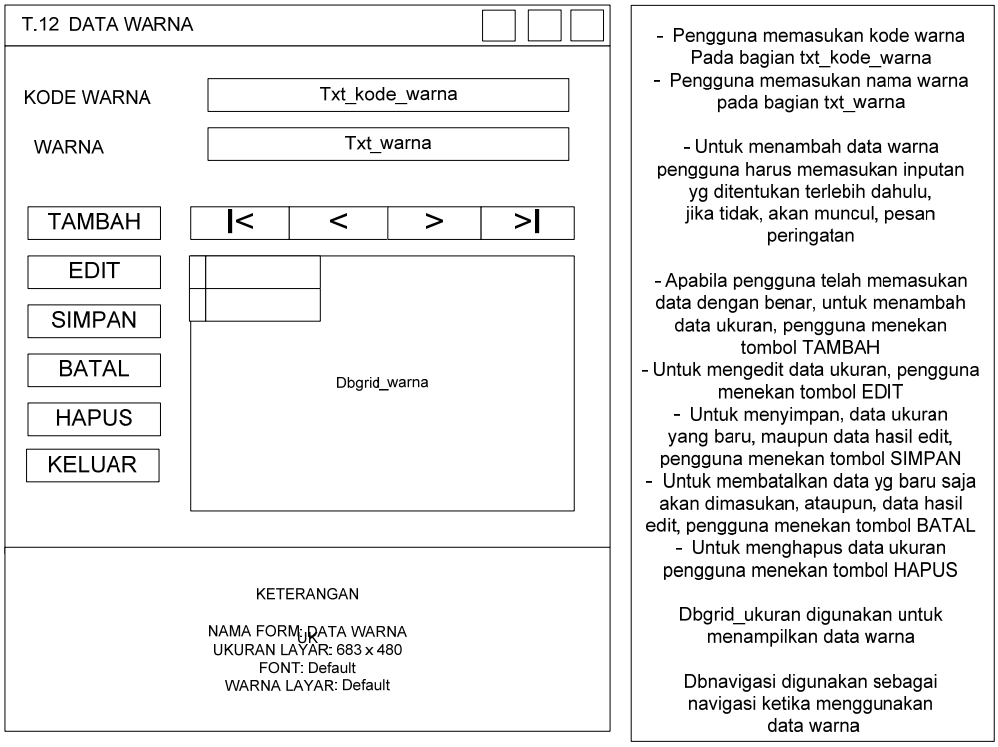
Gambar 3.40 Antar Muka Detail Penjualan



Gambar 3.41 Antar Muka Data Bahan



Gambar 3.42 Antar Muka Data Bentuk



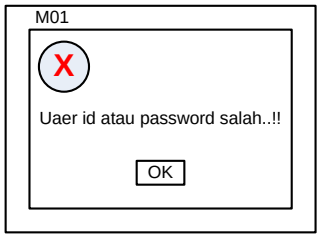
Gambar 3.43 Antar Muka Data Warna

3.2.6 Perancangan pesan

Rancangan pesan adalah rancangan antar muka dari pesan yang akan ditampilkan apabila terjadi kesalahan dalam proses yang dilakukan. Rancangan pesan dari sistem yang akan dibangun adalah sebagai berikut:

1) Rancangan pesan koneksi tidak dapat dilakukan

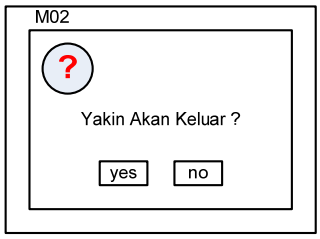
Pesan ini muncul saat *user* melakukan *login database* namun data yang dimasukan salah atau ada masalah dengan komputer *server*.



Gambar 3.44 pesan permintaan user tidak dapat dilakukan

2) Rancangan pesan keluar dari program

Pesan ini muncul ketika *user* akan keluar dari aplikasi



Gambar 3.45 pesan keluar dari program

3) Rancangan pesan nama dan *password* untuk *login* masuk

Pesan ini muncul ketika *user* melakukan *login berhasil*.



Gambar 3.46 pesan nama dan *password* untuk *login*

4) Rancangan pesan untuk mengisi kata kunci saat melakukan pencarian

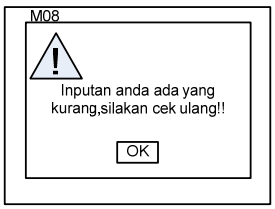
Pesan ini muncul ketika *user* melakukan pencarian atau penyaringan data tanpa memasukan kata kunci.



Gambar 3.47 pesan untuk mengisi kata kunci saat melakukan pencarian

5) Rancangan pesan untuk mengisi semua data

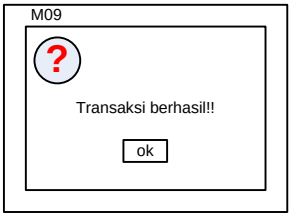
Pesan ini muncul ketika *user* tidak mengisi semua data yang harus diisi.



Gambar 3.48 pesan untuk mengisi semua data

6) Rancangan pesan data berhasil disimpan

Pesan ini muncul ketika data transaksi berhasil disimpan



Gambar 3.49 pesan data berhasil disimpan

7) Rancangan pesan permintaan *user* tidak dapat dilakukan

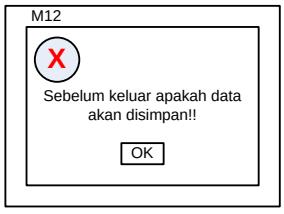
Pesan ini muncul ketika perintah ataupun proses transaksi yang dilakukan *user* tidak bisa dilakukan oleh aplikasi



Gambar 3.50 pesan permintaan *user* tidak dapat dilakukan

8) Rancangan pesan untuk data yang akan diubah

Pesan ini muncul ketika *user mengisi data yang* diubah .



Gambar 3.51 pesan untuk mengisi data yang akan diubah

9) Rancangan pesan no faktur yang tidak ditemukan

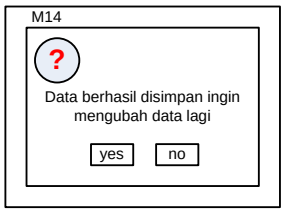
Pesan ini muncul ketika no faktur yang dimasukan *user* tidak ditemukan dalam *database*.



Gambar 3.52 pesan no faktur tidak ditemukan

10) Rancangan pesan data berhasil diubah

Pesan ini muncul ketika data berhasil diubah



Gambar 3.53 pesan data berhasil diubah

11) Rancangan pesan *stock* barang untuk distribusi kurang

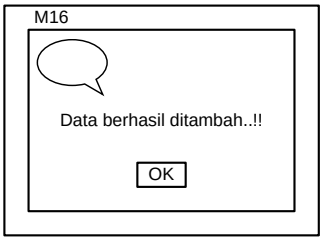
Pesan ini muncul ketika *user* memasukan jumlah barang yang akan didistribusikan lebih besar dari *stock* yang ada.



Gambar 3.54 pesan *stock* barang untuk distribusi kurang

12) Rancangan pesan data berhasil ditambah

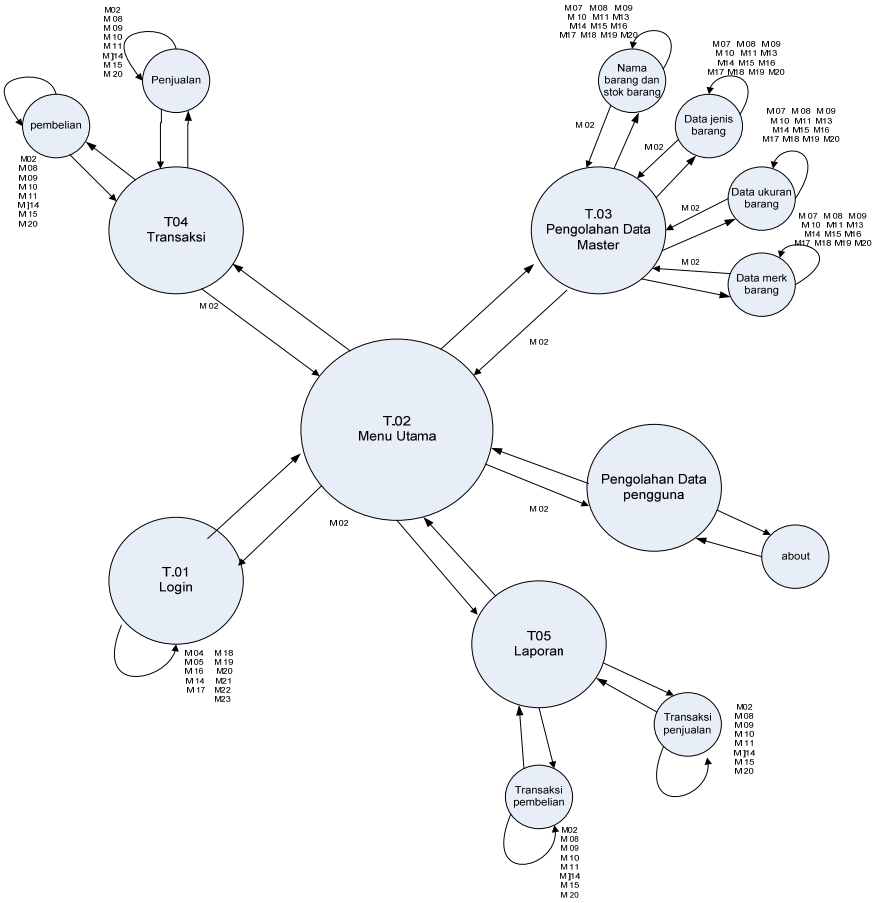
Pesan ini muncul ketika data berhasil disimpan.



Gambar 3.55 pesan data berhasil ditambah

3.2.7 Jaringan Semantik

Setelah melakukan perancangan antar muka dan perancangan pesan maka dilakukanlah perancangan terhadap hubungan antar *form* dan pesan. Untuk menggambarkan hubungannya digambarkan dalam sebuah jaringan semantik.



Gambar 3.56 jaringan semantik

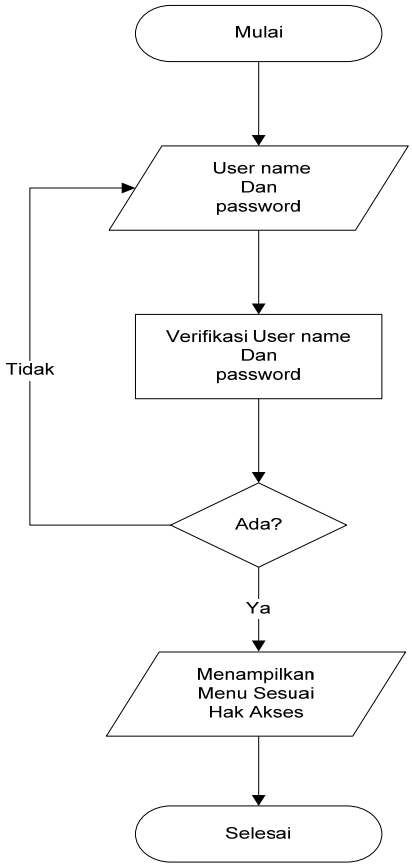
3.2.8 Perancangan Prosedural

Sebagai langkah terakhir dalam perancangan yaitu perancangan prosedural yang akan diimplementasikan ke dalam sistem. Prosedural ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada.

Adapun perancangan prosedural sistem informasi penjualan dan pembelianadalah sebagai berikut :

1. Prosedur *login*

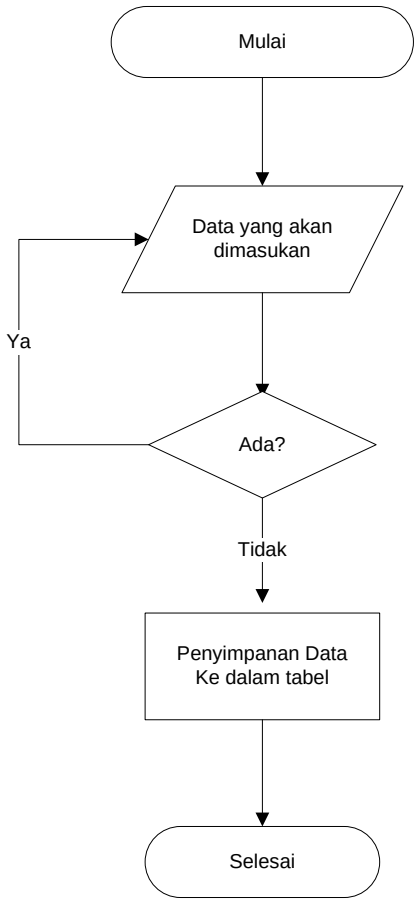
Diagram alir pada prosedur *login* adalah sebagai berikut:



Gambar 3.57 Prosedur *login*

2. Prosedur tambah data

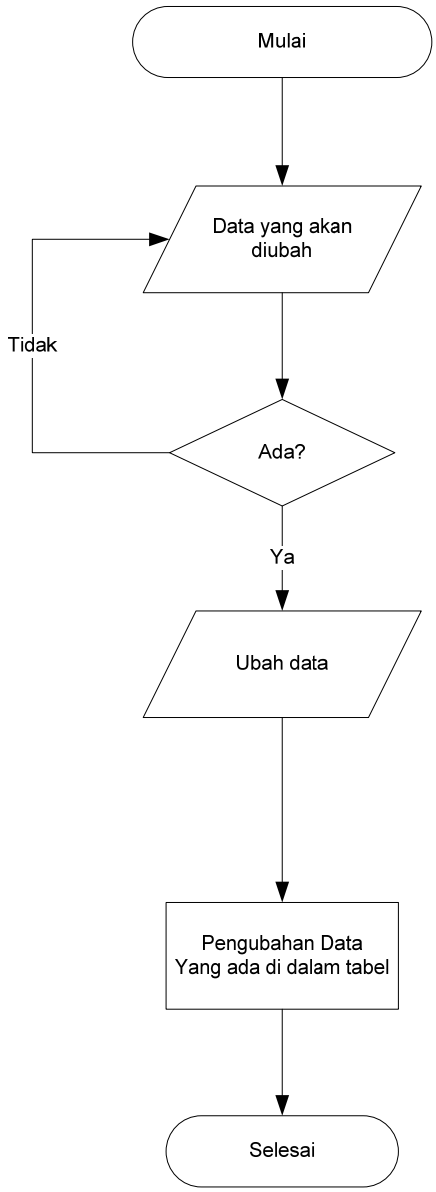
Diagram alir pada prosedur tambah data adalah sebagai berikut:



Gambar 3.58 Prosedur tambah data

3. Prosedur ubah data

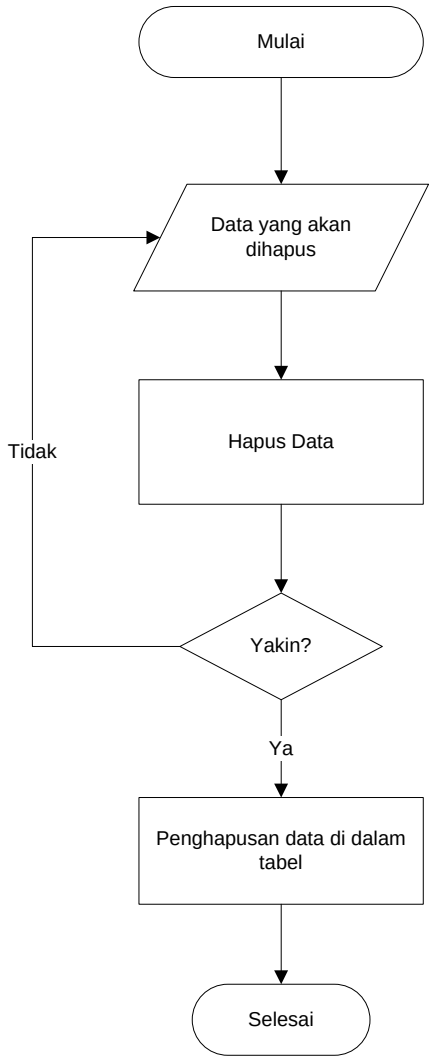
Diagram alir pada prosedur ubah data adalah sebagai berikut:



Gambar 3.59 Prosedur ubah data

4. Prosedur hapus data

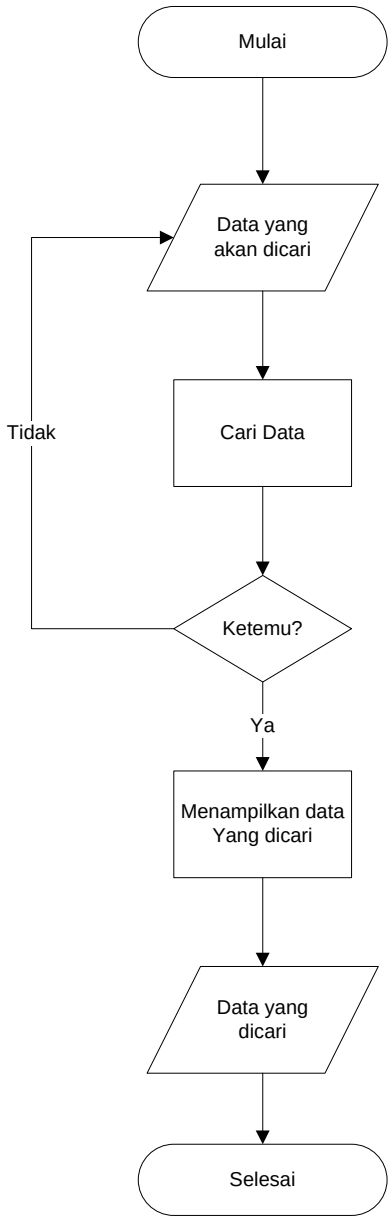
Diagram alir pada prosedur hapus data adalah sebagai berikut:



Gambar 3.60 Prosedur hapus data

5. Prosedur cari data

Diagram alir pada prosedur cari data adalah sebagai berikut:



Gambar 3.61 Prosedur cari data

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Tahap perancangan dan implementasi sudah dilewati, untuk menerjemahkan kebutuhan dari aplikasi yang telah dirancang kedalam representasi perangkat lunak nantinya agar sesuai dengan kegiatan analisis yang telah dilakukan sebelumnya. Dalam implementasi ini meliputi implementasi prosedur dan data, dan implementasi perangkat lunak, sedangkan untuk pengujian akan meliputi pengujian pada perangkat lunak yang telah dimplementasikan sebelumnya.

