SISTEM INFORMASI PENJUALAN KORAN DAN MAJALAH DI MAJU TERUS AGENCY

SKRIPSI

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

ELVETTA HAYATUNNISA BARNITAN
10105132

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

LEMBAR PENGESAHAN

SISTEM INFORMASI PENJUALAN KORAN DAN MAJALAH DI MAJU TERUS AGENCY

ELVETTA HAYATUNNISA BARNITAN
10105132

Pembimbing

Linda Salma A, S.Si., M.T.
NIP. 41277006004

Menyetuji,

Ketua Jurusan Teknik Informatika

Mira Kania Sabariah, S.T., M.T.
NIP. 41277006008

LEMBAR PENGESAHAN

SISTEM INFORMASI PENJUALAN KORAN DAN MAJALAH DI MAJU TERUS AGENCY

ELVETTA HAYATUNNISA BARNITAN
10105132

Penguji 1

Irawan Afrianto, S.T.
NIP. 41277006009

Penguji 2

Linda Salma A, S.Si., M.T.
NIP. 41277006004

Penguji 3

Dian Dharmayanti, S.T.
NIP. 41277006005

Bandung, Agustus 2009

Perihal: Plagiat Tugas Akhir

Saya yang bertanda tangan di bawah ini:

Nama : Elvetta Hayatunnisa Barnitan
Nim : 10105132
Judul Tugas Akhir : Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency

Menyatakan bahwa saya tidak melakukan tindakan meniru, menyalin atau menjiplak tugas akhir/karya ilmiah yang telah ada. 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 pertanyaan

Materai
6000

(Elvetta Hayatunnisa Barnitan)

Tak ada manusia yang terlahir sempurna,
Jangan kau sesali segala yang telah terjadi…
Kita pasti pernah dapatkan cobaan yang berat
Seakan hidup ini tak ada artinya lagi…

Syukuri apa yang ada, Hidup adalah anugerah
Tetap jalani hidup ini, Melakukan yang terbaik…

Tuhan pasti kan menunjukkan Kebesaran dan
KuasaNya, bagi hambanya yang sabar dan tak
kenal putus asa…
…Jangan Menyerah…

D'masiv (Jangan Menyerah - 2009)

Dedicated to my beloved parents

ABSTRAK

SISTEM INFORMASI PENJUALAN KORAN DAN MAJALAH DI MAJU TERUS AGENCY

Oleh

Elvetta Hayatunnisa Barnitan
10105132

Pada era globalisasi seperti sekarang ini, teknologi informasi pun mulai sangat dibutuhkan oleh masyarakat luas. Perkembangan teknologi tidak hanya merambah di perusahaan yang sudah maju saja tapi di perusahaan yang sedang berkembang pun teknologi sudah menjadi kebutuhan yang wajib dipenuhi. Seiring dengan perkembangan teknologi yang ada pada saat ini, Sistem Infomasi merupakan salah satu teknologi yang paling dibutuhkan oleh perusahaan untuk melakukan pengolahan data, dimana data merupakan asset penting dalam menghadapi persaingan di dunia bisnis saat ini.

Pemanfaatan informasi yang tepat dapat mendukung keberlangsungan organisasi dalam menghadapi persaingan. Informasi mutlak diperlukan oleh setiap organisasi yang bergerak dibidang apapun tak terkecuali Maju Terus Agency.

Maju Terus Agency adalah salah satu perusahaan yang bergerak dibidang penyaluran Koran dan majalah. Salah satu layanan yang diberikan oleh Maju Terus Agency adalah penjualan dalam jumlah besar juga kecil. Semua barang yang ada di Maju Terus Agency didapat dari beberapa *supplier* (perusahaan penerbit). Banyak masalah yang timbul di Maju Terus Agency khususnya masalah pendataan barang, *supplier*, pelanggan, transaksi penjualan, dan transaksi pembelian. Mengelola data tersebut bukanlah suatu pekerjaan yang mudah. Banyaknya data yang harus diolah setiap hari sehingga memerlukan suatu sistem yang teratur.

Oleh karena itu dengan dibuatnya Sistem Informasi Penjualan Koran dan Majalah ini akan mempermudah dan mempercepat dalam mendapatkan informasi mengenai *supplier*, pelanggan, informasi data barang, pembuatan catatan transaksi pembelian dan penjualan serta pembuatan laporan dan tentunya mendukung kinerja Maju Terus Agency sebagai salah satu perusahaan peyalur surat kabar dengan memberikan fasilitas aplikasi yang lengkap.

Kata Kunci: Sistem Informasi, Penjualan

ABSTRACT

INFORMATION SYSTEM of NEWSPAPER AND MAGAZINE SALE IN MAJU TERUS AGENCY

By

Elvetta Hayatunnisa Barnitan
10105132

In the globalization era, people begin to need information system technology very much. The development of technology is not only spreading in a big company but also in a developing company. This technology has been a compulsory need. Along with the development of technology nowadays, information system becomes a technology which is needed by company to do data processing, where the data is an important asset in facing the competition in business world today.

The efficiently of information can support the organization persistence in facing the competition. Information is absolutely needed by every organization which active in any kind of field include in Maju Terus Agency.

Maju Terus Agency is a company which active in the distributor of newspaper and magazine. One of the services given by Maju Terus Agency is a big or small amount of sale all items in Maju Terus Agency come from supplier (distributor). Many problem rises in Maju Terus Agency, especially about data collection supplier, customer, and also sale and purchase transaction. Data processing is not an easy job. Many data have to be processed every day so it needs an arranged system.

Because of that, the production of the Information System of Newspaper and Magazine Sale will make the process of getting information of supplier, customer, data collection, the production of sale and purchase transaction note, and also the report production easier and faster. This also support the activity of Maju Terus Agency as a newspaper distributor which gives a full application facilities.

Keywords: Information System, Sale

KATA PENGANTAR

Laporan tugas akhir ini merupakan hasil dari penelitian yang telah dilakukan di Maju Terus Agency Bandung, penyusunan laporan tugas akhir ini merupakan salah satu syarat kelulusan akademik pada program studi strata 1 teknik informatika fakultas teknik dan ilmu komputer universitas komputer Indonesia.

Selama waktu pengerjaan tugas akhir ini banyak sekali pengalaman berharga yang di dapatkan, serta bantuan dan bimbingan, baik bimbingan dari pihak Maju Terus Agency, maupun bimbingan dari pihak Universitas Komputer Indonesia serta nasihat dari orang-orang terdekat. Oleh karena itu pada kesempatan ini tak lupa juga untuk mengucapkan terima kasih kepada:

1. Allah SWT atas segala rahmat dan karuniaNya dan bimbinganNya, sehingga dalam melakukan penelitian serta, penyusunan laporan tugas akhir ini terlaksana dengan baik.
2. Orang tua dan keluarga tercinta, yang memberikan dukungan moril maupun materil, dalam penelitian dan penyusunan laporan tugas akhir.
3. Bapak H. Aan Rohmana, selaku pemilik dari Maju Terus Agency, yang telah mengijinkan saya melakukan penelitian.
4. Ibu Linda Salma A, S.Si, M.T. selaku pembimbing, yang Alhamdulillah telah membimbing saya dengan sangat sabar.
5. Ibu Mira Kania Sabariah, S.T, M.T. selaku ketua jurusan teknik informatika dan dosen wali kelas IF-3 2005.

6. Aa Rian, yang selalu menemani saya dan memberikan dukungan moril dalam penelitian dan penyusunan laporan tugas akhir.
7. A' Husni, yang telah memberikan dukungan dalam mengerjakan program tugas akhir.
8. Teh Tia, yang telah bersedia meminjamkan laptop selama masa pengerjaan dan penyusunan laporan tugas akhir.
9. A' Atjep "Agency", yang telah bersedia diwawancara untuk proyek penelitian tugas akhir.
10. Nadya, Galih, Mmy, Heri "Kiplie", Yudi "Onde", teman-teman yang tiada hentinya mendukung dan memberi bantuan serta informasi selama masa pengerjaan proyek tugas akhir.
11. Debora, Eka "Abok", Dinda, Jelly, Sheni, Irfan, Ecep, teman-teman if-3 seperjuangan dalam melancarkan *event* wisuda 2009.
12. Seluruh anak-anak kelas IF-3 2005 yang sangat kompak.
13. Terakhir kepada semua pihak yang telah membantu yang tidak bisa saya sebutkan satu-persatu. Terima kasih.

Dalam pengerjaan laporan tugas akhir ini, tidak terlepas dari kekurangan, oleh karena itu sangat diharapkan sekali kritik dan saran yang sifatnya membangun, untuk menciptakan laporan ini lebih baik lagi, semoga laporan ini dapat bermanfat khususnya bagi khalayak ramai.

Bandung, Agustus 2009

Penyusun

DAFTAR ISI

LEMBAR JUDUL

LEMBAR PENGESAHAN

ABSTRAK i

ABSTRACT ii

KATA PENGANTAR iii

DAFTAR ISI v

DAFTAR TABEL x

DAFTAR GAMBAR xiii

DAFTAR SIMBOL xvi

DAFTAR LAMPIRAN xviii

BAB I PENDAHULUAN 1

1.1 Latar Belakang Masalah 1

1.2 Identifikasi Masalah 2

1.3 Maksud dan Tujuan 2

1.4 Batasan Masalah 3

1.5 Metodologi Penelitian 4

1.6 Sistematika Penulisan 7

BAB II TINJAUAN PUSTAKA 9

2.1 Tinjauan Perusahaan 9

2.1.1 Sejarah Perusahaan 9

2.1.2 Visi dan Misi 10
2.1.3 Struktur Organisasi 11
2.2 Landasan Teori.....11
2.2.1 Konsep Dasar Sistem.....11
2.2.1.1. Karakteristik Sistem 13
2.2.1.2. Klasifikasi Sistem.....16
2.2.1.3. Pelaku Sistem 20
2.2.1.4. Hal Mendasar Dalam Pengembangan Sistem 22
2.2.2 Konsep Dasar Informasi 23
2.2.3 Konsep Dasar Sistem Informasi 25
2.2.3.1. Komponen Fisik Sistem Informasi.....27
2.2.3.2. Hubungan Pengelola Dengan Sistem Informasi.....28
2.2.3.3. Pendekatan Sistem Informasi 29
2.2.4 Alat Pengembangan Sistem.....30
2.2.4.1. Diagram Arus Data (Data Flow Diagram) 30
2.2.4.2. Entity Relationship Diagram (ERD) 30
2.2.4.3. Kamus Data (Data Dictionary).....31
2.2.5 Konsep Dasar Pengolahan Data 31
2.2.6 Konsep Basis Data.....32
2.2.7 Software Pendukung.....37
2.2.1. Delphi 7 37
2.2.2. MySQL.....39
2.2.8. Metode Pengujian.....43

BAB III ANALISIS DAN PERANCANGAN SISTEM 45
3.1 Analisis Sistem.....45
3.1.1 Analisis Sistem yang sedang Berjalan.....46
3.1.1.1. Prosedur Pembelian Barang 47
3.1.1.2. Prosedur Penjualan Barang 49
3.1.1.3. Prosedur Pembuatan Laporan 51
3.1.2 Analisis Kebutuhan Non Fungsional.....53
3.1.2.1. Analisis Pengkodean 53
3.1.2.2. Analisis Karakteristik *User* 55
3.1.2.3. Analisis Kebutuhan Perangkat Lunak 56
3.1.2.4. Analisis Kebutuhan Perangkat Keras 57
3.1.2.5. Analisis Jaringan 58
3.1.3 Analisis *Database*.....59
3.1.3.1. Entity Relationship Diagram 59
3.1.3.2. Kamus Data 61
3.1.4 Analisis Kebutuhan Fungsional.....64
3.1.4.1. Diagram Konteks.....65
3.1.4.2. DFD Level 0.....66
3.1.4.3. DFD Level 1 Proses 2 67
3.1.4.4. DFD Level 1 Proses 3 68
3.1.4.5. DFD Level 1 Proses 4 69
3.1.4.6. DFD Level 1 Proses 5 70
3.1.4.7. DFD Level 2 Proses 2.1 71

3.1.4.8. DFD Level 2 Proses 2.2 72
3.1.4.9. DFD Level 2 Proses 2.3 73
3.1.4.10. DFD Level 2 Proses 2.4 74
3.1.4.11. DFD Level 2 Proses 3.3 75
3.1.4.12. DFD Level 2 Proses 4.3 75
3.1.5 Spesifikasi Proses 76
3.2 Perancangan Sistem 88
3.2.1 Perancangan Basis Data.....88
3.1.3.3. Skema Relasi 88
3.1.3.4. Struktur Tabel.....90
3.2.2 Perancangan Struktur Menu 93
3.2.3 Perancangan Antarmuka.....96
3.2.4 Perancangan Pesan 113
3.2.5 Jaringan Semantik 114

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM.....115
4.1 Implementasi.....115
4.1.1 Implementasi Perangkat Lunak 115
4.1.2 Implementasi Perangkat Keras 115
4.1.3 Implementasi Basisdata (*Database*) 116
4.1.4 Implementasi Antarmuka 123
4.2 Pengujian Alpha.....126
4.2.1. Metode Pengujian.....126

4.2.2. Rencana Pengujian 127
4.2.2.1. Pengujian Login 128
4.2.2.2. Pengujian Data User 129
4.2.2.3. Pengujian Data Barang 132
4.2.2.4. Pengujian Data Supplier 134
4.2.2.5. Pengujian Data Pelanggan 137
4.2.2.6. Pengujian Data Pembelian 139
4.2.2.7. Pengujian Data Penjualan 144
4.2.3. Kesimpulan Hasil Pengujian Alpha 149
4.3 Pengujian Beta 149
4.3.1. Kesimpulan Hasil Pengujian Beta 152

BAB V KESIMPULAN DAN SARAN 154
5.1 Kesimpulan 154
5.2 Saran 154
DAFTAR PUSTAKA 155
LAMPIRAN

DAFTAR TABEL

Tabel III.1. Rincian Karakteristik Pengguna.....55

Tabel III.2. Spesifikasi Komputer yang Ada 57

Tabel III.3. Spesifikasi Minimum Komputer yang Dibutuhkan 58

Tabel III.4. Kamus Data.....61

Tabel III.5. Spesifikasi Proses.....76

Tabel III.6. Struktur Tabel user.....90

Tabel III.7. Struktur Tabel barang.....90

Tabel III.8. Struktur Tabel supplier.....90

Tabel III.9. Struktur Tabel pelanggan.....90

Tabel III.10. Struktur Tabel pembelian.....91

Tabel III.11. Struktur Tabel penjualan.....91

Tabel III.12. Struktur Tabel detail_pembelian.....91

Tabel III.13. Struktur Tabel detail_penjualan.....91

Tabel III.14. Struktur Tabel barang_detail.....92

Tabel III.15. Struktur Tabel buku_hutang.....92

Tabel III.16. Struktur Tabel buku_piutang 92

Tabel III.17. Struktur Tabel returbeli.....92

Tabel III.18. Struktur Tabel returjual.....92

Tabel III.19. Struktur Tabel returbelidetail.....93

Tabel III.20. Struktur Tabel returjualdetail.....93

Tabel IV.1. Tabel kebutuhan perangkat keras 116

Tabel IV.2. Nama Tabel beserta *file* implementasinya 116

Tabel IV.3. Implementasi antar muka Sistem Informasi Penjualan Koran dan Majalah 123

Tabel IV.4. Rencana Pengujian.....127

Tabel IV.5. Verifikasi Username dan Password 128

Tabel IV.6. Pengujian Cari Data User 129

Tabel IV.7. Pengujian Tambah Data User 129

Tabel IV.8. Pengujian Hapus Data User 131

Tabel IV.9. Pengujian Ubah Data User.....131

Tabel IV.10. Pengujian Cari Data Barang 132

Tabel IV.11. Pengujian Tambah Data Barang 132

Tabel IV.12. Pengujian Hapus Data Barang 133

Tabel IV.13. Pengujian Ubah Data Barang.....134

Tabel IV.14. Pengujian Cari Data Supplier 134

Tabel IV.15. Pengujian Tambah Data Supplier 135

Tabel IV.16. Pengujian Hapus Data Supplier 136

Tabel IV.17. Pengujian Ubah Data Supplier.....136

Tabel IV.18. Pengujian Cari Data Pelanggan 137

Tabel IV.19. Pengujian Tambah Data Pelanggan 137

Tabel IV.20. Pengujian Hapus Data Pelanggan 138

Tabel IV.21. Pengujian Ubah Data Pelanggan.....139

Tabel IV.22. Pengujian Cari Data Pembelian 139

Tabel IV.23. Pengujian Tambah Data Pembelian 140

Tabel IV.24. Pengujian Ubah Data Pembelian 141

Tabel IV.25. Pengujian Tambah Retur Pembelian.....142

Tabel IV.26. Pengujian Pembayaran Hutang.....143

Tabel IV.27. Pengujian Cari Data Penjualan 144

Tabel IV.28. Pengujian Tambah Data Penjualan.....145

Tabel IV.29. Pengujian Tambah Retur Penjualan.....146

Tabel IV.30. Pengujian Pembayaran Piutang 148

DAFTAR GAMBAR

Gambar I.1. *Waterfall Model* 6

Gambar II.1. Struktur Organisasi Maju Terus Agency 11

Gambar II.2 Karakteristik Suatu Sistem 16

Gambar II.3. Siklus Informasi.....24

Gambar III.1. Prosedur Pembelian Barang 48

Gambar III.2. Prosedur Penjualan Barang 50

Gambar III.3. Prosedur Pembuatan Laporan.....52

Gambar III.4. Analisis Jaringan Maju Terus Agency 59

Gambar III.5. ER-Diagram 60

Gambar III.6. Diagram Konteks Sistem Informasi Penjualan Koran dan Majalah.....65

Gambar III.7. DFD Level 0 Sistem Informasi Penjualan Koran dan Majalah.....66

Gambar III.8. DFD Level 1 Proses Pengelolaan Data Master 67

Gambar III.9. DFD Level 1 Proses Pengelolaan Pembelian.....68

Gambar III.10. DFD Level 1 Proses Pengelolaan Penjualan.....69

Gambar III.11. DFD Level 1 Proses Pembuatan Laporan 70

Gambar III.12. DFD Level 2 Proses Pengolahan Data User.....71

Gambar III.13. DFD Level 2 Proses Pengolahan Data Barang.....72

Gambar III.14. DFD Level 2 Proses Pengolahan Data Supplier.....73

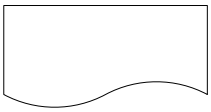
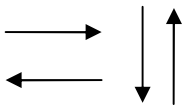
Gambar III.15. DFD Level 2 Proses Pengolahan Data Pelanggan 74

Gambar III.16. DFD Level 2 Proses Pengelolaan Detail Pembelian 75
Gambar III.17. DFD Level 2 Proses Pengelolaan Detail Penjualan 75
Gambar III.18. Skema Relasi 89
Gambar III.19. Struktur Menu Bagian Pembukuan (Administrator) 94
Gambar III.20. Struktur Menu Bagian Pembelian 95
Gambar III.21. Struktur Menu Bagian Penjualan 95
Gambar III.22. Perancangan Tampilan Menu Login 96
Gambar III.23. Perancangan Tampilan Menu setelah Login untuk Bagian Pembukuan (Administrator).....96
Gambar III.24. Perancangan Tampilan Menu setelah Login untuk Bagian Pembelian.....97
Gambar III.25. Perancangan Tampilan Menu setelah Login untuk Bagian Penjualan.....97
Gambar III.26. Perancangan Tampilan Form Data User 98
Gambar III.27. Perancangan Tampilan Form Tambah Data User 98
Gambar III.28. Perancangan Tampilan Form Ubah Data User.....99
Gambar III.29. Perancangan Tampilan Form Data Barang 99
Gambar III.30. Perancangan Tampilan Form Tambah Data Barang 100
Gambar III.31. Perancangan Tampilan Form Ubah Data Barang.....100
Gambar III.32. Perancangan Tampilan Form Data Supplier 101
Gambar III.33. Perancangan Tampilan Form Tambah Data Supplier 101
Gambar III.34. Perancangan Tampilan Form Ubah Data Supplier.....102
Gambar III.35. Perancangan Tampilan Form Data Pelanggan 102

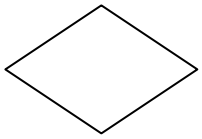
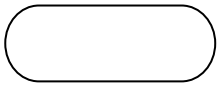
Gambar III.36. Perancangan Tampilan Form Tambah Data Pelanggan 103
Gambar III.37. Perancangan Tampilan Form Ubah Data Pelanggan.....103
Gambar III.38. Perancangan Tampilan Form Pembelian.....104
Gambar III.39. Perancangan Tampilan Form Tambah Pembelian 104
Gambar III.40. Perancangan Tampilan Form Penjualan.....105
Gambar III.41. Perancangan Tampilan Form Tambah Penjualan 105
Gambar III.42. Perancangan Tampilan Form Laporan Data Barang.....106
Gambar III.43. Perancangan Tampilan Form Laporan Data Supplier.....106
Gambar III.44. Perancangan Tampilan Form Laporan Data Pelanggan.....107
Gambar III.45. Perancangan Tampilan Form Laporan Data Pembelian.....107
Gambar III.46. Perancangan Tampilan Form Laporan Data Penjualan.....108
Gambar III.47. Perancangan Tampilan Form Laporan Data Hutang ke Supplier108
Gambar III.48. Perancangan Tampilan Form Laporan Data Piutang Pelanggan.109
Gambar III.49. Perancangan Tampilan Form Retur Pembelian.....109
Gambar III.50. Perancangan Tampilan Form Tambah Retur Pembelian 110
Gambar III.51. Perancangan Tampilan Form Pembayaran Hutang.....110
Gambar III.52. Perancangan Tampilan Form Retur Penjualan.....111
Gambar III.53. Perancangan Tampilan Form Tambah Retur Penjualan.....111
Gambar III.54. Perancangan Tampilan Form Pembayaran Piutang 112
Gambar III.55. Perancangan Tampilan Pesan.....113
Gambar III.56. Jaringan Semantik 114

DAFTAR SIMBOL

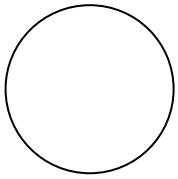
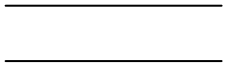
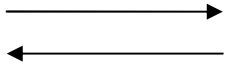
A. Simbol Flowmap

NO	SIMBOL	KETERANGAN
1		Simbol pengolahan (processing symbol): Digunakan untuk pengolahan data dan pemindahan data
2		Simbol dokumen (document symbol):
3		Simbol operasi manual (manual operation symbol):
4		Simbol arsip manual:
5		Simbol tempat penyimpanan data (direct data symbol):
6		Aliran Data: Menunjukan aliran data antar proses.

B. Diagram Hubungan Entitas

NO	SIMBOL	KETERANGAN
1		Data Entitas: Segala sesuatu baik yang nyata mau pun 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.
4		Field: Menyatakan bagian dari suatu tabel.

C. Data Flow Diagram

NO	SIMBOL	KETERANGAN
1		Proses: Menunjukan transformasi dari masukan menjadi keluaran, dalam hal ini sejumlah masukan dapat menjadi hanya satu keluaran atau pun sebaliknya
2		Terminator: Mewakili entitas luar dimana sistem berkomunikasi.
3		Penyimpanan: Untuk memodelkan kumpulan data, atau paket data.
4		Aliran: Menggambarkan gerakan paket data atau informasi dari suatu bagian lain dari sistem dimana sistem penyimpanan mewakili lokasi penyimpanan data.

DAFTAR LAMPIRAN

Lampiran A Tampilan Antar Muka..... A-1

Lampiran B Listing Program B-1

Lampiran C Hasil Kuesioner.....C-1

Lampiran D Dokumen Manual dari Sistem Lama D-1

Lampiran E Surat Penelitian E-1

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada era globalisasi seperti sekarang ini, teknologi informasi pun mulai sangat dibutuhkan oleh masyarakat luas. Perkembangan teknologi tidak hanya merambah di perusahaan yang sudah maju saja tapi di perusahaan yang sedang berkembang pun teknologi sudah menjadi kebutuhan yang wajib dipenuhi. Seiring dengan perkembangan teknologi yang ada pada saat ini, Sistem Infomasi merupakan salah satu teknologi yang paling dibutuhkan oleh perusahaan untuk melakukan pengolahan data, dimana data merupakan asset penting dalam menghadapi persaingan di dunia bisnis saat ini.

Pemanfaatan informasi yang tepat dapat mendukung keberlangsungan organisasi dalam menghadapi persaingan. Informasi mutlak diperlukan oleh setiap organisasi yang bergerak dibidang apapun tak terkecuali Maju Terus Agency.

Maju Terus Agency adalah salah satu perusahaan yang bergerak dibidang penyaluran Koran dan majalah. Salah satu layanan yang diberikan oleh Maju Terus Agency adalah penjualan dalam jumlah besar juga kecil. Semua barang yang ada di Maju Terus Agency didapat dari beberapa *supplier* (perusahaan penerbit). Banyak masalah yang timbul di Maju Terus Agency khususnya masalah pendataan barang, *supplier*, pelanggan, transaksi penjualan, dan transaksi pembelian. Mengelola data tersebut bukanlah suatu pekerjaan yang mudah. Banyaknya data yang harus diolah setiap hari sehingga memerlukan suatu sistem yang teratur.

Selain itu, keterlambatan dalam pembuatan laporan juga sering terjadi karena kesulitan dalam pencarian dan pengolahan data. Semua permasalahan yang timbul disebabkan oleh semua data yang ada hanya berbentuk arsip tertulis baik dalam bentuk buku ataupun faktur. Meskipun dalam melakukan pembuatan laporan sudah menggunakan aplikasi *office*, tetapi resiko munculnya kesalahan dalam pencatatan data sering terjadi.

Berdasarkan latar belakang masalah tersebut, penulis tertarik untuk membangun sebuah aplikasi sistem informasi penjualan koran dan majalah di Maju Terus Agency. Aplikasi ini diharapkan dapat mengurangi permasalahan yang ada dan membantu karyawan dalam mengolah data transaksi penjualan dan pembelian. Serta mempermudah dan menghemat waktu karyawan dalam membuat laporan pendataan koran dan majalah untuk pimpinan.

1.2 Identifikasi Masalah

Berdasarkan uraian latar belakang diatas maka dirumuskan Bagaimana Membangun Aplikasi Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency.

1.3 Maksud dan Tujuan

Berdasarkan permasalahan yang diteliti, maka maksud dari penulisan tugas akhir ini adalah untuk membangun aplikasi sistem informasi untuk mengelola data pembelian dan penjualan koran dan majalah pada Maju Terus Agency.

Sedangkan tujuan yang akan dicapai setelah pembuatan perangkat lunak ini adalah:

1. Membantu proses kerja karyawan dalam mendapatkan informasi tentang persediaan barang, penjualan, pembelian, *supplier* dan pelanggan.
2. Untuk memudahkan *user* pada saat melakukan penambahan data, penghapusan data dan pencarian data serta pembuatan laporan dan pencetakan faktur.
3. Untuk mendukung kinerja agency dengan memberikan fasilitas aplikasi yang lengkap.
4. Untuk memperkecil resiko hilangnya data dari hal yang tidak diinginkan karena pada aplikasi akan dibuatkan *login user*.

1.4 Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, maka dibuat batasan masalah agar ruang lingkup perangkat lunak ini jelas batasannya. Adapun batasan masalah yang dibuat adalah sebagai berikut:

1. Data yang dikelola merupakan seluruh data yang berkaitan dengan penjualan koran dan majalah. Seperti data penerbit (supplier), data pelanggan (sub-agen maupun outlet-toko), data transaksi pembelian dan penjualan, data laporan saldo piutang pelanggan, data laporan saldo hutang ke penerbit, dan data laporan stock opname.
2. Sumber data berasal dari Maju Terus Agency.

3. Sistem yang akan dibangun merupakan sistem informasi berbasis *client-server*.
4. User yang terlibat dalam sistem ini adalah:
    a. Bagian Pembukuan untuk pengolahan data master dan pelaporan
    b. Bagian Pembelian untuk pengolahan pembelian
    c. Bagian Penjualan untuk pengolahan penjualan
5. User yang akan mengakses program aplikasi ini harus login terlebih dahulu, sehingga terjamin keamanan pada saat melakukan pengolahan data.
6. *Tools* yang digunakan adalah Borland Delphi 7.0 sebagai aplikasi dalam membangun *interface* dengan database MySQL Server 5.0.
7. Dalam memodelkan data akan digunakan ERD dan model proses akan menggunakan model terstruktur yaitu Diagram Context dan *Data Flow Diagram* (DFD).

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam penulisan tugas akhir ini adalah sebagai berikut:

1. Tahap pengumpulan data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah sebagai berikut :

a. Studi Literatur.

Pengumpulan data dengan cara mengumpulkan literatur, jurnal, *paper* dan bacaan-bacaan yang ada kaitannya dengan judul penelitian.

b. Observasi.

Teknik pengumpulan data dengan mengadakan penelitian dan peninjauan langsung terhadap permasalahan yang diambil. Digunakan sebagai alat pengumpul data untuk obyek yang belum banyak diketahui (eksploratif), syarat melakukan observasi yaitu:

a) Pengamatan harus dilakukan secara sistematis,
b) Obyek yang diamati harus dalam keadaan wajar (keadaan sebenarnya), tidak dipengaruhi / diatur ataupun dimanipulasi,
c) Obyek yang diamati harus representatif, valid, dan reliabel.

2. Tahap pembuatan perangkat lunak.

Teknik analisis data dalam pembuatan perangkat lunak menggunakan paradigma perangkat lunak secara *waterfall*, yang meliputi beberapa proses diantaranya:

a. System / Information Engineering

Merupakan bagian dari sistem yang terbesar dalam pengerjaan suatu proyek, dimulai dengan menetapkan berbagai kebutuhan dari semua elemen yang diperlukan dan mengalokasikannya kedalam pembentukan perangkat lunak.

b. Analisis

Merupakan tahap menganalisis hal-hal yang diperlukan dalam pelaksanaan proyek pembuatan perangkat lunak.

c. *Design*

Tahap penerjemahan dari data yang dianalisis kedalam bentuk yang mudah dimengerti oleh user.

d. *Coding*

Tahap penerjemahan data atau pemecahan masalah yang telah dirancang keadalam bahasa pemrograman tertentu.

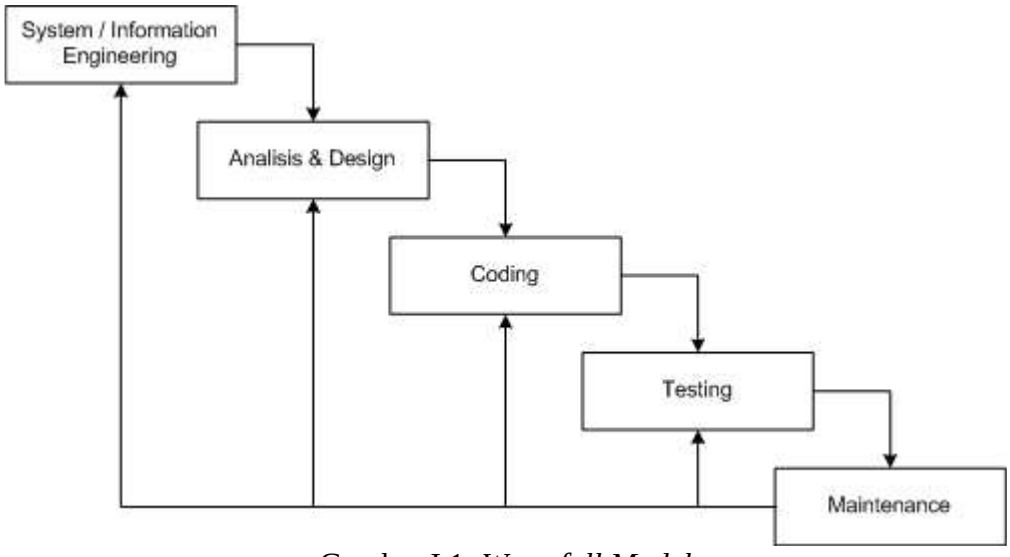
e. *Testing*

Merupakan tahap pengujian terhadap perangkat lunak yang dibangun.

f. *Maintenance*

Tahap akhir dimana suatu perangkat lunak yang sudah selesai dapat mengalami perubahan–perubahan atau penambahan sesuai dengan permintaan *user*.

Alur pada *waterfall model*, bisa dilihat seperti tercantum pada gambar I.1 berikut ini:



Gambar I.1. *Waterfall Model*

1.6 Sistematika Penulisan

Sistematika penulisan laporan skripsi ini disusun untuk memberikan gambaran umum tentang perangkat lunak yang akan dibangun. Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini berisi tentang uraian latar belakang permasalahan, mengidentifikasi permasalahan yang dihadapi, menentukan maksud dan tujuan pembangunan perangkat lunak, pembatasan masalah, menjelaskan metodologi penyusunan laporan, serta sistematika penulisan.

BAB II. LANDASAN TEORI

Bab ini berisi tentang tinjauan umum perusahaan dan pembahasan berbagai konsep dasar mengenai sistem informasi, konsep pengolahan data, dan teori-teori pendukung lainnya yang berkaitan dengan topik pembangunan perangkat lunak yang dilakukan.

BAB III. ANALISIS DAN PERANCANGAN

Bab ini berisi analisis dari sistem yang sedang berjalan dimulai dari prosedur penjualan barang, prosedur pembelian barang, prosedur pembuatan laporan, analisis kebutuhan non fungsional, analisis perangkat lunak, perangkat keras, karakteristik pengguna, analisis jaringan, analisis kebutuhan fungsional termasuk diagram konteks dan *data flow diagram* serta perancangan sistem seperti relasi antar tabel, perancangan basis data, struktur menu, perancangan antarmuka, perancangan pesan dan jaringan semantik.

BAB IV. IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi implementasi dari perangkat lunak, perangkat keras, implementasi basisdata, implementasi antarmuka, serta pengujian dari sistem, yang meliputi pengujian alpha dan pengujian beta.

BAB V. KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil pembahasan bab-bab sebelumnya, serta saran yang merupakan tindak lanjut dari kesimpulan.

BAB II

TINJAUAN PUSTAKA

2.1 Tinjauan Perusahaan

Tinjauan perusahaan berisi tentang sejarah mengenai perusahaan, visi dan misi perusahaan serta sturktur organisasi perusahaan.

2.1.1 Sejarah Singkat

Maju Terus Agency merupakan perusahaan perorangan yang bergerak dibidang penyelenggara informasi serta sarana telekomunikasi berupa surat kabar dan majalah serupa.