4.1 Implementasi

Pada tahap ini hasil dari tahap-tahap analisis dan perancangan yang telah dilakukan akan dituangkan kedalam sebuah program. Program ini adalah Penjualan dan pembelian bahan bangunan di Pd. Pembengunan Raya. Menu yang ada pada program ini akan ditampilkan seluruhnya pada tahap implementasi ini.

4.1.1 Analisis Software

Analisis yang telah dilakukan dalam kebutuhan Software untuk pengembangan dan menjalankan yang disarankan agar aplikasi ini dapat berjalan dengan baik, adalah:

1. Software yang ada saat ini adalah:
    a. Windows XP SP I
    b. Microsoft Office

2. Software yang dibutuhkan untuk menjalankan aplikasi ini adalah:

   a. Windows XP SP I

   b. DBMS menggunakan mysql

   c. Borland Delphi versi 6.0

Jadi setelah dilakukan analisis terhadap software, saat ini masih dirasa kurang untuk menjalankan aplikasi ini nantinya. Software yang harus ditambahkan adalah mysql yang nantinya akan digunakan sebagai DBMS serta Borland Delphi versi 6.

4.1.2 Perangkat Keras Pendukung

Selain perangkat lunak (*Software*) juga dibutuhkan keberadaan perangkat keras (*Hardware*), kebutuhan minimum yang diperlukan agar program ini dapat berjalan dengan baik, dengan spesifikasi sebagai berikut:

   a. Processor 933Mhz.

   b. Hardisk 40 Gb.

   c. Ram 512 Mb.

   d. CD – RW.

   e. VGA 32 Mb.

   f. Printer

4.1.3 Implementasi Data Base dan Antar Muka

Dalam pembuatan sistem Pengolahan Tagihan Piutang Karyawan Pada Rumah Sakit Islam Jakarta ini menggunakan Borland Delphi 7.0 sebagai aplikasi pembangun dan Microsoft Office Access sebagai DBMS

Tabel 4.1 Nama tabel beserta *file* implentasinya

No.	Nama tabel	Nama *file*
1	Pengguna	dtpengguna.sql
2	supplier	dtsupplier.sql
3	konsumen	dtkonsumen.sql
4	Pembelian	Dtpembelian.sql
5	Detail pembelian	Dtdetailpembelian.sql
6	Barang	Dtbarang.sql
7	Penjualan	Dtpenjualan.sql
8	Detail penjualan	Dtdetailpenjualan.sql
9	Ukuran	Dtukuran.sql
10	Jenis	Dtjenis.sql
11	Merk	Dtmerk.sql
12	Ukuran jenis	Dtdetailpenjualan.sql
13	Bahan jenis	Dtukuran.sql
14	Bentuk jenis	Dtjenis.sql
15	Merk jenis	Dtmerk.sql

4.1.3.1 Implementasi Database

Langkah pertama dalam pembuatan aplikasi ini yaitu pembuatan databasenya berikut adalah tabel - tabel yang dibentuk :

1. Tabel barang

Tabel ini berfungi untuk menyimpan data barang yang dimiliki oleh Pd. Pembangunan Raya.

```
CREATE TABLE `t_barang` (  `kd_barang` char(8) NOT NULL default '0',
`kd_merk` char(6) default NULL,
`kd_ukuran` char(6) default NULL,
`kd_jenis` char(8) default NULL,
`nama_barang` char(30) default NULL,
`warna` char(20) default NULL,
`bentuk` char(20) default NULL,
`bahan` char(15) default NULL,
`stok` int(3) default '0',
`harga_barang` int(3) unsigned default '0',
PRIMARY KEY  (`kd_barang`),
UNIQUE KEY `kd_barang` (`kd_barang`)) TYPE=MyISAM;
 );
```

2. tabel detail pembelian

Tabel ini berfungi untuk menyimpan data detail pembelian yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_detail_pemb` (`no_faktur` char(10) default '0',
`kd_barang` char(8) default '0',
`jml_barang` int(3) unsigned default '0',
`harga` int(3) unsigned default '0',
`total_bayar` int(3) unsigned default '0') TYPE=MyISAM;
);
```

3. tabel detail penjualan

Tabel ini berfungi untuk menyimpan data detail penjualan yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_detail_penj` (`no_faktur` char(10) default '0',
`kd_barang` char(8) default '0',
`jml_barang` int(3) unsigned default '0',
`harga` int(3) unsigned default '0',
`total_bayar` int(3) unsigned default '0') TYPE=MyISAM;
);
```

4. tabel jenis

Tabel ini berfungi untuk menyimpan data jenis yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_jenis` ( `kd_jenis` char(15) NOT NULL default '0',
  `nama_jenis` char(30) default '0',
   PRIMARY KEY  (`kd_jenis`),
   UNIQUE KEY `kd_jenis` (`kd_jenis`)
) TYPE=MyISAM;
);
```

5. tabel konsumen

Tabel ini berfungi untuk menyimpan data konsumen yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_konsumen` (`kode_konsumen` char(6) NOT NULL default '0',
  `nama` char(30) default '0',
  `alamat` char(30) default '0',
  `no_telp` char(15) default '0',
  PRIMARY KEY  (`kode_konsumen`),
  UNIQUE KEY `kode_konsumen` (`kode_konsumen`)) TYPE=MyISAM;
);
```

6. tabel merk

Tabel ini berfungi untuk menyimpan data merk yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_merk` (`kd_merk` char(15) NOT NULL default '0',
  `merk` char(30) default '0',
   PRIMARY KEY  (`kd_merk`),
   UNIQUE KEY `kd_merk` (`kd_merk`)) TYPE=MyISAM;
);
```

7. tabel pembelian

Tabel ini berfungi untuk menyimpan data detail pembelian yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_pembelian` ( `no_faktur` char(10) NOT NULL default '0',
  `tgl_pemb` char(15) default '0',
  `kode_supplier` char(6) default '0',
  `total_bayar` int(3) unsigned default '0',
  `nama_pengguna` char(30) default '0',
  PRIMARY KEY  (`no_faktur`),
  UNIQUE KEY `no_faktur` (`no_faktur`)) TYPE=MyISAM;
);
```

8. tabel pengguna

Tabel ini berfungi untuk menyimpan data pengguna yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_pengguna` (`nama_pengguna` char(30) NOT NULL default '0',
  `password` char(10) default '0',
  `jenis_pengguna` char(20) default '0',
  PRIMARY KEY  (`nama_pengguna`),
  UNIQUE KEY `nama_pengguna` (`nama_pengguna`)) TYPE=MyISAM;
);
```

9. tabel penjualan

Tabel ini berfungi untuk menyimpan data penjualan yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_penjualan` (
  `no_faktur` char(10) NOT NULL default '0',
  `tgl_penj` char(15) default '0',
  `kode_konsumen` char(15) default '0',
  `total_bayar` int(3) unsigned default '0',
  `nama_pengguna` char(30) default '0',
   PRIMARY KEY  (`no_faktur`),
   UNIQUE KEY `no_faktur` (`no_faktur`)
```

```
) TYPE=MyISAM;
);
```

10. tabel supplier

Tabel ini berfungi untuk menyimpan data supplier yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_supplier` (
  `kode_supplier` char(6) NOT NULL default '0',
  `nama_supplier` char(30) default '0',
  `no_telp` char(15) default '0',
  `alamat` char(30) default '0',
  `kontak_person` char(30) default '0',
   PRIMARY KEY  (`kode_supplier`),
   UNIQUE KEY `kode_supplier` (`kode_supplier`)
) TYPE=MyISAM;
);
```

11. tabel ukuran

Tabel ini berfungi untuk menyimpan data ukuran yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_ukuran` (
  `kd_ukuran` char(15) NOT NULL default '0',
  `ukuran` char(15) default '0',
   PRIMARY KEY  (`kd_ukuran`),
   UNIQUE KEY `kd_ukuran` (`kd_ukuran`)
) TYPE=MyISAM;
);
```

12. tabel bahan

Tabel ini berfungi untuk menyimpan data bahan yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_bahan` (
  `kd_bahan` char(15) NOT NULL default '0',
  `bahan` char(15) default '0',
   PRIMARY KEY  (`kd_bahan`),
   UNIQUE KEY `kd_bahan` (`kd_bahan`)
) TYPE=MyISAM;
);
```

13. tabel bentuk

Tabel ini berfungi untuk menyimpan data bentuk yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_bentuk` (
  `kd_bentuk` char(15) NOT NULL default '0',
  `bentuk` char(15) default '0',
   PRIMARY KEY  (`kd_bentuk`),
   UNIQUE KEY `kd_bentuk` (`kd_bentuk`)
) TYPE=MyISAM;
);
```

14. tabel warna

Tabel ini berfungi untuk menyimpan data warna yang dimiliki Pd. Pembangunan Raya.

```
CREATE TABLE `t_warna` (
  `kd_warna` char(15) NOT NULL default '0',
  `warna` char(15) default '0',
   PRIMARY KEY  (`kd_warna`),
   UNIQUE KEY `kd_warna` (`kd_warna`)
) TYPE=MyISAM;
);
```

4.1.3.2 Implementasi Antar Muka

Setelah melakukan implementasi data, maka dilakukanlah implemntasi antar muka yang berguna sebagai tampilan untuk pengguna aplikasi ini nantinya. Implementasi antar muka dilakukan pada setiap hasil perancangan antar muka yang telah dibuat ke dalam bentuk aplikasi pada kompiler Delphi 6. Implementasi dalam pengolahan tagihan piutang karyawan ihanya memiliki satu bagian, yaitu:

- Implementasi antar muka bagi user di bagian penagihan

Untuk mengetahui lebih jelas tentang proses yang ada di dalam sistem beserta hasil implementasinya dapat dilihat dalam tabel di bawah ini:

Tabel 4.2 Implementasi antar muka user

Menu	Proses Ke-	Nama File
Menu Login: Dlogin	1	login.dcu
		login.ddp
		login.dfm
		login.pas
Pengolahan Data ukuran: dtambahukuran	2.1.1	tambahukuran.dcu
		tambahukuran.ddp
		tambahukuran.dfm
		tambahukuran.pas
Pengolahan Data ukuran: dubahukuran	2.1.2	ukuran.dcu
		ukuran.ddp
		ukuran.dfm
		ukuran.pas

Menu	Proses Ke-	Nama File
Pengolahan Data ukuran: dhapusukuran	2.1.3	hapusukuran.dcu
		hapusukuran.ddp
		hapusukuran.dfm
		hapusukuran.pas
Pengolahan Data jenis: Dtambahjenis	2.2.1	tambahjenis.dcu
		tambahjenis.ddp
		tambahjenis.dfm
		tambahjenis.pas
Pengolahan Data jenis: dubahjenis	2.2.2	ubahjenis.dcu
		ubahjenis.ddp
		ubahjenis.dfm
		ubahjenis.pas
Pengolahan Data jenis: dhapusjenis	2.2.3	hapusjenis.dcu
		hapusjenis.ddp
		hapusjenis.dfm
		hapusjenis.pas
Pengolahan Data merk: dtambahmerk	2.3.1	tambahmerk.dcu
		tambahmerk.ddp
		tambahmerk.dfm
		tambahmerk.pas
Pengolahan Data merk: dubahmerk	2.3.2	ubahmerk.dcu
		ubahmerk.ddp
		ubahmerk.dfm
		ubahmerk.pas
Pengolahan Data merk: dhapusmerk	2.3.3	hapusmerk.dcu
		hapusmerk.ddp

Menu	Proses Ke-	Nama File
		hapusmerk.dfm
		hapusmerk.pas
Pengolahan Data barang: dtambahbarang	2.4.1	tambahbarang.dcu
		tambahbarang.ddp
		tambahbarang.dfm
		tambahbarang.pas
Pengolahan Data barang: dubahbarang	2.4.2	ubahbarang.dcu
		ubahbarang.ddp
		ubahbarang.dfm
		ubahbarang.pas
Pengolahan Data barang: dhapusbarang	2.4.3	hapusbarang.dcu
		hapusbarang.ddp
		hapusbarang.dfm
		hapusbarang.pas
Pengolahan Data supplier: dtambahsupplier	2.5.1	tambahsupplier.dcu
		tambahsupplier.ddp
		tambahsupplier.dfm
		tambahsupplier.pas
Pengolahan Data supplier: dubahsupplier	2.5.2	ubahsupplier.dcu
		ubahsupplier.ddp
		ubahsupplier.dfm
		ubahsupplier.pas
Pengolahan Data supplier: dhapussupplier	2.5.3	hapussupplier.dcu
		hapussupplier.ddp
		hapussupplier.dfm
		hapussupplier.pas

Menu	Proses Ke-	Nama File
Pengolahan Data konsumen: dtambahkonsumen	2.6.1	tambahkonsumen.dcu
		tambahkonsumen.ddp
		tambahkonsumen.dfm
		tambahkonsumen.pas
Pengolahan Data konsumen: dubahkonsumen	2.6.2	ubahkonsumen.dcu
		ubahkonsumen.ddp
		ubahkonsumen.dfm
		ubahkonsumen.pas
Pengolahan Data konsumen: dhapuskonsumen	2.6.3	hapuskonsumen.dcu
		hapuskonsumen.ddp
		hapuskonsumen.dfm
		hapuskonsumen.pas
Pengolahan Data bahan: dtambahbahan	2.7.1	tambahbahan.dcu
		tambahbahan.ddp
		tambahbahan.dfm
		tambahbahan.pas
Pengolahan Data bahan: dubahkonsumen	2.7.2	ubahbahan.dcu
		ubahbahan.ddp
		ubahbahan.dfm
		ubahbahan.pas
Pengolahan Data bahan: dhapuskonsumen	2.7.3	hapusbahan.dcu
		hapusbahan.ddp
		hapusbahan.dfm
		hapusbahan.pas
Pengolahan Data bentuk: dtambahbentuk	2.8.1	tambah bentuk.dcu
		tambah bentuk.ddp

Menu	Proses Ke-	Nama File
		tambah bentuk.dfm
		tambah bentuk.pas
Pengolahan Data bentuk: dubahbentuk	2.8.2	ubah bentuk.dcu
		ubah bentuk.ddp
		ubahbahan.dfm
		ubah bentuk.pas
Pengolahan Data bentuk: dhapusbentuk	2.8.3	hapus bentuk.dcu
		hapus bentuk.ddp
		hapus bentuk.dfm
		hapus bentuk.pas
Pengolahan Data warna: dtambah warna	2.9.1	tambah warna.dcu
		tambah warna.ddp
		tambah warna.dfm
		tambah warna.pas
Pengolahan Data warna: dubah warna	2.9.2	ubah warna.dcu
		ubah warna.ddp
		ubah warna.dfm
		ubah warna.pas
Pengolahan Data warna: dhapus warna	2.9.3	hapus warna.dcu
		hapus warna.ddp
		hapus warna.dfm
		hapus warna.pas
Pengolahan Data master pembelian: dmasterpembelian	3.1.1	masterpembelian.dcu
		masterpembelian.ddp
		masterpembelian.dfm
		masterpembelian.pas

Menu	Proses Ke-	Nama File
Pengolahan Data detail pembelian: ddetailpembelian	3.1.2	detailpembelian.dcu
		detailpembelian.ddp
		detailpembelian.dfm
		detailpembelian.pas
Pengolahan Data cetak faktur:	3.1.3	cetak faktur:.dcu
		cetak faktur:.ddp
		cetak faktur:.dfm
		cetak faktur:.pas
Pengolahan Data master penjualan: dpenjualan	3.2.1	penjualan.dcu
		penjualan.ddp
		penjualan.dfm
		penjualan.pas
Pengolahan Data detail penjualan: ddetailpenjualan	3.2.2	detailpenjualan.dcu
		detailpenjualan.ddp
		detailpenjualan.dfm
		detailpenjualan.pas
Pengolahan Data cetak faktur penjualan:	3.2.3	cetak faktur penjualan:.dcu
		cetak faktur penjualan:.ddp
		cetak faktur penjualan:.dfm
		cetak faktur penjualan:.pas
Pengolahan Data pengguna: dtambahpengguna	4.1	tambahpengguna.dcu
		tambahpengguna.ddp
		tambahpengguna.dfm
		tambahpengguna.pas
Pengolahan Data pengguna:	4.2	pengguna.dcu
		pengguna.ddp

Menu	Proses Ke-	Nama File
dubahpengguna		pengguna.dfm
		pengguna.pas
Pengolahan Data pengguna: dhapuspengguna	4.3	hapuspengguna.dcu
		hapuspengguna.ddp
		hapuspengguna.dfm
		hapuspengguna.pas
Pengolahan Data pengguna: dcaripengguna	4.4	pengguna:.dcu
		pengguna:.ddp
		pengguna:.dfm
		pengguna:.pas
Pengolahan Data laporan penjualan: dlaporanpenjualan	5.1	laporanpenjualan.dcu
		laporanpenjualan.ddp
		laporanpenjualan.dfm
		laporanpenjualan.pas
Pengolahan Data laporan pembelian: dlaporanpembelian	5.2	laporanpembelian:.dcu
		laporanpembelian:.ddp
		laporanpembelian:.dfm
		laporanpembelian:.pas

Implementasi sistem dimaksudkan untuk menjelaskan secara singkat hasil transformasi dari hasil perancangan kepada program aplikasi.

4.2 Pengujian Alpha

Sebuah aplikasi yang akan diterapkan di instansi haruslah bebas dari kesalahan-kesalahan atau *error*. Oleh karena itu, aplikasi harus diuji coba terlebih

dahulu untuk menemukan kesalahan-kesalahan yang mungkin terjadi, pengujian ini menggunakan metoda pengujian *black box.*

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Pengujian ini berusaha menemukan kesalahan antara lain :

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data
4. Kesalahan kinerja

4.2.1 Pengujian Sistem

Pengujian sistem ini menggunakan metode pengujian black box. Pengujian ini berfokus pada persyaratan fungsional perangkat lunak.

Pengujian perangkat lunak system informasi menggunakan data uji berdasarkan form yang terdapat dalam aplikasi dari sistem.

Tabel 4.3 Rencana Pengujian

Item Uji	Detail Pengujian	Jenis Pengujian
Login	Verifikasi login	*Black box*
Pengolahan data pengguna	Tambah, ubah, dan hapus data pengguna	*Black box*
Pengolahan data supplier	Tambah, ubah, hapus data supplier	*Black box*
Pengolahan data konsumen	Tambah, ubah, hapus konsumen	*Black box*

Pengolahan data Pembelian	Tambah, ubah, cetak data pembelian	*Black box*
Pengolahan data detail pembelian	Cari dan cetak pembelian data pembelian	*Black box*
Pengolahan data barang	Tambah, ubah, hapus barang	*Black box*
Pengolahan data penjualan	Baru, hapus dan hitung	*Black box*
Pengolahan data detail penjualan	Cari dan cetak data penjualan	*Black box*
Pengolahan data ukuran	Tambah, ubah, hapus ukuran	*Black box*
Pengolahan data jenis	Tambah, ubah, hapus ukuran	*Black box*
Pengolahan data merk	Tambah, ubah, hapus ukuran	*Black box*
Pengolahan data bahan	Tambah, ubah, hapus ukuran	*Black box*
Pengolahan data bentuk	Tambah, ubah, hapus ukuran	*Black box*
Pengolahan data warna	Tambah, ubah, hapus ukuran	*Black box*

4.2.1.1 Pengujian Login

Pengujian login terbagi atas dua bagian diantaranya verifikasi nama *user* dan verifikasi *password*.

Tabel 4.4 Pengujian Login

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama user :	Dapat terisi pada	Dapat mengisi	[√] diterima

gugun Password : sukmana	edit box nama *user* dan edit box *password*	nama *user* dan *password* sesuai yang diharapakan	[ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama user : Tessa	Tidak dapat login dan menampilkan pesan "USER ID dan password ada yang salah"	User tidak dapat login dan mendapatkan pesan "USER ID dan password ada yang salah" sesuai yang diharapkan	[√] diterima [ ] ditolak
password : Redite taufana	Tidak dapat login dan menampilkan pesan "USER ID dan password ada yang salah"	User tidak dapat login dan mendapatkan pesan "USER ID dan password ada yang salah" sesuai yang diharapkan	[√] diterima [ ] ditolak
Data nama *user* dan *password*	Tidak dapat login dan menampilkan	User tidak dapat login dan	[√] diterima [ ] ditolak

salah atau kosong	pesan "USER ID dan password ada yang salah"	mendapatkan pesan "USER ID dan password ada yang salah" sesuai yang diharapkan	

4.2.1.2 Pengujian Pengolahan Data pengguna

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data pengguna.

Untuk melakukan pengubahan klik tombol tambah data pengguna maka akan tampil form tambah data pengguna.

a. Tambah Data pengguna

Tabel 4.5 Tambah Data pengguna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : gugun password : sukmana Pengguna : Bagian counter	Data pada nama, password dan pengguna dapat terisi	Dapat mengisi nama, password dan pengguna sesuai dengan yang diharapakan	[√] diterima [ ] ditolak

Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : Tessa Password : Redite taufana pengguna : bagian counter	Data tidak dapat disimpan pada tabel pengguna dan menampilkan pesan " Kode bagian Belum terisi"	Penambahan data gagal dan menampilkan pesan bahwa " kode bagian belum terisi"	[√] diterima [ ] ditolak

b. Pengubahan Data pengguna

Untuk melakukan pengubahan klik tombol ubah data pengguna maka akan tampil form ubah data pengguna.

Tabel 4. 6 Pengubahan Data pengguna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : galih password : dwiutama Pengguna :	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak

Bagian counter	dan Data yang lama akan terganti dengan data baru.	maka data yang lama akan terganti dengan yang baru	
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama, password, data pengguna : kosong atau salah	Data tidak dapat disimpan pada tabel pengguna dan menampilkan pesan " user Id atau password salah"	Penambahan data gagal dan menampilkan pesan bahwa " data belum diisi lengkap"	[√] diterima [ ] ditolak

c. Hapus Data pengguna

Untuk melakukan pengubahan klik tombol hapus data pengguna maka akan tampil form hapus data pengguna.

Tabel 4.7 Hapus data pengguna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : tessa password : redit	Menampilkan pesan	Memberikan pesan "Anda	[√] diterima [ ] ditolak

Pengguna : Bagian counter	“Anda yakin data akan dihapus?” jika ya maka data akan dihapus.	yakin data akan dihapus?” jika ya maka data dihapus. Sesuai yang diinginkan.	
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama, password : Pengguna : Kosong	Saat menekan tombol batal data bagian tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.3 Pengujian Pengolahan Data supplier

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data supplier.

Untuk melakukan pengubahan klik tombol tambah data supplier maka akan tampil form tambah data supplier.

a. Tambah Data supplier

Tabel 4.8 Tambah Data supplier

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode supplier : KS0001 nama supplier : PT.dwiutama no.telp : 022743245 alamat : Jln tubagus ismail no.9 kontak person : 0222505432	Data pada Kode supplier, nama supplier, no.telp, alamat, kontak person dapat terisi	Dapat mengisi Kode supplier, nama supplier, no.telp, alamat, kontak person sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode, nama, no.telp, alamat, kontak person supplier :	Data tidak dapat disimpan pada tabel supplier dan menampilkan	Penambahan data gagal dan menampilkan pesan “ data	[√] diterima [ ] ditolak

Salah atau kosong	pesan “ data supplier Belum lengkap terisi“	supplier Belum lengkap terisi“	

`

b. Pengubahan Data supplier

Untuk melakukan pengubahan klik tombol ubah data supplier maka akan tampil form ubah data supplier.

Tabel 4. 9 Pengubahan data supplier

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode, nama, no.telp, alamat, kontak person supplier	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak
Masukan data yang akan diubah	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**

Kode, nama, no.telp, alamat, kontak person supplier : Salah atau kosong	Data tidak dapat disimpan pada tabel supplier dan menampilkan pesan " data supplier Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data supplier Belum lengkap terisi"	[√] diterima [ ] ditolak

c. Hapus Data supplier

Untuk melakukan pengubahan klik tombol hapus data pengguna maka akan tampil form ubah data konsumen.

Tabel 4.10 Hapus data supplier

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode :KS0002 nama : rizky no.telp : 0228766549 alamat : jln dipati ukur kontak person supplier :	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak

Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode, nama, no.telp, alamat, kontak person supplier : Salah atau kosong	Saat menekan tombol batal data bagian tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.4 Pengujian Pengolahan Data konsumen

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data konsumen.

Untuk melakukan pengubahan klik tombol tambah data supplier maka akan tampil form tambah data konsumen.

a. Tambah Data konsumen

Tabel 4.11 Tambah Data konsumen

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode konsumen : KS0001 nama : ahmad wildan no.telp :	Data pada Kode konsumen, nama no.telp dan alamat dapat terisi	Dapat mengisi Kode konsumen, nama, no.telp dan alamat,	[√] diterima [ ] ditolak

02276543245 alamat : Jln dipati ukur no.145		sesuai dengan yang diharapakan	
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode konsumen, nama , no.telp, dan alamat : salah atau Kosong	Data tidak dapat disimpan pada tabel konsumen dan menampilkan pesan " data konsumen Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data konsumen Belum lengkap terisi"	[√] diterima [ ] ditolak

b. Pengubahan Data konsumen

Untuk melakukan pengubahan klik tombol ubah data konsumen maka akan tampil form ubah data konsumen.

Tabel 4. 12 Pengubahan Data konsumen

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode konsumen : KS0001 nama :	Data dapat diinputkan, yang	Data yang diubah tampil sesuai	[√] diterima [ ] ditolak

ahmad wildan no.telp : 02276543245 alamat : Jln dipati ukur no.145	tertera pada dbgrid keluar pada form pengubahan data	dengan editbox yang tersedia	
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode konsumen : KS0001 nama : ahmad wildan no.telp : 02276543245 alamat : Jln dipati ukur no.145 : data salah atau kosong	Saat menekan tombol batal data konsumen tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.5 Pengujian Pengolahan Data Pembelian

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data pembelian.

Untuk melakukan pengubahan klik tombol tambah data pembelian maka akan tampil form tambah data pembelian.

a. Tambah Data pembelian

Tabel 4.14 Tambah Data pembelian

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur : FB090709001 tgl pembelian : 090709 kode supplier : KS0001 total bayar : Rp250000 user name : gugun	Data pada No. Faktur, tgl pembelian, kode supplier,total bayar, user name dapat terisi	Dapat mengisi No. Faktur, tgl pembelian, kode supplier,total bayar, user name sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**

No. Faktur, tgl pembelian, kode supplier,total bayar, user name Salah atau Kosong	Data tidak dapat disimpan pada tabel pembelian dan menampilkan pesan " data pembelian Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data konsumen Belum lengkap terisi"	[√] diterima [ ] ditolak

b. Pengubahan Data pembelian

Untuk melakukan pengubahan klik tombol ubah data pembelian maka akan tampil form ubah data pembelian.

Tabel 4. 15 Pengubahan Data pembelian

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data yang akan diubah No. Faktur : FB090709007 tgl pembelian : 090709 kode supplier : KS0008	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak

total bayar : Rp200000 user name : ahmad			
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, tgl pembelian, kode supplier,total bayar, user name Salah atau Kosong	Data tidak dapat disimpan pada tabel pembelian dan menampilkan pesan " data pembelian Belum lengkap terisi"	Pengubahan data gagal dan menampilkan pesan " data konsumen Belum lengkap terisi"	[√] diterima [ ] ditolak

c. Hapus Data pembelian

Untuk melakukan pengubahan klik tombol hapus data pembelian maka akan tampil form hapus data pembelian.

Tabel 4.16 Hapus data pembelian

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data yang akan dihapus No. Faktur : FB090709007 tgl pembelian : 090709 kode supplier : KS0008 total bayar : Rp200000 user name : ahmad	Menampilkan pesan “Anda yakin data akan dihapus?” jika ya maka data akan dihapus.	Memberikan pesan “Anda yakin data akan dihapus?” jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data yang akan di hapus No. Faktur : FB090709010	Saat menekan tombol batal data pembelian tidak	Tombol batal bekerja sesuai dengan yang	[√] diterima [ ] ditolak

tgl pembelian : 090709 kode supplier : KS0009 total bayar : Rp2500000 user name : ahmad data bagian yang tidak akan dihapus data salah atau kosong	akan terhapus	diharapkan	

4.2.1.6 Pengujian Pengolahan Data detail pembelian

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data detail pembelian.

Untuk melakukan pengubahan klik tombol tambah data detail pembelian maka akan tampil form tambah data detail pembelian.

a. cari Data detail pembelian

Tabel 4.17 Tambah Data detail pembelian

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Cari No. Faktur, kode barang, jumlah, harga dan total bayar	Data yang diharapkan No. Faktur, kode barang, jumlah, harga dan total bayar dapat diisi	Dapat ditemukan No. Faktur, kode barang, jumlah, harga dan total bayar sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
cari Kode, nama, no.telp, alamat, kontak person supplier : salah atau kosong	Data tidak dapat disimpan pada tabel detail pembelian dan menampilkan pesan " data detail pembelian Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data detail pembelian Belum lengkap terisi"	[√] diterima [ ] ditolak

b. cetak Data detail pembelian

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Cari No. Faktur, kode barang, jumlah, harga dan total bayar	Data yang diharapkan No. Faktur, kode barang, jumlah, harga dan total bayar dapat dicetak	Data masukan dapat dicetak secara keseluruhan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Cari No. Faktur, kode barang, jumlah, harga dan total bayar : Data Salah atau kosong	Data yang diharapkan No. Faktur, kode barang, jumlah, harga dan total bayar tidak dapat dicetak	Data masukan tidak dapat dicetak secara keseluruhan	[√] diterima [ ] ditolak

4.2.1.7 Pengujian Pengolahan Data barang

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data barang.

Untuk melakukan pengubahan klik tombol tambah data barang maka akan tampil form tambah data barang.

a. Tambah Data barang

Tabel 4.18 Tambah Data barang

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan	Data pada Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan dapat terisi	Dapat mengisi Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**

Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan : Salah atau Kosong	Data tidak dapat disimpan pada tabel barang dan menampilkan pesan "Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan Belum terisi"	Penambahan data gagal dan menampilkan pesan bahwa "Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan belum terisi"	[√] diterima [ ] ditolak

b. Pengubahan Data barang

Untuk melakukan pengubahan klik tombol ubah data barang maka akan tampil form ubah data barang.

Tabel 4. 19 Pengubahan Data barang

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data yang akan diubah	Data yang lama akan terganti	Data yang lama terlihat sudah	[√] diterima [ ] ditolak

Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan :	dengan data baru.	terganti dengan yang baru	
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan : Salah atau kosong	Data tidak dapat disimpan pada tabel barang dan menampilkan pesan “ data belum diisi lengkap“	Penambahan data gagal dan menampilkan pesan bahwa “ data belum diisi lengkap”	[√] diterima [ ] ditolak

c. Hapus Data barang

Untuk melakukan pengubahan klik tombol hapus data barang maka akan tampil form ubah data barang.

Tabel 4.20 Hapus data pengguna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data yang akan dihapus Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan	Menampilkan pesan “Anda yakin data akan dihapus?” jika ya maka data akan dihapus.	Memberikan pesan “Anda yakin data akan dihapus?” jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data	Saat menekan	Tombol batal	[√] diterima

yang akan di hapus Kode barang, nama barang, warna, harga, stok, bentuk, kode ukuran, kode jenis, kode merk, bahan : Salah atau kosong	tombol batal data barang tidak akan terhapus	bekerja sesuai dengan yang diharapkan	[ ] ditolak

4.2.1.8 Pengujian Pengolahan Data penjualan

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data penjualan.

Untuk melakukan pengubahan klik tombol tambah data penjualan maka akan tampil form tambah data penjualan.

a. Tambah Data penjualan

Tabel 4.21 Tambah Data penjualan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur :	Data pada No.	Dapat mengisi No.	[√] diterima

FJ0809090004 tgl penjualan : 080909 kode konsumen : KK0004 total bayar : Rp.450000 user name : gugun	Faktur, tgl penjualan, kode konsumen, total bayar, user name dapat terisi	Faktur, tgl penjualan, kode konsumen, total bayar, user name sesuai dengan yang diharapakan	[ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, tgl penjualan, kode konsumen, total bayar, user name : Salah atau kosong	Data tidak dapat disimpan pada tabel penjualan dan menampilkan pesan " data penjualan Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data penjualan Belum lengkap terisi"	[√] diterima [ ] ditolak

b. Pengubahan Data penjualan

Untuk melakukan pengubahan klik tombol ubah data pejualan maka akan tampil form ubah data penjualan.