Maju Terus Agency pertama kali didirikan oleh Bpk. H. Aan Rohmana pada tahun 1987 yang bertempat di jalan Sukagalih Gg. H. Gojali RT 02 RW 07. Pada mulanya Maju Terus Agency hanya sebagai penyulpai satu jenis surat kabar saja yaitu Koran harian Pikiran Rakyat.

Seiring dengan perkembangan teknologi, mulai banyak jenis-jenis surat kabar yang bermunculan. Tidak ingin ketinggalan, Maju Terus Agency pun mulai gencar menjalin kerja sama dengan agen-agen lain untuk mendapatkan informasi dan suplai barang secara langsung.

Karena semakin meningkatnya permintaan dari pelanggan, akhirnya pada tahun 1996, Maju Terus Agency berpindah kantor ke tempat yang lebih besar dan mulai mendistribusikan Koran dan majalah serupa. Pada tahun yang sama, Maju Terus Agency mendaftarkan perusahaannya sebagai Perusahaan

perseorangan dengan nomor TDP: 101155216418 atas nama H. Aan Rohmana yang terletak di jalan Sukagalih nomor 42 Bandung 40162, yang kegiatan usaha pokoknya sebagai usaha perdagangan hasil percetakan.

Krisis moneter yang melanda Indonesia tahun 1998 tidak menyurutkan usaha Maju Terus Agency. Di saat agency lain mulai gulung tikar dan kehilanggan pelanggannya, Maju Terus Agency tetap dipercaya oleh para pelanggan dan distributor sehingga Maju Terus Agency bisa berkembang pesat seperti sekarang.

2.1.1 Visi dan Misi

2.1.2.1. Visi

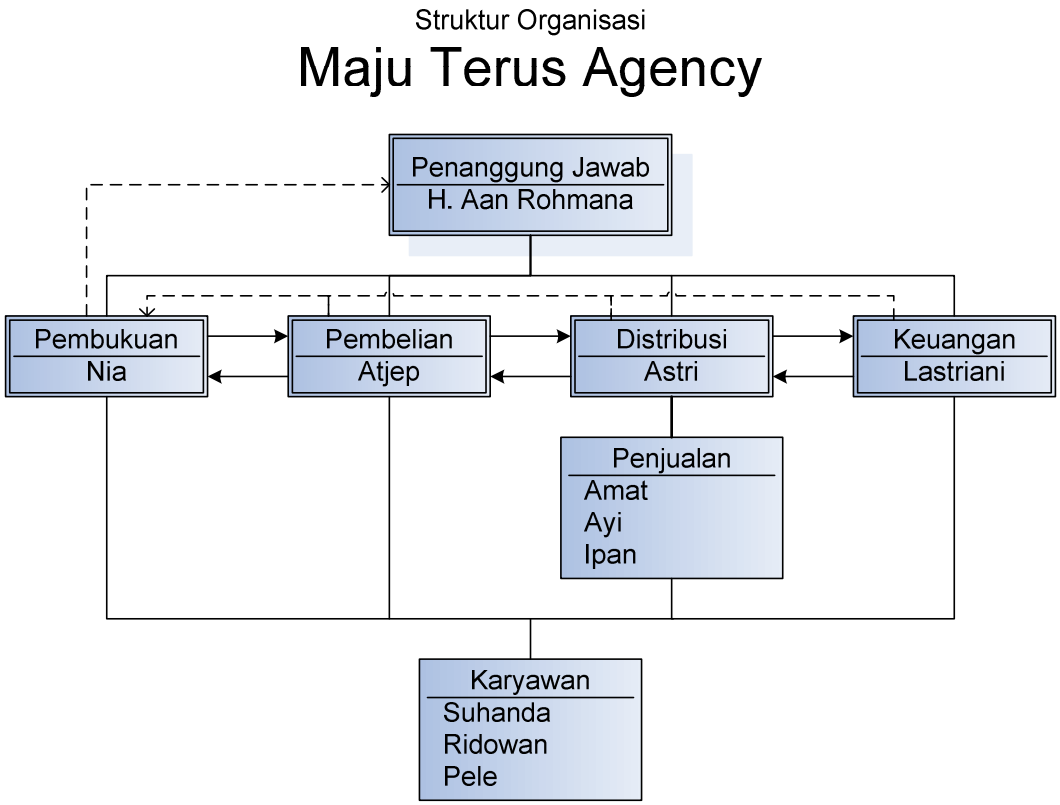
“Menjadi agency yang dipercaya oleh masyarakat”

Maju Terus Agency berupaya untuk menempatkan diri sebagai perusahaan penyalur Koran yang dipercaya dalam pendistribusiannya.

2.1.2.2. Misi

Maju Terus Agency mempunyai misi “Memperluas wawasan bangsa lewat surat kabar dan selalu optimis untuk Maju dalam mengembangkan kepercayaan masyarakat” dengan jaminan bahwa pelanggan akan mendapatkan pelayanan yang baik, berupa kemudahan dalam mendapatkan surat kabar yang diinginkan, dengan harga kompetitif.

2.1.2 Struktur Organisasi



Gambar II.1. Struktur Organisasi Maju Terus Agency

2.2.Landasan Teori

Landasan teori merupakan suatu acuan penting berkenaan dengan teori-teori yang dibutuhkan dalam mengerjakan suatu laporan. Dalam bab ini akan diuraikan beberapa teori pendukung seperti konsep dasar sistem informasi serta teori mengenai software pendukung pembangunan perangkat lunak.

2.2.1 Konsep Dasar Sistem

Pengertian dan definisi sistem pada berbagai bidang berbeda-beda, tetapi meskipun istilah *sistem* yang digunakan bervariasi, semua sistem pada

bidang-bidang tersebut mempunyai beberapa persyaratan umum, yaitu sistem harus mempunyai elemen, lingkungan, interaksi Antar elemen, interaksi antara elemen dengan lingkungannya, dan yang terpenting adalah sistem harus mempunyai tujuan yang akan dicapai.

Berdasarkan persyaratan ini, sistem dapat didefinisikan sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama.

Kumpulan elemen terdiri dari :

a. manusia
b. mesin
c. prosedur
d. dokumen
e. data
f. elemen lain yang terorganisir dari elemen-elemen tersebut

Elemen sistem disamping berhubungan satu sama lain, juga berhubungan dengan lingkungannya untuk mencapai tujuan yang telah ditentukan sebelumnya.

Terdapat beberapa definisi sistem yaitu:

Menurut Gordon B. Davis (1984):
“ Sebuah sistem terdiri dari bagian-bagian yang saling berkaitan yang beroperasi bersama untuk mencapai beberapa sasaran atau maksud “.

Menurut Raymond Mcleod (2001):
“ Sistem adalah himpunan dari unsur-unsur yang saling berkaitan sehingga membentuk suatu kesatuan yang utuh dan terpadu “.

Menurut Jerry FithGerald:
“sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.”

Menurut Ludwig Von Bartalanfy:
“Sistem merupakan seperangkat unsur yang saling terikat dalam suatu antar relasi diantara unsur-unsur tersebut dengan lingkungan.”

Menurut Anatol Raporot:
“Sistem adalah suatu kumpulan kesatuan dan perangkat hubungan satu sama lain.”

Menurut L. Ackof:
“Sistem adalah setiap kesatuan secara konseptual atau fisik yang terdiridari bagian-bagian dalam keadaan saling tergantung satu sama lainnya.”

2.2.1.1. Karakteristik Sistem

Suatu sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu :

1. Komponen-komponen Komponen sistem atau elemen sistem dapat berupa :
    a. Elemen-elemen yang lebih kecil yang disebut *sub sistem*, misalkan sistem komputer terdiri dari sub sistem perangkat keras, perangkat lunak dan manusia.
    b. Elemen-elemen yang lebih besar yang disebut *supra sistem*. Misalkan bila perangkat keras adalah sistem yang memiliki sub sistem CPU, perangkat I/O dan memori, maka supra sistem perangkat keras adalah sistem komputer.
2. Batas sistem

    Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas

sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan, karena dengan batas sistem ini fungsi dan tugas dari subsistem yang satu dengan lainnya berbeda tetapi tetap saling berinteraksi. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan luar sistem

   Segala sesuatu diluar dari batas sistem yang mempengaruhi operasi dari suatu sistem. Lingkungan luar sistem ini dapat bersifat menguntungkan atau merugikan. Lingkungan luar yang menguntungkan harus dipelihara dan dijaga agar tidak hilang pengaruhnya, sedangkan lingkungan luar yang merugikan harus dimusnahkan dikendalikan agar tidak mengganggu operasi sistem.

4. Penghubung

   Penghubung merupakan media perantara antar subsistem. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem lainnya. Output dari satu subsistem akan menjadi input untuk subsistem yang lainnya dengan melalui penghubung.

5. Masukkan

   Merupakan energi yang dimasukkan ke dalam sistem. Masukan dapat berupa Masukan Perawatan (*Maintenance Input*) adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Masukan Sinyal (*Signal Input*) adalah energi yang diproses untuk didapatkan keluaran.

Sebagai contoh didalam sistem komputer, program adalah maintanance input yang digunakan untuk mengoperasikan komputernya dan data adalah signal input untuk diolah menjadi informasi.

6. Keluaran

   Merupakan hasil dari energi yang diolah oleh sistem. Meliputi : Keluaran yangberguna, contohnya Informasi yang dikeluarkan oleh komputer. Dan Keluaran yang tidak berguna yang dikenal sebagai sisa pembuangan, contohnya panas yang dikeluarkan oleh komputer.

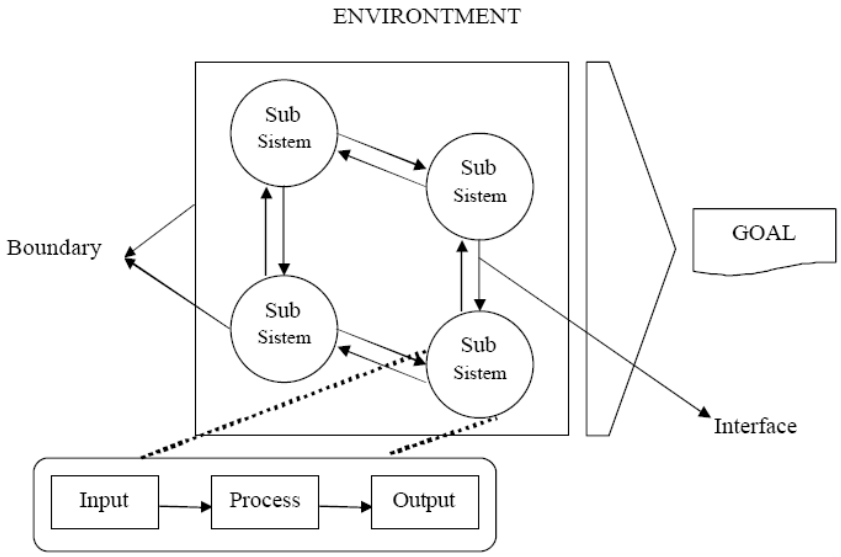
7. Pengolah

   Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya. Pengolah yang akan merubah masukan menjadi keluaran.

   Contoh CPU pada Komputer, Bagian Produksi yang mengubah bahan baku menjadi barang jadi, Bagian akuntansi yang mengolah data transaksi menjadi laporan keuangan.

8. Sasaran atau tujuan

   Setiap sistem pasti mempunyai tujuan ataupun sasaran yang mempengaruhi input yang dibutuhkan dan output yang dihasilkan. Dengan kata lain suatu sistem akan dikatakan berhasil kalau pengoperasian sistem itu mengenai sasaran atau tujuannya. Sistem yang tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya.



Gambar II.2 Karakteristik Suatu Sistem

2.2.1.2. Klasifikasi Sistem

Sistem dapat diklasifikasikan sebagai berikut:

1. Sistem Abstrak (Abstract System) ; sistem yang berupa pemikiran atau ide- ide yang tidak tampak secara fisik (Sistem Teologia yang merupakan suatu sistem yang menggambarkan hubungan Tuhan dengan Manusia)
2. Sistem Fisik (Physical System) ; merupakan sistem yang ada secara fisik sehingga setiap makhluk dapat melihatnya (Sistem Komputer, Sistem Akuntansi, Sistem Produksi dll.)
3. Sistem Alamiah (Natural System) ; sistem yang terjadi melalui proses alam dalam artian tidak dibuat oleh manusia. (Sistem Tata Surya, Sistem Galaxi, Sistem Reproduksi dll.)
4. Sistem Buatan Manusia (Human Made System) ; sistem yang dirancang oleh manusia. Sistem buatan manusia yang melibatkan interaksi manusia dengan mesin disebut human machine system (contoh Sistem Informasi)

5. Sistem Tertentu (Deterministic System) ; beroperasi dengan tingkah laku yang sudah dapat diprediksi. Interaksi bagian-bagiannya dapat dideteksi dengan pasti sehingga keluaran dari sistem dapat diramalkan (contoh ; Sistem Komputer)
6. Sistem Tak Tentu (Probabilistic System) ; sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probabilitas. (Contoh : Sistem Manusia)
7. Sistem Tertutup (Close System) ; sistem yang tidak berhubungan dan tidak terpengaruh dengan sistem luarnya. Sistem ini bekerja secara otomatis tanpa adanya turut campur tangan dari pihak luarnya. Secara teoritis sistem tersebut ada, tetapi kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanyalah relatively closed system (secara relatif tertutup, tidak benar-benar tertutup).
8. Sistem Terbuka (Open System) ; sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Lebih spesifik dikenal juga yang disebut dengan sistem terotomasi ; yang merupakan bagian dari sistem buatan manusia dan berinteraksi dengan kontrol oleh satu atau lebih komputer sebagai bagian dari sistem yang digunakan dalam masyarakat modern.

Sistem Terotomasi Mempunyai Sejumlah Komponen yaitu ;

1. Perangkat keras (CPU, disk, printer, tape).

2. Perangkat lunak (sistem operasi, sistem database, program pengontrol komunikasi, program aplikasi).
3. Personil (yang mengoperasikan sistem, menyediakan masukan, mengkonsumsi keluaran dan melakukan aktivitas manual yang mendukung sistem).
4. Data (yang harus tersimpan dalam sistem selama jangka waktu tertentu).
5. Prosedur (instruksi dan kebijakan untuk mengoperasikan sistem).

Sistem Terotomasi Terbagi Dalam Sejumlah Katagori :

1. On-line Systems. Sistem on-line adalah sistem yang menerima langsung input pada area dimana input tersebut direkam dan menghasilkan output yang dapat berupa hasil komputasi pada area dimana mereka dibutuhkan. Area sendiri dapat dipisah-pisah dalam skala, misalnya ratusan kilometer. Biasanya digunakan bagi reservasi angkutan udara, reservasi kereta api, perbankan dll.
2. Real-time Systems. Sistem real-time adalah mekanisme pengontrolan, perekaman data, pemrosesan yang sangat cepat sehinga output yang dihasilkan dapat diterima dalam waktu yang relatif sama. Perbedaan dengan sistem on- line adalah satuan waktu yang digunakan real-time biasanya seperseratus atau seperseribu detik sedangkan online masih dalah skala detik atau bahkan kadang beberapa menit. Perbedaan lainnya, on-line biasanya hanya berinteraksi dengan pemakai, sedangkan real-time berinteraksi langsung dengan pemakai dan lingkungan yang dipetakan.

3. Decision Support System + Strategic Planning System. Sistem yang memproses transaksi organisasi secara harian dan membantu para manajer mengambil keputusan, mengevaluasi dan menganalisa tujuan organisasi. Digunakan untuk sistem penggajian, sistem pemesanan, sistem akuntansi dan sistem produksi. Biasanya berbentuk paket statistik, paket pemasaran dll. Sistem ini tidak hanya merekam dan menampilkan data tetapi juga fungsi- fungsi matematik, data analisa statistik dan menampilkan informasi dalam bentuk grafik (tabel, chart) sebagaimana laporan konvensional.
4. Knowledge-based System. Program komputer yang dibuat mendekati kemampuan dan pengetahuan seorang pakar. Umumnya menggunakan perangkat keras dan perangkat lunak khusus seperti LISP dan PROLOG.

Sistem Berdasarkan Prinsip Dasar Secara Umum Terbagi Dalam :

1. Sistem Terspesialisasi ; adalah sistem yang sulit diterakpan pada lingkungan yang berbeda (misalnya sistem biologi; ikan yang dipindahkan ke darat)
2. Sistem Besar ; adalah sistem yang sebagian besar sumber dayanya berfungsi melakukan perawatan harian (misalnya dinosaurus sebagai sistem biologi menghabiskan sebagian besar masa hidupnya dengan makan dan makan).

3. Sistem Sebagai Bagian Dari Sistem Lain ; sistem selalu merupakan bagian dari sistem yang lebih besar, dan dapat terbagi menjadi sistem yang lebih kecil.
4. Sistem Berkembang ; walaupun tidak berlaku bagi semua sistem tetapi hamper semua sistem selalu berkembang.

2.2.1.3. Pelaku Sistem

Pelaku sistem terdiri dari 7 kelompok :

1. Pemakai ; Pada umumnya 3 ada jenis pemakai, yaitu operasional, pengawas dan
2. eksekutif.
3. Manajemen ; Umumnya terdiri dari 3 jenis manajemen, yaitu manajemen pemakai yang bertugas menangani pemakaian dimana sistem baru diterapkan, manajemen sistem yang terlibat dalam pengembangan sistem itu sendiri dan manajemen umum yang terlibat dalam strategi perencanaan sistem dan sistem pendukung pengambilan keputusan. Kelompok manajemen biasanya terlibat dengan keputusan yang berhubungan dengan orang, waktu dan uang, misalnya ; “ sistem tersebut harus mampu melakukan fungsi x,y,z, selain itu harus dikembangkan dalam waktu enam bulan dengan melibatkan programmer dari departemen w, dengan biaya sebesar x”.
4. Pemeriksa ; Ukuran dan kerumitan sistem yang dikerjakan dan bentuk alami organisasi dimana sistem tersebut diimplementasikan dapat

menentukan kesimpulan perlu tidaknya pemeriksa. Pemeriksa biasanya menentukan segala sesuatunya berdasarkan ukuran-ukuran standar yang dikembangkan pada banyak perusahaan sejenis.

5. Penganalisa sistem ; Fungsi- fungsinya antara lain sebagai :
   a. Arkeolog ; yaitu yang menelusuri baga imana sebenarnya sistem lama berjalan, bagaimana sistem tersebut dijalankan dan segala hal yang menyangkut sistem lama.
   b. Inovator ; yaitu yang membantu mengembangkan dan membuka wawasan pemakai bagi kemungkinan-kemungkinan lain.
   c. Mediator ; yaitu yang menjalankan fungsi komunikasi dari semua level, antara lain pemakai, manajer, programmer, pemeriksa dan pelaku sistem yang lainnya yang mungkin belum punya sikap dan cara pandang yang sama.
   d. Pimpinan proyek ; Penganalisa sistem haruslah personil yang lebih berpengalaman dari programmer atau desainer. Selain itu mengingat penganalisa sistem umumnya ditetapkan terlebih dahulu dalam suatu pekerjaan sebelum yang lain bekerja, adalah hal yang wajar jika penanggung jawab pekerjaan menjadi porsi penganalisa sistem.
6. Pendesain sistem ; Pendesain sistem menerima hasil penganalisa sistem berupa kebutuhan pemakai yang tidak berorientasi pada teknologi tertentu, yang kemudian ditransformasikan ke desain arsitektur tingkat tinggi dan dapat diformulasikan oleh programmer.

7. Programmer ; Mengerjakan dalam bentuk program dari hasil desain yang telah diterima dari pendesain.
8. Personel pengoperasian ; Bertugas dan bertanggungjawab di pusat komputer misalnya jaringan, keamanan perangkat keras, keamanan perangkat lunak, pencetakan dan backup. Pelaku ini mungkin tidak diperlukan bila sistem yang berjalan tidak besar dan tidak membutuhkan klasifikasi khusus untuk menjalankan sistem.

2.2.1.4. Hal Mendasar Dalam Pengembangan Sistem

Penganalisa sistem merupakan bagian dari tim yang berfungsi mengembangkan sistem yang memiliki daya guna tinggi dan memenuhi kebutuhan pemakai akhir. Pengembangan ini dipengaruhi sejumlah hal,yaitu :

1. Produktifitas, saat ini dibutuhkan sistem yang lebih banyak, lebih bagus dan lebih cepat. Hal ini membutuhkan lebih banyak programmer dan penganalisa sistem yang berkualitas, kondisi kerja ekstra, kemampuan pemakai untuk mengambangkan sendiri, bahasa pemrograman yang lebih baik, perawatan sistem yang lebih baik (umumnya 50 % sampai 70 % sumber daya digunakan untuk perawatan sistem), disiplin teknis pemakaian perangkat lunak dan perangkat pengembangan sistem yang terotomasi.
2. Realibilitas, waktu yang dihabiskan untuk testing sistem secara umum menghabiskan 50% dari waktu total pengembangan sistem. Dalam kurun waktu 30 tahun sejumlah sistem yang digunakan di berbagai perusahaan

mengalami kesalahan dan ironisnya sangat tidak mudah untuk mengubahnya. Jika terjadi kesalahan, ada dua cara yang bisa dilakukan, yaitu melakukan pelacakan sumber kesalahan dan harus menemukan cara untuk mengoreksi kesalahan tersebut dengan mengganti program, menghilangkan sejumlah statement lama atau menambahkan sejumlah statement baru.

3. Maintabilitas, perawatan mencakup ;
   a. modifikasi sistem sesuai perkembangan perangkat keras untuk meningkatkan kecepatan pemrosesan (yang memegang peranan penting dalam pengoperasian sistem),
   b. modifikasi sistem sesuai perkembangan kebutuhan pemakai. Antara 50% sampai 80% pekerjaan yang dilakukan pada kebanyakan pengembangan sistem dilakukan untuk revisi, modifikasi, konversi,peningkatan dan pelacakan kesalahan.

2.2.2 Konsep Dasar Informasi

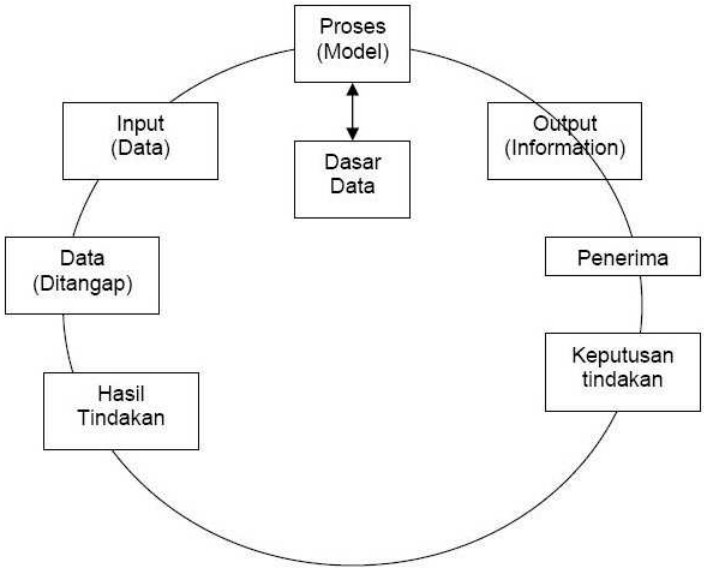
Secara umum informasi dapat didefinisikan sebagai data yang diolah menjadi bentuk yang lebih berarti bagi yang menerimanya.

Menurut Raymond Mcleod, :
“ Informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang ”

Secara umum informasi dapat didefinisikan sebagai hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang

digunakan untuk pengambilan keputusan.

Sumber dari informasi adalah **data**. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyak, gambar, dsb.



Gambar II.3. Siklus Informasi

Adapun fungsi-fungsi informasi adalah sebagai berikut :

1. Untuk meningkatkan pengetahuan bagi si pemakai
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal.

Informasi yang berkualitas harus :

1. akurat,
2. tepat pada waktunya
3. relevan.

2.2.3 Konsep Dasar Sistem Informasi

Definisi Sistem Informasi:

1. “Suatu sistem terintegrasi yang mampu menyediakan informasi yang bermanfaat bagi penggunanya”.
2. “Sebuah sistem terintegrasi atau sistem manusia-mesin, untuk menyediakan informasi untuk mendukung operasi, manajemen dalam suatu organisasi”.
3. “Sekumpulan prosedur manual atau terkomputerisasi yang mengumpulkan/mengambil, mengolah, menyimpan dan menyebarkan informasi dalam mendukung pengambilan dan kendali keputusan”
4. “Sekelompok orang, prosedur, input, output dan pengolahannya secara bersama-sama menghasilkan informasi yang akurat, tepat waktu dan relevan bagi penggunanya”

Menurut Robert A. Leitch;
“sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan- laporan yang diperlukan.”

Ada 4 operasi dasar dari sistem informasi yaitu mengumpulkan, mengolah, menyimpan dan menyebarkan informasi. Informasi mungkin dikumpulkan

dari lingkungan dalam atau luar dan memungkinkan didistribusikan ke dalam atau ke luar organisasi.

Dari definisi di atas terdapat beberapa kata kunci :

1. Berbasis komputer dan Sistem Manusia/Mesin
   a. Berbasis komputer: perancang harus memahami pengetahuan komputer dan pemrosesan informasi
   b. Sistem manusia mesin: ada interaksi antara manusia sebagai pengelola dan mesin sebagai alat untuk memroses informasi. Ada proses manual yang harus dilakukan manusia dan ada proses yang terotomasi oleh mesin. Oleh karena itu diperlukan suatu prosedur/manual sistem.
2. Sistem basis data terintegrasi

   Adanya penggunaan basis data secara bersama-sama (sharing) dalam sebuah data base manajemen system.
3. Mendukung Operasi

   Informasi yang diolah dan di hasilkan digunakan untuk mendukung operasi organisasi.

   Istilah Sistem Informasi

   = *Manajemen Information System*

   = *Information Processing System*

   = *Information Decision System*

   = *Information System.*

Semuanya mengacu pada sebuah sistem informasi berbasis komputer yang dirancang untuk mendukung operasi, manajemen dan fungsi pengambilan keputusan suatu organisasi.

2.2.3.1. Komponen Fisik Sistem Informasi

Komponen fisik sistem informasi antara lain adalah:

1. Perangkat keras komputer: CPU, Storage, perangkat Input/Output, Terminal untuk interaksi, Media komunikasi data
2. Perangkat lunak komputer: perangkat lunak sistem (sistem operasi dan utilitinya), perangkat lunak umum aplikasi (bahasa pemrograman), perangkat lunak aplikasi (aplikasi akuntansi dll).
3. Basis data: penyimpanan data pada media penyimpan komputer.
4. Prosedur: langkah-langkah penggunaan sistem
5. Personil untuk pengelolaan operasi (SDM), meliputi:
   a. *Clerical Personnel* (untuk menangani transaksi dan pemrosesan data dan melakukan inquiry = operator);
   b. *First level manager*: untuk mengelola pemrosesan data didukung dengan perencanaan, penjadwalan, identifikasi situasi out-of-control dan pengambilan keputusan level menengah ke bawah.
   c. *Staff specialist* : digunakan untuk analisis untuk perencanaan dan pelaporan.

d. *Management*: untuk pembuatan laporan berkala, permintaan khsus, analisis khusus, laporan khsusus, pendukung identifikasi masalah dan peluang. Aplikasi = program + prosedur pengoperasian.

Sistem informasi dapat diklasifikasikan sebagai sistem informasi formal dan informal.

Sistem Informasi Formal adalah sistem informasi yang memiliki prosedur dan kebijaksanaan tertulis dalam dokumen. contoh Undang-undang pajak penjualan.

Sistem Informasi Informal adalah sistem informasi yang memiliki prosedur dan kebijaksanaan tidak tertulis dalam dokumen, contoh pemberian diskon penjualan secara tidak terduga.

2.2.3.2. Hubungan Pengelola Dengan Sistem Informasi

Pada bagian 1 sudah disebutkan bahwa salah satu komponen dari sistem informasi adalah personel sebagai pengelola informasi. Oleh karena itu hubungan antara sistem informasi dengan pengelolanya sangat erat. Sistem informasi yang dibutuhkan sangat tergantung dari kebutuhan pengelolanya.

Pengelola sistem informasi terorganisasi dalam suatu struktur manajemen. Oleh karena itu bentuk / jenis sistem informasi yang diperlukan sesuai dengan level manajemennya.

Manajemen Level Atas: untuk perencanaan strategis, kebijakan dan pengambilan keputusan.

Manejemen Level Menengah: untuk perencanaan taktis.

Manejemen Level Bawah: untuk perencanan dan pengawasan operasi

Operator: untuk pemrosesan transaksi dan merespon permintaan.

Untuk pengembangan sebuah sistem informasi diperlukan struktur manajemen organisasi personil. Strutktur dasarnya:

Direktur Sistem Informasi
Manajer Pengembangan Sistem
Analis Sistem
Programmer
Manejer Komputer dan Operasi.

Variasi struktur manajemen sangat tergantung pada Managerial Efficiency vs User Service.

2.2.3.3. Pendekatan Sistem Informasi

Sistem Informasi dipelajari di berbagai disiplin ilmu, seperti :

1. Ilmu Komputer ; yang berkaitan dengan teori-teori perhitungan, metode komputasi dan metode penyimpanan dan akses data yang efisien.
2. Ilmu Manajemen ; yang memberikan penekanan pada model-model normatif dari pembuatan/pengambilan keputusan dan praktek manajemen.
3. Riset Operasional ; yang berfokus pada teknik-teknik matematis untuk mengoptimalkan parameter-parameter organisasi yang terpilih, seperti biaya transportasi, kendali persediaan dan biaya transaksi.
4. SIM ; yang memusatkan pada sistem informasi yang berkaitan dengan komputer dan merupakan gabungan/kombinasi dari ilmu komputer, ilmu manajemen dan riset operasional untuk membentuk sistem dan aplikasi.

5. Sosiologi ; yang berpusat pada pengaruh-pengaruh sosial group dan organisasi dan kegunaan sistem.
6. Ilmu Politik ; yang berkaitan dengan pengaruh-pengaruh politik dan kegunaan informasi. (mencoba untuk mengubah distribusi keuntungan dalam masyarakat)
7. Psikologi ; yang menekankan pada respon-respon individual terhadap realitas sistem dan model- model cognitive dari alasan manusia pekerjaan-pekerjaan, tugas seperti aplikasi mikrokomputer, database, client personal

2.2.4. Alat Pengembangan Sistem

2.2.4.1. Diagram Arus Data (Data Flow Diagram)

Data flow diagram (DFD) digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan DFD memperlihatkan aliran data dan pengembangan suatu sistem yang ditinjau dari segi data yang ditampilkan dengan simbol dan aliran tertentu.

2.2.4.2. Entity Relationship Diagram (ERD)

E-R model didasarkan atas persepsi terhadap dunia nyata yanga terdiri dari sekumpulan objek, disebut entity dan hubungan antar objek tersebut, disebut relationship. Entity adalah objek di dunia yang bersifat unik. Setiap entity mempunyai atribut yang membedakannya dengan entity lainnya.

Pemodelan data dengan model E-R menggunakan diagram E-R. Diagram E-R terdiri dari :

1. Kotak persegi panjang, menggambarkan himpunan entity
2. Elip, menggambarkan atribut-atribut entity
3. Diamon, menggambarkan hubungan antara himpunan entity
4. Garis, yang menghubungkan antar objek dalam diagram E-R

2.2.4.3. Kamus Data (Data Dictionary)

Kamus data adalah katalog tentang data kebutuhan-kebutuhan informasi dari suatu sistem informasi kamus data dapat mendefinisikan data yang mengalir pada sistem dengan lengkap. Kamus data dapat digunakan pada tahap analisa dan perancangan sistem, kamus data disunakan untuk merancang masukan, merancang laporan-laporan dan database.

Dengan adanya kamus data di dapat definisi definisi dari bentuk-bentuk yang tidak dimengerti dalam DFD bersifat global hanya ditujukan nama arus data saja. Keterangan lebih lanjut tentang struktur dari arus data dapat dilihat dikamus data.

2.2.5 Konsep Dasar Pengelolaan Data

Definisi: Pengolahan data adalah proses operasi sistematis terhadap data. Selama operasi, [misal kalkulasi atau operasi logika] sedang berlangsung, data disimpan sementara dalam prosesor.

Yang termasuk dalam proses pengolahan data, antara lain:

1. verifikasi,
2. pengorganisasian data,
3. pencarian kembali,

4. transformasi,
5. penggabungan,
6. pengurutan,
7. perhitungan / kalkulasi
8. ekstraksi data untuk membentuk informasi, dan
9. pembentukan pengetahuan

Secara semantik, isi atau nilai data orisinil harus tidak berubah. Sedangkan isi atau nilai data yang diolah atau diproses bisa berubah [menjadi informasi].

2.2.6 Konsep Basis Data

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili denga menggunakan nilai yang sama antar tabel. Model yang lain seperti model hierarkis dan model

jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel.

Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut.

Database atau Basis Data adalah kumpulan informasi yang menyangkut suatu topik tertentu. Database dapat pula diartikan sebagai kumpulan data tentang suatu benda atau kejadian yang saling berhubungan satu sama lain. Salah satu definisi tentang database dikemukakan oleh Elmasri (1994) sebagai berikut:

> "basis data mempunyai berbagai sumber data dalam pengumpulan data, bervariasi derajad interaksi kejadian dari dunia nyata, dirancang dan dibangun agar dapat digunakan oleh beberapa pemakai untuk berbagai kepentingan."

Data merupakan fakta yang mewakili suatu obyek, seperti : manusia, peristiwa, konsep, keadaan, dan sebagainya, yang dapat dicatat dan mempunyai arti yang implisit. Data dicatat atau direkam dalam bentuk angka, huruf, simbol, gambar, atau kombinasinya. Informasi adalah Data yang telah diolah lebih lanjut untuk tujuan tertentu. Melalui database, data dan informasi dikumpulkan, diolah, dan digunakan untuk berbagai tujuan tertentu.

Pengelolaan basis data dapat dilakukan secara manual ataupun dengan komputer. Salah satu model database secara manual adalah penyimpanan berkas-berkas dokumen dalam rak atau lemari arsip, dimana berkas dokumen disusun secara teratur menurut suatu aturan tertentu, agar mudah untuk diambil dan dimanfaatkan kembali pada waktu dibutuhkan.