Tabel 4. 22 Pengubahan Data penjualan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data yang akan diubah No. Faktur : FJ0809090004 tgl penjualan : 080909 kode konsumen : KK0004 total bayar : Rp.350000 user name : risky	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data yang akan diubah	Data tidak dapat disimpan pada	Penambahan data gagal dan	[√] diterima [ ] ditolak

No. Faktur : tgl penjualan : kode konsumen : total bayar : user name : salah atau kosong	tabel penjualan dan menampilkan pesan " data penjualan Belum lengkap terisi"	menampilkan pesan " data penjualan Belum lengkap terisi"	

c. Hapus Data penjualan

Untuk melakukan pengubahan klik tombol hapus data penjualan maka akan tampil form hapus data penjualan.

Tabel 4.23 Hapus data penjualan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data yang akan dihapus No. Faktur : FJ0809090004 tgl penjualan : 080909 kode konsumen :	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak

KK0004 total bayar : Rp.350000 user name : risky			
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data yang akan dihapus No. Faktur : tgl penjualan : kode konsumen : total bayar : user name : salah atau kosong	Data tidak dapat disimpan pada tabel penjualan dan menampilkan pesan " data penjualan Belum lengkap terisi"	Penambahan data gagal dan menampilkan pesan " data penjualan Belum lengkap terisi"	[√] diterima [ ] ditolak

4.2.1.9 Pengujian Pengolahan Data detail penjualan

Pengujian data bagian ini terbagi kedalam 2 bagian yaitu proses pencarian data cetak detail penjualan.

Untuk melakukan pengubahan klik tombol cari data detail pejualan maka akan tampil form data detail penjualan.

a. cari Data detail penjualan

Tabel 4.24 cari Data detail penjualan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, kode barang, jumlah, nama barang harga dan total bayar	Data yang diharapkan No. Faktur, kode barang, jumlah, nama barang harga dan total bayar dapat ditemukan	Dapat ditemukan No. Faktur, kode barang, jumlah, nama barang harga dan total bayar sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, kode barang, jumlah, nama barang harga dan total bayar : Salah atau kosong	Data tidak dapat disimpan pada tabel detail penjualan dan menampilkan pesan “ data detail penjualan Belum lengkap terisi“	Pencarian data gagal dan menampilkan pesan “ data detail penjualan Belum lengkap terisi“	[√] diterima [ ] ditolak

b. cetak Data detail penjualan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, kode barang, jumlah, nama barang harga dan total bayar	Data yang diharapkan No. Faktur, kode barang, jumlah, nama barang harga dan total bayar dapat dicetak	Data masukan dapat dicetak secara keseluruhan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
No. Faktur, kode barang, jumlah, nama barang harga dan total bayar : Salah atau kosong	Data yang diharapkan No. Faktur, kode barang, jumlah, nama barang harga dan total bayar tidak dapat dicetak	Data masukan tidak dapat dicetak secara keseluruhan	[√] diterima [ ] ditolak

4.2.1.10 Pengujian Pengolahan Data ukuran

Pengujian data bagian ini terbagi kedalam 4 bagian yaitu proses penambahan, pengubahan ,penghapusan dan pencarian data ukuran.

Untuk melakukan pengubahan klik tombol tambah data ukuran maka akan tampil form tambah data ukuran.

a. Tambah Data ukuran

Tabel 4.25 Tambah Data ukuran

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode ukuran : KU0001 Ukuran : 24 * 36 cm	Data pada Kode ukuran dan ukuran dapat terisi	Dapat mengisi Kode ukuran dan ukuran sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode ukuran dan ukuran : Salah atau kosong	Data tidak dapat disimpan pada tabel ukuran dan menampilkan pesan “ data	Penambahan data gagal dan menampilkan pesan “ data ukuran Belum	[√] diterima [ ] ditolak

	ukuran Belum lengkap terisi“	lengkap terisi“	

b. Pengubahan Data ukuran

Untuk melakukan pengubahan klik tombol ubah data ukuran maka akan tampil form ubah data ukuran.

Tabel 4. 26 Pengubahan Data ukuran

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data ukuran : 45 * 67 cm	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak
Masukan data yang akan diubah	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak

c. Hapus Data ukuran

Untuk melakukan pengubahan klik tombol hapus data ukuran maka akan tampil form ubah data ukuran.

Tabel 4.27 Hapus data ukuran

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data ukuran : 34 * 40 cm	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
ukuran yang tidak akan dihapus : data salah atau kosong	Saat menekan tombol batal data ukuran tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.11 Pengujian Pengolahan Data jenis

Pengujian data bagian ini terbagi kedalam 3 bagian yaitu proses penambahan, pengubahan dan penghapusan data jenis.

Untuk melakukan pengubahan klik tombol tambah data jenis maka akan tampil form tambah data jenis.

a. Tambah Data jenis

Tabel 4.28 Tambah Data jenis

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode jenis : KD0002 Jenis : bahan bangunan	Data pada Kode jenis dan jenis dapat terisi	Dapat mengisi Kode jenis dan jenis sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode jenis dan jenis : Salah atau kosong	Data tidak dapat disimpan pada tabel jenis dan menampilkan pesan “ data jenis	Penambahan data gagal dan menampilkan pesan “ data jenis Belum lengkap	[√] diterima [ ] ditolak

	Belum lengkap terisi“	terisi“	

b. Pengubahan Data jenis

Untuk melakukan pengubahan klik tombol ubah data jenis maka akan tampil form ubah data jenis.

Tabel 4. 29 Pengubahan Data jenis

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data jenis yang akan diubah : Jenis : bahan bangunan	Menampilkan pesan “Anda yakin data akan dihapus?” jika ya maka data akan dihapus.	Data yang diubah tampil sesuai dengan editbox yang tersedia Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data jenis yang akan diubah : Jenis : bahan	Menampilkan pesan “Anda yakin data	Memberikan pesan “Anda yakin data akan	[√] diterima [ ] ditolak

bangunan	akan dihapus?" jika ya maka data akan dihapus.	dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	

c. Hapus Data jenis

Untuk melakukan pengubahan klik tombol hapus data jenis maka akan tampil form ubah data jenis.

Tabel 4.30 Hapus data jenis

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Hapus data jenis	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data jenis yang	Saat menekan	Tombol batal	[√] diterima

tidak akan dihapus	tombol batal data jenis tidak akan terhapus	bekerja sesuai dengan yang diharapkan	[ ] ditolak

4.2.1.12 Pengujian Pengolahan Data merk

Pengujian data bagian ini terbagi kedalam 3 bagian yaitu proses penambahan, pengubahan dan penghapusan data merk.

Untuk melakukan pengubahan klik tombol tambah data merk maka akan tampil form tambah data merk.

a. Tambah Data merk

Tabel 4.31Tambah Data merk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode merk : KM0003 Merk : grasia	Data pada Kode merk dan merk dapat terisi	Dapat mengisi Kode merk dan merk sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode merk dan	Data tidak dapat	Penambahan data	[√] diterima

merk Salah atau kosong	disimpan pada tabel merk dan menampilkan pesan " data merk Belum lengkap terisi"	gagal dan menampilkan pesan " data merk Belum lengkap terisi"	[ ] ditolak

b. Pengubahan Data merk

Untuk melakukan pengubahan klik tombol ubah data merk maka akan tampil form ubah data merk.

Tabel 4. 32 Pengubahan Data merk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data yang akan diubah data merk : toto	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak
Data merk : toto Salah atau kosong	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak

c. Hapus Data merk

Untuk melakukan pengubahan klik tombol hapus data merk maka akan tampil form ubah data merk.

Tabel 4.33 Hapus data merk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data merk : toto	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data merk yang tidak akan dihapus atau data kosong	Saat menekan tombol batal data merk tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.13 Pengujian Pengolahan Data Bahan

Pengujian data bagian ini terbagi kedalam 3 bagian yaitu proses penambahan, pengubahan dan penghapusan data bahan.

Untuk melakukan pengubahan klik tombol tambah data bahan maka akan tampil form tambah data bahan.

a. Tambah Data Bahan

Tabel 4.34 Tambah Data bahan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode bahan : KN0003 Bahan : stainles	Data pada Kode m bahan dan bahan dapat terisi	Dapat mengisi Kode bahan dan bahan sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode bahan dan bahan : Salah atau kosong	Data tidak dapat disimpan pada tabel bahan dan menampilkan	Penambahan data gagal dan menampilkan pesan “ data	[√] diterima [ ] ditolak

	pesan “ data bahan Belum lengkap terisi“	bahan Belum lengkap terisi“	

b. Pengubahan Data Bahan

Untuk melakukan pengubahan klik tombol ubah data bahan maka akan tampil form ubah data bahan.

Tabel 4. 35 Pengubahan Data Bahan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data bahan yang akan diubah : kayu	Data yang tertera pada dbgrid keluar pada form pengubahan data atau Data yang lama akan terganti dengan data baru.	Data yang diubah tampil sesuai dengan editbox yang tersedia. Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak
Masukan data bahan yang akan diubah: kosaong	.	[√] diterima [ ] ditolak	

c. Hapus Data Bahan

Untuk melakukan pengubahan klik tombol hapus data Bahan maka akan tampil form hapus data Bahan.

Tabel 4.36 Hapus data bahan

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data Bahan : plastik	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data Bahan yang tidak akan dihapus atau kosong	Saat menekan tombol batal data Bahan tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.14 Pengujian Pengolahan Data Bentuk

Pengujian data bagian ini terbagi kedalam 3 bagian yaitu proses penambahan, pengubahan dan penghapusan data bentuk.

Untuk melakukan pengubahan klik tombol tambah data bentuk maka akan tampil form tambah data bentuk.

a. Tambah Data Bentuk

Tabel 4.37 Tambah Data bentuk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode bentuk : KK0004 Bentuk : lancip	Data pada Kode bentuk dan bentuk dapat terisi	Dapat mengisi Kode bentuk dan bentuk sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode bentuk dan bentuk : salah atau kosong	Data tidak dapat disimpan pada tabel bentuk dan menampilkan	Penambahan data gagal dan menampilkan pesan “ data	[√] diterima [ ] ditolak

	pesan “ data bentuk Belum lengkap terisi“	bentuk Belum lengkap terisi“	

b. Pengubahan Data Bentuk

Untuk melakukan pengubahan klik tombol ubah data bentuk maka akan tampil form ubah data bentuk.

Tabel 4. 38 Pengubahan Data Bentuk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data bentuk : lancip	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak
data bentuk yang akan diubah	Data yang lama akan terganti dengan data baru.	Data yang lama terlihat sudah terganti dengan yang baru	[√] diterima [ ] ditolak

c. Hapus Data Bentuk

Untuk melakukan pengubahan klik tombol hapus data bentuk maka akan tampil form hapus data bentuk.

Tabel 4.39 Hapus data bentuk

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data bentuk : oval	Menampilkan pesan "Anda yakin data akan dihapus?" jika ya maka data akan dihapus.	Memberikan pesan "Anda yakin data akan dihapus?" jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
data bentuk yang tidak akan dihapus,salah, atau kosong	Saat menekan tombol batal data bentuk tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.2.1.15 Pengujian Pengolahan Data Warna

Pengujian data bagian ini terbagi kedalam 3 bagian yaitu proses penambahan, pengubahan dan penghapusan data warna.

Untuk melakukan pengubahan klik tombol tambah data warna maka akan tampil form tambah data warna.

a. Tambah Data Warna

Tabel 4.40 Tambah Data Warna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna : KW0005 Warna : putih	Data pada Kode warna dan warna dapat terisi	Dapat mengisi Kode warna dan warna sesuai dengan yang diharapakan	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna dan warna : Salah atau kosong	Data tidak dapat disimpan pada tabel warna dan menampilkan	Penambahan data gagal dan menampilkan pesan “ data	[√] diterima [ ] ditolak

	pesan “ data warna Belum lengkap terisi“	warna Belum lengkap terisi“	

b. Pengubahan Data Warna

Untuk melakukan pengubahan klik tombol ubah data warna maka akan tampil form ubah data warna.

Tabel 4. 41 Pengubahan Data warna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna : KW0005 Warna : putih	Data yang tertera pada dbgrid keluar pada form pengubahan data	Data yang diubah tampil sesuai dengan editbox yang tersedia	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna : KW0005 Warna : putih : Salah atau kosong	Saat menekan tombol batal data warna tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

c. Hapus Data Warna

Untuk melakukan pengubahan klik tombol hapus data warna maka akan tampil form hapus data warna.

Tabel 4.42 Hapus data warna

Kasus dan hasil uji (data normal)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna : KW0005 Warna : putih	Menampilkan pesan “Anda yakin data akan dihapus?” jika ya maka data akan dihapus.	Memberikan pesan “Anda yakin data akan dihapus?” jika ya maka data dihapus. Sesuai yang diinginkan.	[√] diterima [ ] ditolak
Kasus dan hasil uji (data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode warna : KW0005 Warna : putih Data salah atau kosong	Saat menekan tombol batal data warna tidak akan terhapus	Tombol batal bekerja sesuai dengan yang diharapkan	[√] diterima [ ] ditolak

4.3 Pengujian Betha

Pengujian betha merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, yaitu dengan membuat kuisioner yang ditujukan kepada pengguna dari sistem informasi pengolahan data barang di perusahaan terhadap sistem yang telah dibangun. Sampel yang diambil sebanyak 1 orang yang merupakan calon pengguna aplikasi ini.

Kuesioner ini terdiri dari 5 pertanyaan dengan menggunakan lima pilihan jawaban yang mewakili dari tujuan akhir yang ingin dicapai dalam pembangunan sistem informasi penjualan dan pembelian di perusahaan ini, dimana daftar pertanyaan yang diberikan adalah sebagai berikut :

1. Sistem Informasi Penjualan dan Pembelian yang dibangun memudahkan bagi pengguna untuk mengolah data penjualan dan pembelian tiap karyawan?

    a. Sangat Mudah

    b. Cukup Mudah

    c. Mudah

    d. Kurang Mudah

    e. Sangat Tidak Mudah

2. Apakah aplikasi yang dibangun sudah dapat membantu Mempercepat dalam menghasilkan informasi agar tidak terjadi keterlambatan dalam memperoleh informasi?

   a. Sangat Membantu

   b. Cukup Membantu

   c. Membantu

   d. Kurang Membantu

   e. Sangat Tidak Membantu

3. Apakah aplikasi yang dibangun sudah dapat mengurangi kesalahan dalam pengolahan penjualan dan pembelian?

   a. Sangat Mengurangi

   b. Cukup Mengurangi

   c. Mengurani

   d. Kurang Mengurangi

   e. Sangat Tidak Mengurangi

4. Apakah aplikasi yang dibangun sudah dapat menyajikan data penjualan dan pembelian secara cepat dan akurat?

a. Sangat akurat

b. Cukup akurat

c. akurat

d. Kurang akurat

e. Sangat Tidak akurat

5. Sistem Informasi Penjualan dan Pembelian yang dibangun mudah untuk dipelajari dan digunakan oleh pengguna?

a. Sangat mudah

b. Cukup mudah

c. mudah

d. Kurang mudah

e. Sangat Tidak mudah

Keterangan:

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai Prosentase

Berikut adalah hasil yang didapat dari kuesioner.

1. Sistem Informasi Penjualan dan Pembelian yang dibangun memudahkan bagi pengguna untuk mengolah data penjualan dan pembelian tiap karyawan?

No	Keterangan	Responden	Prosentase (%)
1	Sangat Mudah	-	-
2	Cukup Mudah	1	100%
3	Mudah	-	-
4	Kurang Mudah	-	-
5	Sangat Tidak Mudah	-	-

Berdasarkan hasil prosentase dapat disimpulkan bahwa pengguna menyatakan cukup mudah menggunakan aplikasi ini.

2. Apakah aplikasi yang dibangun sudah dapat membantu Mempercepat dalam menghasilkan informasi agar tidak terjadi keterlambatan dalam memperoleh informasi?

No	Keterangan	Responden	Prosentase (%)
1	Sangat Membantu	-	-
2	Cukup Membantu	1	100%
3	Membantu	-	-

4	Kurang Membantu	-	-
5	Sangat Tidak Membantu	-	-

Berdasarkan hasil prosentase dapat disimpulkan bahwa pengguna menyatakan informasi yang disediakan cukup membantu.

3. Apakah aplikasi yang dibangun sudah dapat mengurangi kesalahan dalam pengolahan data penjualan dan pembelian?

No	Keterangan	Responden	Prosentase (%)
1	Sangat Mengurangi	-	-
2	Cukup Mengurangi	-	-
3	mengurangi	1	100%
4	Kurang mengurangi	-	-
5	Sangat Tidak mengurangi	-	-

Berdasarkan hasil prosentase dapat disimpulkan bahwa pengguna menyatakan *dapat mengurangi kesalahan dalam pengolahan data penjualan dan pembelian.*

5. Apakah aplikasi yang dibangun sudah dapat menyajikan penolahan data penjualan dan pembelian secara cepat dan akurat?

No	Keterangan	Responden	Prosentase (%)
1	Sangat akurat	-	-
2	Cukup akurat	-	-
3	Akurat	1	100%
4	Kurang akurat	-	-
5	Sangat Tidak akurat	-	-

Berdasarkan hasil prosentase dapat disimpulkan bahwa pengguna menyatakan bahwa aplikasi ini akurat dalam pengolahan.

5. Sistem Informasi Penjualan dan Pembelian yang dibangun mudah untuk dipelajari dan digunakan oleh pengguna

No	Keterangan	Responden	Prosentase (%)
1	Sangat Mudah	-	-
2	Cukup Mudah	1	100%
3	Mudah	-	-
4	Kurang Mudah	-	-
5	Sangat Tidak Mudah	-	-

Berdasarkan hasil prosentase dapat disimpulkan bahwa pengguna menyatakan bahwa aplikasi yang dibangun Cukup Mudah digunakan.

BAB V

KESIMPULAN DAN SARAN

Pada bagian terakhir ini akan dikemukakan kesimpulan sistem informasi yang dapat diperoleh dari pembahasan bab-bab sebelumnya serta saran yang bersifat membangun.

5.1 Kesimpulan

Sistem Informasi Penjualan dan Pembelian Bahan Baku Bangunan Di. Pd. Pembangunan Raya telah dibangun dan dilakukan pengujianterhadap sistemnya, maka dapat disimpulkan bahwa:

1. Sistem ini memudahkan pengguna untuk mengolahan data penjualan dan pembelian.
2. Hasil laporan yang dikeluarkan cukup membantu pengguna dalam mengelola informasi yang ditampilkan
3. Hasil pengolahan sesuai dengan yang diharapkan dan menapilkan nya dengan cepat.

5.2 Saran

Sistem Informasi Penjualan dan Pembelian Bahan Baku Bangunan Di. PD. Pembangunan Raya ini masih dapat dikembangkan lebih lanjut dengan perkembangan spesifikasi kebutuhan pengguna sistem yang harus dipenuhi dalam mencapai tahap yang lebih tinggi dan kinerja sistem yang lebih baik. Berikut adalah beberapa saran untuk pengembangan lebih lanjut :

1. Diharapkan untuk pengembangan selanjutnya yaitu sistem informasi penjualan dan pembelian berbasis *WEB*
2. Diharapkan untuk pengembangan selanjutnya tampilan antarmuka dibuat lebih menarik.

DAFTAR PUSTAKA

1. Budi Susanto, (4 April 2008), *Model Entity-Relationship*, http://www.sikd.djapk.go.id/sakpd/Buku2bab2
2. Jogianto HM, (1999). *Analisis dan Desain Sistem Informasi*, Andi. Yogyakarta.
3. MADCOMS.(2002). *Pemrograman Borland Delphi 7 (jilid1*).Andi. Yogyakarta.
4. Pressman, Roger S.(2002). *Rekayasa Perangkat Lunak Pendekatan Praktisi.* Andi. Yogyakarta.
5. Teddy, Marcus, Agus, Prijono, Josef Widiadhi.(2005). *Pemrograman Delphi Dengan AdoExpress*. Informatika. Bandung.
6. WAHANA KOMPUTER.(2003). *Panduan Praktis Pemrograman Borland Delphi 7.0*. Andi. Yogyakarta.
7. WAHANA KOMPUTER.(2003). *Tip dan Trik Pemrograman Delphi 7.0,*Andi, Yogyakarta.
8. -. (14 Maret 2008), *Unique Key*, http://en.wikipedia.org/wiki/Unique_key

LAMPIRAN A

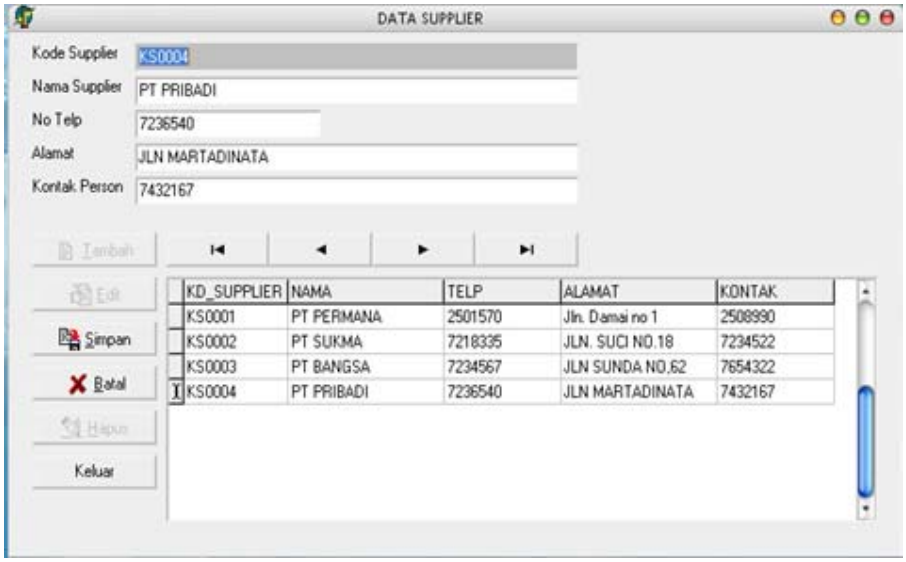
TAMPILAN ANTAR MUKA



Tampilan Login



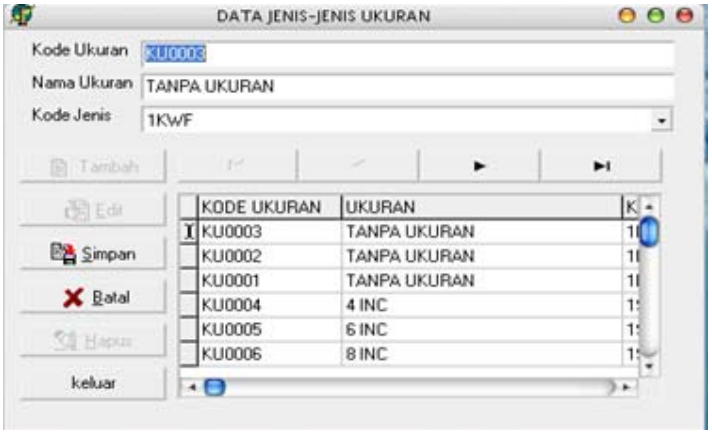
Tampilan Menu Utama



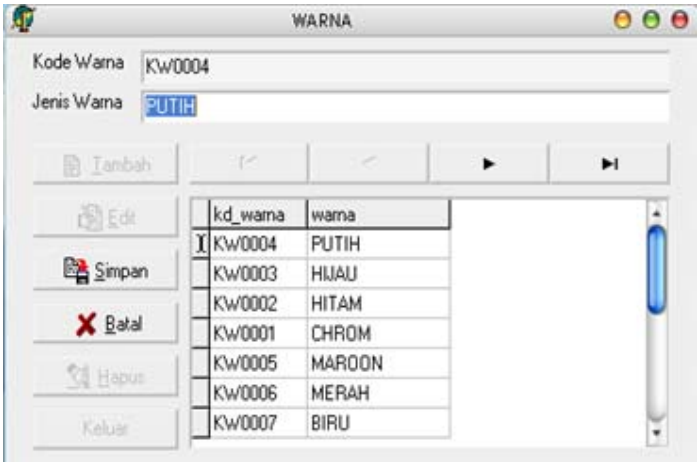
Tampilan Data Supplier



Tampilan Jenis Barang



Tampilan Jenis Ukuran



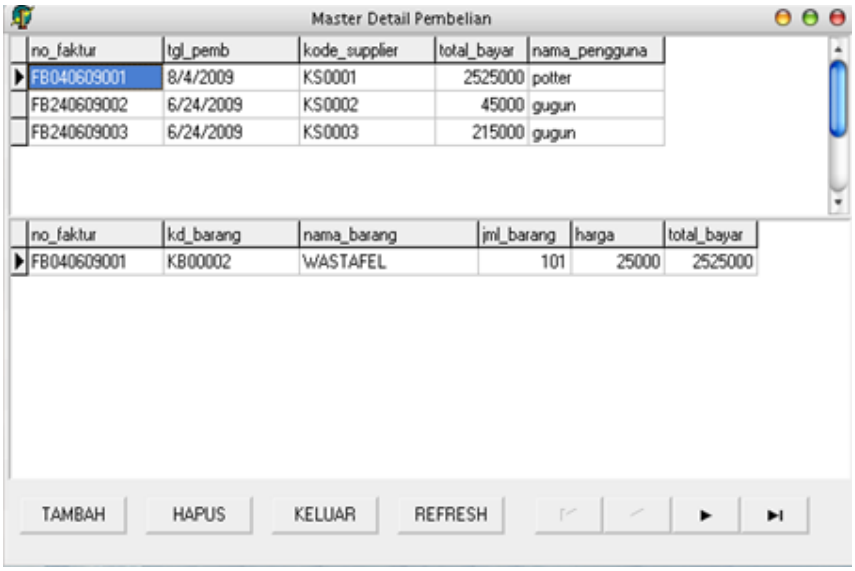
Tampilan Warna



Tampilan Bentuk



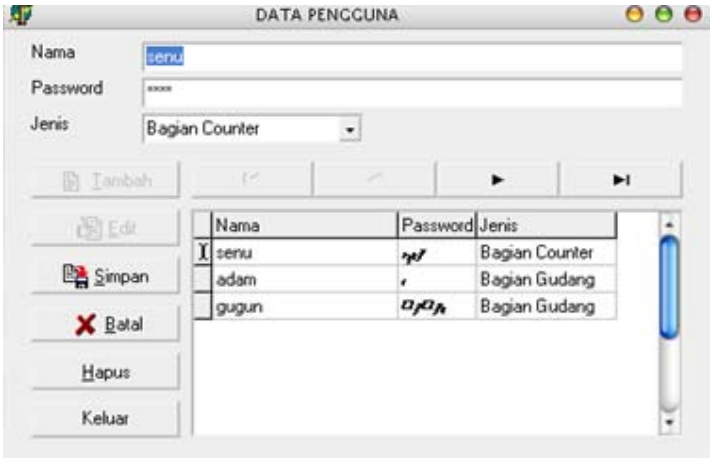
Tampilan Bahan



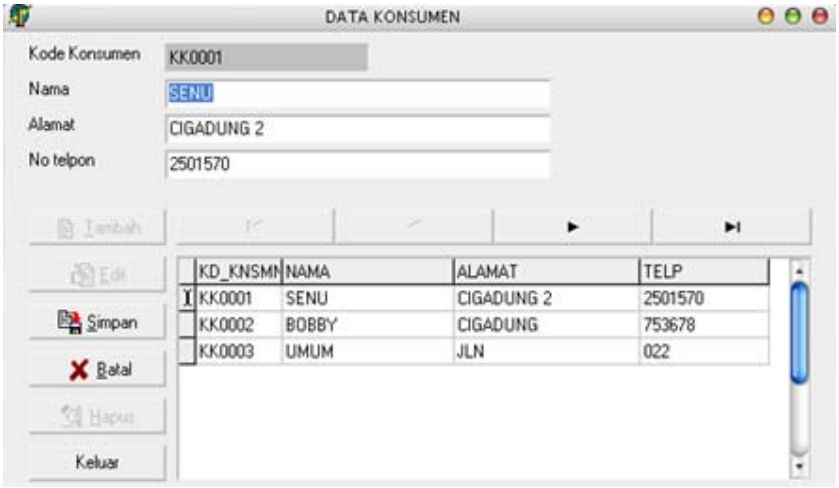
Tampilan Master Penjualan



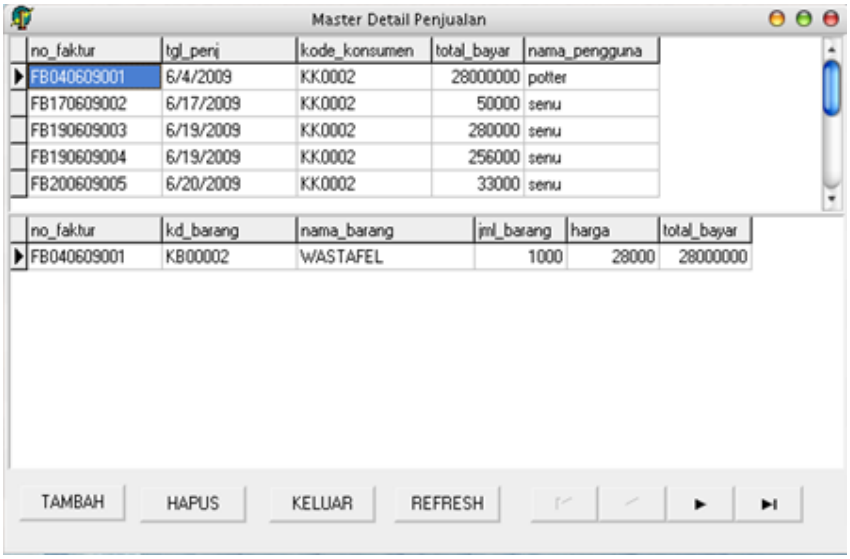
Tampilan Transaksi Pembelian



Tampilan Data pengguna



Tampilan Data Konsumen



Tampilan Master Detail Penjualan

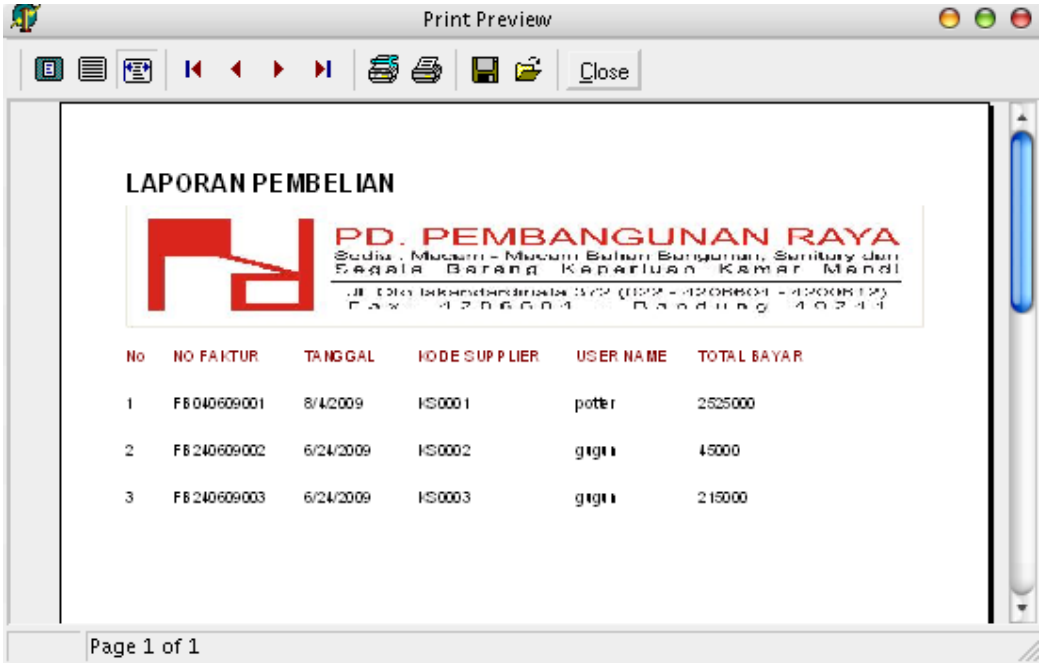


Tampilan Transaksi Penjualan



No	FAKTUR	TANGGAL	KODE KONSUMEN	USER NAME	TOTAL
1	FB0406090013/4/2009		KK0002	potter	28000000
2	FB1706090023/17/2009		KK0002	senu	50000
3	FB1906090033/19/2009		KK0002	senu	280000
4	FB1906090043/19/2009		KK0002	senu	256000
5	FB2006090053/20/2009		KK0002	senu	33000
6	FB2406090063/24/2009		KK0002	senu	11000

Tampilan laporan Transaksi Penjualan



No	NO FAKTUR	TANGGAL	KODE SUPPLIER	USER NAME	TOTAL BAYAR
1	FB040609001	8/4/2009	KS0001	potter	2525000
2	FB240609002	6/24/2009	KS0002	gugun	45000
3	FB240609003	6/24/2009	KS0003	gugun	215000