Model database yang banyak dikembangkan sekarang adalah database terkomputerisasi, dimana data dikumpulkan dan disimpan dengan berbagai cara dan aturan tertentu untuk selanjutnya diproses guna menghasilkan output informasi tertentu sesuai kebutuhan penggunanya. Basis data berbasis komputer dapat dikelola, baik oleh sekumpulan program aplikasi untuk suatu kepentingan atau oleh Sistem Manajemen Basis Data (Database Management System). Sistem Manajemen Basis Data (SMBD) adalah kumpulan program yang digunakan untuk membuat dan mengelola basis data. Suatu SMBD merupakan sistem perangkat lunak yang secara umum dapat digunakan untuk melakukan pemrosesan dalam hal pendefinisian, penyusunan, dan manipulasi basis data untuk berbagai aplikasi.

Program database sederhana, misalnya seperti komponen database pada Microsoft Excel, bisa menyimpan informasi hanya dalam satu tabel yang seringkali disebut dengan database flat. Program database yang lebih komplek, misalnya Microsoft Access, MySQL, Oracle, dll, bisa menyimpan informasi dalam banyak tabel yang saling berhubungan, yang karena pembuatannya tersebut maka disebut database relational.

Apabila sebuah database tidak melakukan apapun selain informasi yang tersimpan dalam sebuah tabel, database tersebut akan berguna sebagai daftar kertas. Sebagai contoh adalah sebuah buku telepon. Ketika informasi dipublikasikan dalam sebuah buku telepon yang tersimpan dalam sebuah database, informasi tersebut butuh tempat yang lebih kecil, mudah diperbanyak dan didistribusikan, dan jika database dirancang dengan benar, informasinya akan bisa diambil dalam banyak cara. Namun bila database buku telepon tersebut disusun dan disimpan dalam format elektronik, maka kita bisa memanipulasi informasi dalam cara yang handal untuk meningkatkan manfaatnya.

Kekuatan database sesungguhnya bukanlah pada kemampuan menyimpan informasi, melainkan kemampuan anda mengambil informasi yang tepat secara cepat dari database.

Adapun Struktur Database adalah:

1. Database File/Table
2. Record Elemen data/Field

Dari pengertian diatas dapat disimpulkan bahwa basis data mempunyai beberapa kriteria penting, yaitu :

1. Bersifat data oriented dan bukan program oriented.
2. Dapat digunakan oleh beberapa program aplikasi tanpa perlu mengubah basis datanya. Dapat dikembangkan dengan mudah, baik volume maupun strukturnya.

3. Dapat memenuhi kebutuhan sistem-sistem baru secara mudah
4. Dapat digunakan dengan cara-cara yang berbeda.

Prinsip utama Data Base adalah pengaturan data dengan tujuan utama fleksibelitas dan kecepatan pada saat pengambilan data kembali.

Adapun ciri-ciri basis data diantaranya adalah sebagai berikut :

1. Efisiensi meliputi kecepatan, ukuran, dan ketepatan
2. Data dalam jumlah besar.
3. Berbagi Pakai (dipakai bersama sama/Sharebility).
4. Mengurangi bahkan menghilangkan terjadinya duplikasi dan ketidakkonsistenan data.

Software atau aplikasi yang bertugas untuk mengatur, menyimpan, memodifikasi data disebut dengan software database engine dan lebih resminya disebut dengan DBMS (Database Management System). Ada banyak sekali aplikasi DBMS ini mulai yang berjalan di komputer personal (PC) sampai ke komputer skala mainframe. Contoh-contoh dari aplikasi database engine misalnya seperti:

1. SQL Server, dibuat oleh Microsoft.
2. MS Access, dibuat oleh Microsoft.
3. Oracle Database, dibuat oleh Oracle.
4. MySQL, dibuat oleh MySQL AB.

5. Firebird, dibuat oleh komunitas open source berdasarkan dari kode nterbase.
6. PostgreSQL, dibuat oleh komunitas open source.
7. DB2, dibuat oleh IBM.

2.2.7 Software Pendukung

Software utama yang digunakan dalam pembuatan perangkat lunak ini adalah Delphi 7.0 sebagai perangkat lunak pembangun dan MySQL sebagai *database*.

2.2.7.1. Delphi 7

Borland Delphi atau yang disebut delphi saja, merupakan sarana pemrograman aplikasi visual. Bahasa pemrograman yang digunakan adalah bahasa pemrograman pascal atau yang kemudian juga disebut bahasa pemrograman Delphi. Delphi merupakan generasi penerus dari tubo pascal. Tirbo pascal yang diluncurkan pada tahun 1983 dirancang utntuk dijalankan pada system operasi DOS (DISCK OPERATING SYSTEM) yang merupakan syatem operasi yang banyak digunakan pada say itu. Sedangkan Delphi diluncurkan pada tahun 1995 dirancang untuk beroprasi dibawa system operasi windows.

Kelebihan Borland Delphi 7, Borland dephi 7 merupakan pilihan dari sebagian kalangan programmer untuk membuat aplikasi. Hal ini disebabkan kelebiha yang ada pada Borland, berikut ini sebagian kecil dari banyak kelebihan Borland Delphi 7:

1. Berorientasi Object Oriented Programig. Setiap bagian yang ada pada program dipanang sebagai suatu object yagn mempunyai sifat-sifat yang dapat diubah dan diatur.
2. Satu file EXE, setelah anda merancang program dalam IDE Delphi, Delphi akan mengkompilasinya menjadi sebua file executable tunggal. Program yang anda buat dapat langsung dijalankan dan didistribusikan pada computer lain tanpa perlu menyertakan file DLL dari luar. Ini merupakan sebuah kelebihan yang sangat berarti.
3. Borland Delphi 7 hadir bersama Borland Kylix 3 yang berbasiskan Linux, sehingga memungkinkan anda untuk membuat aplikasi multi-platform.

Konfigurasi komputer untuk sistem: untuk dapat melakukan intalasi dan menggunakan Borland Delphi 7 dengan normal, anda harus menuai persyaratan sebagai berikut :

1. Prosesor: Prosesor Pentium 166MHz atau yang lebih tinggi.
2. System Operasi: System operasi Windows 98, windows 98 Se, Windows ME, Windows 2000, Windows 2003 dan Windows XP

Sejarah Borland Delphi

1. Delphi versi 1 (berjalan pada windows 3.1 atau windows 16 bit)
2. Delphi versi 2 (Berjalan pada windows 95 atau delphi 32 bit)
3. Delphi versi 3 (berjalan pada windows 95 keatas dengan tambahan fitur internet atua web)

4. Perkembangan selanjutnya diikuti dengan Delphi versi 4, 5 dan 6.
5. Versi terkini dari delphi adalahversi 7 dengan tambahan vitur .net dengan tambahan file XML

2.2.7.2. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya

merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh user maupun program-program aplikasinya. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan Interbase.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Open Source.MySQL didistribusikan secara *open source,* dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. 'Multiuser'. MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. 'Performance tuning'. MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Jenis Kolom. MySQL memiliki tipe kolom yang sangat kompleks, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (tool)yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

Terdapat beberapa API tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain: bahasa pemrograman C, C++, C#, bahasa pemrograman Eiffel, bahasa pemrograman Smalltalk, bahasa pemrograman Java, bahasa pemrograman Lisp, Perl, PHP, bahasa pemrograman Python, Ruby, REALbasic dan Tcl. Sebuah antarmuka ODBC memanggil MyODBC yang memungkinkan setiap bahasa pemrograman yang mendukung ODBC untuk berkomunikasi dengan basis data MySQL. Kebanyakan kode sumber MySQL dalam ANSI C.

MySQL sangat populer dalam aplikasi web seperti MediaWiki (perangkat lunak yang dipakai Wikipedia dan proyek-proyek sejenis) dan PHP-Nuke dan berfungsi sebagai komponen basis data dalam LAMP. Popularitas sebagai aplikasi web dikarenakan kedekatannya dengan popularitas PHP, sehingga seringkali disebut sebagai *Dynamic Duo*.

2.2.8 Metode Pengujian

Pengujian adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Pentingnya pengujian perangkat lunak dan implikasinya yang mengacu pada kualitas perangkat lunak tidak dapat terlalu ditekan karena melibatkan sederetan aktivitas produksi di mana peluang terjadinya kesalahan manusia sangat besar dan arena ketidakmampuan manusia untuk melakukan dan berkomunikasi dengan sempurna maka pengembangan perangkat lunak diiringi dengan aktivitas jaminan kualitas.

Pendekatan Desain Test Case terbagi menjadi :

1. Pengujian white-box

   Pengujian white-box merupakan pendekatan terhadap pengujian yang diturunkan dari pengetahuan struktur dan implementasi perangkat lunak.

   Pengujian white-box biasanya diterapkan untuk unit program yang relatif kecil seperti subrutin atau operasi yang terkait dengan suatu objek. Penguji dapat menganalisis kode dan menggunakan pengetahuan mengenai struktur komponen untuk menurunkan data uji. Analisis kode dapat digunakan untuk menemukan berapa kasus uji yang dibutuhkan untuk menjamin bahwa semua statement pada program atau komponen dieksekusi paling tidak satu kali pada proses pengujian. Pengetahuan mengenai algoritma yang digunakan untuk implementasi beberapa fungsi dapat dipakai untuk mengidentifikasi partisi ekuivalensi lebih lanjut.

2. Pengujian black-box

   Pengujian black-box merupakan pendekatan pengujian yang ujinya diturunkan dari spesifikasi program atau komponen. Sistem merupakan ‘kotak hitam’ yang perilakunya hanya dapat ditentukan dengan mempelajari input dan output yang berkaitan. Nama lain untuk cara ini adalah pengujian fungsional karena penguji hanya berkepentingan dengna fungsionalitas dan bukan implementasi perangkat lunak.

   Pengujian black-box dapat diterapkan pada sistem yang disusun sebagai sistem maupun sebagai objek. Penguji memberikan input kepada komponen atau sistem dan meneliti output yang dihasilkan. Jika output bukan merupakan yang dikehendaki, berarati pengujian tersebut telah mendeteksi masalah dengan perangkat lunak tersebut.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem memberikan gambaran tentang sistem yang diamati yang saat ini sedang berjalan, kelebihan dan kekurangan sistem tersebut dapat diketahui dan diidentifikasi, sehingga dalam membangun perangkat lunak menjadi lebih mudah.

Dari hasil analisis sistem lama, maka akan ditemukan beberapa data dan fakta yang akan dijadikan bahan uji dan analisis menuju pengembangan dan penerapan sebuah aplikasi sistem yang diusulkan.

Sesuai dengan metodologi penelitian dalam membangun perangkat lunak, tahap analisis sistem merupakan tahap yang kritis dan sangat penting, karena jika terjadi kesalahan ditahap ini akan menyebabkan kesalahan pula pada tahap selanjutnya. Tujuan global analisis sistem adalah mengidentifikasi hal-hal apa saja yang diperlukan sistem yang akan dibangun dan memahami dengan jelas proses yang akan dilakukan oleh sistem.

Sebagai analisis pada sistem yang sedang berjalan, akan dibahas bagaimana prosedur dan aliran dokumen yang sedang berjalan yang digambarkan dalam bentuk *flow map*, pengkodean dan analisis sistem non funsional yang meliputi perangkat keras dan perangkat lunak yang digunakan, serta analisis *user* yang terlibat.

3.1.1 Analisis Sistem yang sedang Berjalan

Analisis terhadap sistem yang sedang berjalan bertujuan untuk mengetahui lebih jelas bagaimana kerja suatu sistem dan mengetahui masalah yang dihadapi sistem untuk dapat dijadikan landasan usulan perancangan sistem. Tahap analisa ini merupakan tahapan yang penting di dalam merancang suatu sistem. Analisa sistem yang dilakukan untuk mengetahui masalah apa yang sedang dihadapi oleh perusahaan.

Sistem yang sedang berjalan diartikan sebagai sistem yang sedang dipakai, sedangkan analisa sistem yang sedang berjalan diartikan sebagai cara untuk memahami terlebih dahulu masalah yang dihadapi oleh sistem, seperti mendefinisikan kebutuhan-kebutuhan funsional dari sistem sehingga dapat diketahui apa saja kebutuhan-kebutuhan pemakai yang belum terpenuhi oleh sistem yang sedang berjalan tersebut.

Tujuan dari analisa sistem untuk menentukan bentuk dari rancangan sistem yang diterapkan. Analisa tersebut juga dapat menentukan langkah-langkah perancangan yang akan dibuat sehingga rancangan sistem sesuai dengan kebutuhan pemakai dan sistem mempunyai hak unjuk kerja yang efisien dan efektif, dapat menghasilkan informasi yang cepat, tepat dan akurat.

Prosedur merupakan urutan langkah-langkah yang terjadi atau yang dilakukan dalam sistem yang sedang berjalan. Adapun beberapa prosedur yang terlibat dalam sistem ini adalah sebagi berikut:

1. Prosedur Pembelian Barang

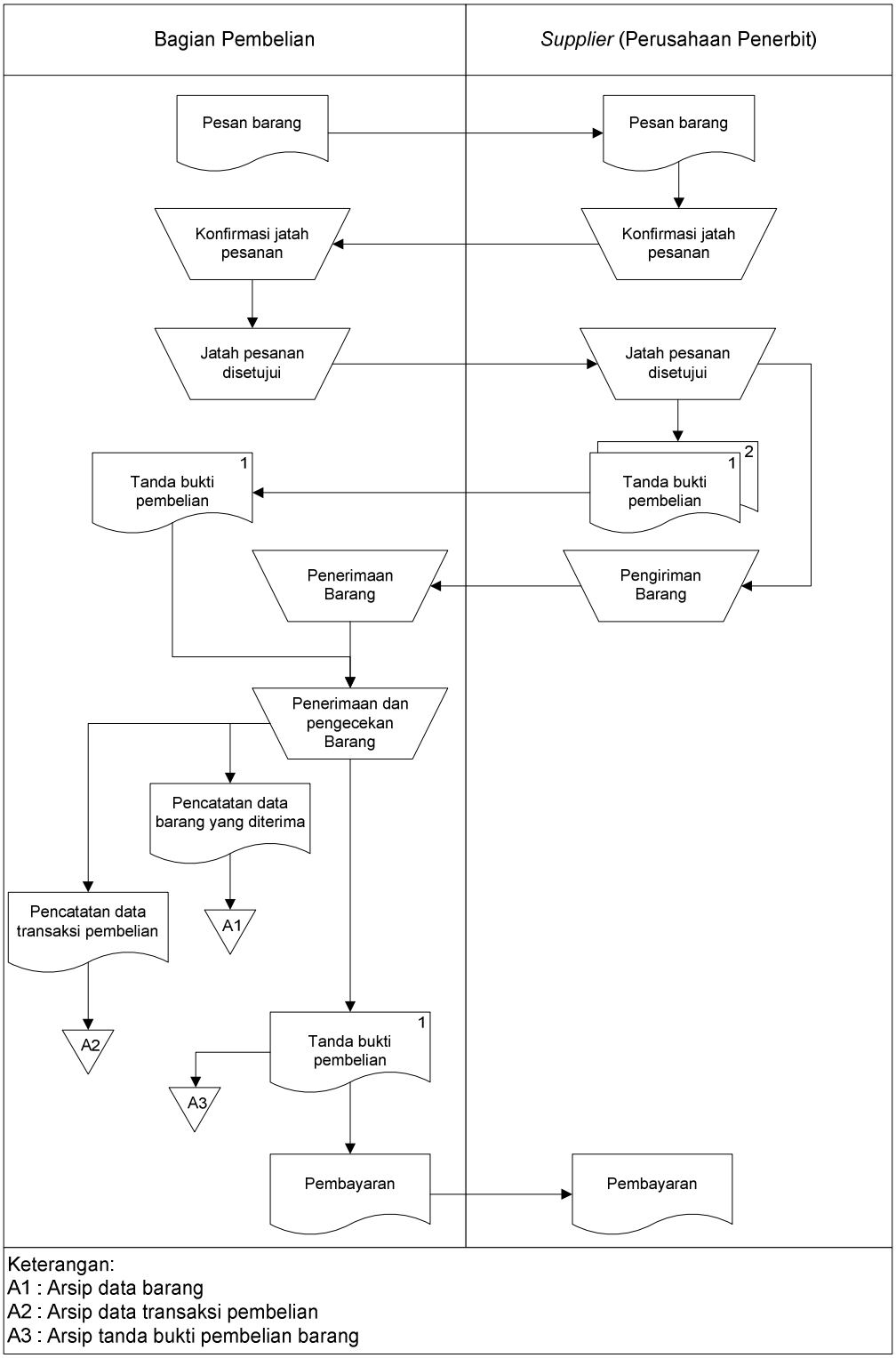
2. Prosedur Penjualan Barang
3. Prosedur Pembuatan Laporan

3.1.1.1. Prosedur Pembelian Barang

Dalam tahap ini prosedur pembelian barang yang dilakukan ketika:

1. Bagian Pembelian memesan barang ke *supplier* (perusahaan penerbit)
2. Bagian Pembelian menunggu konfirmasi jatah pemesanan barang dari *supplier* (perusahaan penerbit)
3. Setelah jatah pemesanan dikonfirmasi *supplier* (perusahaan penerbit), maka *supplier* (perusahaan penerbit) akan melakukan pengiriman barang dan tanda bukti pembelian
4. Setelah barang dan tanda bukti pembelian diterima, maka Bagian Pembelian melakukan pengecekan barang sesuai dengan tanda bukti pembeliannnya
5. Bagian Pembelian mencatatkan seluruh data transaksi pembelian barang pada buku secara manual
6. Setelah proses pencatatan selesai maka akan dilakukan pembayaran ke *supplier* (perusahaan penerbit)

Adapun prosedur pembelian barang yang sedang berjalan dapat kita lihat pada gambar III.1. berikut.



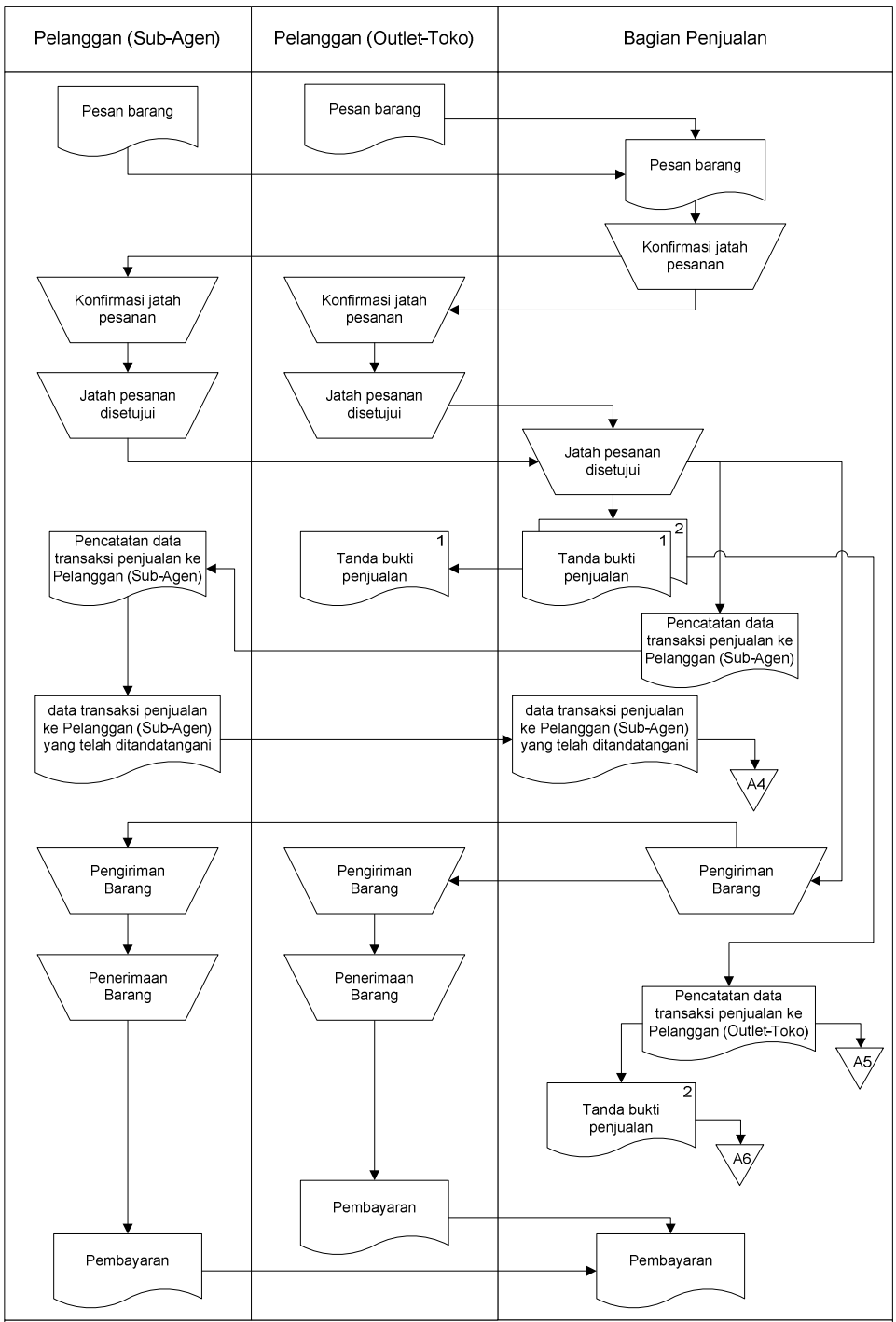
Gambar III.1. Prosedur Pembelian Barang

3.1.1.2. Prosedur Penjualan Barang

Dalam tahap ini prosedur penjualan barang yang dilakukan ketika:

1. Pelanggan (Sub-Agen), Pelanggan (Outlet-Toko) memesan barang ke Bagian Penjualan
2. Pelanggan (Sub-Agen), Pelanggan (Outlet-Toko) menunggu konfirmasi jatah pemesanan barang dari Bagian Penjualan
3. Setelah jatah pemesanan dikonfirmasi Bagian Penjualan, maka akan dilakukan pengiriman barang ke Pelanggan (Sub-Agen), Pelanggan (Outlet-Toko)
4. Barang dikirim ke Pelanggan (Sub-Agen), maka Bagian Penjualan mencatatkan seluruh transaksi penjualan barang pada buku secara manual dan ditandatangani oleh pihak Pelanggan (Sub-Agen) sebagai bukti telah dilakukan transaksi penjualan ke Pelanggan (Sub-Agen)
5. Barang dikirim ke Pelanggan (Outlet-Toko), maka Bagian Penjualan akan mencatatkan transaksi penjualan barang pada buku terpisah dan membuatkan tanda bukti penjualan rangkap dua (halaman 1 untuk Pelanggan (Outlet-Toko), halaman 2 sebagai arsip Maju Terus Agency)
6. Setelah barang telah diterima pelanggan, maka pelanggan akan meyetorkan pembayaran ke Bagian Penjualan

Adapun prosedur penjualan barang yang sedang berjalan dapat kita lihat pada gambar III.2. berikut.



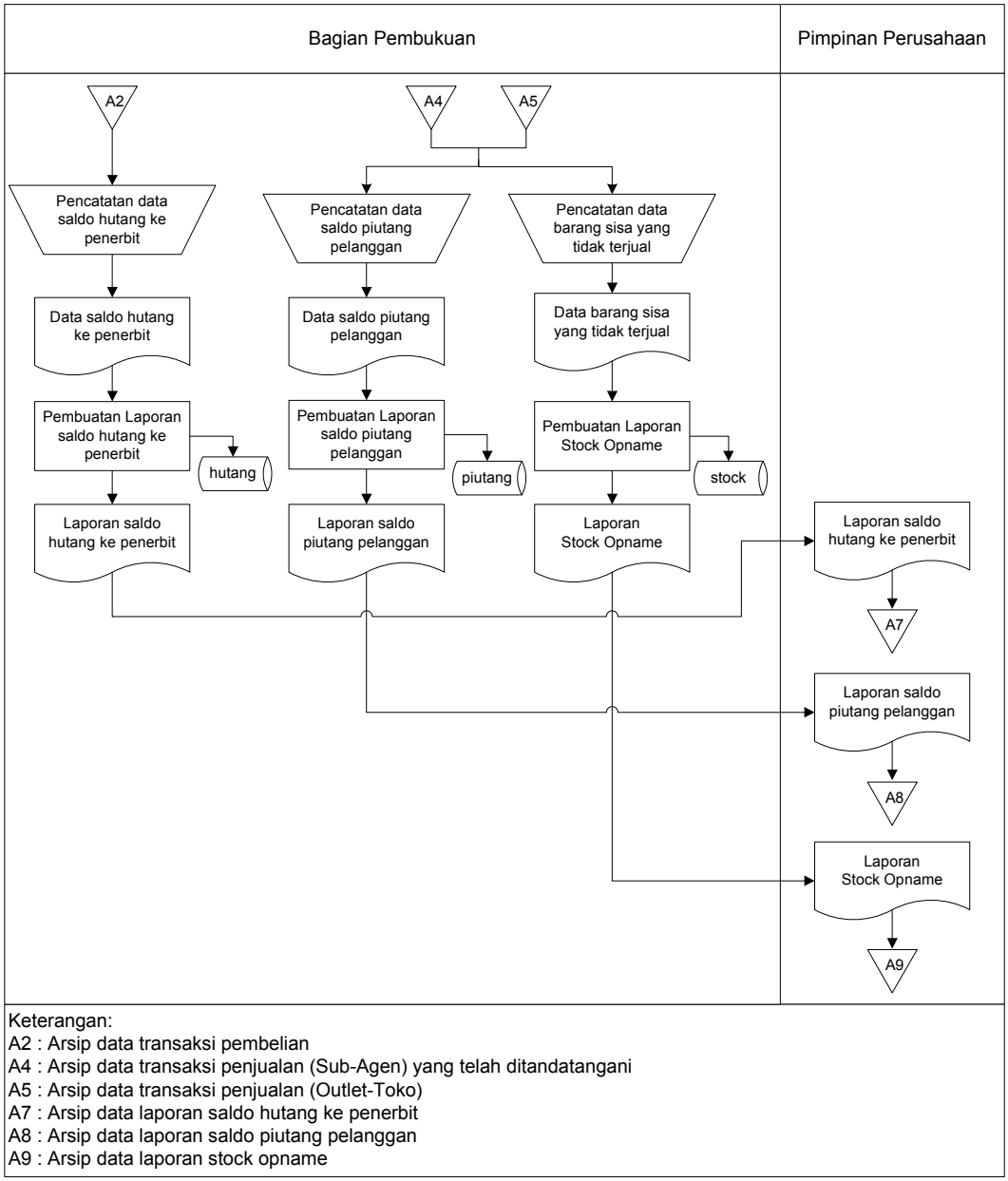
Gambar III.2. Prosedur Penjualan Barang

3.1.1.3. Prosedur Pembuatan Laporan

Dalam tahap ini prosedur pembuatan laporan dilakukan ketika:

1. Dari arsip data transaksi pembelian, oleh Bagian Pembukuan akan dikumpulkan seluruh data saldo hutang ke *supplier* (perusahaan penerbit) untuk laporan saldo hutang ke penerbit
2. Dari arsip data transaksi penjualan, oleh Bagian Pembukuan akan dikumpulkan seluruh data saldo piutang pelanggan untuk laporan saldo piutang pelanggan dan data barang sisa yang tidak terjual untuk laporan *stock opname*
3. Setelah data terkumpul maka akan dilakukan pembuatan laporan. Data diketik kedalam aplikasi *office* (*excel*)
4. Setelah data selesai diketik maka akan dicetak sebagai laporan untuk pimpinan perusahaan

Prosedur pembuatan laporan yang sedang berjalan dapat kita lihat pada gambar III.3. berikut.



Gambar III.3. Prosedur Pembuatan Laporan

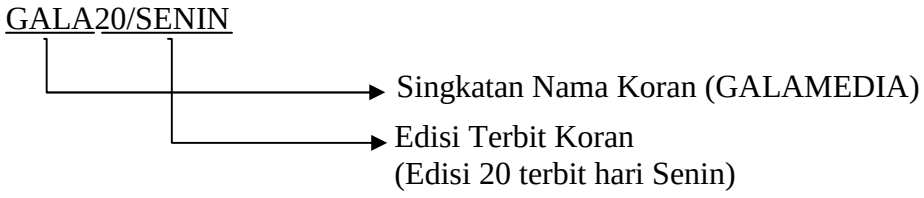
3.1.2 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan analisis terhadap sistem yang sedangberjalan, yaitu sistem pengolahan penjualan dan pembelian barang. Sistem berjalan dianalisis berdasarkan beberapa hal, yaitu analisis pengkodean, analisis user, analisis perangkat lunak, analisis perangkat keras, dan analisis jaringan.

3.1.2.1. Analisis Pengkodean

Kode yang saat ini digunakan di maju terus agency hanya untuk kode barang saja sementara untuk pelanggan dan *supplier* belum mempunyai kode. Semua pengolahan transaksi pembelian dan penjualan di Maju Terus Agency masih menggunakan data *supplier* dan pelanggan secara langsung tanpa pengkodean karena pencatatan datanya masih menggunakan cara manual.

Contoh pengkodean untuk Koran yang ada sekarang adalah sebagai berikut:

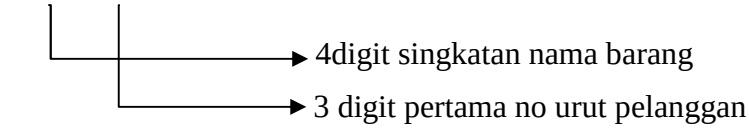


Dalam menjalankan aplikasi ini bukan hanya kode barang yang diperlukan. Kode pelanggan dan *supplier* juga dibutuhkan. Maka perlu ditambahkan Id pelanggan dan Id *supplier*. Format kode yang disarankan adalah sebagai berikut:

1. Kode Barang

   Format

   BBBB 999

   Contoh: GALA0001

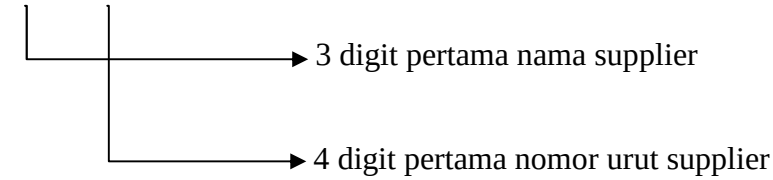
   GALA: Menerangkan 4 digit nama barang Galamedia

   0001 : Menerangkan 4 digit nomor urut penerbit

2. Id *Supplier*

   Format

   SSS 9999

   Contoh: INF0001

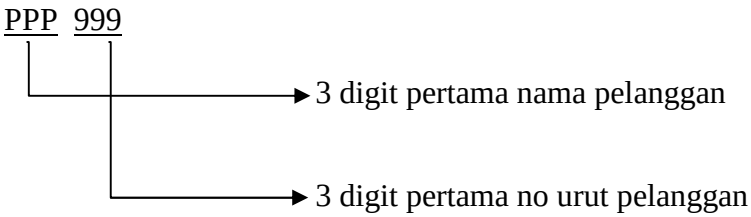
   INF : Menerangkan 3 digit nama penerbit Infomedia

   0001 : Menerangkan 4 digit nomor urut penerbit

3. Id Pelanggan

   Format

   PPP 999

   Contoh: KIN001

   INF : Menerangkan 3 digit nama pelanggan King TB.

   001 : Menerangkan 3 digit nomor urut pelanggan

3.1.2.2. Analisis Karakteristik *User*

Aplikasi ini akan digunakan oleh *user* yang merupakan karyawan di Maju Terus Agency. *User* dapat melakukan kegiatan yang berhubungan dengan data yang terdapat pada database aplikasi. Adapun tujuan dari analisis ini adalah untuk mengetahui siapa saja *user* yang terlibat beserta karakteristiknya sehingga dapat diketahui tingkat pemahaman dan pengalaman *user* terhadap komputer.

User akan dibagi menjadi tiga yaitu *user* bagian pembukuan, *user* bagian pembelian dan *user* bagian penjualan. Adapun rinciannya dapat dilihat pada tabel III.1 berikut.