Tampilan laporan Transaksi Pembelian

LAMPIRAN B

LISTING PROGRAM

```
unit u_login;

interface

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

type
  Tf_login = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Button1: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender:
TObject);
    procedure FormClose(Sender: TObject;
var Action: TCloseAction);
    procedure FormCreate(Sender:
TObject);
    procedure Button2Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_login: Tf_login;

implementation

uses u_dm, u_menu_utama;

{$R *.dfm}

procedure Tf_login.Button1Click(Sender:
TObject);
begin
with DM do begin
ADOT_PENGGUNA.First;
repeat
if
(Edit1.Text=ADOT_PENGGUNA['nama_
pengguna'])and
(Edit2.Text=ADOT_PENGGUNA['passwo
rd']) then
begin
ShowMessage('Selamat Bekerja
'+Edit1.Text+'!!');
Tf_utama_sales.Create(self);
if
(dm.ADOT_PENGGUNA['jenis_pengguna']='bag
ian counter') then
begin
with f_utama_sales do begin
      l_username.Caption:=Edit1.Text;

l_status.Caption:=DM.ADOT_PENGGUNA['jeni
s_pengguna'];
   Menu1.Enabled:=true;
   Transaksi1.Enabled:=true;
   Mn_jenis.Visible:=false;
   Mn_merk.Visible:=false;
   Mn_ukuran.Visible:=false;
   Mn_supplier.Visible:=false;
   Mn_pembelian.Visible:=false;
   Mn_pencarian.Enabled:=true;
   Mn_pengguna.Enabled:=true;
   Laporan1.Enabled:=true;
   ambahEditHapus1.visible:=false;
   Logout.Enabled:=true;
end;
end
else
if
(dm.ADOT_PENGGUNA['jenis_pengguna']='gud
ang') then
begin
 with f_utama_sales do begin
      l_username.Caption:=Edit1.Text;

l_status.Caption:=DM.ADOT_PENGGUNA['jeni
s_pengguna'];
   Menu1.Enabled:=true;
   Transaksi1.Enabled:=true;
   Mn_jenis.Enabled:=true;
   Mn_merk.Enabled:=true;
   Mn_ukuran.Enabled:=true;
   Mn_supplier.Enabled:=true;
   Mn_pembelian.Enabled:=true;
   Mn_pencarian.Enabled:=true;
   Mn_pengguna.Enabled:=true;
   Laporan1.Enabled:=true;
   ambahEditHapus1.Enabled:=true;
   Logout.Enabled:=true;
end;
```

```
end;
close;
exit;
end;
ADOT_PENGGUNA.Next;
until ADOT_PENGGUNA.Eof;
ShowMessage('User ID atau Password
salah!!');
end;
end;

procedure Tf_login.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
Action:=caFree;
end;

procedure Tf_login.FormCreate(Sender:
TObject);
begin
DM.ADOT_PENGGUNA.Active:=true
end;

procedure Tf_login.Button2Click(Sender:
TObject);
begin
application.Terminate;
end;

end.


unit u_barang;

interface

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

type
  Tf_barang = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    Label3: TLabel;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    Label6: TLabel;
    DBEdit6: TDBEdit;
    DBEdit7: TDBEdit;
    Label7: TLabel;
    Label8: TLabel;
    DBEdit8: TDBEdit;
    Label9: TLabel;
    DBEdit9: TDBEdit;
    DBEdit10: TDBEdit;
    Label10: TLabel;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender: TObject);
    procedure BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender: TObject);
    procedure BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_barang: Tf_barang;

implementation

uses u_dm, u_data_pengguna,
u_detail_pembelian, u_detail_penjualan, u_jenis,
```

```
  u_konsumen, u_login, u_menu_utama,
u_merk, u_pemb_langsung,
   u_supplier, u_ukuran;


{$R *.dfm}

procedure
TF_barang.enable_bitbtn_assisten(enable:b
oolean);
begin
      keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;

BitBtn_Edit_Assisten.Enabled:=enable;

BitBtn_Hapus_Assisten.Enabled:=enable;
      BitBtn_Batal_Assisten.Enabled:=not
enable;

BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_barang.keadaan_awal(keadaan:boolean
);
begin
      DBEdit1.Enabled:=keadaan;
      DBEdit2.Enabled:=keadaan;
      DBEdit3.Enabled:=keadaan;
      DBEdit4.Enabled:=keadaan;
      DBEdit5.Enabled:=keadaan;
      DBEdit6.Enabled:=keadaan;
      DBEdit7.Enabled:=keadaan;
      DBEdit8.Enabled:=keadaan;
      DBEdit9.Enabled:=keadaan;
      DBEdit10.Enabled:=keadaan;

end;

procedure Tf_barang.FormShow(Sender:
TObject);
begin
      enable_bitbtn_assisten(True);
end;

procedure
Tf_barang.BitBtn_Tambah_AssistenClick(Sender
: TObject);
begin
      DM.ADOT_BARANG.Append;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_barang.BitBtn_Edit_AssistenClick(Sender:
TObject);
begin
      DM.ADOT_BARANG.Edit;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_barang.BitBtn_Simpan_AssistenClick(Sender:
TObject);
begin
      if (DBEdit1.Text='')and(DBEdit2.Text='')
then
      begin
        MessageDlg('Isi data terlebih
dahulu',mtConfirmation,[mbOK],0);
        DBEdit1.SetFocus;
      end
      else
      begin
        try
          DM.ADOT_BARANG.Post;
        except
          DM.ADOT_BARANG.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_barang.BitBtn_Batal_AssistenClick(Sender:
TObject);
begin
      DM.ADOT_BARANG.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_barang.BitBtn_Hapus_AssistenClick(Sender:
TObject);
```

```
begin
      if MessageDlg('Apakah yakin datanya
akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then
       DM.ADOT_BARANG.Delete
      else
       DM.ADOT_BARANG.Cancel;
end;

procedure Tf_barang.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_BARANG.State=DsInsert)or(
DM.ADOT_BARANG.State=DsEdit) then
      if MessageDlg('Sebelum keluar,
apakah data akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then

BitBtn_Simpan_AssistenClick(Sender)
      else

BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_BARANG.Close;
f_utama_sales.Mn_DataBarang1.Enabled:
=true;
end;

end.


unit u_data_pengguna;

interface

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

type
  Tf_data_pengguna = class(TForm)
    Nama: TLabel;
    Password: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender: TObject);
    procedure BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender: TObject);
    procedure BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_data_pengguna: Tf_data_pengguna;

implementation

uses u_barang, u_detail_pembelian,
u_detail_penjualan, u_dm, u_jenis,
  u_konsumen, u_login, u_menu_utama, u_merk,
u_pemb_langsung,
   u_supplier, u_ukuran;



{$R *.dfm}

procedure
Tf_data_pengguna.enable_bitbtn_assisten(enable:
boolean);
```

```
begin
      keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;

BitBtn_Edit_Assisten.Enabled:=enable;

BitBtn_Hapus_Assisten.Enabled:=enable;
      BitBtn_Batal_Assisten.Enabled:=not
enable;

BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_data_pengguna.keadaan_awal(keadaan:
boolean);
begin
      DBEdit1.Enabled:=keadaan;
      DBEdit2.Enabled:=keadaan;
      DBEdit3.Enabled:=keadaan;

end;

procedure
Tf_data_pengguna.FormShow(Sender:
TObject);
begin
      enable_bitbtn_assisten(True);
end;

procedure
Tf_data_pengguna.BitBtn_Tambah_Assist
enClick(Sender: TObject);
begin
      DM.ADOT_PENGGUNA.Append;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_data_pengguna.BitBtn_Edit_AssistenCl
ick(Sender: TObject);
begin
      DM.ADOT_PENGGUNA.Edit;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_data_pengguna.BitBtn_Simpan_AssistenClick
(Sender: TObject);
begin
      if (DBEdit1.Text='')and(DBEdit2.Text='')
then
      begin
        MessageDlg('Isi data terlebih
dahulu',mtConfirmation,[mbOK],0);
        DBEdit1.SetFocus;
      end
      else
      begin
        try
          DM.ADOT_PENGGUNA.Post;
        except
          DM.ADOT_PENGGUNA.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_data_pengguna.BitBtn_Batal_AssistenClick(S
ender: TObject);
begin
      DM.ADOT_PENGGUNA.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_data_pengguna.BitBtn_Hapus_AssistenClick(
Sender: TObject);
begin
      if MessageDlg('Apakah yakin datanya akan
dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
        DM.ADOT_PENGGUNA.Delete
      else
        DM.ADOT_PENGGUNA.Cancel;
end;

procedure Tf_data_pengguna.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_PENGGUNA.State=DsInsert)or(DM
.ADOT_PENGGUNA.State=DsEdit) then
```

```
      if MessageDlg('Sebelum keluar,
apakah data akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then

BitBtn_Simpan_AssistenClick(Sender)
      else

BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_PENGGUNA.Close;
f_utama_sales.ambahEditHapus1.Enabled:
=true;
end;

end.
```

```
unit u_detail_pembelian;

interface

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

type
  Tf_detail_pembelian = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Splitter1: TSplitter;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    Panel1: TPanel;
    Label1: TLabel;
    Button1: TButton;
    DBNavigator1: TDBNavigator;
    edit1: TEdit;
    BitBtn1: TBitBtn;
    procedure Button1Click(Sender:
TObject);
    procedure Q_pembelian;
    procedure Q_detail_pemb;
    procedure FormClose(Sender: TObject;
var Action: TCloseAction);
    procedure BitBtn1Click(Sender:
TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_detail_pembelian: Tf_detail_pembelian;

implementation

uses u_barang, u_data_pengguna,
u_detail_penjualan, u_dm, u_jenis,
  u_konsumen, u_login, u_menu_utama, u_merk,
u_pemb_langsung,
   u_supplier, u_ukuran, u_cetak_det_pemb;
```

```
{$R *.dfm}

procedure Tf_detail_pembelian.Q_pembelian;
var
   DBASE_PATH:string;
   Objsearch:string;
   petik:string;
begin
DBASE_PATH:='t_pembelian';
petik:='';
Objsearch:='no_faktur';
with DM.ADOQ_pembelian do begin
if Length(Edit1.Text)<>0 then begin
        Close;
          with SQl do begin
            Clear;
             Add('select * from '+DBASE_PATH);
             Add('where '+ObjSearch+' like
'+petik+'%'+Edit1.Text+'%'+petik);
          end;
        Open;
end;
if IsEmpty then ShowMessage('Data tidak di
temukan...');
Edit1.SetFocus;
end;
end;


procedure Tf_detail_pembelian.Q_detail_pemb;
var
   DBASE_PATH:string;
```

```
  Objsearch:string;
  petik:string;
begin
DBASE_PATH:='t_detail_pemb';
petik:="";
Objsearch:='no_faktur';
with DM.ADOQ_DETAIL_PEMB do
begin
if Length(Edit1.Text)<>0 then begin
      Close;
        with SQl do begin
          Clear;
           Add('select * from
'+DBASE_PATH);
           Add('where '+ObjSearch+' like
'+petik+'%'+Edit1.Text+'%'+petik);
        end;
      Open;
end;
end;
end;

procedure
Tf_detail_pembelian.Button1Click(Sender:
TObject);
begin
Q_pembelian;
Q_detail_pemb;
end;

procedure
Tf_detail_pembelian.FormClose(Sender:
TObject;
 var Action: TCloseAction);
begin
Action:=caFree;
f_utama_sales.Mn_detail_pemb.Enabled:=t
rue;

end;

procedure
Tf_detail_pembelian.BitBtn1Click(Sender:
TObject);
begin
f_cetak_det_pemb.QuickRep1.Preview;
end;

end.
```

```
unit u_detail_penjualan;

interface

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

type
  Tf_detail_penjualan = class(TForm)
    GroupBox1: TGroupBox;
    GroupBox2: TGroupBox;
    Splitter1: TSplitter;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    Panel1: TPanel;
    Label1: TLabel;
    Button1: TButton;
    DBNavigator1: TDBNavigator;
    edit1: TEdit;
    BitBtn1: TBitBtn;

    procedure Button1Click(Sender: TObject);
    procedure Q_penjualan;
    procedure Q_detail_penj;
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_detail_penjualan: Tf_detail_penjualan;

implementation

uses u_barang, u_data_pengguna,
u_detail_pembelian, u_dm, u_jenis,
  u_konsumen, u_login, u_menu_utama, u_merk,
u_pemb_langsung,
    u_supplier, u_ukuran, u_cetak_det_penj;



{$R *.dfm}
procedure Tf_detail_penjualan.Q_penjualan;
```

```
var
  DBASE_PATH:string;
  Objsearch:string;
  petik:string;
begin
DBASE_PATH:='t_penjualan';
petik:="";
Objsearch:='no_faktur';
with DM.ADOQ_penjualan do begin
if Length(Edit1.Text)<>0 then begin
      Close;
        with SQl do begin
          Clear;
           Add('select * from
'+DBASE_PATH);
           Add('where '+ObjSearch+' like
'+petik+'%'+Edit1.Text+'%'+petik);
        end;
      Open;
end;
if IsEmpty then ShowMessage('Data tidak
di temukan...');
Edit1.SetFocus;
end;
end;


procedure
Tf_detail_penjualan.Q_detail_penj;
var
  DBASE_PATH:string;
  Objsearch:string;
  petik:string;
begin
DBASE_PATH:='t_detail_penj';
petik:="";
Objsearch:='no_faktur';
with DM.ADOQ_DETAIL_PENJ do begin
if Length(Edit1.Text)<>0 then begin
      Close;
        with SQl do begin
          Clear;
           Add('select * from
'+DBASE_PATH);
           Add('where '+ObjSearch+' like
'+petik+'%'+Edit1.Text+'%'+petik);
        end;
      Open;
end;
end;
end;
```

```
procedure
Tf_detail_penjualan.Button1Click(Sender:
TObject);
begin
Q_penjualan;
Q_detail_penj;
end;

procedure
Tf_detail_penjualan.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
Action:=caFree;
f_utama_sales.Mn_detail_penj.Enabled:=true;

end;

procedure
Tf_detail_penjualan.BitBtn1Click(Sender:
TObject);
begin
f_cetak_det_penj.QuickRep1.Preview;
end;

end.


unit u_jenis;

interface

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

type
  Tf_jenis = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
```

```
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender:
TObject);
    procedure FormClose(Sender: TObject;
var Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure
keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_jenis: Tf_jenis;

implementation

uses u_data_pengguna, u_detail_penjualan,
u_konsumen, u_login, u_merk,
  u_pemb_langsung,
  u_supplier, u_ukuran, u_dm,
u_menu_utama;



{$R *.dfm}

procedure
Tf_jenis.enable_bitbtn_assisten(enable:boo
lean);
begin
       keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;
       BitBtn_Edit_Assisten.Enabled:=enable;
       BitBtn_Hapus_Assisten.Enabled:=enable;
       BitBtn_Batal_Assisten.Enabled:=not enable;
       BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_jenis.keadaan_awal(keadaan:boolean);
begin
       DBEdit1.Enabled:=keadaan;
       DBEdit2.Enabled:=keadaan;

end;

procedure Tf_jenis.FormShow(Sender: TObject);
begin
       enable_bitbtn_assisten(True);
end;

procedure
Tf_jenis.BitBtn_Tambah_AssistenClick(Sender:
TObject);
begin
       DM.ADOT_JENIS.Append;
       enable_bitbtn_assisten(False);
       DBEdit1.SetFocus;
end;

procedure
Tf_jenis.BitBtn_Edit_AssistenClick(Sender:
TObject);
begin
       DM.ADOT_JENIS.Edit;
       enable_bitbtn_assisten(False);
       DBEdit1.SetFocus;
end;

procedure
Tf_jenis.BitBtn_Simpan_AssistenClick(Sender:
TObject);
begin
       if (DBEdit1.Text='')and(DBEdit2.Text='')
then
       begin
         MessageDlg('Afawan isi data Assisten-nya
dulu !',mtConfirmation,[mbOK],0);
         DBEdit1.SetFocus;
       end
       else
       begin
```

```
        try
          DM.ADOT_JENIS.Post;
        except
          DM.ADOT_JENIS.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_jenis.BitBtn_Batal_AssistenClick(Send
er: TObject);
begin
      DM.ADOT_JENIS.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_jenis.BitBtn_Hapus_AssistenClick(Sen
der: TObject);
begin
      if MessageDlg('Apakah Antum yakin
datanya akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then
        DM.ADOT_JENIS.Delete
      else
        DM.ADOT_JENIS.Cancel;
end;

procedure Tf_jenis.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_JENIS.State=DsInsert)or(DM
.ADOT_JENIS.State=DsEdit) then
      if MessageDlg('Sebelum keluar,
apakah data akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then

BitBtn_Simpan_AssistenClick(Sender)
      else

BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_JENIS.Close;
f_utama_sales.Mn_jenis.Enabled:=true;
end;
```

```
end.


unit u_konsumen;

interface

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

type
  Tf_konsumen = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    DBEdit3: TDBEdit;
    Label3: TLabel;
    Label4: TLabel;
    DBEdit4: TDBEdit;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender: TObject);
    procedure BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender: TObject);
    procedure BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  f_konsumen: Tf_konsumen;

implementation

uses u_data_pengguna, u_detail_penjualan,
u_jenis, u_login, u_merk,
  u_pemb_langsung,
  u_supplier, u_ukuran, u_dm,
u_menu_utama;



{$R *.dfm}

procedure
Tf_konsumen.enable_bitbtn_assisten(enabl
e:boolean);
begin
      keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;

BitBtn_Edit_Assisten.Enabled:=enable;

BitBtn_Hapus_Assisten.Enabled:=enable;
      BitBtn_Batal_Assisten.Enabled:=not
enable;

BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_konsumen.keadaan_awal(keadaan:bool
ean);
begin
      DBEdit1.Enabled:=keadaan;
      DBEdit2.Enabled:=keadaan;
      DBEdit3.Enabled:=keadaan;
      DBEdit4.Enabled:=keadaan;
end;

procedure
Tf_konsumen.FormShow(Sender:
TObject);
begin
      enable_bitbtn_assisten(True);
end;

procedure
Tf_konsumen.BitBtn_Tambah_AssistenClick(Sen
der: TObject);
begin
      DM.ADOT_KONSUMEN.Append;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_konsumen.BitBtn_Edit_AssistenClick(Sender:
TObject);
begin
      DM.ADOT_KONSUMEN.Edit;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_konsumen.BitBtn_Simpan_AssistenClick(Sen
der: TObject);
begin
      if (DBEdit1.Text='')and(DBEdit2.Text='')
then
      begin
        MessageDlg('Afawan isi data Assisten-nya
dulu !',mtConfirmation,[mbOK],0);
        DBEdit1.SetFocus;
      end
      else
      begin
        try
          DM.ADOT_KONSUMEN.Post;
        except
          DM.ADOT_KONSUMEN.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_konsumen.BitBtn_Batal_AssistenClick(Sender
: TObject);
begin
      DM.ADOT_KONSUMEN.Cancel;
      enable_bitbtn_assisten(True);
end;
```

```
procedure
Tf_konsumen.BitBtn_Hapus_AssistenClic
k(Sender: TObject);
begin
      if MessageDlg('Apakah Antum yakin
datanya akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then
        DM.ADOT_KONSUMEN.Delete
      else
        DM.ADOT_KONSUMEN.Cancel;
end;

procedure
Tf_konsumen.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_KONSUMEN.State=DsInsert
)or(DM.ADOT_KONSUMEN.State=DsEd
it) then
      if MessageDlg('Sebelum keluar,
apakah data akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then

BitBtn_Simpan_AssistenClick(Sender)
      else

BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_KONSUMEN.Close;
f_utama_sales.Mn_konsumen.Enabled:=tru
e;
end;

end.


unit u_merk;

interface

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

type
  Tf_merk = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender: TObject);
    procedure BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender: TObject);
    procedure BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender: TObject);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_merk: Tf_merk;

implementation

uses u_data_pengguna, u_detail_penjualan,
u_jenis, u_konsumen, u_login,
  u_pemb_langsung,
  u_supplier, u_ukuran, u_dm, u_menu_utama;



{$R *.dfm}

procedure
TF_merk.enable_bitbtn_assisten(enable:boolean);
begin
```

```
      keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;

BitBtn_Edit_Assisten.Enabled:=enable;

BitBtn_Hapus_Assisten.Enabled:=enable;
      BitBtn_Batal_Assisten.Enabled:=not
enable;

BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_merk.keadaan_awal(keadaan:boolean);
begin
      DBEdit1.Enabled:=keadaan;
      DBEdit2.Enabled:=keadaan;
end;

procedure Tf_merk.FormShow(Sender:
TObject);
begin
      enable_bitbtn_assisten(True);
end;

procedure
Tf_merk.BitBtn_Tambah_AssistenClick(S
ender: TObject);
begin
      DM.ADOT_MERK.Append;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_merk.BitBtn_Edit_AssistenClick(Sende
r: TObject);
begin
      DM.ADOT_MERK.Edit;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_merk.BitBtn_Simpan_AssistenClick(Se
nder: TObject);
begin
      if (DBEdit1.Text='')and(DBEdit2.Text='')
then
      begin
        MessageDlg('Afawan isi data Assisten-nya
dulu !',mtConfirmation,[mbOK],0);
        DBEdit1.SetFocus;
      end
      else
      begin
        try
          DM.ADOT_MERK.Post;
        except
          DM.ADOT_MERK.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_merk.BitBtn_Batal_AssistenClick(Sender:
TObject);
begin
      DM.ADOT_MERK.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_merk.BitBtn_Hapus_AssistenClick(Sender:
TObject);
begin
      if MessageDlg('Apakah Antum yakin
datanya akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
        DM.ADOT_MERK.Delete
      else
        DM.ADOT_MERK.Cancel;
end;

procedure Tf_merk.FormClose(Sender: TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_MERK.State=DsInsert)or(DM.ADO
T_MERK.State=DsEdit) then
      if MessageDlg('Sebelum keluar, apakah data
akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