Tabel III.1. Rincian Karakteristik Pengguna

Keterangan	Pengguna	Tanggung Jawab	Hak Akses
Jabatan	**Bagian Pembukuan**	Melakukan kegiatan pencatatan serta pengelolaan data master dan data laporan	Login admin
Nama	Nia Kaniasari		
Umur	33		
Pengalaman menggunakan komputer	8 tahun		
Latar belakang pendidikan	S1		
Sistem Operasi yang pernah/sering digunakan	Win 2000, Win XP		
Software yang pernah/sering digunakan	Ms. Word, Ms. Excel		
Jabatan	**Bagian Pembelian**	Melakukan kegiatan pencatatan serta pengelolaan data transaksi pembelian ke	Login pembelian
Nama	Atjep		
Umur	27		
Pengalaman menggunakan	4 tahun		

komputer		supplier	
Latar belakang pendidikan	SMA		
Sistem Operasi yang pernah/sering digunakan	Win 2000, Win XP		
Software yang pernah/sering digunakan	Ms. Word, Ms. Excel		
Jabatan	**Bagian Penjualan (Distribusi)**	Melakukan kegiatan pencatatan serta pengelolaan data transaksi penjualan ke sub-agen dan outlet-toko	Login penjualan
Nama	Astri		
Umur	23		
Pengalaman menggunakan komputer	3 tahun		
Latar belakang pendidikan	SMA		
Sistem Operasi yang pernah/sering digunakan	Win 2000, Win XP		
Software yang pernah/sering digunakan	Ms. Word, Ms. Excel		

3.1.2.3. Analisis Kebutuhan Perangkat Lunak

Pada sistem yang berjalan perangkat lunak yang digunakan adalah *Ms.Office, Windows XP* untuk Sistem Operasi dan perangkat lunak yang dibutuhkan untuk membangun Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency adalah *Borland Delphi 7.0, MySQL Server 5.0* sebagai *Database Management Storage* (DBMS), mysql-connector-odbc-5.1.5, SQLYog Enterprise v7.14

3.1.2.4. Analisis Kebutuhan Perangkat Keras

Analisa perangkat keras merupakan proses analisis yang lebih menekankan kepada aspekpemanfaatan perangkat keras yang selama ini telah dimiliki. Maju Terus Agency memiliki 3 buah komputer yang terletak dibagian pembukuan, pembelian dan penjualan. Spesifikasi komputer yang ada di Maju Terus Agency dapat dilihat pada tabel III.2 berikut:

Tabel III.2. Spesifikasi Komputer yang Ada

No.	Komputer	Spesifikasi
1.	Komputer bagian pembukuan	Monitor
		Procesor Core 2 Duo 2.8 GHz
		Harddisk 200 GB
		RAM 1 GB
		VGA 256 MB
		DVD-ROM
		Mouse dan Keyboard
		Printer
2.	Komputer bagian pembelian dan bagian penjualan	Monitor
		Procesor Intel Celeron ® 1.8 GHz
		Harddisk 80 GB
		RAM 446 MB
		VGA 64 MB
		DVD-ROM
		Mouse dan Keyboard
		Printer

Dan perangkat keras yang digunakan untuk pemakai sistem informasi penjualan ini secara optimal memerlukan spesifikasi minimum komputer yang bapat dilihat pada tabel III.3 sebagai berikut:

Tabel III.3. Spesifikasi Minimum Komputer yang Dibutuhkan

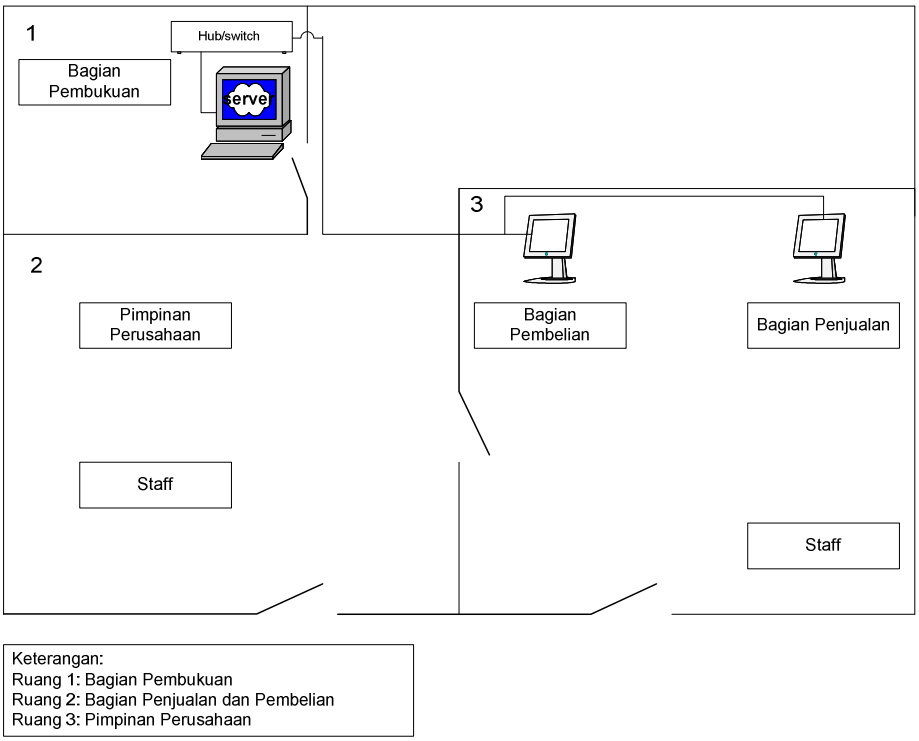
No.	Komputer	Spesifikasi
1.	Komputer yang dibutuhkan untuk menjalankan aplikasi	Monitor VGA
		Procesor dengan kecepatan 1.7 GHz (Pentium III)
		Kapasitas Harddisk 20 GB
		Kapasitas RAM 128 GB
		VGA 256 MB
		CD-ROM
		Mouse dan Keyboard
		Printer

Secara keseluruhan spesifikasi komputer yang ada di Maju Terus Agency telah mencukupi untuk menjalankan sistem yang dibuat.

3.1.2.5. Analisis Jaringan

Analisis jaringan komputer ini diperlukan karena aplikasi ini dibuat dengan sistem *client server*. Usulan jaringan *client server* dalam pendataan ini dibuat 3 komputer dan 1 *hub/switch* dimana 1 komputer untuk *server* dan 2 komputer untuk *client*.

Denah penempatan komputer di perusahaan dapat kita liha pada gambar III.4 berikut.



Gambar III.4. Analisis Jaringan Maju Terus Agency

Secara keseluruhan jaringan komputer yang ada di Maju Terus Agency telah mencukupi untuk menjalankan sistem yang dibuat.

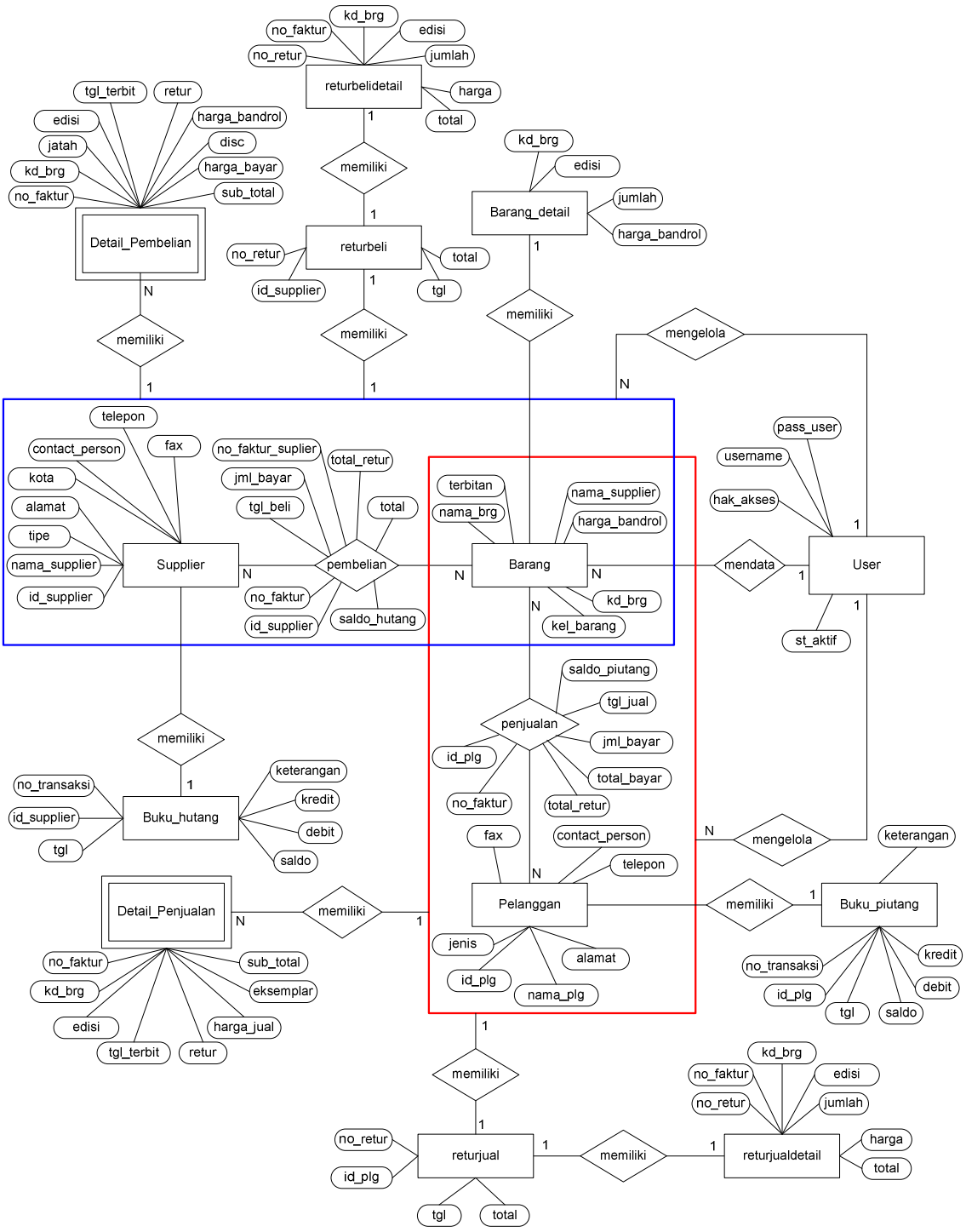
3.1.3 Analisis *Database*

Database merupakan kumpulan data yang saling berkaitan satu sama lain yang direalisasikan dengan *relation key* yang digambarkan dalam *Entity Relationship Diagram*.

3.1.3.1. Entity Relationship Diagram

Dalam memodelkan data dan menggambarkan hubungan antara data yang ada pada sistem yang sedang berjalan di Maju terus Agency digunakan alat bantu yaitu *Entity Relationship Diagram* (ERD). ERD

yang diusulkan di Maju Terus Agency dapat dilihat pada gambar III.5 berikut.



Gambar III.5. ER-Diagram

3.1.3.2. Kamus Data

Kamus data merupakan katalog fakta tentang data dan kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analisis sistem dapat mendefinisikan data yang mengalir pada sistem dengan lengkap.

Tabel III.4. Kamus Data

Nama	**User**
Deskripsi	Berisi data user
Struktur Data	Username+Pass_user+Hak-akses+St_aktif
Username Pass_user Hak_akses St_aktif	[a…z | A…Z] [0…9 | a…z | A…Z] [A…Z] [0 / 1]
Nama	**Barang**
Deskripsi	Berisi data barang
Struktur Data	Kd_brg+Nama_brg+ Terbitan+Nama_supplier +Harga_bandrol+Kel_barang
Kd_brg Nama_brg Terbitan Nama_supplier Harga_bandrol Kel_barang	[0…9 | A…Z] [a…z | A…Z] [a…z | A…Z] [a…z | A…Z] [0…9] [a…z | A…Z]
Nama	**Barang_detail**
Deskripsi	Berisi data barang_detail
Struktur Data	Kd_brg+Edisi+Jumlah +Harga_bandrol
Kd_brg Edisi Jumlah Harga_bandrol	[0…9 | A…Z] [0…9 | a…z | A…Z] [0…9] [0…9]
Nama	**Supplier**
Deskripsi	Berisi data supplier
Struktur Data	Id_supplier+Nama_supplier+Tipe+Alamat+Kota +Contact_person+Telepon+Fax
Id_supplier Nama_supplier Tipe Alamat Kota	[0…9 | A…Z] [a…z | A…Z] [A…Z] [0…9 | a…z | A…Z] [a…z | A…Z]

Contact_person	[a…z | A…Z]
Telepon	[0…9]
Fax	[0…9]
Nama	**Pelanggan**
Deskripsi	Berisi data pelanggan
Struktur Data	Id_plg+Nama_plg+Jenis+Alamat+ Contact_person +Telepon+Fax
Id_plg	[0…9 | A…Z]
Nama_plg	[a…z | A…Z]
Jenis	[A…Z]
Alamat	[0…9 | a…z | A…Z]
Contact_person	[a…z | A…Z]
Telepon	[0…9]
Fax	[0…9]
Nama	**Pembelian**
Deskripsi	Berisi data pembelian
Struktur Data	No_faktur+ No_faktur_suplier +Tgl_beli+Id_supplier +Saldo_hutang+Jml_bayar+Total+Total_retur
No_faktur	[0…9 | A…Z]
No_faktur_suplier	[0…9 | A…Z]
Tgl_beli	[0…9]
Id_supplier	[0…9 | A…Z]
Saldo_hutang	[0…9]
Jml_bayar	[0…9]
Total	[0…9]
Total_retur	[0…9]
Nama	**Penjualan**
Deskripsi	Berisi data penjualan
Struktur Data	No_faktur+Tgl_jual+Id_plg+Saldo_piutang+Jml_bayar +Total+Total_retur
No_faktur	[0…9 | A…Z]
Tgl_jual	[0…9]
Id_plg	[0…9 | A…Z]
Saldo_piutang	[0…9]
Jml_bayar	[0…9]
Total	[0…9]
Total_retur	[0…9]
Nama	**Detail_pembelian**
Deskripsi	Berisi detail_pembelian
Struktur Data	No_faktur +Kd_brg+ Edisi+Tgl_terbit+ Jatah+Retur +Harga_bandrol+Disc+Harga_bayar+Sub_total
No_faktur	[0…9 | A…Z]
Kd_brg	[0…9 | A…Z]
Edisi	[0…9 | a…z | A…Z]
Tgl_terbit	[0…9]

Jatah	[0…9]
Retur	[0…9]
Harga_bandrol	[0…9]
Disc	[0…9]
Harga_bayar	[0…9]
Sub_total	[0…9]
Nama	**Detail_penjualan**
Deskripsi	Berisi detail_penjualan
Struktur Data	No_faktur+Kd_brg+Edisi+Tgl_terbit+Eksemplar +Retur+Harga_jual+Sub_total
No_faktur	[0…9 | A…Z]
Kd_brg	[0…9 | A…Z]
Edisi	[0…9 | a…z | A…Z]
Tgl_terbit	[0…9]
Eksemplar	[0…9]
Retur	[0…9]
Harga_jual	[0…9]
Sub_total	[0…9]
Nama	**Returbeli**
Deskripsi	Berisi retur data pembelian
Struktur Data	No_retur+Tgl+Id_supplier+Total
No_retur	[0…9 | A…Z]
Tgl	[0…9]
Id_supplier	[0…9 | A…Z]
Total	[0…9]
Nama	**Returjual**
Deskripsi	Berisi retur data penjualan
Struktur Data	No_retur+Tgl+Id_plg+Total
No_retur	[0…9 | A…Z]
Tgl	[0…9]
Id_plg	[0…9 | A…Z]
Total	[0…9]
Nama	**Returbelidetail**
Deskripsi	Berisi detail retur data pembelian
Struktur Data	No_retur+No_faktur+Kd_brg+Edisi+Jumlah+Harga +Total
No_retur	[0…9 | A…Z]
No_faktur	[0…9 | A…Z]
Kd_brg	[0…9 | A…Z]
Edisi	[0…9 | a…z | A…Z]
Jumlah	[0…9]
Harga	[0…9]
Total	[0…9]
Nama	**Returjualdetail**
Deskripsi	Berisi detail retur data penjualan

Struktur Data	No_retur+No_faktur+Kd_brg+Edisi+Jumlah+Harga +Total
No_retur No_faktur Kd_brg Edisi Jumlah Harga Total	[0…9 | A…Z] [0…9 | A…Z] [0…9 | A…Z] [0…9 | a…z | A…Z] [0…9] [0…9] [0…9]
Nama	**Buku_Hutang**
Deskripsi	Berisi data hutang supplier
Struktur Data	No_transaksi+Tgl+Id_supplier+Debit+Kredit+Saldo +Keterangan
No_transaksi Tgl Id_supplier Debit Kredit Saldo Keterangan	[0…9 | A…Z] [0…9] [0…9 | A…Z] [0…9] [0…9] [0…9] [a…z | A…Z]
Nama	**Buku_Piutang**
Deskripsi	Berisi data piutang pelanggan
Struktur Data	No_transaksi+Tgl+Id_plg+Debit+Kredit+Saldo +Keterangan
No_transaksi Tgl Id_plg Debit Kredit Saldo Keterangan	[0…9 | A…Z] [0…9] [0…9 | A…Z] [0…9] [0…9] [0…9] [a…z | A…Z]

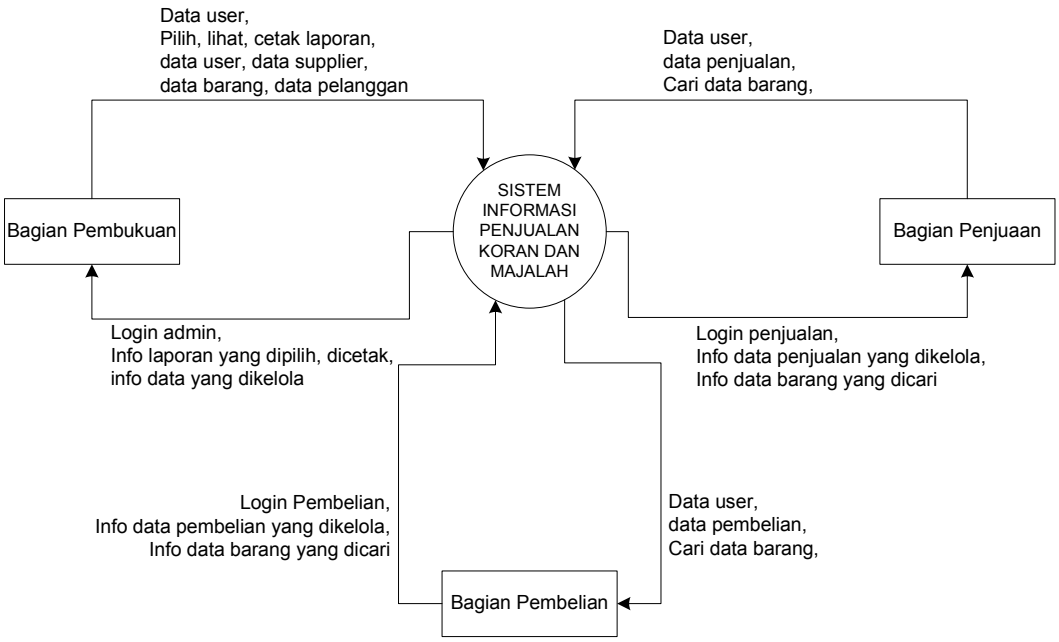
3.1.4 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional merupakan analisa dari aplikasi penjualan koran dan majalah di Maju Terus Agency yang akan dibangun. Analisis kebutuhan fungsional ini dapat digambarkan dalam bentuk diagram konteks dan *Data Flow Diagram* (DFD).

3.1.4.1. Diagram Konteks

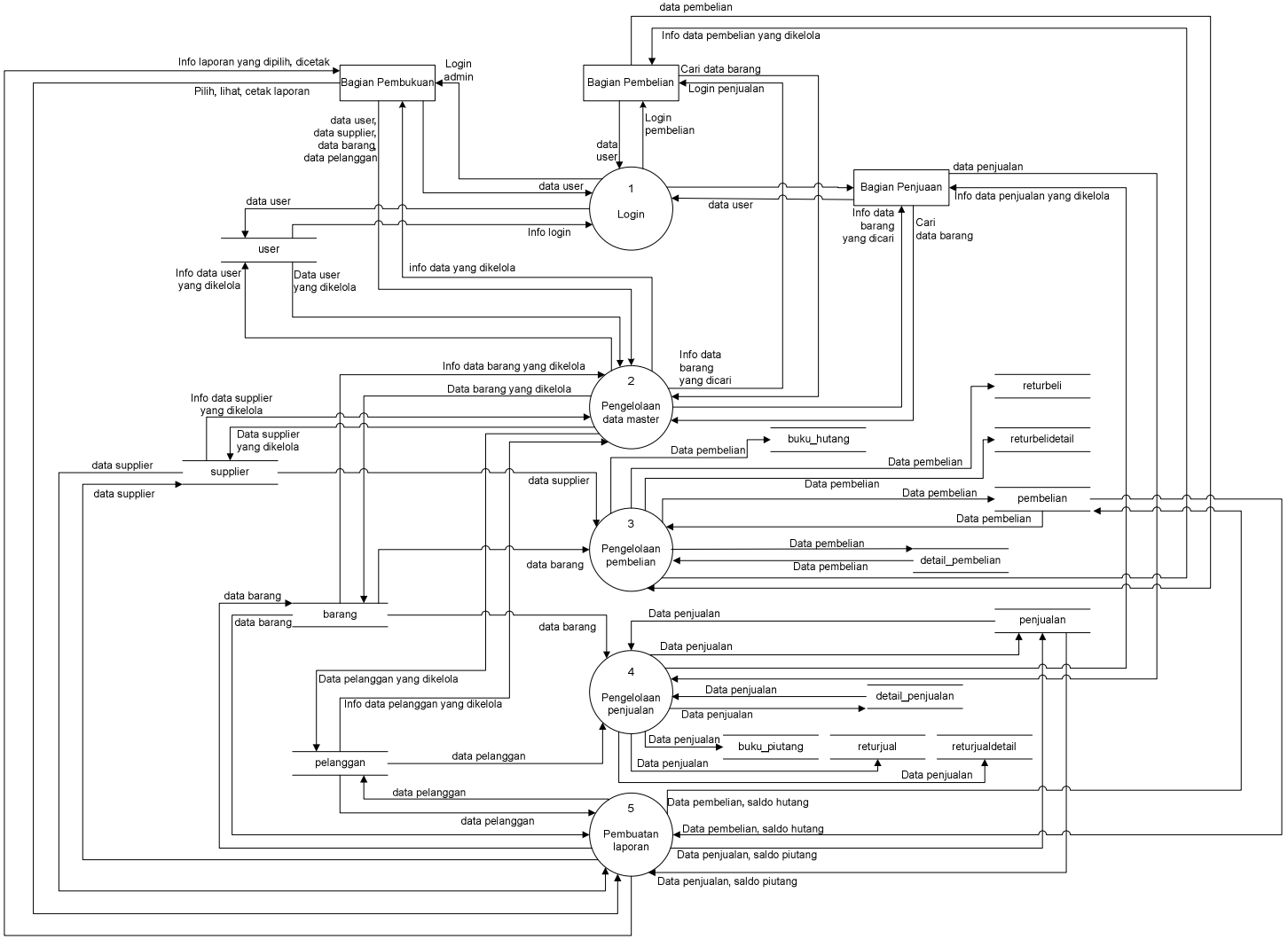
Diagram konteks merupakan suatu model yang menjelaskan secara global bagaimana data digunakan dan ditransformasikan untuk proses atau yang menggambarkan aliran data kedalam dan keluar sistem.

Berikut adalah diagram konteks yang diusulkan pada sistem informasi penjualan koran dan majalah di Maju Terus Agency, yang bisa kita lihat pada gambar III.6.



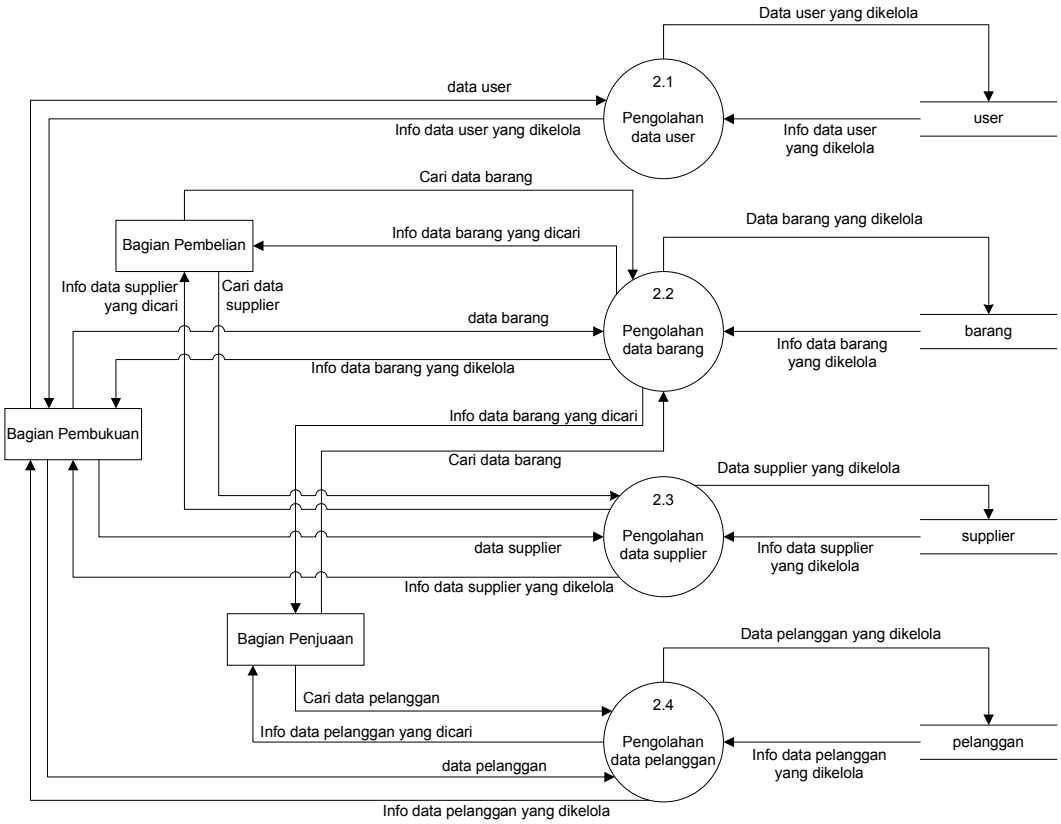
Gambar III.6. Diagram Konteks Sistem Informasi Penjualan Koran dan Majalah

3.1.4.1. DFD Level 0



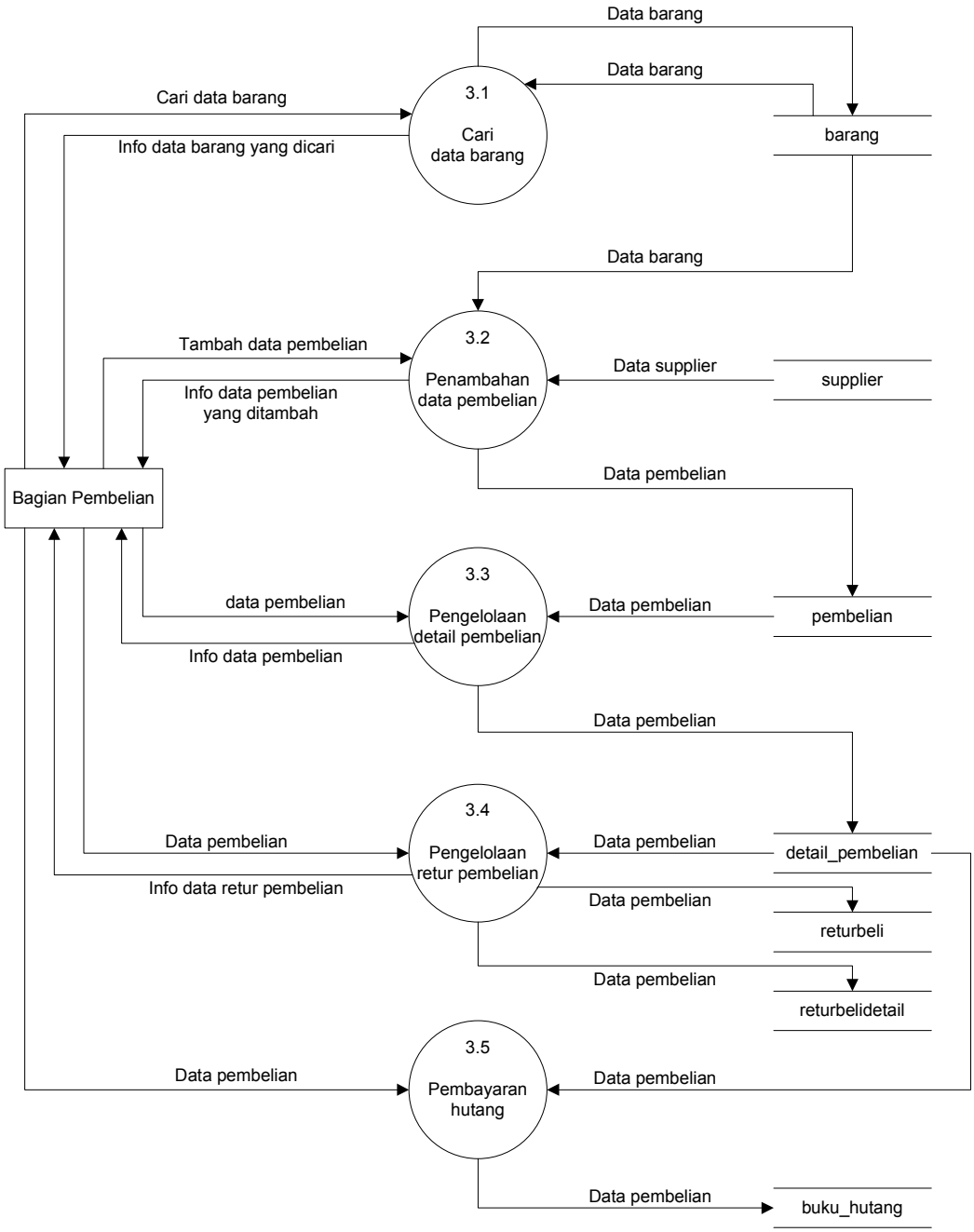
Gambar III.7. DFD Level 0 Sistem Informasi Penjualan Koran dan Majalah

3.1.4.3. DFD Level 1 Proses 2



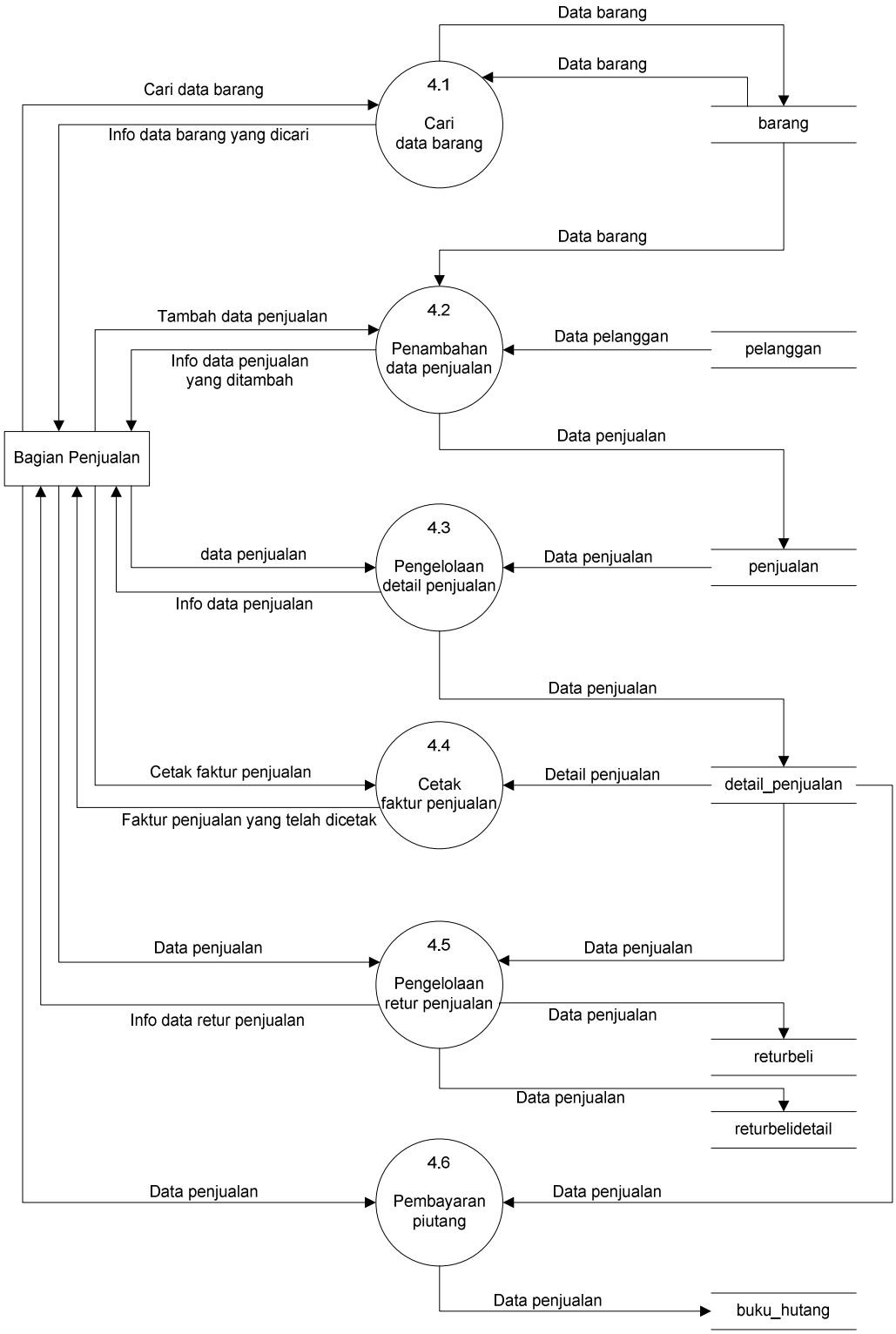
Gambar III.8. DFD Level 1 Proses Pengelolaan Data Master

3.1.4.4. DFD Level 1 Proses 3



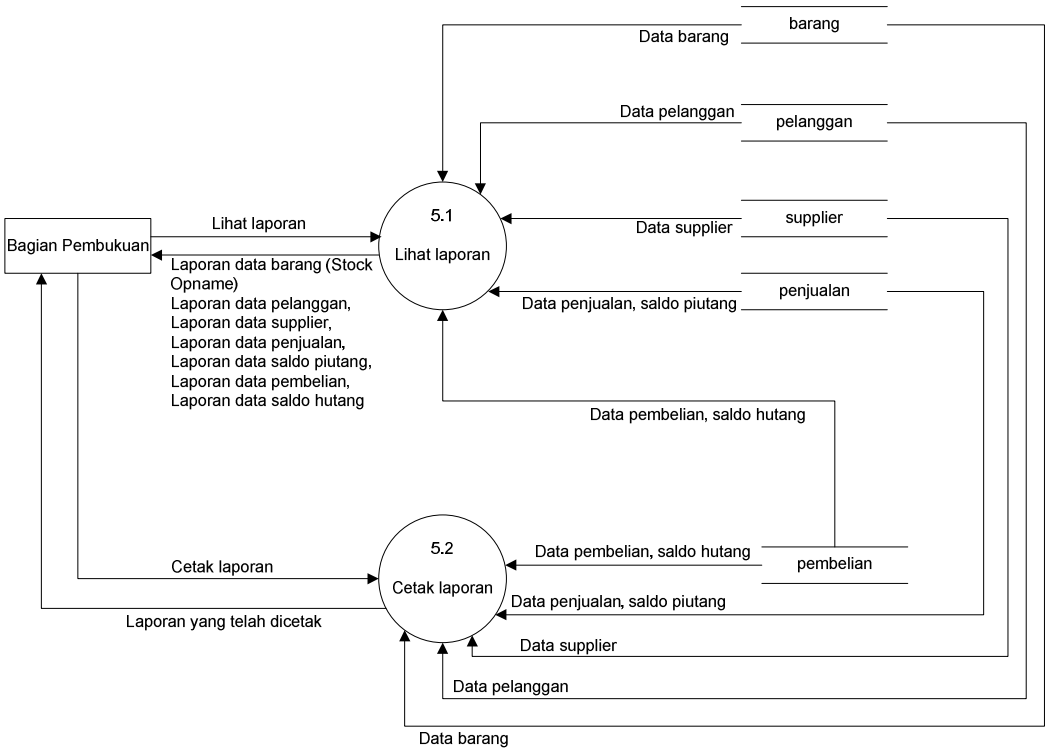
Gambar III.9. DFD Level 1 Proses Pengelolaan Pembelian

3.1.4.5. DFD Level 1 Proses 4



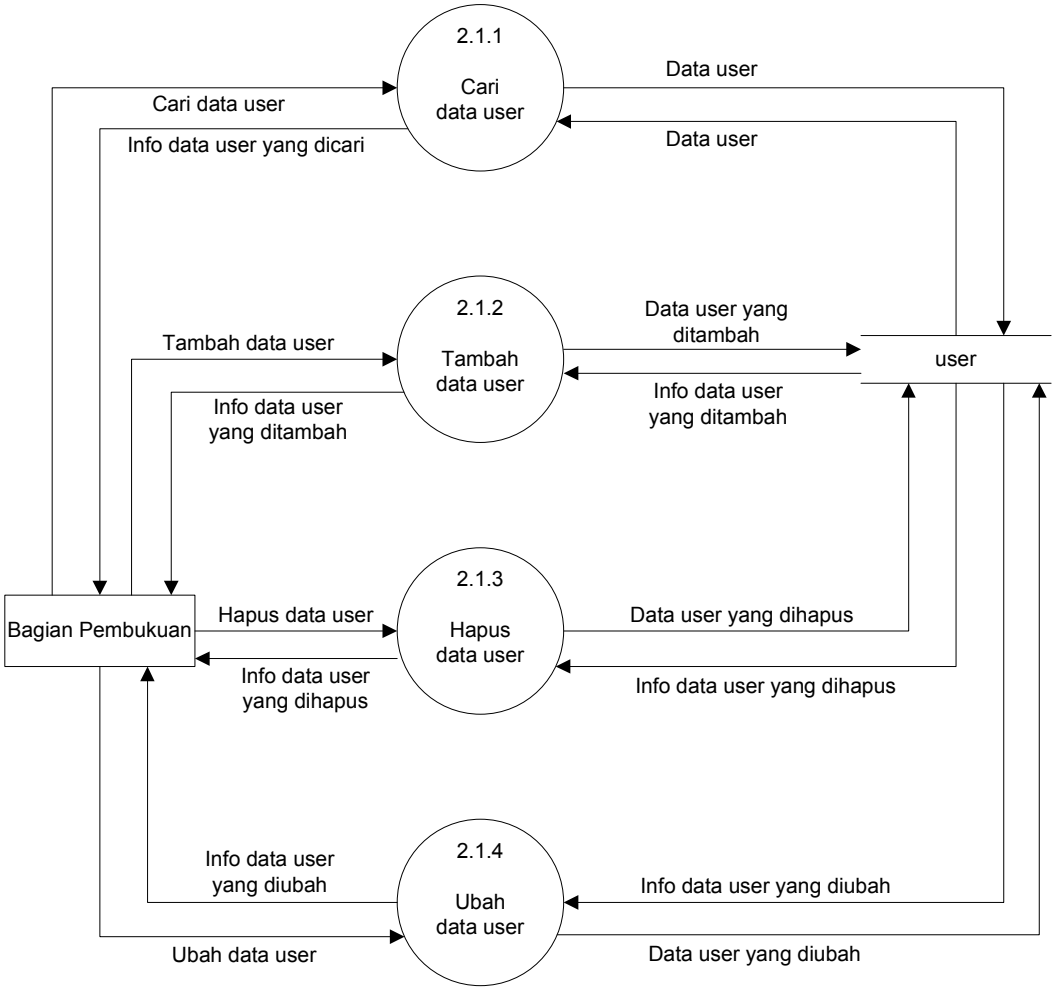
Gambar III.10. DFD Level 1 Proses Pengelolaan Penjualan

3.1.4.6. DFD Level 1 Proses 5



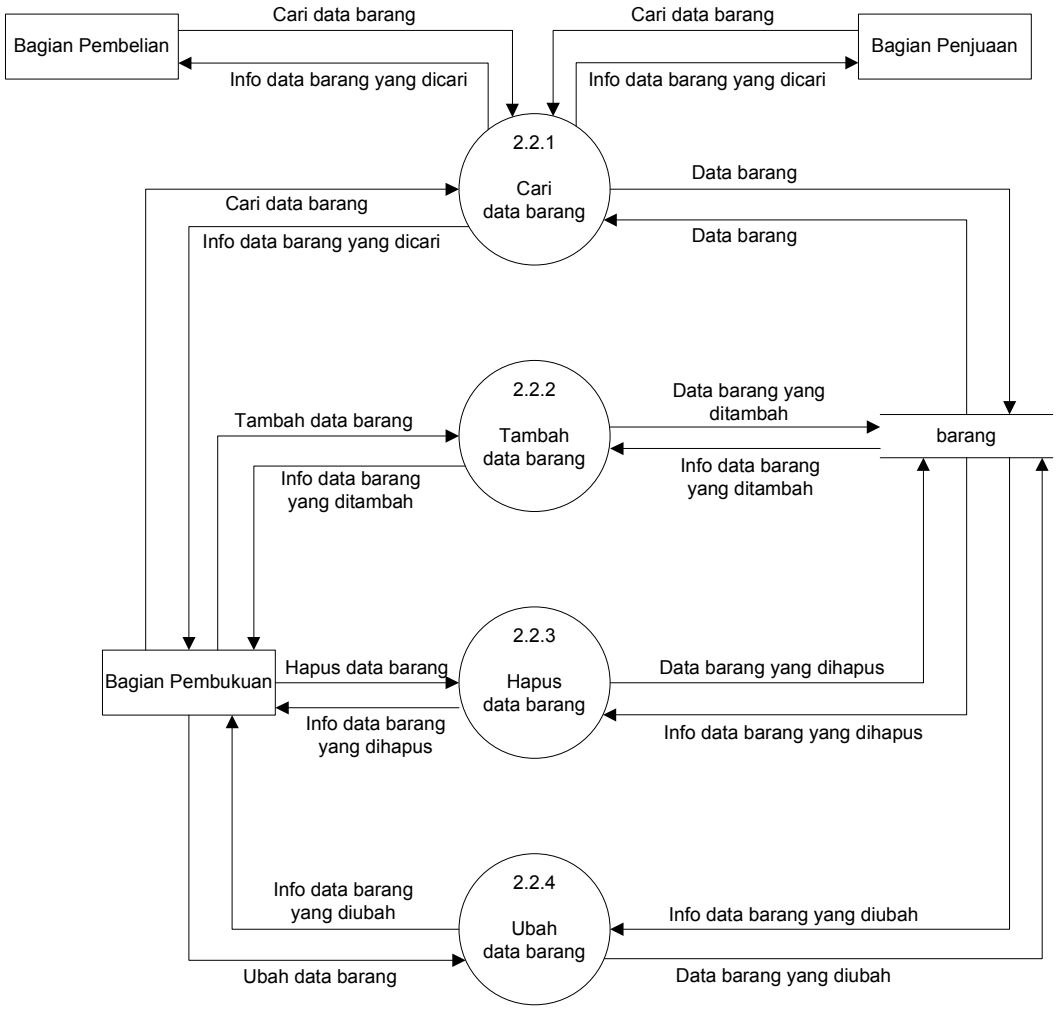
Gambar III.11. DFD Level 1 Proses Pembuatan Laporan

3.1.4.7. DFD Level 2 Proses 2.1



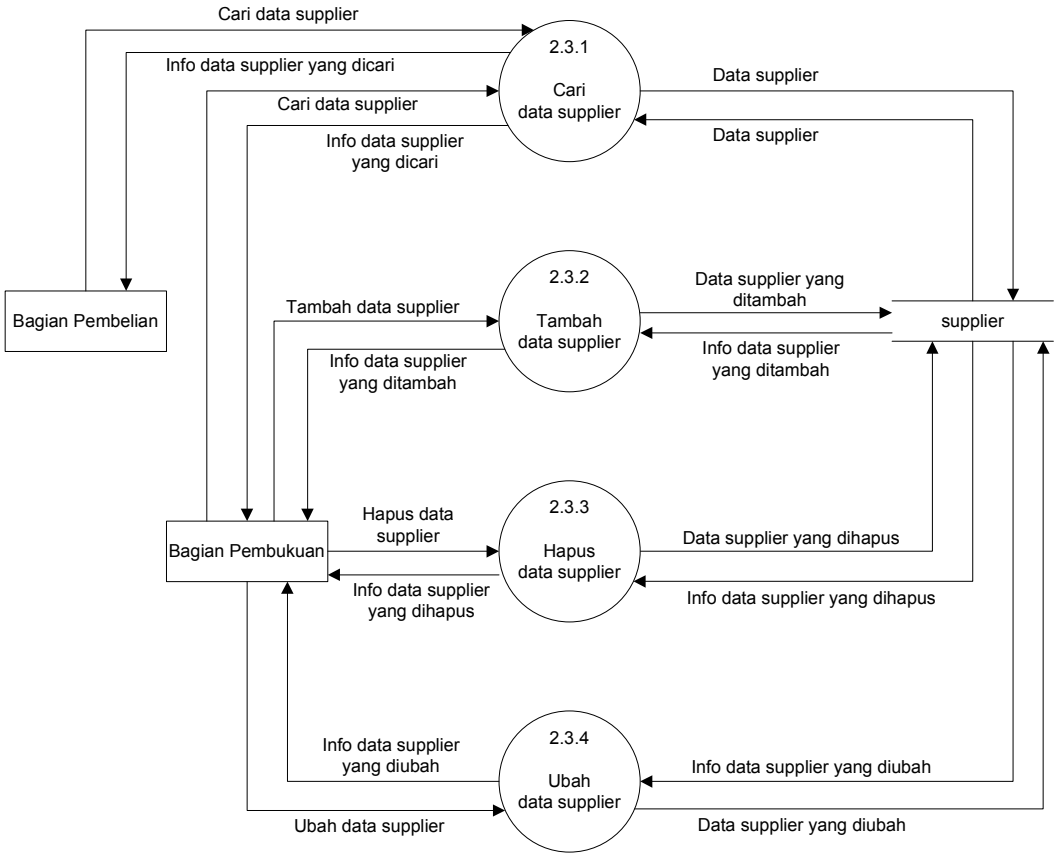
Gambar III.12. DFD Level 2 Proses Pengolahan Data User

3.1.4.8. DFD Level 2 Proses 2.2



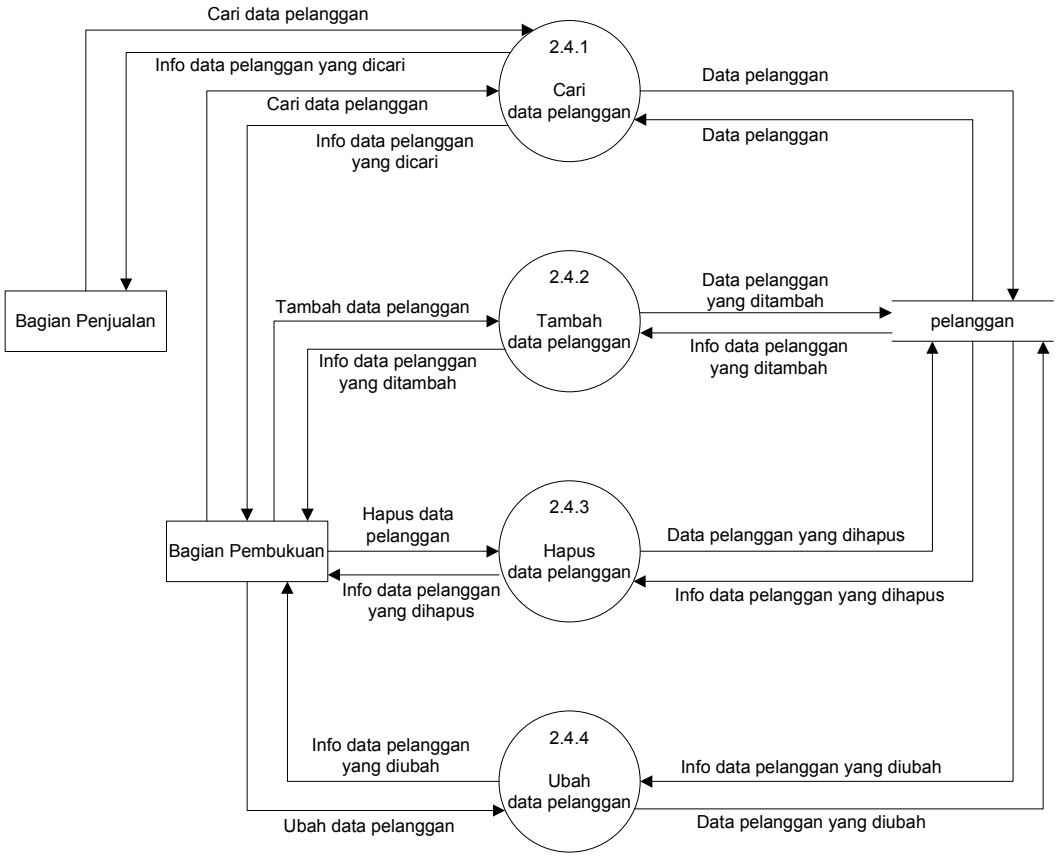
Gambar III.13. DFD Level 2 Proses Pengolahan Data Barang

3.1.4.9. DFD Level 2 Proses 2.3



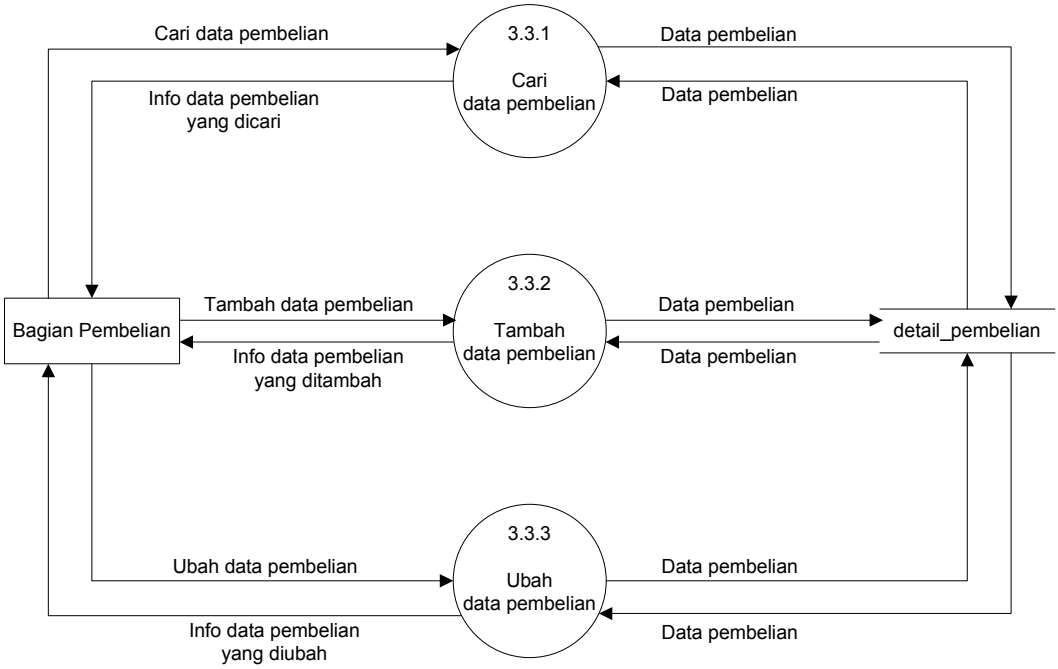
Gambar III.14. DFD Level 2 Proses Pengolahan Data Supplier

3.1.4.10.DFD Level 2 Proses 2.4



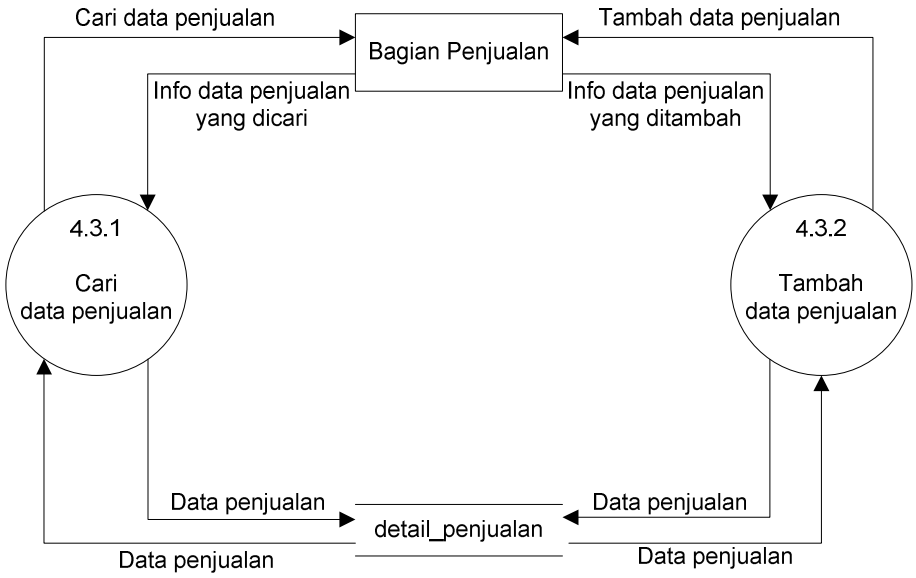
Gambar III.15. DFD Level 2 Proses Pengolahan Data Pelanggan

3.1.4.11.DFD Level 2 Proses 3.3



Gambar III.16. DFD Level 2 Proses Pengelolaan Detail Pembelian

3.1.4.12.DFD Level 2 Proses 4.3



Gambar III.17. DFD Level 2 Proses Pengelolaan Detail Penjualan

3.1.5 Spesifikasi Proses

Dari penggambaran DFD sebelumnya, dapat diketahui banyaknya aliran data baik arah serta tujuannya. Untuk lebih mengetahui kegunaannya dan hasil keluarannya dapat dilihat pada tabel III.5 berikut.

Tabel III.5. Spesifikasi Proses

No.	Proses	Keterangan
1	No. Proses	2.1
	Nama Proses	Pengolahan data user
	Source (Sumber)	Bagian Pembukuan
	Input	data user Info data user yang dikelola
	Output	Data user yang dikelola Info data user yang dikelola
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian Pembukuan mengelola data user sesuai dengan data user yang ada di *database*} *End*
2.	No. Proses	2.2
	Nama Proses	Pengolahan data barang
	Source (Sumber)	Bagian Pembukuan Bagian Pembelian Bagian Penjualan
	Input	Cari data barang Cari data barang data barang Info data barang yang dikelola
	Output	Info data barang yang dicari Info data barang yang dicari Data barang yang dikelola Info data barang yang dikelola
	Destination	Bagian Pembukuan Bagian Pembelian Bagian Penjualan
	Logika Proses	*Begin* {Bagian Pembukuan mengelola data barang sesuai dengan data barang yang ada di *database*} *End*
3.	No. Proses	2.3
	Nama Proses	Pengolahan data supplier

	Source (Sumber)	Bagian Pembukuan Bagian Pembelian
	Input	Cari data supplier data supplier Info data supplier yang dikelola
	Output	Info data supplier yang dicari Data supplier yang dikelola Info data supplier yang dikelola
	Destination	Bagian Pembukuan Bagian Pembelian
	Logika Proses	*Begin* {Bagian Pembukuan mengelola data supplier sesuai dengan data supplier yang ada di *database*} *End*
4.	No. Proses	2.4
	Nama Proses	Pengolahan data pelanggan
	Source (Sumber)	Bagian Pembukuan Bagian Penjualan
	Input	Cari data pelanggan data pelanggan Info data pelanggan yang dikelola
	Output	Info data pelanggan yang dicari Data pelanggan yang dikelola Info data pelanggan yang dikelola
	Destination	Bagian Pembukuan Bagian Penjualan
	Logika Proses	*Begin* {Bagian Pembukuan mengelola data pelanggan sesuai dengan data pelanggan yang ada di *database*} *End*
5.	No. Proses	3.1
	Nama Proses	Cari data barang
	Source (Sumber)	Bagian Pembelian
	Input	Cari data barang Data barang
	Output	Info data barang yang dicari Data barang
	Destination	Bagian Pembelian
	Logika Proses	*Begin* { Bagian Pembelian, melihat data barang *If* data barang ada *then* tampil data barang yang dicari *else* info data barang yang telah dicari} *End*
6.	No. Proses	3.2

	Nama Proses	Penambahan data pembelian
	Source (Sumber)	Bagian Pembelian
	Input	Data barang Data supplier Tambah data pembelian
	Output	Info data pembelian yang ditambah Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {*Insert* data pembelian sesuai dengan data pembelian yang baru, simpan pembelian yang baru ke *database*} *End*
7.	No. Proses	3.3
	Nama Proses	Pengolahan detail pembelian
	Source (Sumber)	Bagian Pembelian
	Input	data pembelian Data pembelian
	Output	Info data pembelian Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {*Insert* data detail pembelian sesuai dengan data detail pembelian yang baru, simpan detail pembelian yang baru ke *database*} *End*
8.	No. Proses	3.4
	Nama Proses	Pengolahan retur pembelian
	Source (Sumber)	Bagian Pembelian
	Input	data pembelian Data pembelian
	Output	Info data pembelian Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {*Insert* data retur pembelian sesuai dengan data retur pembelian yang baru, simpan retur pembelian yang baru ke *database*} *End*
9.	No. Proses	3.5
	Nama Proses	Pembayaran Hutang
	Source (Sumber)	Bagian Pembelian
	Input	data pembelian Data pembelian
	Output	Info data pembelian

		Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {*Insert* data pembayaran hutang sesuai dengan data hutang yang baru, simpan data hutang yang baru ke *database*} *End*
10.	No. Proses	4.1
	Nama Proses	Cari data barang
	Source (Sumber)	Bagian Penjualan
	Input	Cari data barang Data barang
	Output	Info data barang yang dicari Data barang
	Destination	Bagian Penjualan
	Logika Proses	*Begin* { Bagian Penjualan, melihat data barang *If* data barang ada *then* tampil data barang yang dicari *else* info data barang yang telah dicari} *End*
11.	No. Proses	4.2
	Nama Proses	Penambahan data penjualan
	Source (Sumber)	Bagian Penjualan
	Input	Data barang Data pelanggan Tambah data penjualan
	Output	Info data penjualan yang ditambah Data penjualan
	Destination	Bagian Penjualan
	Logika Proses	*Begin* {*Insert* data penjualan sesuai dengan data penjualan yang baru, simpan penjualan yang baru ke *database*} *End*
12.	No. Proses	4.3
	Nama Proses	Pengelolaan detail penjualan
	Source (Sumber)	Bagian Penjualan
	Input	Data penjualan data penjualan
	Output	Info data penjulan Data penjualan
	Destination	Bagian Penjualan
	Logika Proses	*Begin* {*Insert* data detail penjualan sesuai dengan data detail penjualan yang baru, simpan detail

		penjualan yang baru ke *database*} *End*
13.	No. Proses	4.4
	Nama Proses	Cetak faktur penjualan
	Source (Sumber)	Bagian Penjualan
	Input	Detail penjualan Cetak faktur penjualan
	Output	Faktur penjualan yang telah dicetak
	Destination	Bagian Penjualan
	Logika Proses	*Begin* {Bagian Penjualan mencetak faktur penjualan sesuai dengan detail penjualan yang ada di *database*} *End*
14.	No. Proses	4.5
	Nama Proses	Pengolahan retur penjualan
	Source (Sumber)	Bagian penjualan
	Input	Data penjualan Data penjualan
	Output	Info data penjualan Data penjualan
	Destination	Bagian penjualan
	Logika Proses	*Begin* {*Insert* data retur penjualan sesuai dengan data retur penjualan yang baru, simpan retur penjualan yang baru ke *database*} *End*
15.	No. Proses	4.6
	Nama Proses	Pembayaran Piutang
	Source (Sumber)	Bagian penjualan
	Input	data penjualan Data penjualan
	Output	Info data penjualan Data penjualan
	Destination	Bagian penjualan
	Logika Proses	*Begin* {*Insert* data pembayaran piutang sesuai dengan data piutang yang baru, simpan data piutang yang baru ke *database*} *End*
16.	No. Proses	5.1
	Nama Proses	Lihat laporan
	Source (Sumber)	Bagian Pembukuan
	Input	Lihat laporan Data barang

		Data supplier Data pelanggan Data pembelian, saldo hutang Data penjualan, saldo piutang
	Output	Laporan data barang (stock opname) Laporan data supplier Laporan data pelanggan Laporan data pembelian Laporan saldo hutang Laporan data penjualan Laporan saldo piutang
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian Pembukuan melihat laporan yang ada di *database*} *End*
17.	No. Proses	5.2
	Nama Proses	Cetak laporan
	Source (Sumber)	Bagian Pembukuan
	Input	Cetak laporan Data barang Data supplier Data pelanggan Data pembelian, saldo hutang Data penjualan, saldo piutang
	Output	Laporan yang telah dicetak
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian Pembukuan mencetak laporan yang ada di *database*} *End*
18.	No. Proses	2.1.1
	Nama Proses	Cari data user
	Source (Sumber)	Bagian Pembukuan
	Input	Cari data user Data user
	Output	Info data user yang dicari Data user
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian Pembukuan melihat data user *If* data user ada *then* tampil data user yang dicari *else* info data user yang telah dicari} *End*
19.	No. Proses	2.1.2

	Nama Proses	Tambah data user
	Source (Sumber)	Bagian Pembukuan
	Input	Tambah data user Data user yang ditambah
	Output	Info data user yang ditambah Info data user yang ditambah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {*Insert* data user sesuai dengan data user yang baru, simpan user yang baru ke *database*} *End*
20.	No. Proses	2.1.3
	Nama Proses	Hapus data user
	Source (Sumber)	Bagian Pembukuan
	Input	Hapus data user Data user yang dihapus
	Output	Info data user yang dihapus Info data user yang dihapus
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi user *Delete* data user sesuai dengan data user yang akan dihapus yang ada di *database*} *End*
21.	No. Proses	2.1.4
	Nama Proses	Ubah data user
	Source (Sumber)	Bagian Pembukuan
	Input	Ubah data user Data user yang diubah
	Output	Info data user yang diubah Info data user yang diubah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi user *Edit* data user sesuai dengan data user yang akan diubah, simpan data user yang telah diubah kedalam *database*} *End*
22.	No. Proses	2.2.1
	Nama Proses	Cari data barang
	Source (Sumber)	Bagian Pembukuan Bagian Pembelian Bagian Penjualan
	Input	Cari data barang Data barang

	Output	Info data barang yang dicari Data barang
	Destination	Bagian Pembukuan Bagian Pembelian Bagian Penjualan
	Logika Proses	*Begin* {Bagian Pembukuan, Bagian Pembelian, Bagian Penjualan melihat data barang *If* data barang ada *then* tampil data barang yang dicari *else* info data barang yang telah dicari} *End*
23.	No. Proses	2.2.2
	Nama Proses	Tambah data barang
	Source (Sumber)	Bagian Pembukuan
	Input	Tambah data barang Data barang yang ditambah
	Output	Info data barang yang ditambah Info data barang yang ditambah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {*Insert* data barang sesuai dengan data barang yang baru, simpan barang yang baru ke *database*} *End*
24.	No. Proses	2.2.3
	Nama Proses	Hapus data barang
	Source (Sumber)	Bagian Pembukuan
	Input	Hapus data barang Data barang yang dihapus
	Output	Info data barang yang dihapus Info data barang yang dihapus
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi barang *Delete* data barang sesuai dengan data barang yang akan dihapus yang ada di *database*} *End*
25.	No. Proses	2.2.4
	Nama Proses	Ubah data barang
	Source (Sumber)	Bagian Pembukuan
	Input	Ubah data barang Data barang yang diubah
	Output	Info data barang yang diubah Info data barang yang diubah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin*

		{Bagian pembukuan melihat informsai barang *Edit* data barang sesuai dengan data barang yang akan diubah, simpan data barang yang telah diubah kedalam *database*} *End*
26.	No. Proses	2.3.1
	Nama Proses	Cari data supplier
	Source (Sumber)	Bagian Pembukuan Bagian Pembelian
	Input	Cari data supplier Data supplier
	Output	Info data supplier yang dicari Data supplier
	Destination	Bagian Pembukuan Bagian Pembelian
	Logika Proses	*Begin* {Bagian Pembukuan, Bagian Pembelian, melihat data supplier *If* data supplier ada *then* tampil data supplier yang dicari *else* info data supplier yang telah dicari} *End*
27.	No. Proses	2.3.2
	Nama Proses	Tambah data supplier
	Source (Sumber)	Bagian Pembukuan
	Input	Tambah data supplier Data supplier yang ditambah
	Output	Info data supplier yang ditambah Info data supplier yang ditambah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {*Insert* data supplier sesuai dengan data supplier yang baru, simpan supplier yang baru ke *database*} *End*
28.	No. Proses	2.3.3
	Nama Proses	Hapus data supplier
	Source (Sumber)	Bagian Pembukuan
	Input	Hapus data supplier Data supplier yang dihapus
	Output	Info data supplier yang dihapus Info data supplier yang dihapus
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi supplier *Delete* data supplier sesuai dengan data supplier

		yang akan dihapus yang ada di *database*} *End*
29.	No. Proses	2.3.4
	Nama Proses	Ubah data supplier
	Source (Sumber)	Bagian Pembukuan
	Input	Ubah data supplier Data supplier yang diubah
	Output	Info data supplier yang diubah Info data supplier yang diubah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi supplier *Edit* data supplier sesuai dengan data supplier yang akan diubah, simpan data supplier yang telah diubah kedalam *database*} *End*
30.	No. Proses	2.4.1
	Nama Proses	Cari data pelanggan
	Source (Sumber)	Bagian Pembukuan Bagian Penjualan
	Input	Cari data pelanggan Data pelanggan
	Output	Info data pelanggan yang dicari Data pelanggan
	Destination	Bagian Pembukuan Bagian Penjualan
	Logika Proses	*Begin* {Bagian Pembukuan, Bagian Penjualan melihat data pelanggan *If* data pelanggan ada *then* tampil data pelanggan yang dicari *else* info data pelanggan yang telah dicari} *End*
31.	No. Proses	2.4.2
	Nama Proses	Tambah data pelanggan
	Source (Sumber)	Bagian Pembukuan
	Input	Tambah data pelanggan Data pelanggan yang ditambah
	Output	Info data pelanggan yang ditambah Info data pelanggan yang ditambah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {*Insert* data pelanggan sesuai dengan data pelanggan yang baru, simpan pelanggan yang baru ke *database*} *End*

32.	No. Proses	2.4.3
	Nama Proses	Hapus data pelanggan
	Source (Sumber)	Bagian Pembukuan
	Input	Hapus data pelanggan Data pelanggan yang dihapus
	Output	Info data pelanggan yang dihapus Info data pelanggan yang dihapus
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi pelanggan *Delete* data pelanggan sesuai dengan data pelanggan yang akan dihapus yang ada di *database*} *End*
33.	No. Proses	2.4.4
	Nama Proses	Ubah data pelanggan
	Source (Sumber)	Bagian Pembukuan
	Input	Ubah data pelanggan Data pelanggan yang diubah
	Output	Info data pelanggan yang diubah Info data pelanggan yang diubah
	Destination	Bagian Pembukuan
	Logika Proses	*Begin* {Bagian pembukuan melihat informasi pelanggan *Edit* data pelanggan sesuai dengan data pelanggan yang akan diubah, simpan data pelanggan yang telah diubah kedalam *database*} *End*
34.	No. Proses	3.3.1
	Nama Proses	Cari data pembelian
	Source (Sumber)	Bagian Pembelian
	Input	Cari data pembelian Data pembelian
	Output	Info data pembelian yang dicari Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {Bagian Pembelian melihat data pembelian *If* data pembelian ada *then* tampil data pembelian yang dicari *else* info data pembelian yang telah dicari} *End*
35.	No. Proses	3.3.2
	Nama Proses	Tambah data pembelian
	Source (Sumber)	Bagian Pembelian

	Input	Tambah data pembelian Data pembelian
	Output	Info data pembelian yang ditambah Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {*Insert* data pembelian sesuai dengan data pembelian yang baru, simpan pembelian yang baru ke *database*} *End*
36.	No. Proses	3.3.3
	Nama Proses	Ubah data pembelian
	Source (Sumber)	Bagian Pembelian
	Input	Ubah data pembelian Data pembelian
	Output	Info data pembelian yang diubah Data pembelian
	Destination	Bagian Pembelian
	Logika Proses	*Begin* {Bagian Pembelian melihat informasi pembelian *Edit* data pembelian sesuai dengan data pembelian yang akan diubah, simpan data pembelian yang telah diubah kedalam *database*} *End*
37.	No. Proses	4.3.1
	Nama Proses	Cari data penjualan
	Source (Sumber)	Bagian Penjualan
	Input	Cari data penjualan Data penjualan
	Output	Info data penjualan yang dicari Data penjualan
	Destination	Bagian Penjualan
	Logika Proses	*Begin* {Bagian Penjualan melihat data penjualan *If* data penjualan ada *then* tampil data penjualan yang dicari *else* info data penjualan yang telah dicari} *End*
38.	No. Proses	4.3.2
	Nama Proses	Tambah data penjualan
	Source (Sumber)	Bagian Penjualan
	Input	Tambah data penjualan Data penjualan
	Output	Info data penjualan yang ditambah Data penjualan

	Destination	Bagian Penjualan
	Logika Proses	*Begin* {*Insert* data penjualan sesuai dengan data penjualan yang baru, simpan penjualan yang baru ke *database*} *End*

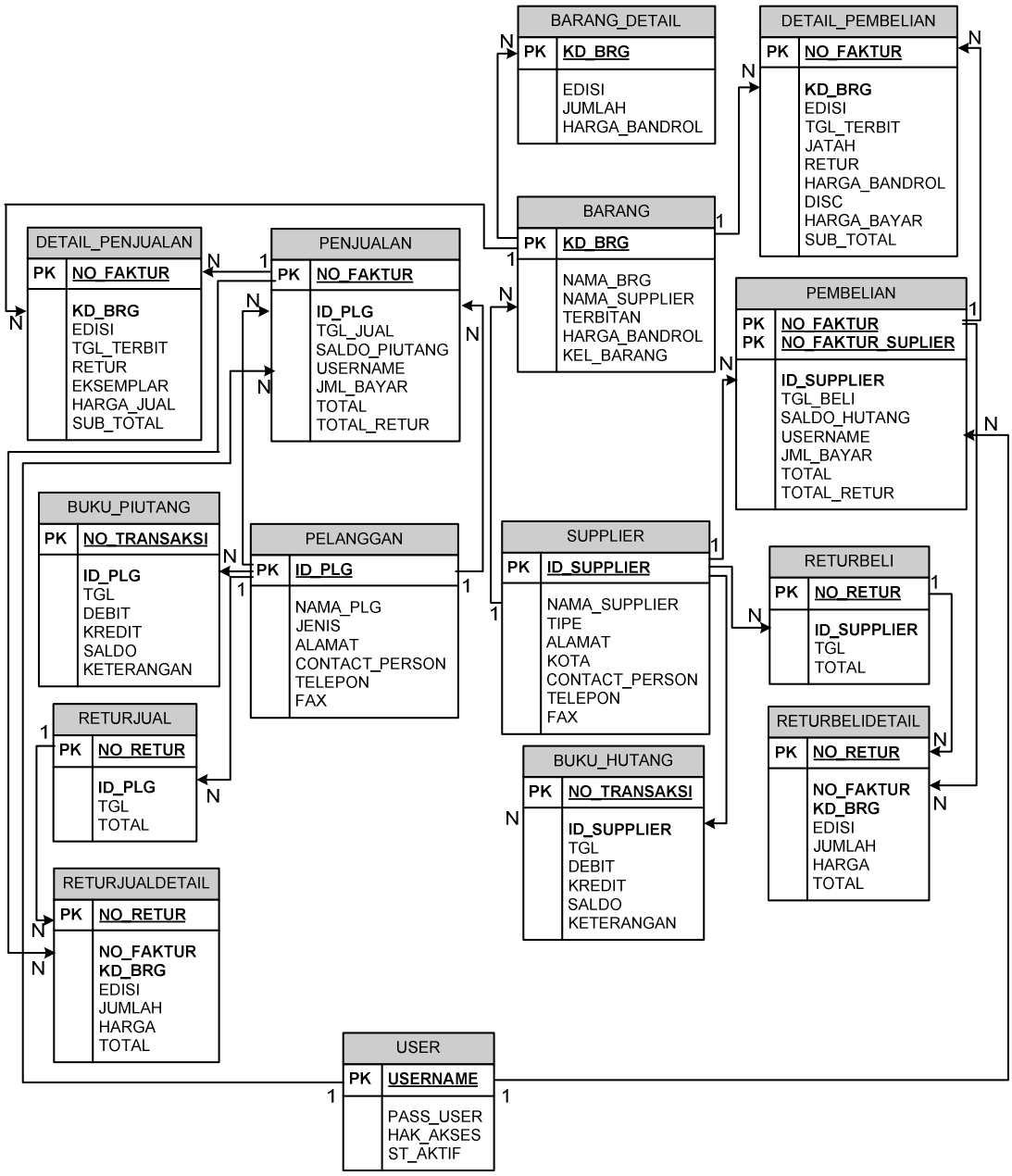
3.1 Perancangan Sistem

Perancangan akan dimulai dengan tahap analisis terhadap sistem selesai dilakukan. Perancangan dapat didefinisikan sebagai penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Tahapan ini menyangkut mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah instalasi dari sistem akan benar-benar memuaskan dari rancang bangunyang telah ditetapkan pada akhir tahap analisis sistem. Alat bantu yang digunakan untuk menggambarkan perancangan sistem secara umum yang akan dibangun, yaitu diagram konteks dan *data flow diagram*. Untuk menjelaskan bagaimana suatu masukan diproses pada sistem maka bv digunakan spesifikasi proses dan kamus data untuk mengetahui aliran data yang mengalir pada sistem.