        BitBtn_Simpan_AssistenClick(Sender)
      else
        BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
```

```
DM.ADOT_MERK.Close;
f_utama_sales.Mn_merk.Enabled:=true;
end;

end.


unit u_pemb_langsung;

interface

uses
  Windows, Messages, SysUtils, Variants,
Classes, Graphics, Controls, Forms,
  Dialogs, Menus, Grids, DBGrids,
ExtCtrls, StdCtrls, Mask, DBCtrls,
  ComCtrls;

type
  Tf_pembelian = class(TForm)
    Label1: TLabel;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    DateTimePicker1: TDateTimePicker;
    Label2: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label6: TLabel;
    Bevel1: TBevel;
    Label3: TLabel;
    Label4: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    DBEdit7: TDBEdit;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit4: TEdit;
    GroupBox1: TGroupBox;
    ListBox1: TListBox;
    GroupBox2: TGroupBox;
    ListBox2: TListBox;
    GroupBox3: TGroupBox;
    ListBox3: TListBox;
    GroupBox4: TGroupBox;
    ListBox4: TListBox;
    DBEdit1: TDBEdit;
    DBLookupComboBox2:
TDBLookupComboBox;
    DBLookupComboBox3:
TDBLookupComboBox;
    DBLookupComboBox1:
TDBLookupComboBox;
    Edit3: TEdit;
    Button3: TButton;
    Button4: TButton;
    Label10: TLabel;
    DBEdit3: TDBEdit;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Edit2Change(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Edit2KeyPress(Sender: TObject; var
Key: Char);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure
DBLookupComboBox3Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var
Key: Char);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_pembelian: Tf_pembelian;
  tampung2:TStringList;
implementation

uses u_dm, u_menu_utama;

{$R *.dfm}


procedure Tf_pembelian.Button1Click(Sender:
TObject);
var jml,stok_awal,stok_akhir:integer;
begin

stok_awal:=StrToInt(DBEdit3.Text);
jml:=StrToInt(Edit2.Text);
stok_akhir:=stok_awal+jml;
```

```
if (Edit1.Text='') or
(DBLookupComboBox1.text='')
or (DBLookupComboBox2.text='') or
(DBLookupComboBox3.text='')
or (Edit2.Text='') or (DBEdit1.Text='')  or
(Edit4.Text='')

then begin
      ShowMessage('Inputan ada yang
kurang!!, silahkan cek ulang');
      exit;
end else if  (Edit2.Text='0') then begin
      ShowMessage('Masukan Jumlah
Pembelian');
      Edit2.SetFocus;
      exit;
end;
tampung2.Add(IntToStr(stok_akhir));
ListBox1.Items.Add(DBLookupComboBo
x3.Text);
ListBox2.Items.add(DBEdit1.Text);
ListBox3.Items.Add(Edit2.Text);
ListBox4.Items.add(Edit4.Text);
Edit2.Text:='0';
Edit4.Text:='0';
end;

procedure
Tf_pembelian.FormCreate(Sender:
TObject);
begin
tampung2:=TStringList.Create;
DateTimePicker1.Date:=now;
end;

procedure
Tf_pembelian.Edit2Change(Sender:
TObject);
var a,b,c:Integer;
begin


if Length(Edit2.Text)=0 then
edit2.text:='0';
  a:=StrToInt(Edit2.Text);
  b:=StrToInt(DBEdit1.Text);
  c:=a*b;
  Edit4.Text:=IntToStr(c);
end;
```

```
procedure Tf_pembelian.Button2Click(Sender:
TObject);
const petik:char='';
var perintah1:string;
i,j,harga,total:integer;
begin
if ListBox1.Items.Count <> 0 then begin

total:=0;


for i:=0 to ListBox4.Items.Count-1 do begin
        harga:=StrToInt(ListBox4.Items[i]);
        total:=total+harga;
end;


   perintah1:='INSERT INTO t_pembelian '+
          'VALUES ( '+
          petik+Edit1.Text+petik+','+#13+

petik+DateToStr(DateTimePicker1.Date)
+petik+','+#13+

petik+DBLookupComboBox1.Text+petik+','+#13
+
          petik+IntToStr(total)+petik+','+#13+

petik+DBLookupComboBox2.Text+petik+#13+
          ' )';


    try

      DM.ADOConnection1.Execute(perintah1);
      for j:=0 to ListBox1.Items.Count-1 do begin

DM.ADOConnection1.Execute('INSERT INTO
t_detail_pemb '+
          'VALUES ( '+
          petik+Edit1.Text+petik+','+#13+
          petik+ListBox1.Items[j] +petik+','+#13+
          petik+ListBox3.Items[j]+petik+','+#13+
          petik+ListBox2.Items[j]+petik+','+#13+
          petik+ListBox4.Items[j]+petik+#13+
          ' )');



DM.ADOConnection1.Execute('UPDATE
t_barang SET '+
```

```
            'stok =
'+tampung2.Strings[j]+#13+
            'WHERE kd_barang =
'+petik+ListBox1.Items[j]+petik);
       end;

    Except
          ShowMessage('NO FAKTUR
SUDAH TERDAFTAR SILAHKAN
GUNAKAN YANG LAIN!');
          Button3Click(sender);
          exit;
    end;

Edit3.Text:=IntToStr(total);
ShowMessage('Transaksi Berhasil Di
Simpan!');
DM.ADOT_BARANG.Refresh;
end else ShowMessage('Pilih barang
terlebih dahulu!');
tampung2.Free;
end;

procedure
Tf_pembelian.FormShow(Sender:
TObject);
begin
Edit1.SetFocus;
end;

procedure
Tf_pembelian.Edit2KeyPress(Sender:
TObject; var Key: Char);
begin
if not (Key in ['0'..'9',#8,#13])  then
Key:=#0;
end;

procedure
Tf_pembelian.Button3Click(Sender:
TObject);
begin
ListBox1.Items.Clear;
ListBox2.Items.Clear;
ListBox3.Items.Clear;
ListBox4.Items.Clear;
Edit1.Clear;
Edit2.Text:='0';
Edit4.Text:='0';
Edit1.SetFocus;
tampung2.Free;
end;

procedure Tf_pembelian.Button4Click(Sender:
TObject);
begin
ListBox2.Items.Delete(ListBox1.Itemindex);
ListBox3.Items.Delete(ListBox1.Itemindex);
ListBox4.Items.Delete(ListBox1.Itemindex);
ListBox1.DeleteSelected;
end;

procedure
Tf_pembelian.DBLookupComboBox3Click(Send
er: TObject);
begin
Edit2.Text:='0';
Edit4.Text:='0';
end;

procedure Tf_pembelian.Edit1KeyPress(Sender:
TObject; var Key: Char);
begin
if not (Key in ['0'..'9','a'..'z','A'..'Z',#8,#13])  then
Key:=#0;
end;

procedure Tf_pembelian.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
Action:=cafree;
DM.ADOT_PEMBELIAN.Close;
DM.ADOT_SUPPLIER.Close;
DM.ADOT_PENGGUNA.Close;
DM.ADOT_BARANG.Close;
f_utama_sales.Mn_pembelian.Enabled:=true;

end;

end.


unit u_penjualan;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,
  Dialogs, Menus, Grids, DBGrids, ExtCtrls,
StdCtrls, Mask, DBCtrls,
```

```
  ComCtrls;

type
  Tf_penjualan = class(TForm)
    Label1: TLabel;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    DateTimePicker1: TDateTimePicker;
    Label2: TLabel;
    Label5: TLabel;
    Label7: TLabel;
    Label6: TLabel;
    Bevel1: TBevel;
    Label3: TLabel;
    Label4: TLabel;
    Label8: TLabel;
    Label9: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    DBEdit7: TDBEdit;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit4: TEdit;
    GroupBox1: TGroupBox;
    ListBox1: TListBox;
    GroupBox2: TGroupBox;
    ListBox2: TListBox;
    GroupBox3: TGroupBox;
    ListBox3: TListBox;
    GroupBox4: TGroupBox;
    ListBox4: TListBox;
    DBEdit1: TDBEdit;
    DBLookupComboBox2:
TDBLookupComboBox;
    DBLookupComboBox3:
TDBLookupComboBox;
    DBLookupComboBox1:
TDBLookupComboBox;
    Edit3: TEdit;
    Button3: TButton;
    Button4: TButton;
    DBEdit3: TDBEdit;
    Label10: TLabel;
    procedure Button1Click(Sender:
TObject);
    procedure FormCreate(Sender:
TObject);
    procedure Edit2Change(Sender:
TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure Edit2KeyPress(Sender: TObject; var
Key: Char);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure
DBLookupComboBox3Click(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var
Key: Char);
    procedure FormClose(Sender: TObject; var
Action: TCloseAction);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_penjualan: Tf_penjualan;
  tampung:TStringList;
implementation

uses u_dm, u_menu_utama;

{$R *.dfm}


procedure Tf_penjualan.Button1Click(Sender:
TObject);
var
jml,stok_awal,stok_akhir:integer;
begin

stok_awal:=StrToInt(DBEdit3.Text);
jml:=StrToInt(Edit2.Text);
stok_akhir:=stok_awal-jml;




if (Edit1.Text='') or
(DBLookupComboBox1.text='')
or (DBLookupComboBox2.text='') or
(DBLookupComboBox3.text='')
or (Edit2.Text='') or (DBEdit1.Text='')  or
(Edit4.Text='')

then begin
      ShowMessage('Inputan ada yang kurang!!,
silahkan cek ulang');
```

```
      exit;
end else if  (Edit2.Text='0') then begin
      ShowMessage('Masukan Jumlah
Penjualan!');
      Edit2.SetFocus;
      exit;
end else if (StrToInt(Edit2.Text) >
StrToInt(DBEdit3.Text)) then begin
  ShowMessage('Stok tidak mencukupi!');
  exit;
end;

tampung.Add(IntToStr(stok_akhir));
ListBox1.Items.Add(DBLookupComboBo
x3.Text);
ListBox2.Items.add(DBEdit1.Text);
ListBox3.Items.Add(Edit2.Text);
ListBox4.Items.add(Edit4.Text);
Edit2.Text:='0';
Edit4.Text:='0';
end;

procedure
Tf_penjualan.FormCreate(Sender:
TObject);
begin
tampung:=TStringList.Create;
DateTimePicker1.Date:=now;
end;

procedure
Tf_penjualan.Edit2Change(Sender:
TObject);
var a,b,c:Integer;
begin


if Length(Edit2.Text)=0 then
edit2.text:='0';
   a:=StrToInt(Edit2.Text);
   b:=StrToInt(DBEdit1.Text);
   c:=a*b;
   Edit4.Text:=IntToStr(c);
end;

procedure
Tf_penjualan.Button2Click(Sender:
TObject);
const petik:char='';
var
perintah1:string;
i,j,harga,total:integer;
begin

if ListBox1.Items.Count > 0  then begin
total:=0;
for i:=0 to ListBox4.Items.Count-1 do begin
          harga:=StrToInt(ListBox4.Items[i]);
          total:=total+harga;
end;


    perintah1:='INSERT INTO t_penjualan '+
           'VALUES ( '+
           petik+Edit1.Text+petik+','+#13+

petik+DateToStr(DateTimePicker1.Date)
+petik+','+#13+

petik+DBLookupComboBox1.Text+petik+','+#13
+
           petik+IntToStr(total)+petik+','+#13+

petik+DBLookupComboBox2.Text+petik+#13+
           ' )';



     try

      DM.ADOConnection1.Execute(perintah1);
      for j:=0 to ListBox1.Items.Count-1 do begin


DM.ADOConnection1.Execute('INSERT INTO
t_detail_penj '+
           'VALUES ( '+
           petik+Edit1.Text+petik+','+#13+
           petik+ListBox1.Items[j] +petik+','+#13+
           petik+ListBox3.Items[j]+petik+','+#13+
           petik+ListBox2.Items[j]+petik+','+#13+
           petik+ListBox4.Items[j]+petik+#13+
           ' )');



DM.ADOConnection1.Execute('UPDATE
t_barang SET '+
           'stok = '+tampung.Strings[j]+#13+
           'WHERE kd_barang =
'+petik+ListBox1.Items[j]+petik);
```

```
      end;

    Except
        ShowMessage('NO FAKTUR
SUDAH TERDAFTAR SILAHKAN
GUNAKAN YANG LAIN!');
        Button3Click(sender);
        EXIT;
    end;

Edit3.Text:=IntToStr(total);
ShowMessage('Transaksi di simpan!');
DM.ADOT_BARANG.Refresh;
end else ShowMessage('Pilih barang
terlebih dahulu!');
tampung.Free;
end;

procedure
Tf_penjualan.FormShow(Sender:
TObject);
begin
Edit1.SetFocus;
end;

procedure
Tf_penjualan.Edit2KeyPress(Sender:
TObject; var Key: Char);
begin
if not (Key in ['0'..'9',#8,#13])  then
Key:=#0;
end;

procedure
Tf_penjualan.Button3Click(Sender:
TObject);
begin
ListBox1.Items.Clear;
ListBox2.Items.Clear;
ListBox3.Items.Clear;
ListBox4.Items.Clear;
Edit1.Clear;
Edit2.Text:='0';
Edit4.Text:='0';
Edit1.SetFocus;
tampung.Free;
end;

procedure Tf_penjualan.Button4Click(Sender:
TObject);
begin
ListBox2.Items.Delete(ListBox1.Itemindex);
ListBox3.Items.Delete(ListBox1.Itemindex);
ListBox4.Items.Delete(ListBox1.Itemindex);
ListBox1.DeleteSelected;
end;

procedure
Tf_penjualan.DBLookupComboBox3Click(Sende
r: TObject);
begin
Edit2.Text:='0';
Edit4.Text:='0';
end;

procedure Tf_penjualan.Edit1KeyPress(Sender:
TObject; var Key: Char);
begin
if not (Key in ['0'..'9','a'..'z','A'..'Z',#8,#13])  then
Key:=#0;
end;

procedure Tf_penjualan.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
Action:=caFree;
      DM.ADOT_PENJUALAN.Close;
      DM.ADOT_KONSUMEN.Close;
      DM.ADOT_BARANG.Close;
      DM.ADOT_PENGGUNA.Close;

f_utama_sales.Mn_Penjualan.Enabled:=true;
end;

end.


unit u_supplier;

interface

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

type
```

```
  Tf_supplier = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    DBEdit3: TDBEdit;
    Label4: TLabel;
    Label3: TLabel;
    DBEdit4: TDBEdit;
    Label5: TLabel;
    DBEdit5: TDBEdit;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender:
TObject);
    procedure FormClose(Sender: TObject;
var Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure
keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_supplier: Tf_supplier;

implementation

uses u_data_pengguna, u_detail_penjualan,
u_jenis, u_konsumen, u_login,
  u_merk, u_pemb_langsung,
  u_ukuran, u_dm, u_barang, u_detail_pembelian,
u_menu_utama;



{$R *.dfm}

procedure
TF_supplier.enable_bitbtn_assisten(enable:boolea
n);
begin
      keadaan_awal(not enable);
      BitBtn_Tambah_Assisten.Enabled:=enable;
      BitBtn_Edit_Assisten.Enabled:=enable;
      BitBtn_Hapus_Assisten.Enabled:=enable;
      BitBtn_Batal_Assisten.Enabled:=not enable;
      BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_supplier.keadaan_awal(keadaan:boolean);
begin
      DBEdit1.Enabled:=keadaan;
      DBEdit2.Enabled:=keadaan;
      DBEdit3.Enabled:=keadaan;
      DBEdit4.Enabled:=keadaan;
      DBEdit5.Enabled:=keadaan;
end;

procedure Tf_supplier.FormShow(Sender:
TObject);
begin
      enable_bitbtn_assisten(True);
end;

procedure
Tf_supplier.BitBtn_Tambah_AssistenClick(Sende
r: TObject);
begin
      DM.ADOT_SUPPLIER.Append;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_supplier.BitBtn_Edit_AssistenClick(Sender:
TObject);
```

```
begin
      DM.ADOT_SUPPLIER.Edit;
      enable_bitbtn_assisten(False);
      DBEdit1.SetFocus;
end;

procedure
Tf_supplier.BitBtn_Simpan_AssistenClick
(Sender: TObject);
begin
      if
(DBEdit1.Text='')and(DBEdit2.Text='')
then
      begin
        MessageDlg('Afawan isi data
Assisten-nya dulu
!',mtConfirmation,[mbOK],0);
        DBEdit1.SetFocus;
      end
      else
      begin
        try
          DM.ADOT_SUPPLIER.Post;
        except
          DM.ADOT_SUPPLIER.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_supplier.BitBtn_Batal_AssistenClick(S
ender: TObject);
begin
      DM.ADOT_SUPPLIER.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_supplier.BitBtn_Hapus_AssistenClick(
Sender: TObject);
begin
      if MessageDlg('Apakah Antum yakin
datanya akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then
        DM.ADOT_SUPPLIER.Delete
      else
        DM.ADOT_SUPPLIER.Cancel;
end;

procedure Tf_supplier.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_SUPPLIER.State=DsInsert)or(DM.
ADOT_SUPPLIER.State=DsEdit) then
      if MessageDlg('Sebelum keluar, apakah data
akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
        BitBtn_Simpan_AssistenClick(Sender)
      else
        BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_SUPPLIER.Close;
f_utama_sales.Mn_supplier.Enabled:=true;
end;

end.


unit u_ukuran;

interface

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

type
  Tf_ukuran = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    BitBtn_Tambah_Assisten: TBitBtn;
    BitBtn_Edit_Assisten: TBitBtn;
    BitBtn_Simpan_Assisten: TBitBtn;
    BitBtn_Batal_Assisten: TBitBtn;
    BitBtn_Hapus_Assisten: TBitBtn;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBNavigator1: TDBNavigator;
    DBGrid1: TDBGrid;
    procedure FormShow(Sender: TObject);
    procedure
BitBtn_Tambah_AssistenClick(Sender: TObject);
    procedure BitBtn_Edit_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Simpan_AssistenClick(Sender: TObject);
```

```
    procedure
BitBtn_Batal_AssistenClick(Sender:
TObject);
    procedure
BitBtn_Hapus_AssistenClick(Sender:
TObject);
    procedure FormClose(Sender: TObject;
var Action: TCloseAction);
  private
     procedure
enable_bitbtn_assisten(enable:boolean);
     procedure
keadaan_awal(keadaan:boolean);
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_ukuran: Tf_ukuran;

implementation

uses u_barang, u_data_pengguna,
u_detail_pembelian, u_detail_penjualan,
  u_dm, u_jenis, u_konsumen, u_login,
u_menu_utama, u_merk,
u_pemb_langsung,
    u_supplier;




{$R *.dfm}

procedure
Tf_ukuran.enable_bitbtn_assisten(enable:b
oolean);
begin
       keadaan_awal(not enable);

BitBtn_Tambah_Assisten.Enabled:=enable
;

BitBtn_Edit_Assisten.Enabled:=enable;

BitBtn_Hapus_Assisten.Enabled:=enable;
       BitBtn_Batal_Assisten.Enabled:=not
enable;
       BitBtn_Simpan_Assisten.Enabled:=not
enable;
end;

procedure
Tf_ukuran.keadaan_awal(keadaan:boolean);
begin
       DBEdit1.Enabled:=keadaan;
       DBEdit2.Enabled:=keadaan;

end;

procedure Tf_ukuran.FormShow(Sender:
TObject);
begin
       enable_bitbtn_assisten(True);
end;

procedure
Tf_ukuran.BitBtn_Tambah_AssistenClick(Sender
: TObject);
begin
       DM.ADOT_UKURAN.Append;
       enable_bitbtn_assisten(False);
       DBEdit1.SetFocus;
end;

procedure
Tf_ukuran.BitBtn_Edit_AssistenClick(Sender:
TObject);
begin
       DM.ADOT_UKURAN.Edit;
       enable_bitbtn_assisten(False);
       DBEdit1.SetFocus;
end;

procedure
Tf_ukuran.BitBtn_Simpan_AssistenClick(Sender:
TObject);
begin
       if (DBEdit1.Text='')and(DBEdit2.Text='')
then
       begin
         MessageDlg('Afawan isi data Assisten-nya
dulu !',mtConfirmation,[mbOK],0);
         DBEdit1.SetFocus;
       end
       else
       begin
         try
           DM.ADOT_UKURAN.Post;
```

```
        except
          DM.ADOT_UKURAN.Cancel;
        end;
          enable_bitbtn_assisten(True);
      end;
end;

procedure
Tf_ukuran.BitBtn_Batal_AssistenClick(Se
nder: TObject);
begin
      DM.ADOT_UKURAN.Cancel;
      enable_bitbtn_assisten(True);
end;

procedure
Tf_ukuran.BitBtn_Hapus_AssistenClick(S
ender: TObject);
begin
      if MessageDlg('Apakah Antum yakin
datanya akan dihapus
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then
        DM.ADOT_UKURAN.Delete
      else
        DM.ADOT_UKURAN.Cancel;
end;

procedure Tf_ukuran.FormClose(Sender:
TObject;
  var Action: TCloseAction);
begin
      if
(DM.ADOT_UKURAN.State=DsInsert)or(
DM.ADOT_UKURAN.State=DsEdit) then
      if MessageDlg('Sebelum keluar,
apakah data akan disimpan
?',mtConfirmation,[mbYes,mbNo],0)=mrY
es then

BitBtn_Simpan_AssistenClick(Sender)
      else

BitBtn_Batal_AssistenClick(Sender);
Action:=caFree;
DM.ADOT_UKURAN.Close;
f_utama_sales.Mn_ukuran.Enabled:=true;
end;

end.
```

```
unit u_cetak_barang;

interface

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

type
  Tf_cetak_barang = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRLabel11: TQRLabel;
    QRLabel12: TQRLabel;
    QRLabel13: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRDBText7: TQRDBText;
    QRDBText8: TQRDBText;
    QRDBText9: TQRDBText;
    QRDBText10: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_cetak_barang: Tf_cetak_barang;

implementation

uses u_dm;
```

```
{$R *.dfm}

end.


unit u_cetak_det_pemb;

interface

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

type
  Tf_cetak_det_pemb = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    procedure
QuickRep1BeforePrint(Sender:
TCustomQuickRep;
      var PrintReport: Boolean);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_cetak_det_pemb: Tf_cetak_det_pemb;

implementation

uses u_dm, u_detail_penjualan;

{$R *.dfm}

procedure
Tf_cetak_det_pemb.QuickRep1BeforePrint(Sende
r: TCustomQuickRep;
  var PrintReport: Boolean);
begin
QRDBText1.DataField:='no_faktur';
QRDBText2.DataField:='kd_barang';
QRDBText3.DataField:='jml_barang';
QRDBText4.DataField:='harga';
QRDBText5.DataField:='total_bayar';
end;

end.


unit u_cetak_det_penj;

interface

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

type
  Tf_cetak_det_penj = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    procedure QuickRep1BeforePrint(Sender:
TCustomQuickRep;
      var PrintReport: Boolean);
  private
    { Private declarations }
  public
```

```
    { Public declarations }
  end;

var
  f_cetak_det_penj: Tf_cetak_det_penj;

implementation

uses u_dm, u_detail_penjualan;

{$R *.dfm}

procedure
Tf_cetak_det_penj.QuickRep1BeforePrint(
Sender: TCustomQuickRep;
  var PrintReport: Boolean);
begin
QRDBText1.DataField:='no_faktur';
QRDBText2.DataField:='kd_barang';
QRDBText3.DataField:='jml_barang';
QRDBText4.DataField:='harga';
QRDBText5.DataField:='total_bayar';
end;

end.


unit u_cetak_konsumen;

interface

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

type
  Tf_cetak_konsumen = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_cetak_konsumen: Tf_cetak_konsumen;

implementation

uses u_dm;

{$R *.dfm}

end.


unit u_cetak_pengguna;

interface

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

type
  Tf_cetak_pengguna = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
```

```
  f_cetak_pengguna: Tf_cetak_pengguna;

implementation

uses u_dm;

{$R *.dfm}

end.


unit u_cetak_supplier;

interface

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

type
  Tf_cetak_suplier = class(TForm)
    QuickRep1: TQuickRep;
    ColumnHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  f_cetak_suplier: Tf_cetak_suplier;

implementation

uses u_dm;
```

```
{$R *.dfm}

end.
```