3.2.1 Perancangan Basis Data

3.1.3.3. Skema Relasi

Skema Relasi Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency dapat dilihat pada gambar III.18 berikut:



Gambar III.18. Skema Relasi

3.1.3.4. Struktur Tabel

Struktur tabel dalam Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency adalah sebagai berikut:

Tabel III.6. Struktur Tabel user

No.	Nama Field	Type	Length	Keterangan
1.	Username	Char	10	Primary Key
2.	Pass_user	Char	32	
3.	Hak_akses	Char	10	
4.	St_aktif	Tinyint		

Tabel III.7. Struktur Tabel barang

No.	Nama Field	Type	Length	Keterangan
1.	Kd_brg	Char	20	Primary Key
2.	Nama_brg	Char	50	
3.	Terbitan	Char	2	
4.	Nama_supplier	Char	30	
5.	Harga_bandrol	Int		
6.	Kel_barang	Enum		

Tabel III.8. Struktur Tabel supplier

No.	Nama Field	Type	Length	Keterangan
1.	Id_supplier	Char	8	Primary Key
2.	Nama_supplier	Char	30	
3.	Tipe	Char	2	
4.	Alamat	Char	50	
5.	Kota	Char	30	
6.	Contact_person	Char	30	
7.	Telepon	Char	15	
8.	Fax	Char	15	

Tabel III.9. Struktur Tabel pelanggan

No.	Nama Field	Type	Length	Keterangan
1.	Id_plg	Char	8	Primary Key
2.	Nama_plg	Char	30	
3.	Jenis	Char	2	
4.	Alamat	Char	50	
5.	Contact_person	Char	30	
6.	Telepon	Char	15	
7.	Fax	Char	15	

Tabel III.10. Struktur Tabel pembelian

No.	Nama Field	Type	Length	Keterangan
1.	No_faktur	Char	20	Primary Key
2.	No_faktur_suplier	Char	15	Primary Key
3.	Tgl_beli	Date		
4.	Id_supplier	Char	8	Foreign Key
5.	Saldo_hutang	Int		
6.	Jml_bayar	Int		
7.	Total	Int		
8.	Total_retur	Int		

Tabel III.11. Struktur Tabel penjualan

No.	Nama Field	Type	Length	Keterangan
1.	No_faktur	Char	20	Primary Key
2.	Tgl_jual	Date		
3.	Id_plg	Char	8	Foreign Key
4.	Saldo_piutang	Int		
5.	Jml_bayar	Int		
6.	Total	Int		
7.	Total_retur	Int		

Tabel III.12. Struktur Tabel detail_pembelian

No.	Nama Field	Type	Length	Keterangan
1.	No_faktur	Char	20	Primary Key
2.	Kd_brg	Char	20	Foreign Key
3.	Edisi	Char	10	
4.	Tgl_terbit	Date		
5.	Jatah	Int		
6.	Retur	Int		
7.	Harga_bandrol	Int		
8.	Disc	Int		
9.	Harga_bayar	Int		
10.	Sub_total	Int		

Tabel III.13. Struktur Tabel detail_penjualan

No.	Nama Field	Type	Length	Keterangan
1.	No_faktur	Char	20	Primary Key
2.	Kd_brg	Char	20	Foreign Key
3.	Edisi	Char	10	
4.	Tgl_terbit	Date		
5.	Eksemplar	Int		
6.	Retur	Int		
7.	Harga_jual	Int		
8.	Sub_total	Int		

Tabel III.14. Struktur Tabel barang_detail

No.	Nama Field	Type	Length	Keterangan
1.	Kode_brg	Char	20	Primary Key
2.	Edisi	Char	10	
3.	Jumlah	Int		
4.	Harga_bandrol	Int		

Tabel III.15. Struktur Tabel buku_hutang

No.	Nama Field	Type	Length	Keterangan
1.	No_transaksi	Varchar	15	Primary Key
2.	Tgl	Date		
3.	Id_supplier	Char	8	Foreign Key
4.	Debit	Int		
5.	Kredit	Int		
6.	Saldo	Int		
7.	Keterangan	Enum		

Tabel III.16. Struktur Tabel buku_piutang

No.	Nama Field	Type	Length	Keterangan
1.	No_transaksi	Char	15	Primary Key
2.	Tgl	Date		
3.	Id_plg	Char	8	Foreign Key
4.	Debit	Int		
5.	Kredit	Int		
6.	Saldo	Int		
7.	Keterangan	Enum		

Tabel III.17. Struktur Tabel returbeli

No.	Nama Field	Type	Length	Keterangan
1.	No_retur	Char	20	Primary Key
2.	Tgl	Date		
3.	Id_supplier	Char	8	Foreign Key
4.	Total	Int		

Tabel III.18. Struktur Tabel returjual

No.	Nama Field	Type	Length	Keterangan
1.	No_retur	Char	20	Primary Key
2.	Tgl	Date		
3.	Id_plg	Char	8	Foreign Key
4.	Total	Int		

Tabel III.19. Struktur Tabel returbelidetail

No.	Nama Field	Type	Length	Keterangan
1.	No_retur	Char	20	Primary Key
2.	No_faktur	Char	20	Foreign Key
3.	Kd_brg	Char	20	Foreign Key
4.	Edisi	Char	10	
5.	Jumlah	Int		
6.	Harga	Int		
7.	Total	Int		

Tabel III.20. Struktur Tabel returjualdetail

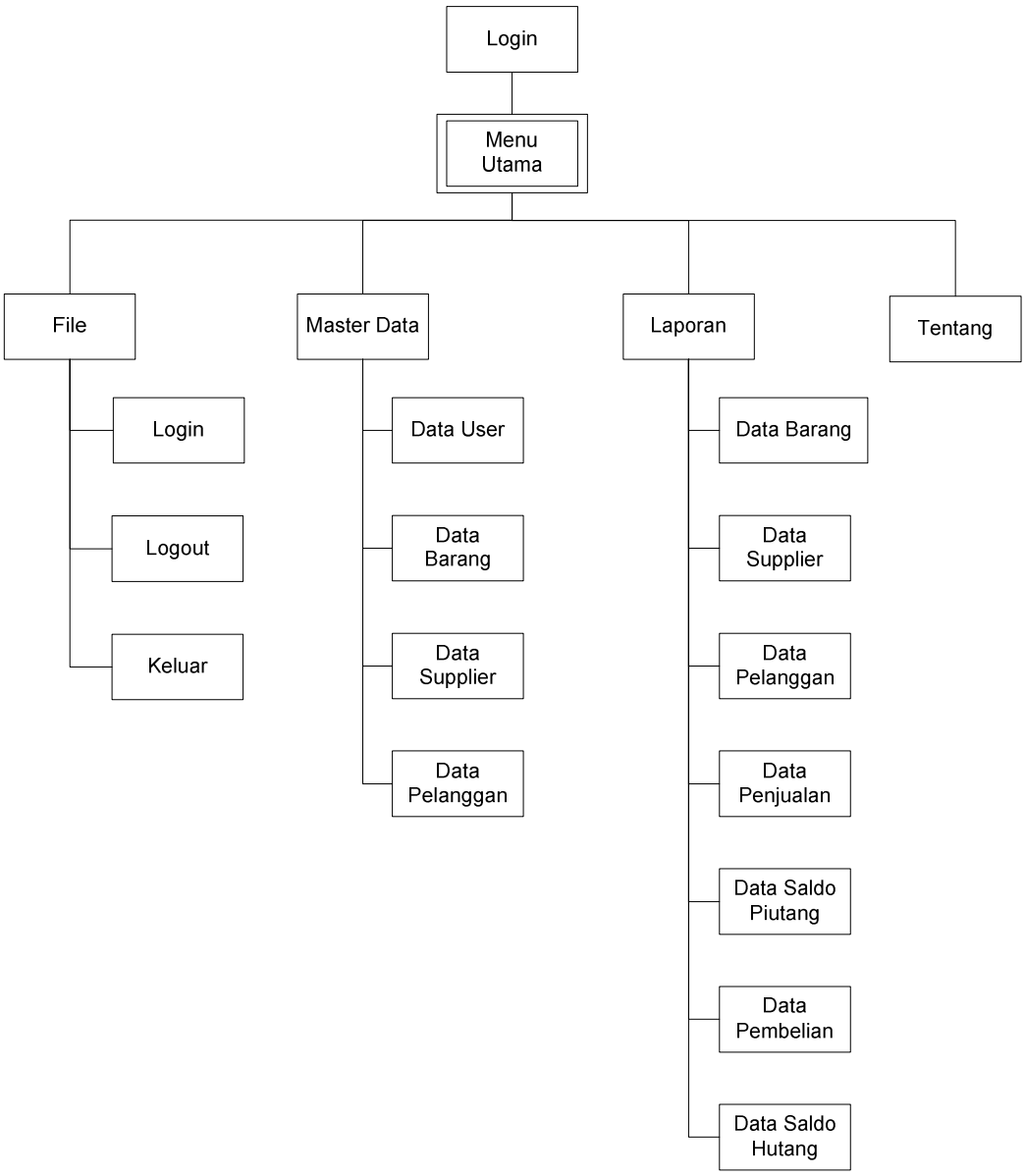
No.	Nama Field	Type	Length	Keterangan
1.	No_retur	Char	20	Primary Key
2.	No_faktur	Char	20	Foreign Key
3.	Kd_brg	Char	20	Foreign Key
4.	Edisi	Char	10	
5.	Jumlah	Int		
6.	Harga	Int		
7.	Total	Int		

3.2.2 Perancangan Struktur Menu

Struktur menu dibangun untuk menjelaskan perancangan menu proses yang dapat digunakan oleh pengguna. Struktur menu secara umum yang dibuat dalam sistem ini terdiri dari menu-menu dan sub menunya.

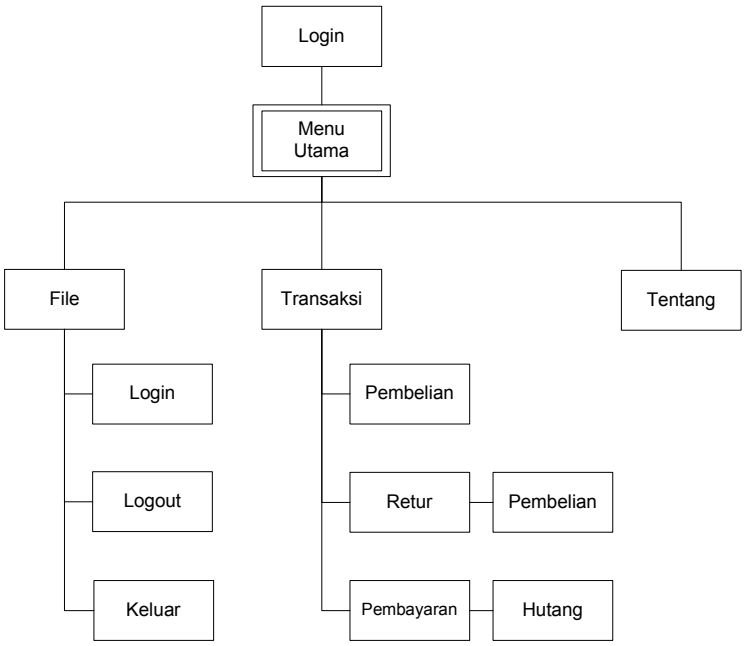
Perancangan struktur menu aplikasi sistem informasi penjualan koran dan majalah ini menggunakan arsitektur menu hierarki, dimana strukturnya akan dibagi tiga sesuai dengan kebutuhan masing-masing *user*, yaitu struktur menu untuk bagian pembukuan (administrator), bagian pembelian dan bagian penjualan.

Struktur menu untuk bagian pembukuan (administrator) dapat dilihat pada gambar III.19. berikut ini.



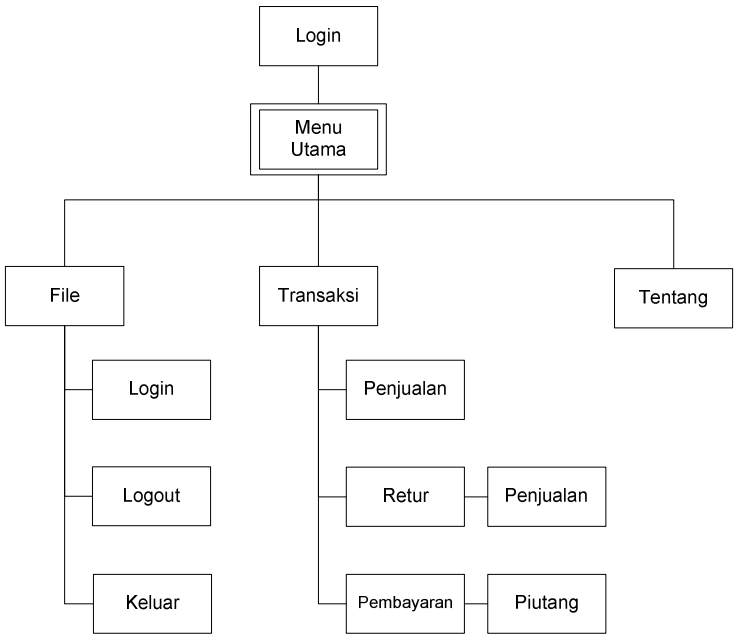
Gambar III.19. Struktur Menu Bagian Pembukuan (Administrator)

Struktur menu untuk bagian pembelian dapat dilihat pada gambar III.20. berikut ini.



Gambar III.20. Struktur Menu Bagian Pembelian

Struktur menu untuk bagian penjualan dapat dilihat pada gambar III.21. berikut ini.

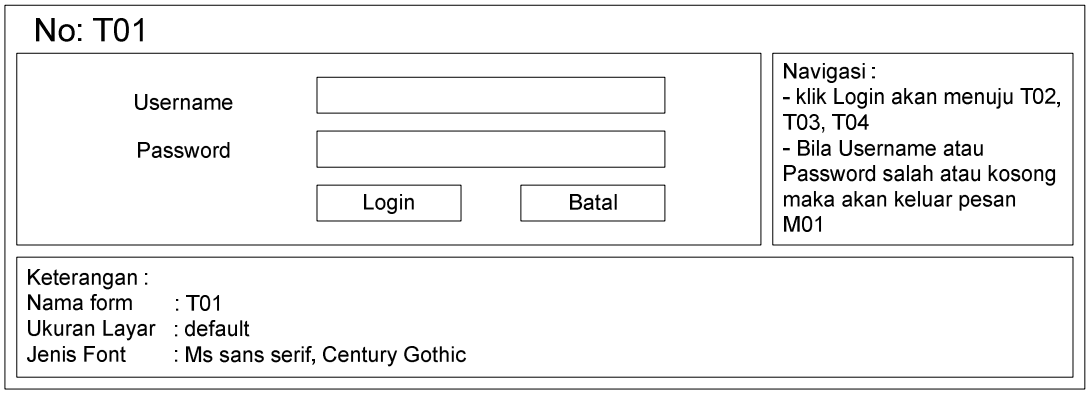


Gambar III.21. Struktur Menu Bagian Penjualan

3.2.3 Perancangan Antarmuka

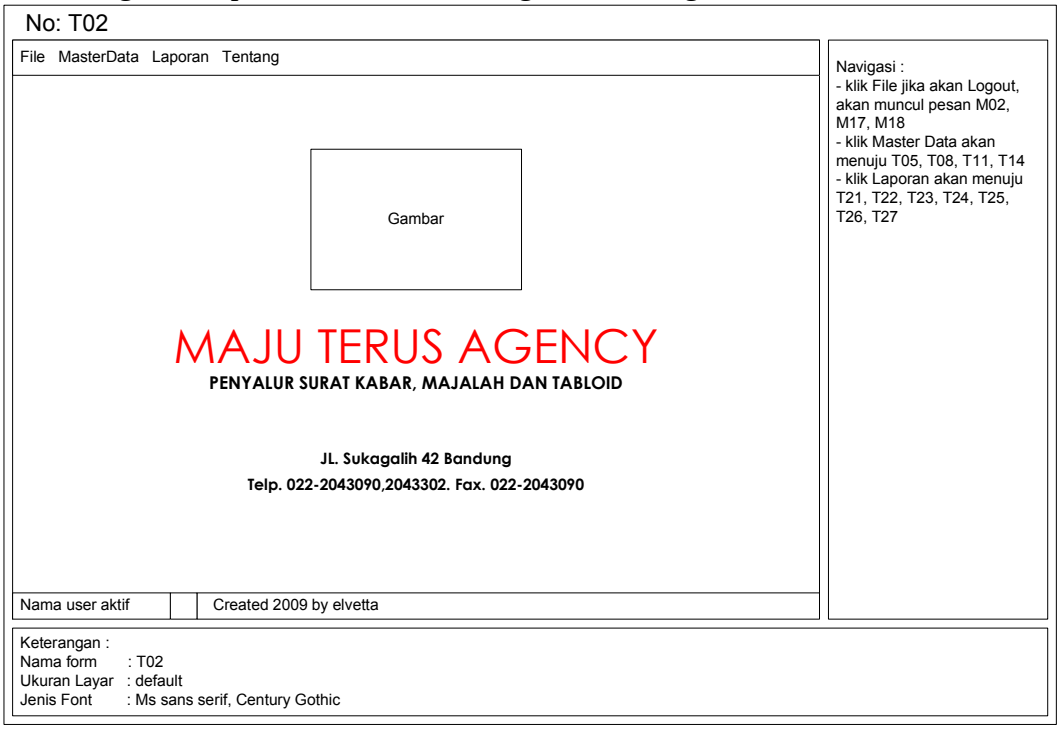
Perancangan antarmuka dibuat untuk menggambarkan tampilan program yang akan digunakan oleh pengguna untuk berinteraksi dengan aplikasi yang dibuat. Perancangan dibuat berdasarkan tampilan antarmuka baik *input* maupun *output* yang akan dihasilkan saat aplikasi diimplementasikan.

1. Perancangan Tampilan Menu Login



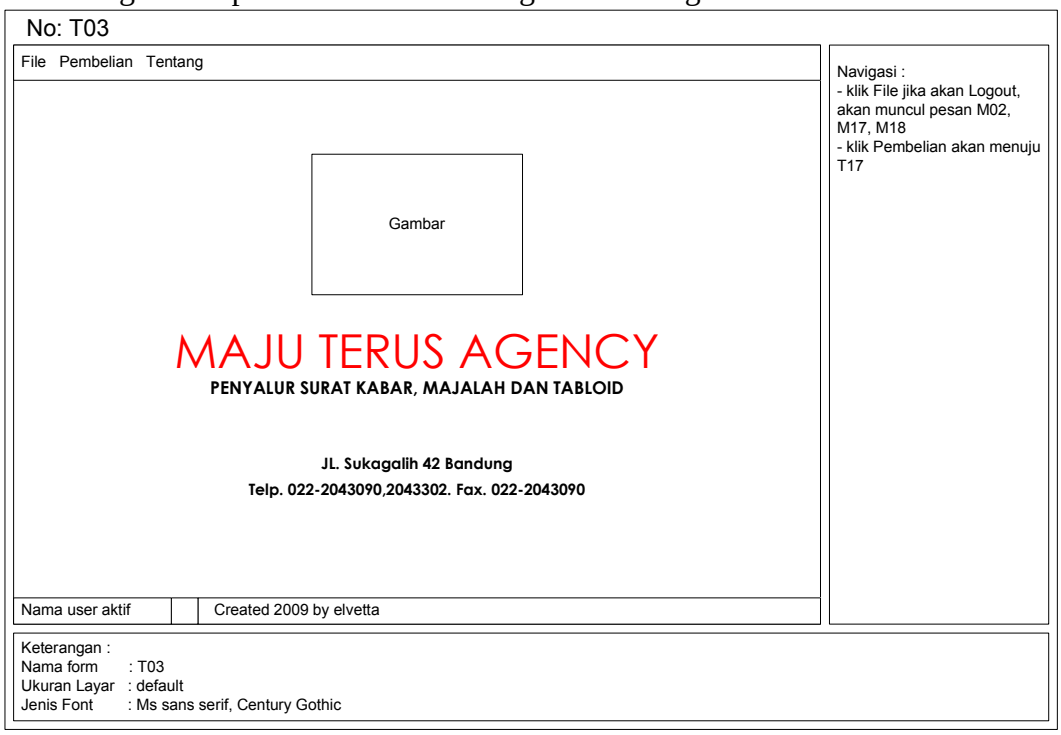
Gambar III.22. Perancangan Tampilan Menu Login

2. Perancangan Tampilan Menu setelah Login untuk Bagian Pembukuan



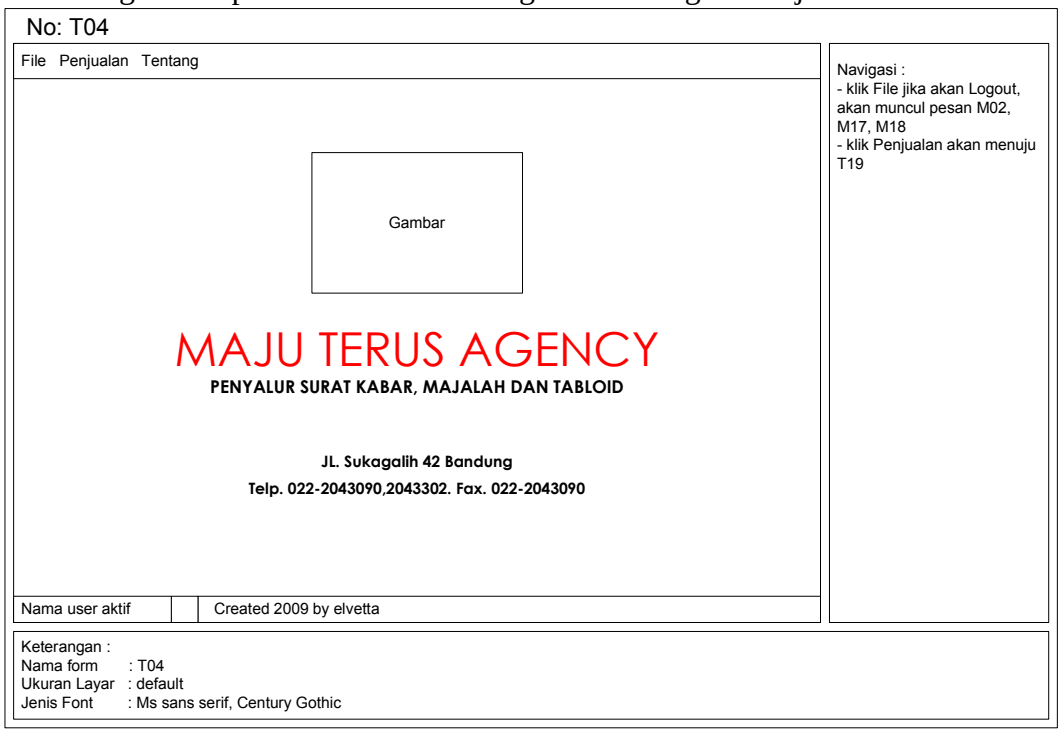
Gambar III.23. Perancangan Tampilan Menu setelah Login untuk Bagian Pembukuan (Administrator)

3. Perancangan Tampilan Menu setelah Login untuk Bagian Pembelian



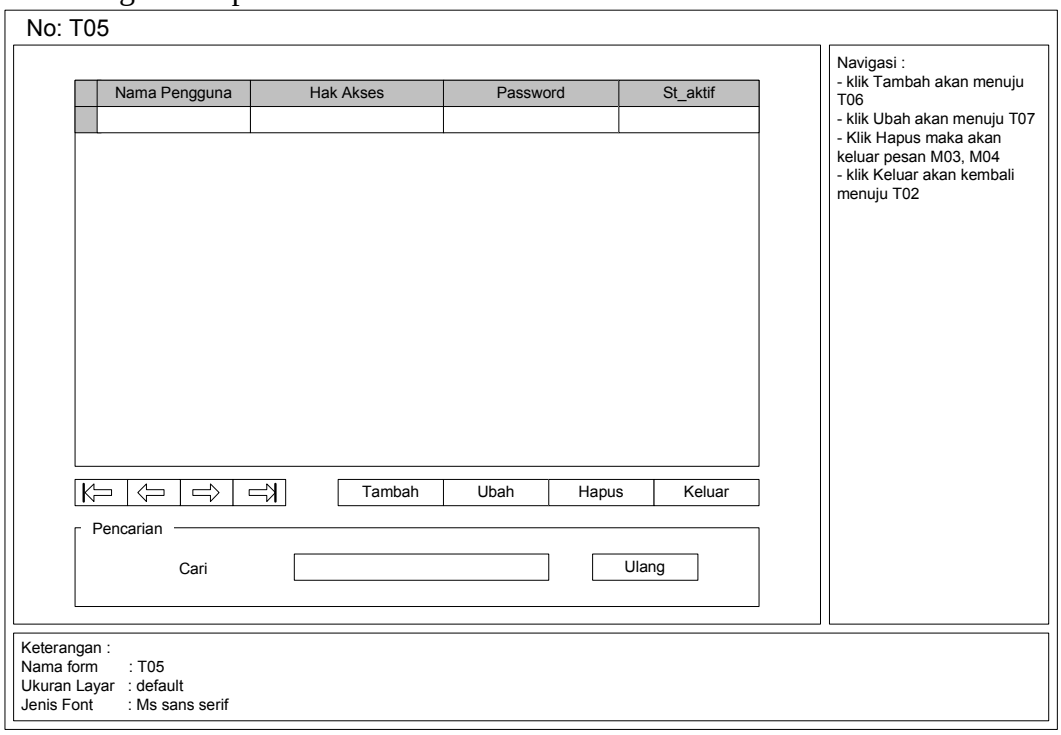
Gambar III.24. Perancangan Tampilan Menu setelah Login untuk Bagian Pembelian

4. Perancangan Tampilan Menu setelah Login untuk Bagian Penjualan



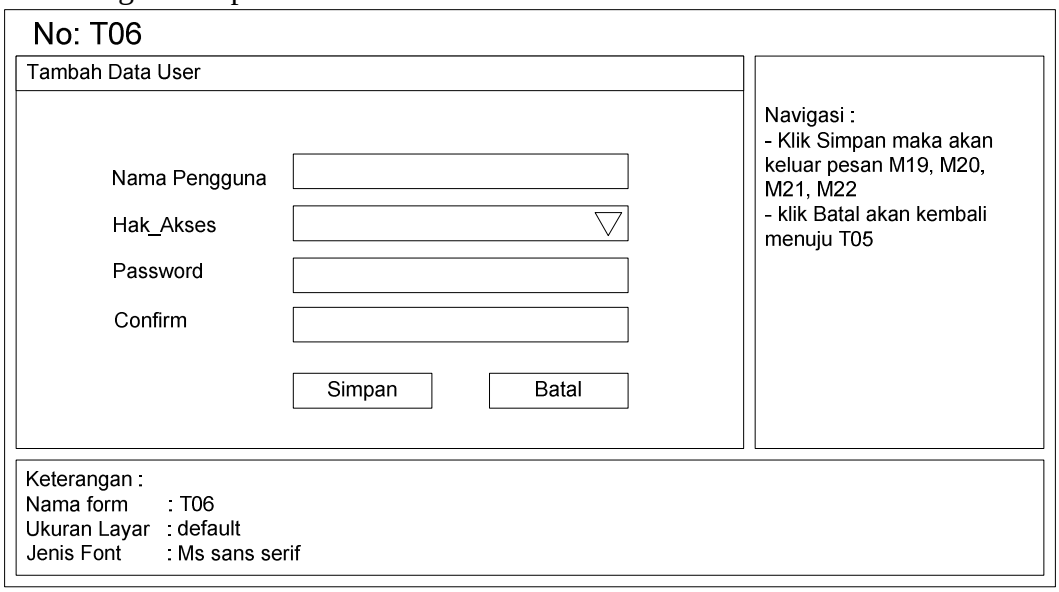
Gambar III.25. Perancangan Tampilan Menu setelah Login untuk Bagian Penjualan

5. Perancangan Tampilan Form Data User



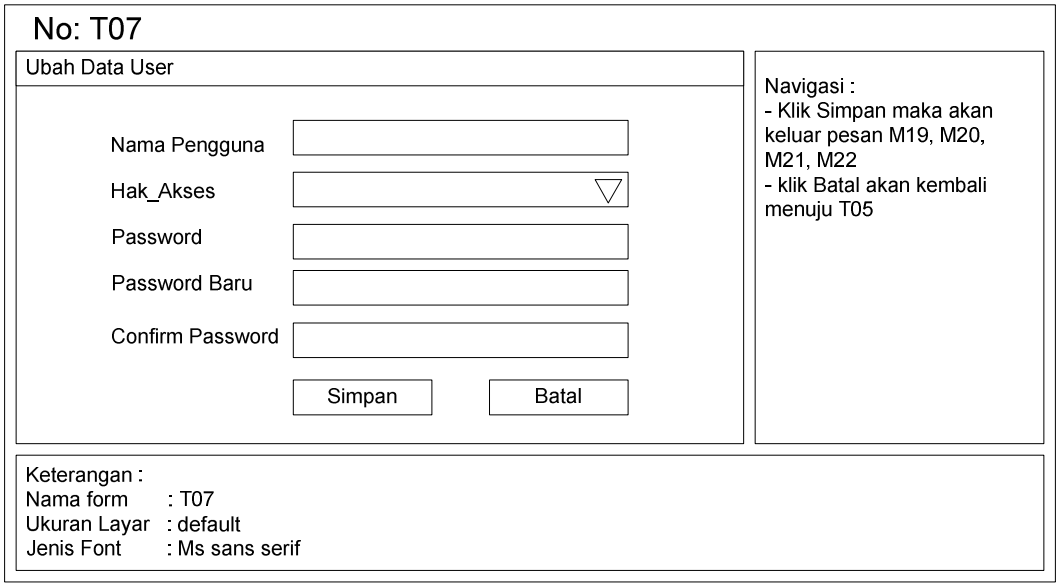
Gambar III.26. Perancangan Tampilan Form Data User

6. Perancangan Tampilan Form Tambah Data User



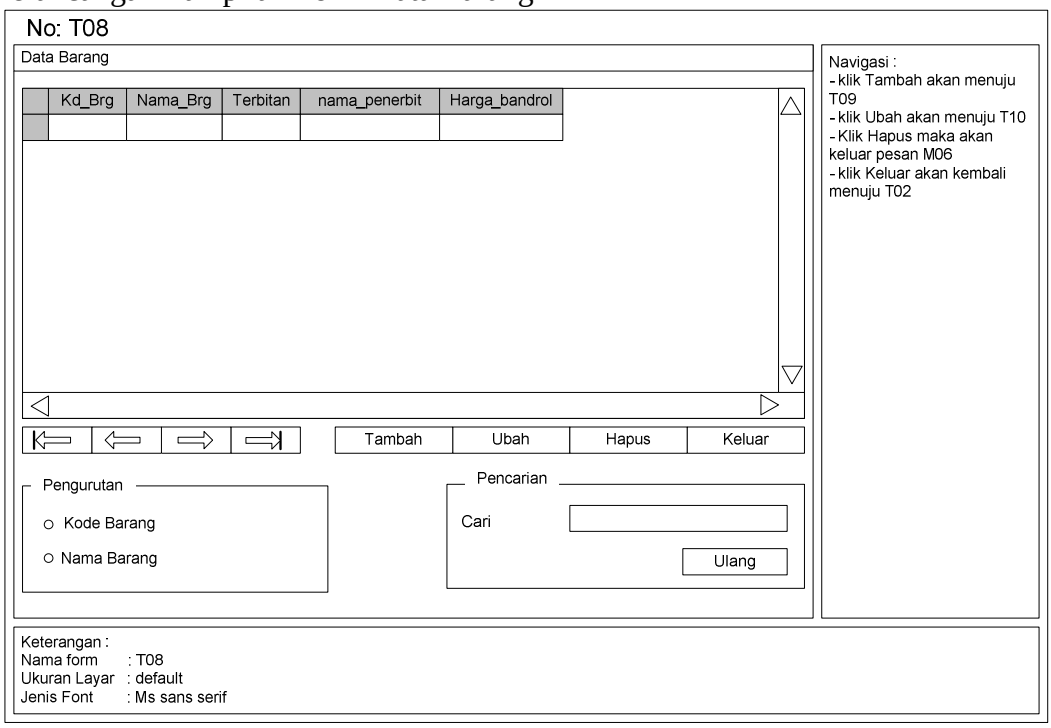
Gambar III.27. Perancangan Tampilan Form Tambah Data User

7. Perancangan Tampilan Form Ubah Data User



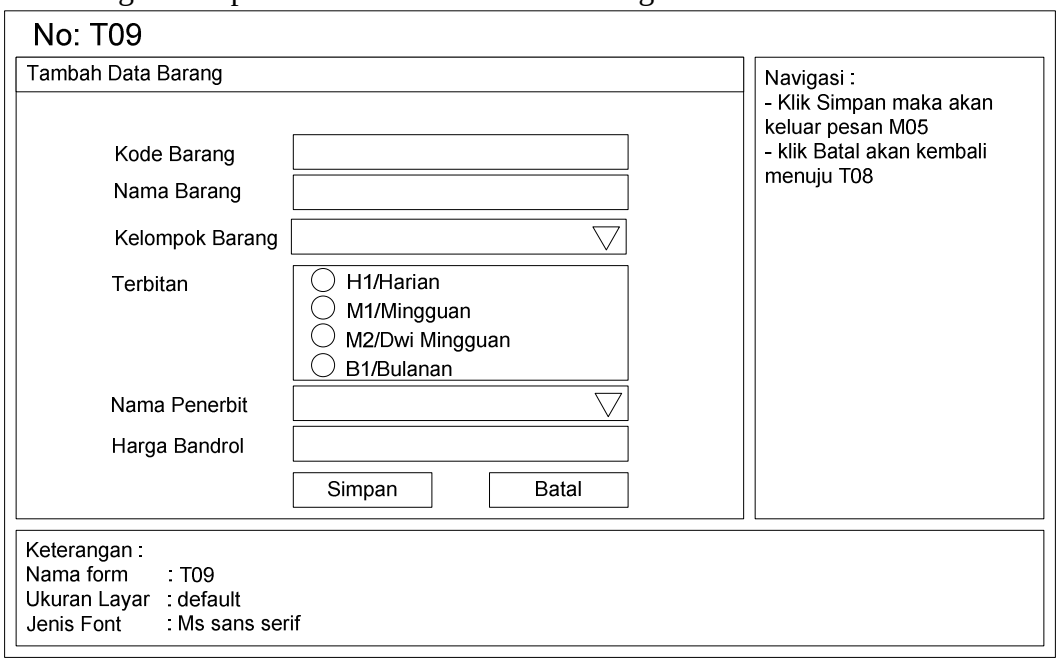
Gambar III.28. Perancangan Tampilan Form Ubah Data User

8. Perancangan Tampilan Form Data Barang



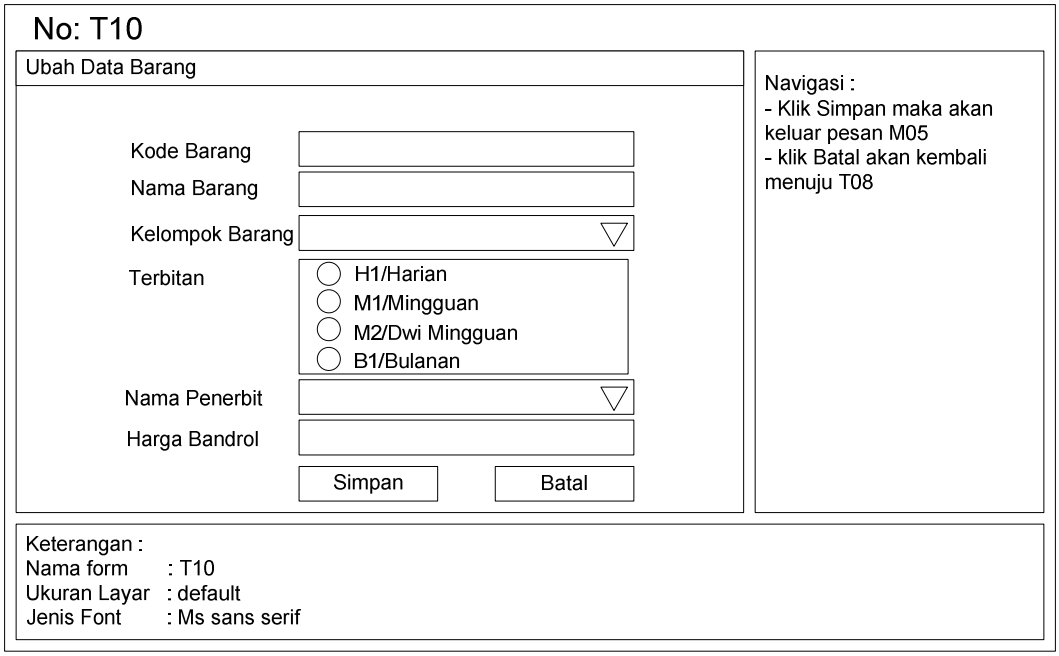
Gambar III.29. Perancangan Tampilan Form Data Barang

9. Perancangan Tampilan Form Tambah Data Barang



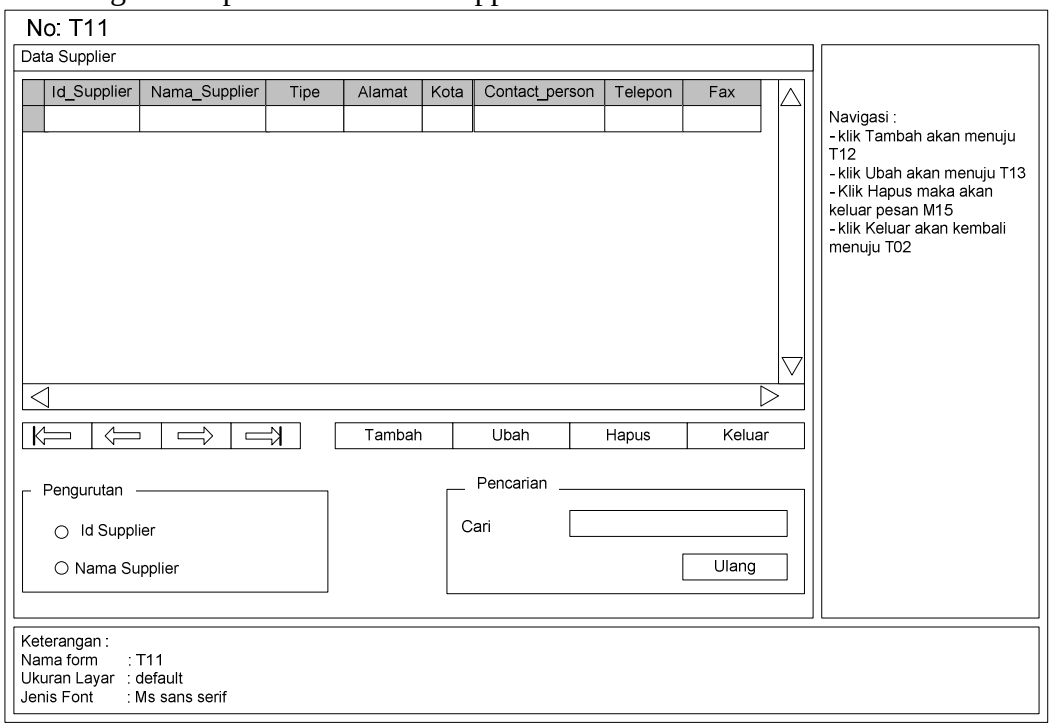
Gambar III.30. Perancangan Tampilan Form Tambah Data Barang

10. Perancangan Tampilan Form Ubah Data Barang



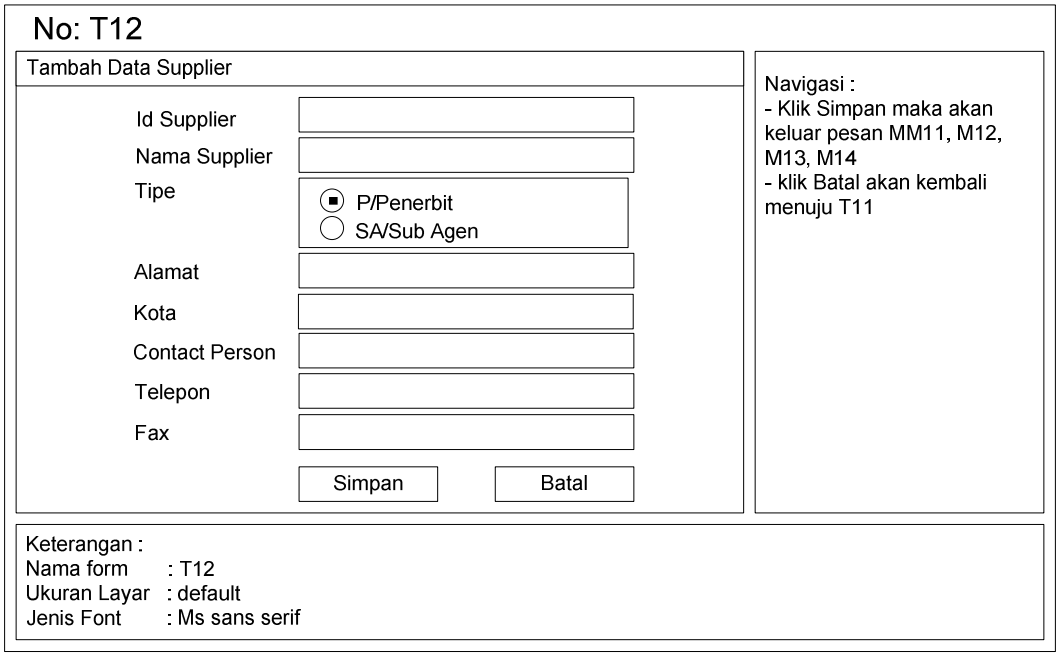
Gambar III.31. Perancangan Tampilan Form Ubah Data Barang

11. Perancangan Tampilan Form Data Supplier



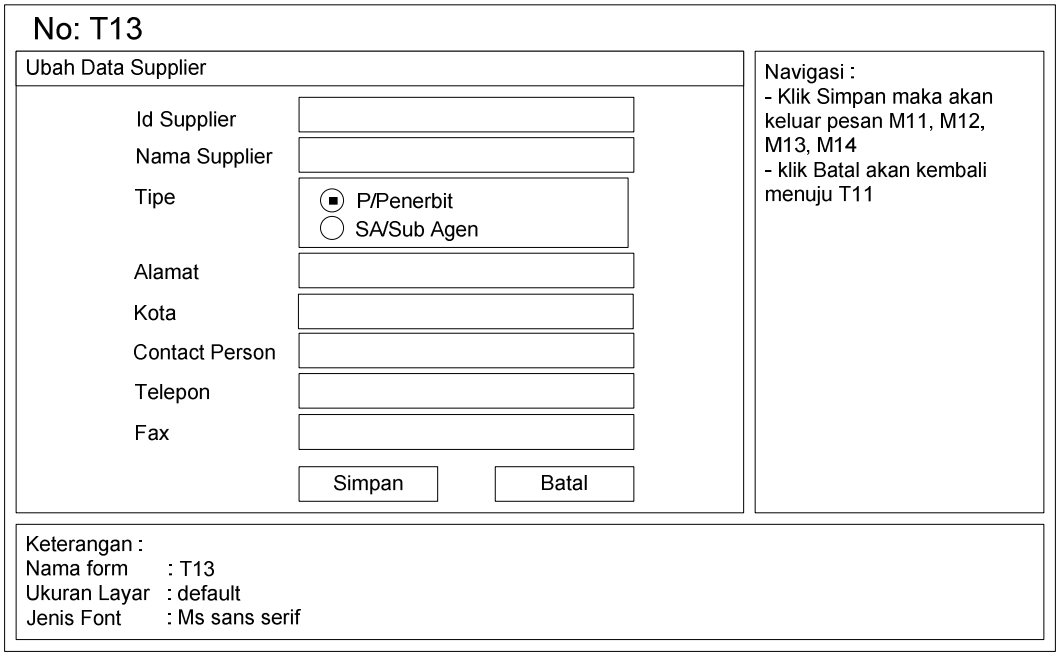
Gambar III.32. Perancangan Tampilan Form Data Supplier

12. Perancangan Tampilan Form Tambah Data Supplier



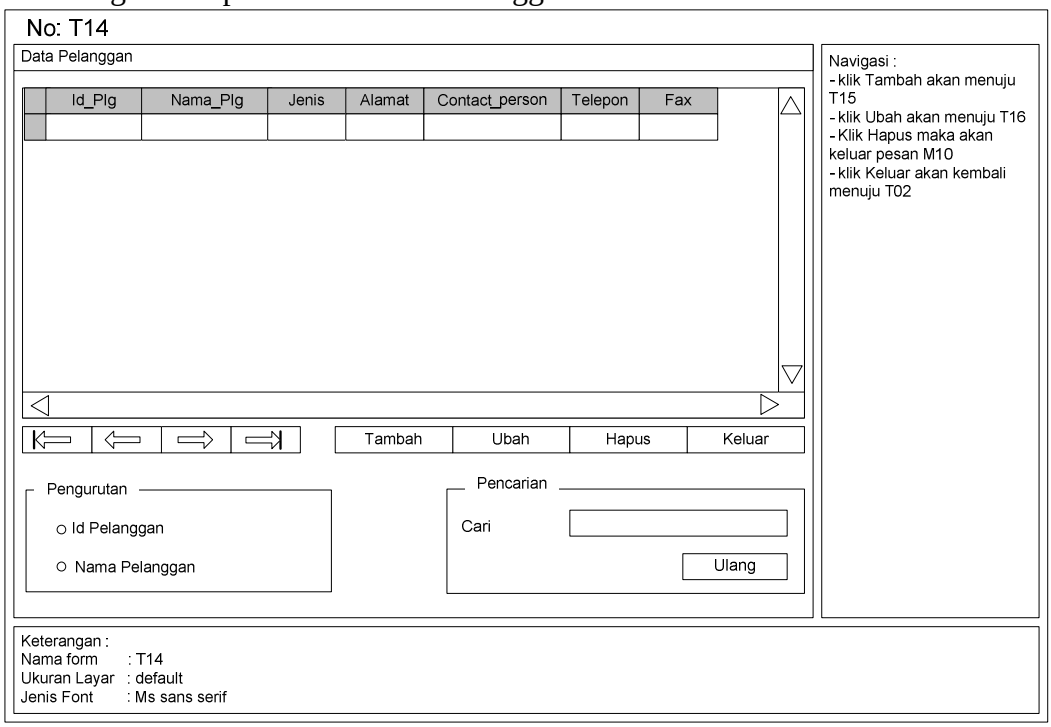
Gambar III.33. Perancangan Tampilan Form Tambah Data Supplier

13. Perancangan Tampilan Form Ubah Data Supplier



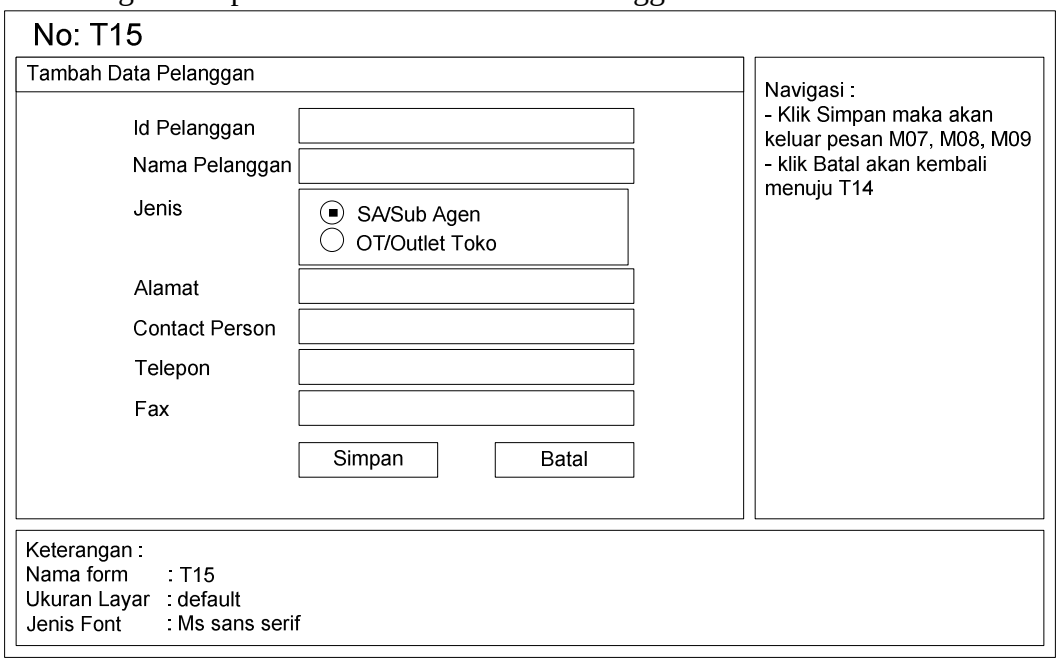
Gambar III.34. Perancangan Tampilan Form Ubah Data Supplier

14. Perancangan Tampilan Form Data Pelanggan



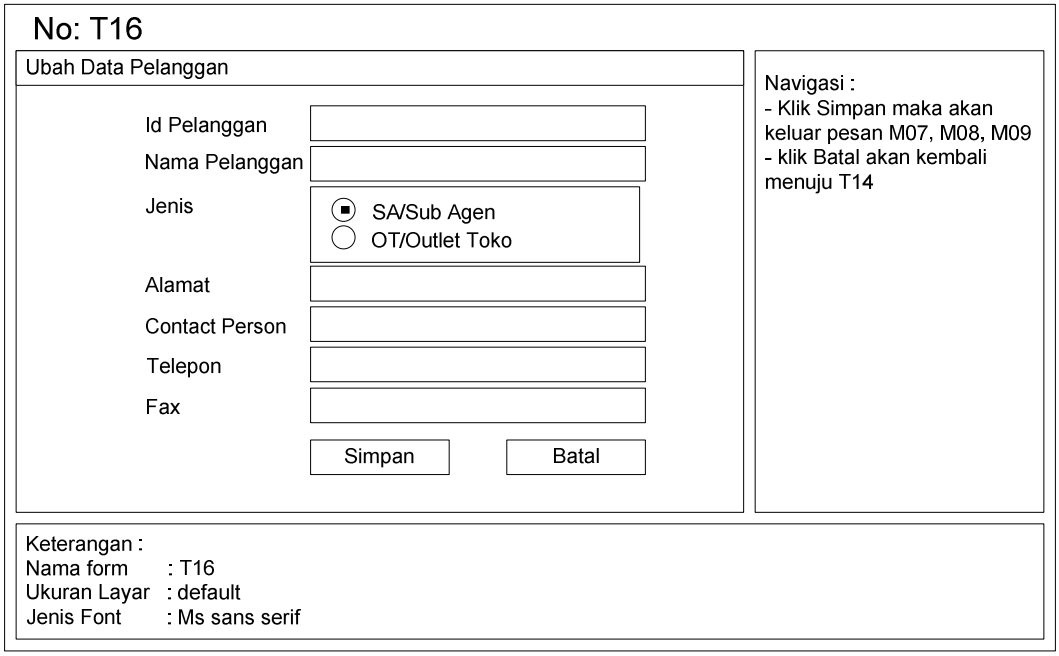
Gambar III.35. Perancangan Tampilan Form Data Pelanggan

15. Perancangan Tampilan Form Tambah Data Pelanggan



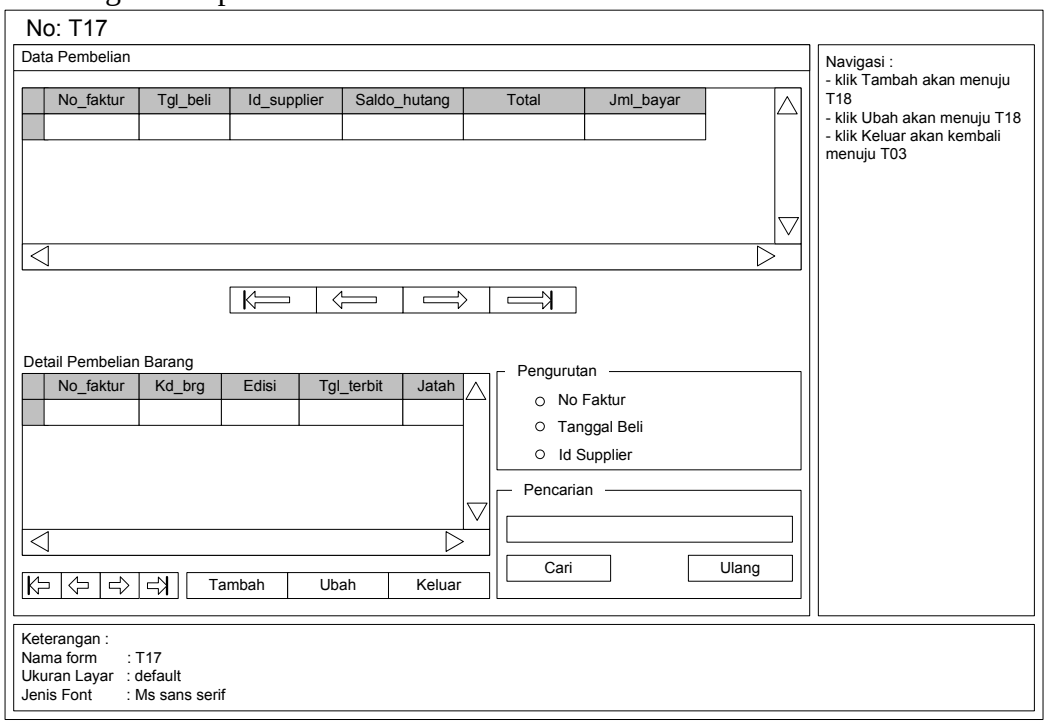
Gambar III.36. Perancangan Tampilan Form Tambah Data Pelanggan

16. Perancangan Tampilan Form Ubah Data Pelanggan



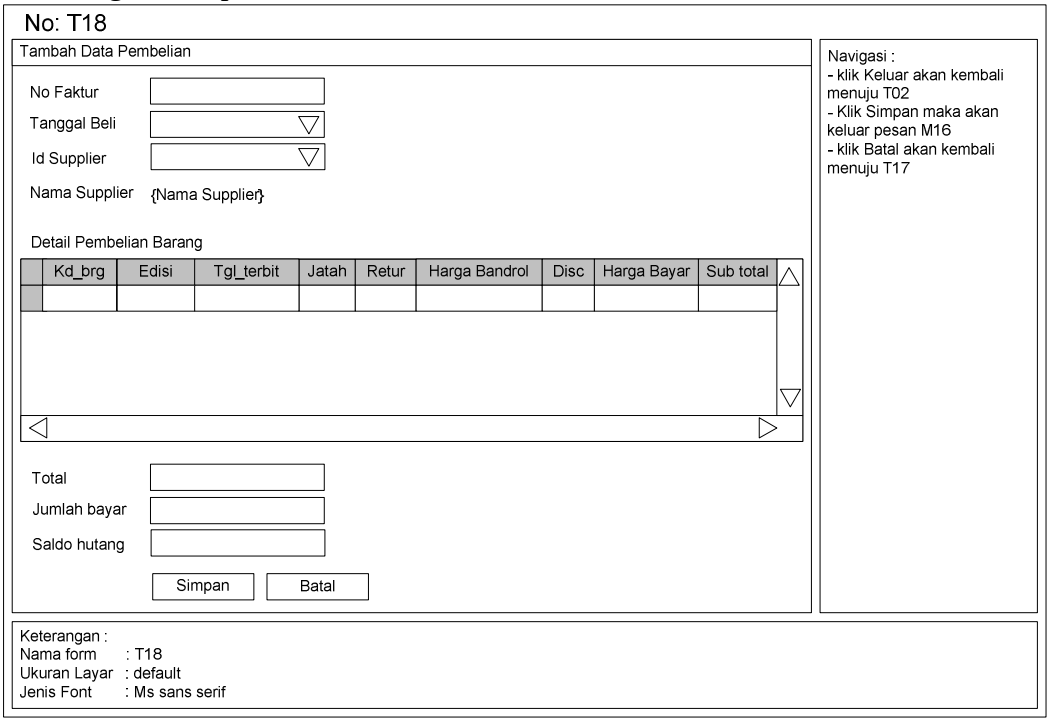
Gambar III.37. Perancangan Tampilan Form Ubah Data Pelanggan

17. Perancangan Tampilan Form Pembelian



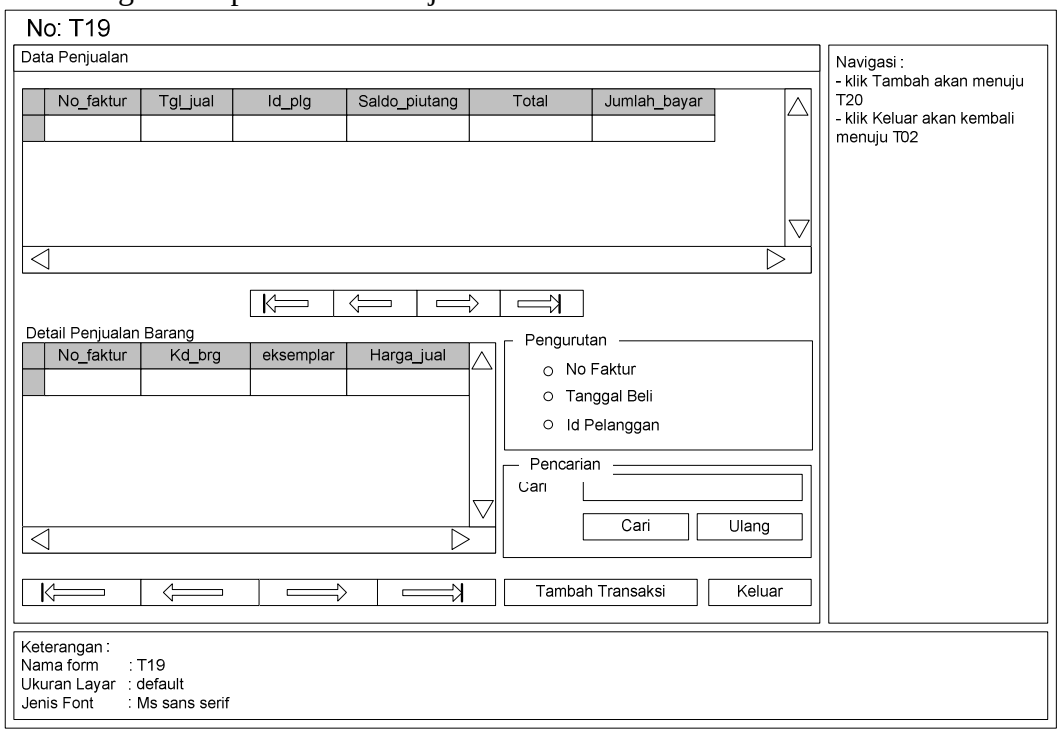
Gambar III.38. Perancangan Tampilan Form Pembelian

18. Perancangan Tampilan Form Tambah Pembelian



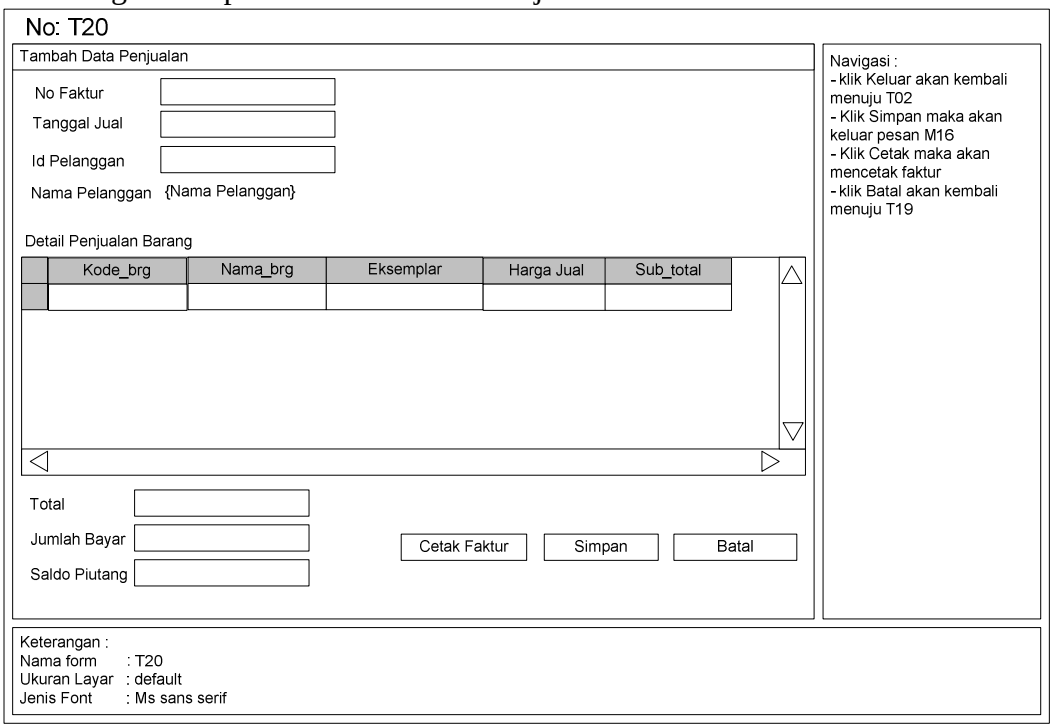
Gambar III.39. Perancangan Tampilan Form Tambah Pembelian

19. Perancangan Tampilan Form Penjualan



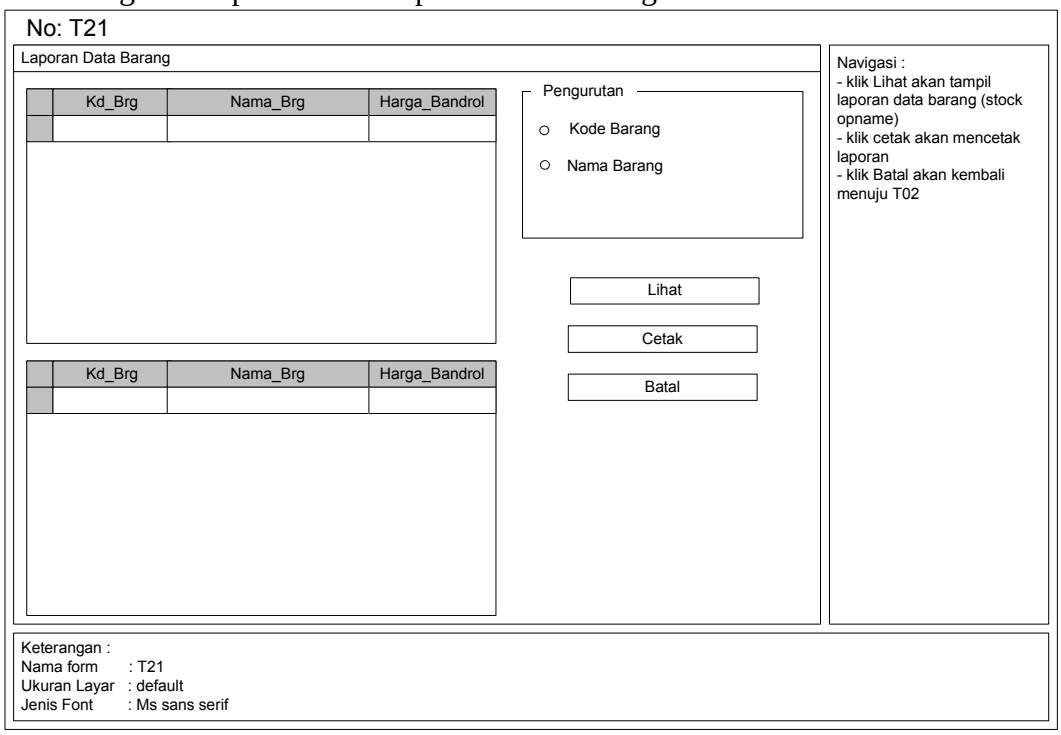
Gambar III.40. Perancangan Tampilan Form Penjualan

20. Perancangan Tampilan Form Tambah Penjualan



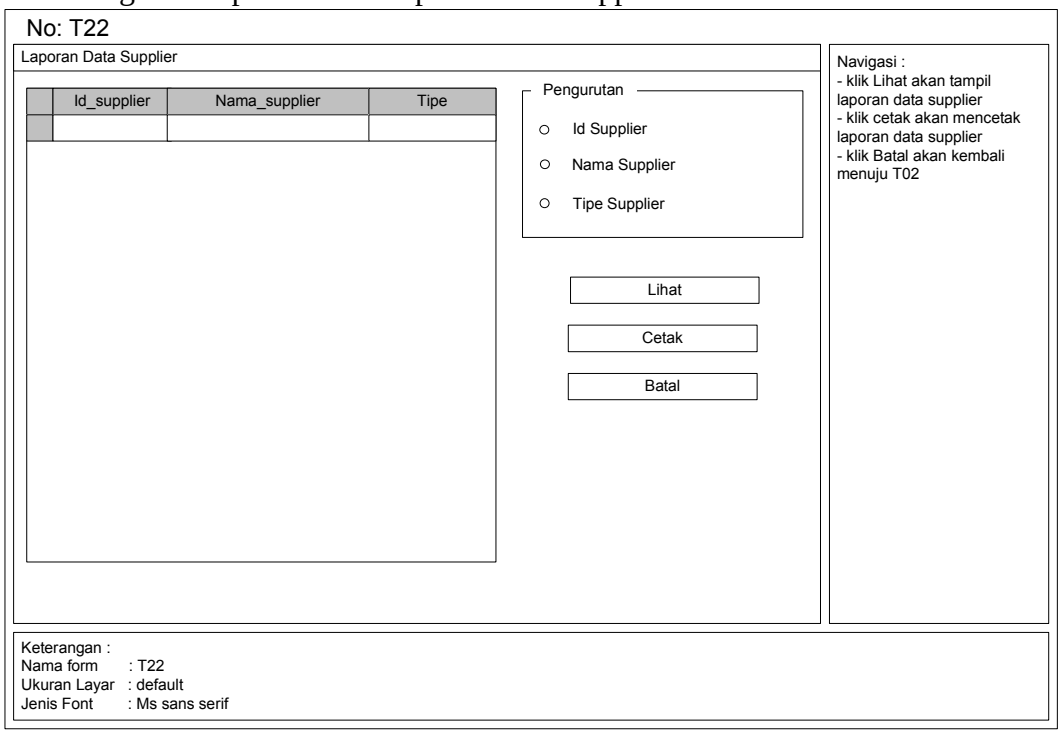
Gambar III.41. Perancangan Tampilan Form Tambah Penjualan