LAMPIRAN C

KUESIONER

KUESIONER

SISTEM INFORMASI PENJUALAN DAN PEMBELIAN BAHAN BAKU BANGUNAN DI. PD. PEMBANGUNAN RAYA BERBASIS *CLIENT_SERVER*

Nama: ……

1. Sistem Informasi Penjualan dan Pembelian yang dibangun memudahkan bagi pengguna untuk mengolah data penjualan dan pembelian tiap karyawan?

   a. Sangat Mudah

   b. Cukup Mudah

   c Mudah

   d. Kurang Mudah

   e. Sangat Tidak Mudah

2. Apakah aplikasi yang dibangun sudah dapat membantu Mempercepat dalam menghasilkan informasi agar tidak terjadi keterlambatan dalam memperoleh informasi?

   a. Sangat Membantu

   b. Cukup Membantu

   c. Membantu

d. Kurang Membantu

e. Sangat Tidak Membantu

3. Apakah aplikasi yang dibangun sudah dapat mengurangi kesalahan dalam pengolahan penjualan dan pembelian?

a. Sangat Mengurangi

b. Cukup Mengurangi

c. Mengurani

d. Kurang Mengurangi

e. Sangat Tidak Mengurangi

4. Apakah aplikasi yang dibangun sudah dapat menyajikan data penjualan dan pembelian secara cepat dan akurat?

a. Sangat akurat

b. Cukup akurat

c. akurat

d. Kurang akurat

e. Sangat Tidak akurat

5. Sistem Informasi Penjualan dan Pembelian yang dibangun mudah untuk dipelajari dan digunakan oleh pengguna?

   a. Sangat mudah

   b. Cukup mudah

   c. mudah

   d. Kurang mudah

   e. Sangat Tidak mudah

Bandung, …… 2009

(……)

LAMPIRAN D
SURAT PENELITIAN

PD. PEMBANGUNAN RAYA
Sedia : Macam - Macam Bahan Bangunan, Sanitary dan
Segala Barang Keperluan Kamar Mandi
Jl. Olo Iskandardinata 372 (022 - 4206604 - 4200612)
Fax. 4206604 - Bandung 40241

Nomor : 002/BLS-KP/2009
Lampiran : -
Perihal : Keterangan Izin Penelitian

Kepada Yth :
Ketua Jurusan Teknik Informatika
Fakultas Teknis Dan Ilmu Komputer
Universitas Komputer Indonesia
Jl. Dipatiukur 112 Bandung

Dengan Hormat,

Dengan hormat,
Sehubungan surat permohonan No. 442 /LP-S1/TIF-FTIK/UNIKOM/IV/2009 yang telah kami terima mengenai permohonan kerja praktek, maka kami sampaikan mahasiswa tersebut di bawah ini :

Nama	: Gugun Sukmana
NIM	: 10104036
Jurusan	: Teknik Informatika
Program Studi	: S1

Dapat diterima untuk melakukan Kerja Praktek di Pd. Pembangunan Raya, mulai 28 April s/d 28 juli 2009 dan kepada mahasiswa tersebut diharapkan dapat mematuhi semua ketentuan serta peraturan yang berlaku di perusahaan kami.
Demikian atas perhatiannya saya ucapkan terimakasih.

Bandung, 23 April 2009

PD. PEMBANGUNAN RAYA

PD. PEMBANGUNAN RAYA
Sedia : Macam - Macam Bahan Bangunan, Sanitary dan
Segala Barang Keperluan Kamar Mandi
Jl. Olo Iskandardinata 372 (022 - 4206604 - 4200612)
Fax. 4206604 - Bandung 40241

Nomor : 004/BLS-KP/2009
Perihal : Pemakaian Software

Kepada Yth :
Ketua Jurusan Teknik Informatika
Fakultas Teknis Dan Ilmu Komputer
Universitas Komputer Indonesia
Jl. Dipatiukur 112 Bandung

Dengan Hormat,

Dengan ini kami menerangkan bahwa mahasiswa di bahwa ini :

Nama	: Gugun Sukmana
NIM	: 10104036
Jurusan	: Teknik Informatika
Program Studi	: S1

Telah selesai melakukan penelitian di Pd. Pembangunan Raya. Sehubungan dengan hal tersebut, kami memberitahukan bahwa software yang telah dibuat oleh mahasiswa tersebut telah digunakan di perusahaan kami.

Atas perhatiannya kami ucapkan terima kasih.

Bandung, 15 Agustus 2009

PD. PEMBANGUNAN RAYA