21. Perancangan Tampilan Form Laporan Data Barang



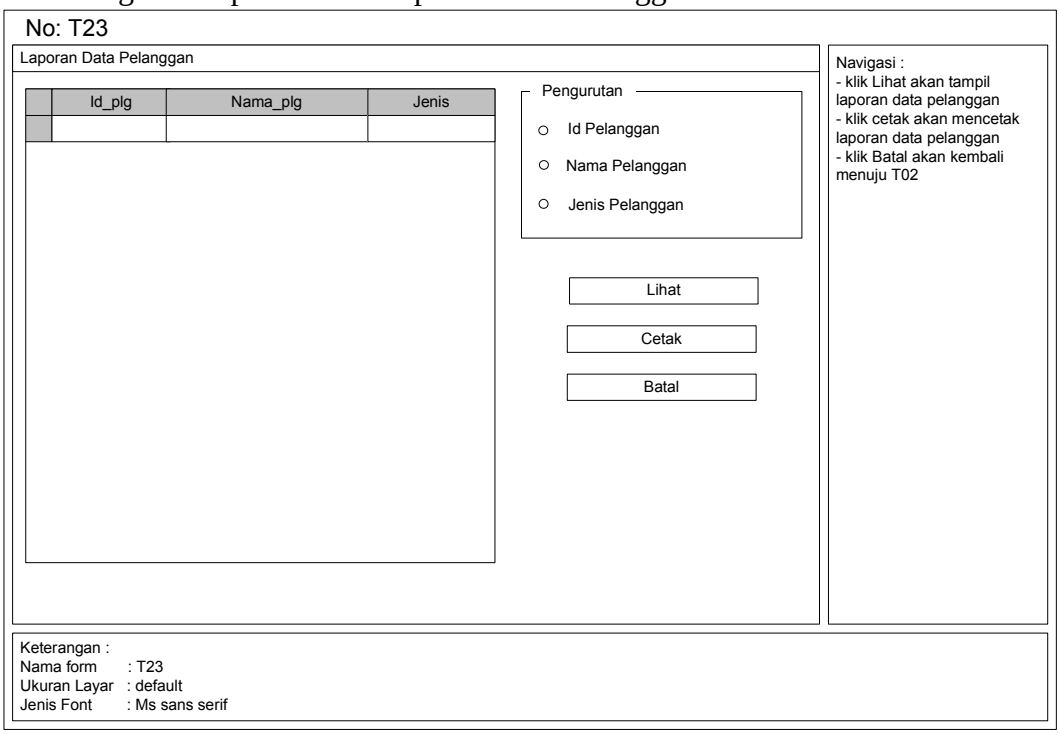
Gambar III.42. Perancangan Tampilan Form Laporan Data Barang

22. Perancangan Tampilan Form Laporan Data Supplier



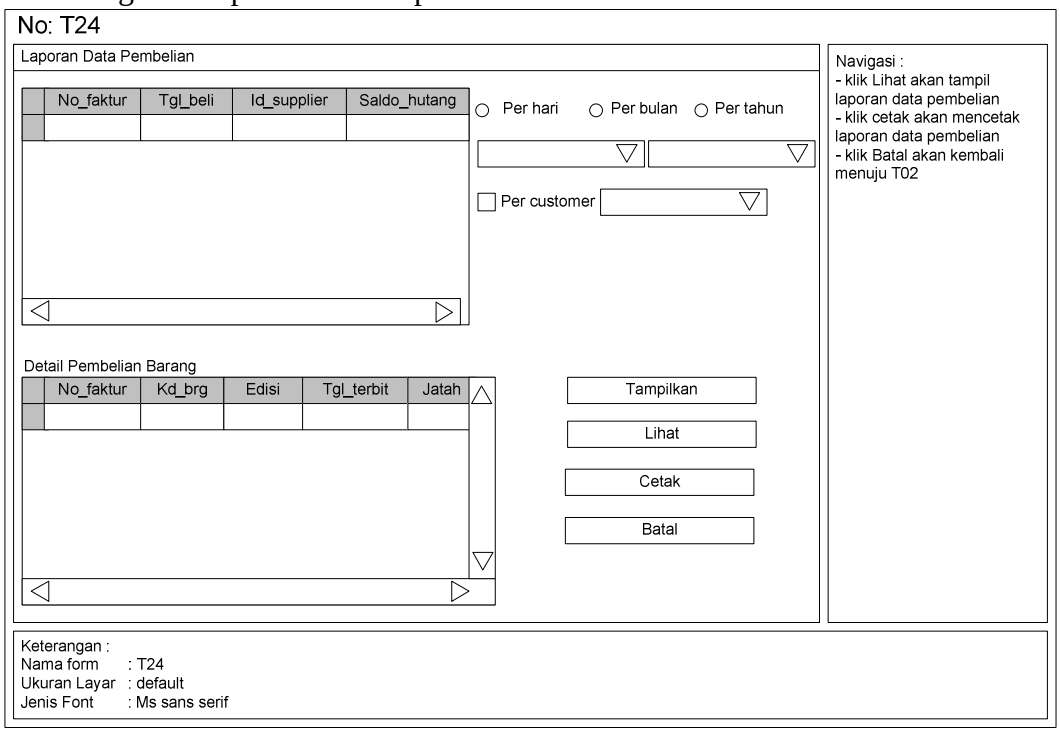
Gambar III.43. Perancangan Tampilan Form Laporan Data Supplier

23. Perancangan Tampilan Form Laporan Data Pelanggan



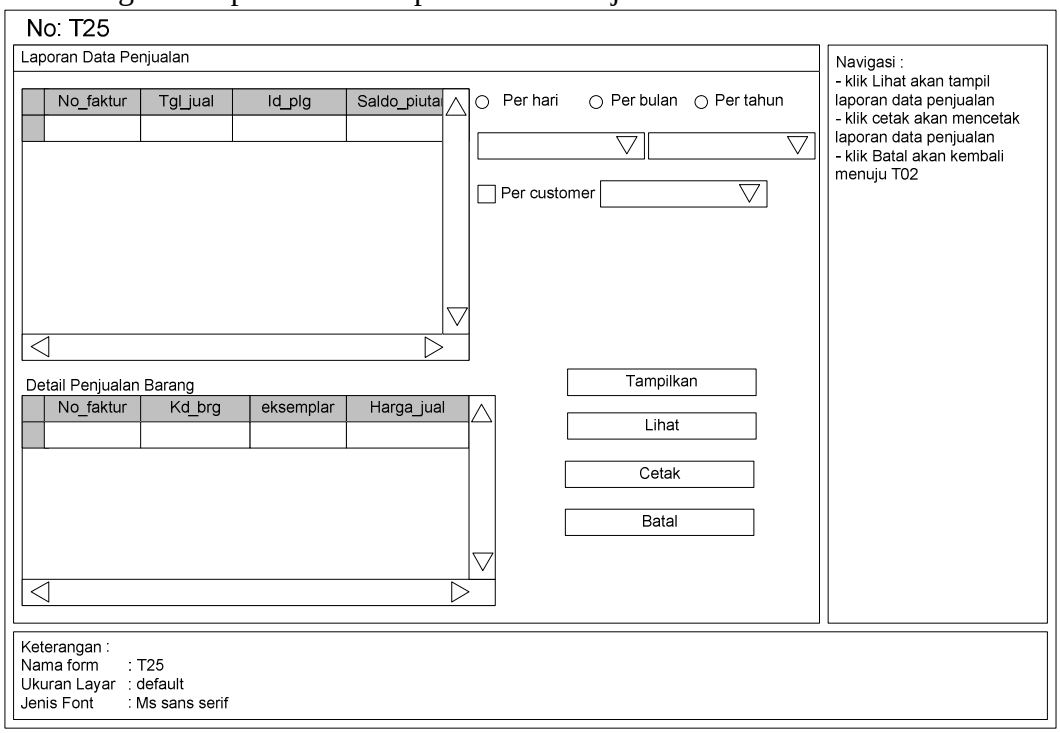
Gambar III.44. Perancangan Tampilan Form Laporan Data Pelanggan

24. Perancangan Tampilan Form Laporan Data Pembelian



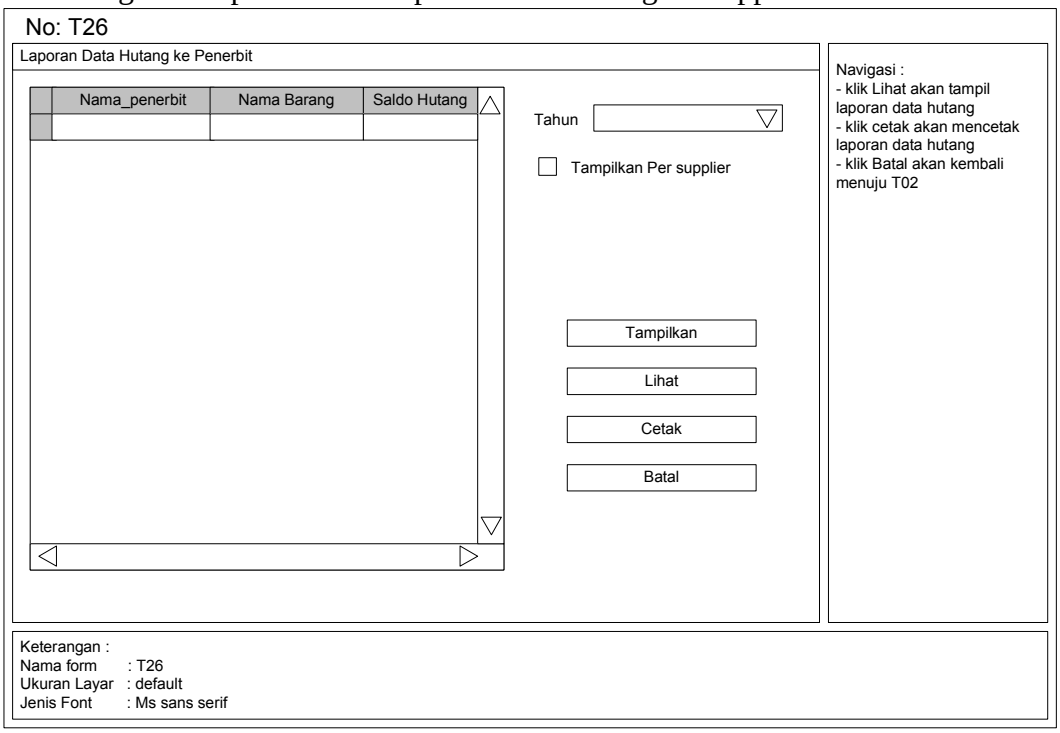
Gambar III.45. Perancangan Tampilan Form Laporan Data Pembelian

25. Perancangan Tampilan Form Laporan Data Penjualan



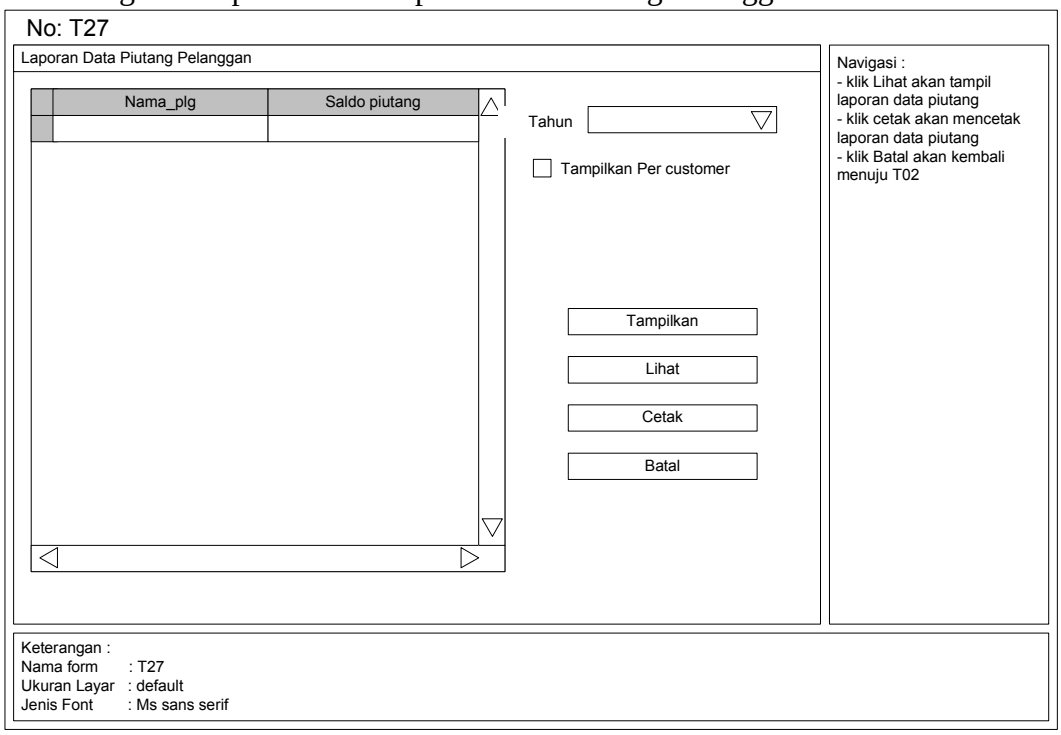
Gambar III.46. Perancangan Tampilan Form Laporan Data Penjualan

26. Perancangan Tampilan Form Laporan Data Hutang ke Supplier



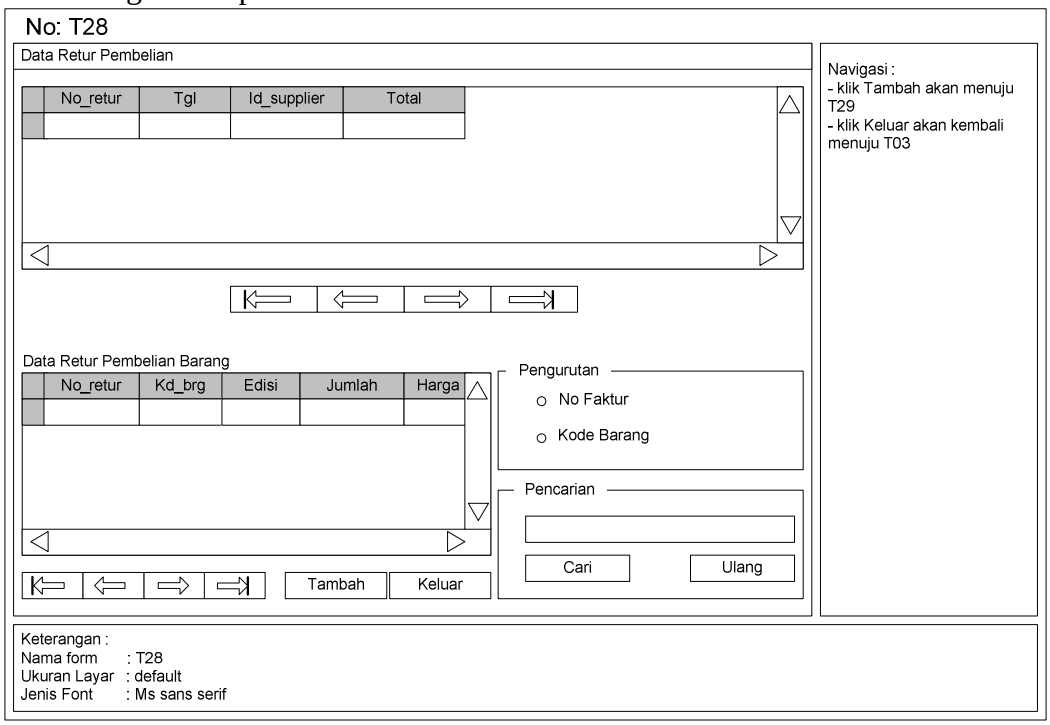
Gambar III.47. Perancangan Tampilan Form Laporan Data Hutang ke Supplier

27. Perancangan Tampilan Form Laporan Data Piutang Pelanggan



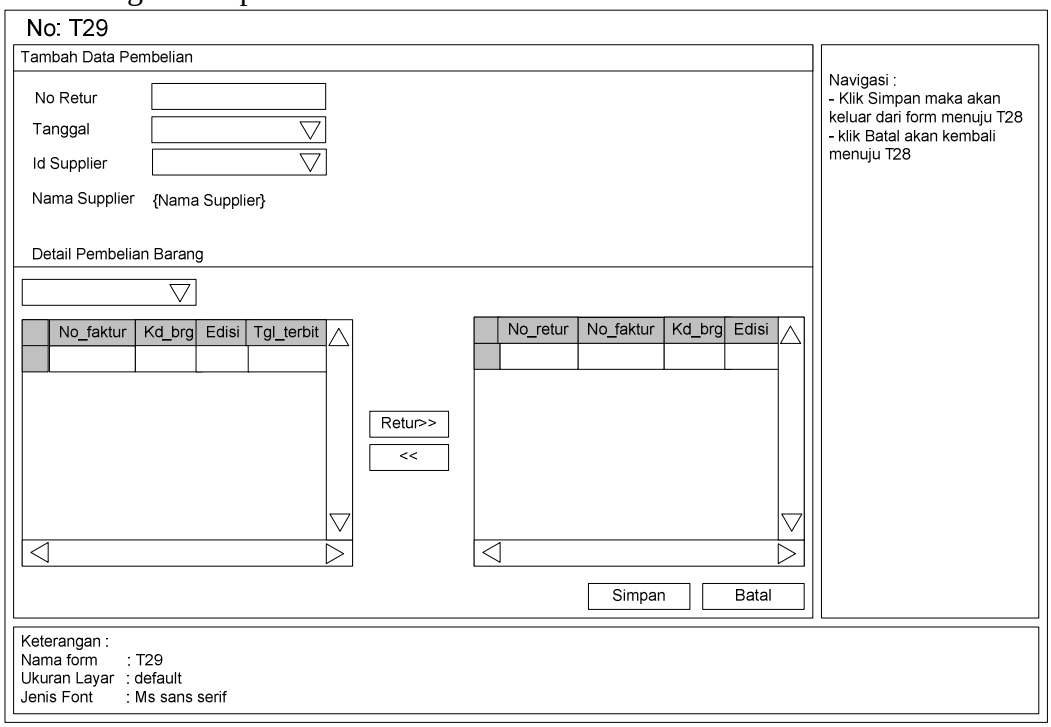
Gambar III.48. Perancangan Tampilan Form Laporan Data Piutang Pelanggan

28. Perancangan Tampilan Form Retur Pembelian



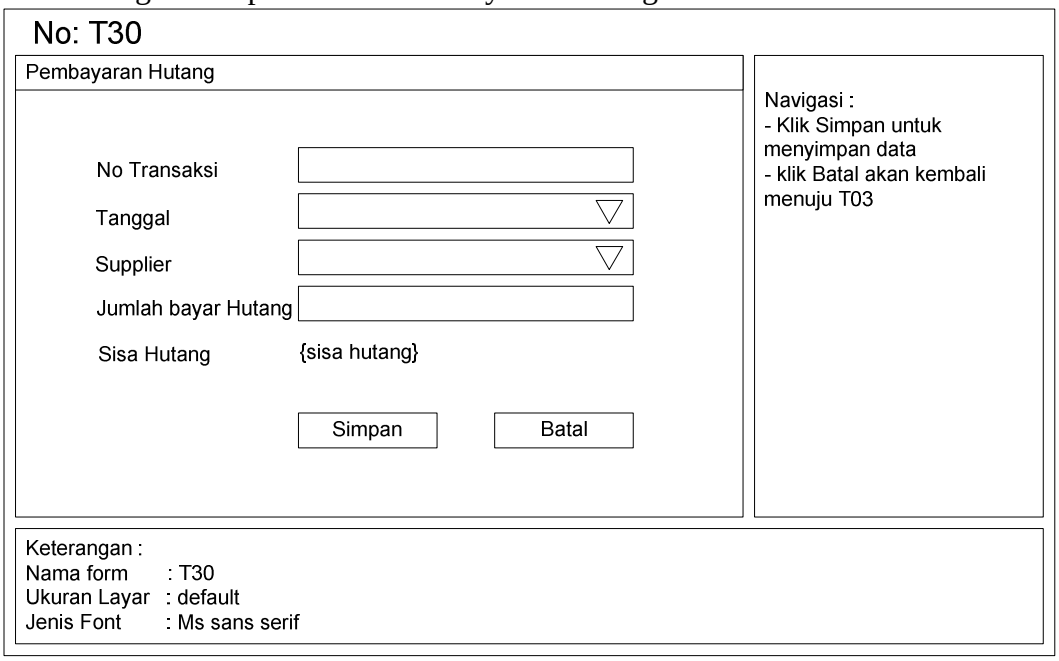
Gambar III.49. Perancangan Tampilan Form Retur Pembelian

29. Perancangan Tampilan Form Tambah Retur Pembelian



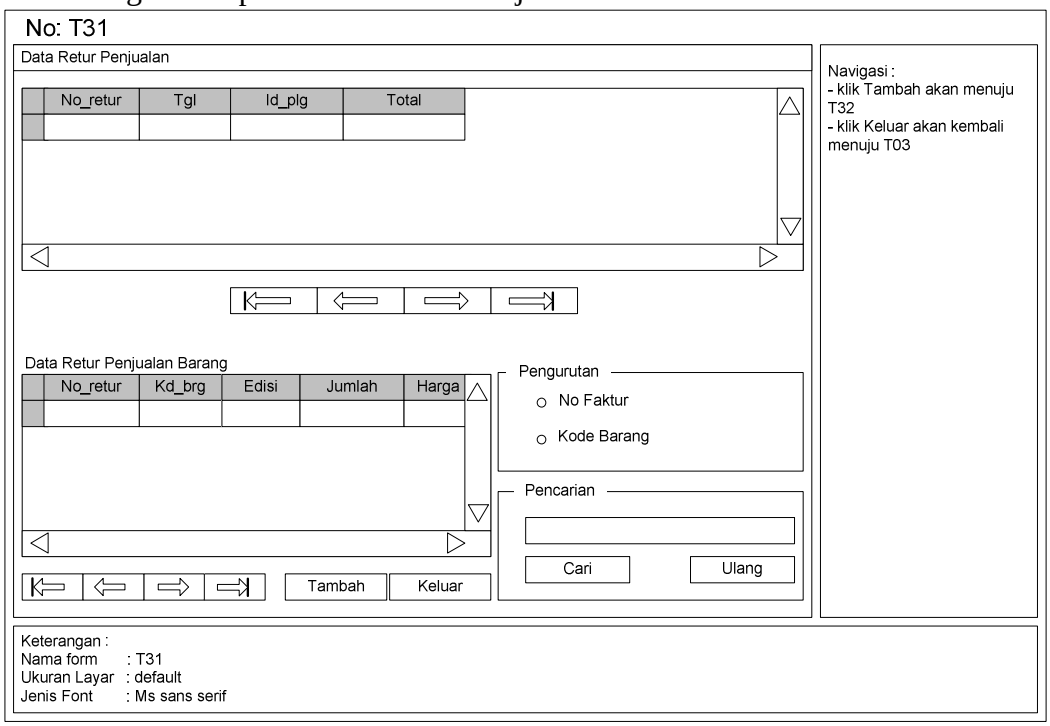
Gambar III.50. Perancangan Tampilan Form Tambah Retur Pembelian

30. Perancangan Tampilan Form Pembayaran Hutang



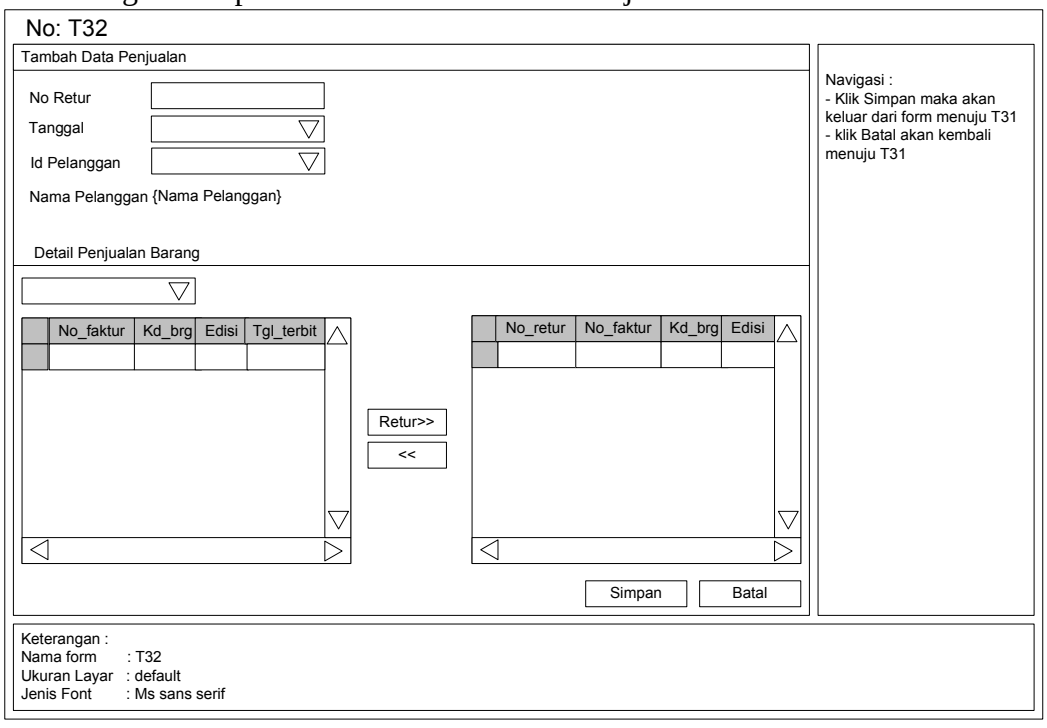
Gambar III.51. Perancangan Tampilan Form Pembayaran Hutang

31. Perancangan Tampilan Form Retur Penjualan



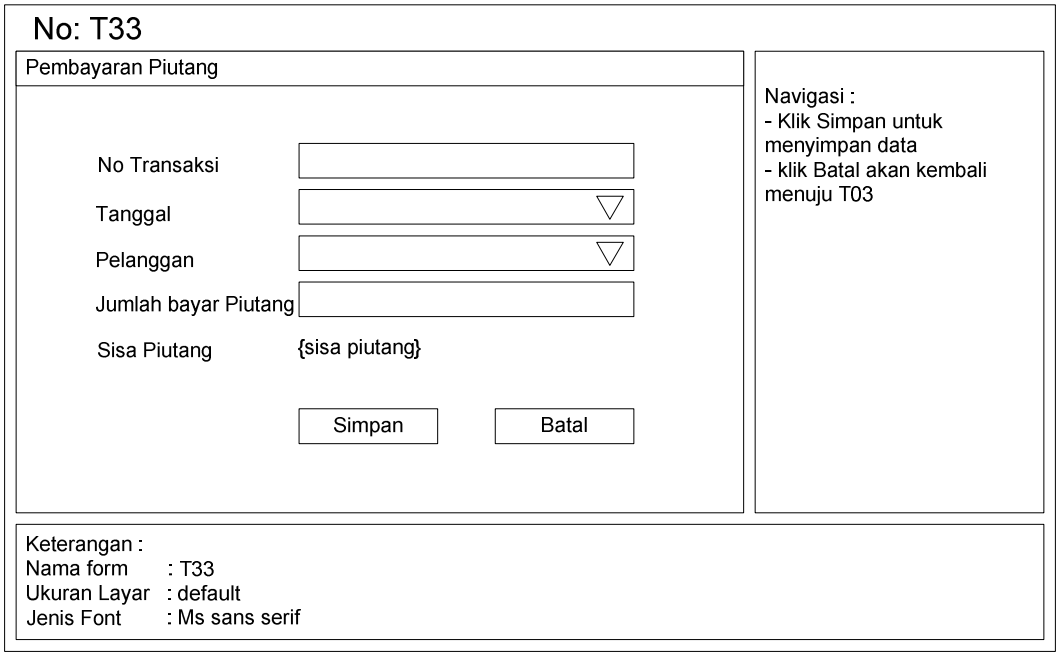
Gambar III.52. Perancangan Tampilan Form Retur Penjualan

32. Perancangan Tampilan Form Tambah Retur Penjualan



Gambar III.53. Perancangan Tampilan Form Tambah Retur Penjualan

33. Perancangan Tampilan Form Pembayaran Piutang



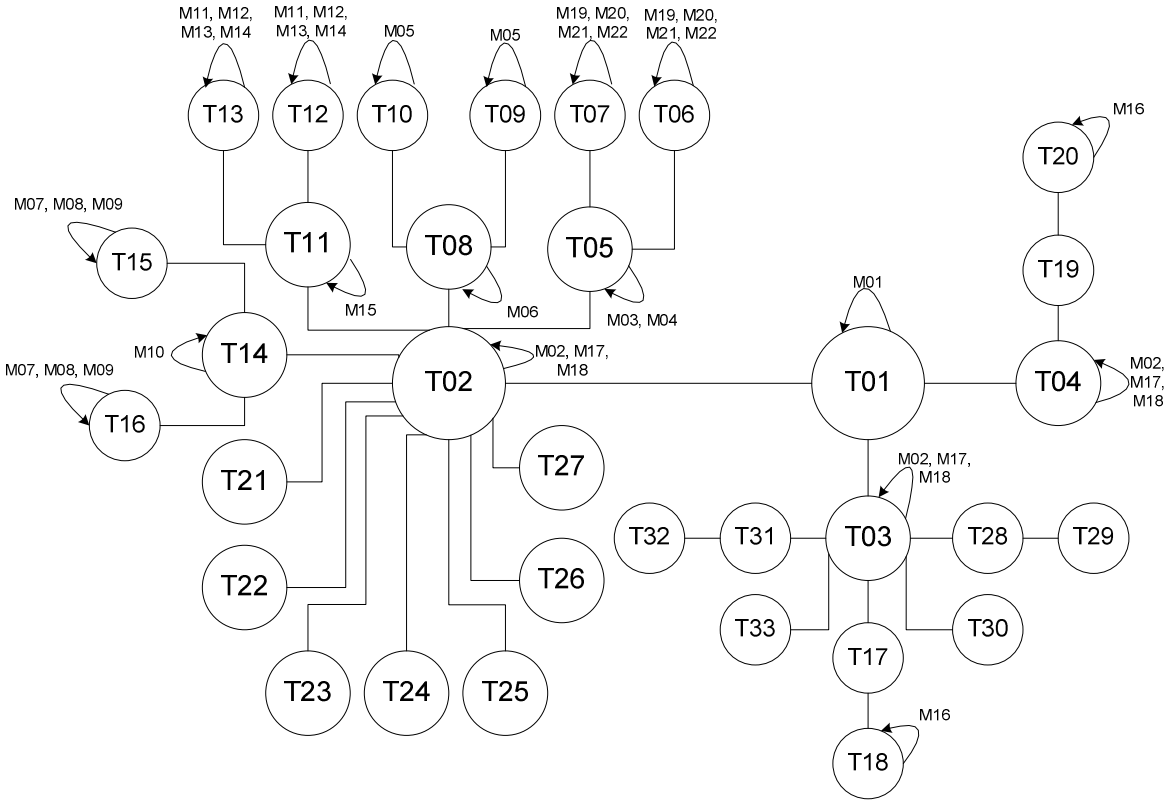
Gambar III.54. Perancangan Tampilan Form Pembayaran Piutang

3.2.4 Perancangan Pesan



Gambar III.55. Perancangan Tampilan Pesan

3.2.5 Jaringan Semantik



Gambar III.56. Jaringan Semantik

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini merupakan tahap penerjemahan kebutuhan pembuatan aplikasi kedalam representasi perangkat lunak sebelum penulisan kode program dimulai sesuai dengan hasil analisis yang telah dilakukan mencakup data-data yang digambarkan dengan tampilan. Sedangkan untuk pengujian meliputi pengujian terhadap perangkat lunak hasil implementasi.

4.1 Implementasi

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis.

4.1.1 Implementasi Perangkat Lunak

Perangkat lunak (*software*) yang dipasang pada sistem komputer yang digunakan untuk membangun Sistem Informasi ini adalah sebagai berikut:

1. Sistem Operasi Windows XP
2. Pemrograman Borland Delphi 7.0
3. Database MySQL Server 5.0
4. mysql-connector-odbc-5.1.5
5. SQLYog Enterprise v7.14

4.1.2 Implementasi Perangkat Keras

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:

Tabel IV.1. Tabel kebutuhan perangkat keras

No.	Nama Bagian	Jumlah	Spesifikasi
1.	Server	1	Monitor
			Procesor Core 2 Duo 2.8 GHz
			Harddisk 200 GB
			RAM 1 GB
			VGA 256 MB
			DVD-ROM
			Mouse dan Keyboard
			Printer
2.	Client	2	Monitor
			Procesor Intel Celeron ® 1.8 GHz
			Harddisk 80 GB
			RAM 446 MB
			VGA 64 MB
			DVD-ROM
			Mouse dan Keyboard
			Printer

4.1.3 Implementasi Basisdata (*Database*)

Dalam pembuatan Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency ini menggunakan MySQL Server 5.0 sebagai DBMS.

Tabel IV.2. Nama Tabel beserta *file* implementasinya

No.	Nama Tabel	Nama *file*
1.	user	user.sql
2.	barang	barang.sql
3.	supplier	supplier.sql
4.	pelanggan	pelanggan.sql
5.	pembelian	pembelian.sql
6.	detail_pembelian	detail_pembelian.sql
7.	penjualan	penjualan.sql
8.	detail_penjualan	detail_penjualan.sql
9.	barang_detail	barang_detail.sql
10.	returbeli	Returbeli.sql
11.	returbelidetail	Returbelidetail.sql
12.	returjual	Returjual.sql
13.	returjualdetail	Returjualdetail.sql
14.	buku_hutang	buku_hutang.sql
15.	buku_piutang	buku_piutang.sql

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

1. Tabel user

   Tabel ini berfungsi untuk menampung data user.

```
/*Table structure for table `user` */
CREATE TABLE `user` (
  `username` char(10) NOT NULL,
  `pass_user` char(32) default NULL,
  `hak_akses` char(10) default NULL,
  `st_aktif` tinyint(4) default '0',
  PRIMARY KEY  (`username`),
  KEY `Idxhak_akses` (`hak_akses`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel barang

   Tabel ini berfungsi untuk menampung data barang

```
/*Table structure for table `barang` */
CREATE TABLE `barang` (
 `kd_brg` varchar(10) NOT NULL default '',
 `nama_brg` varchar(50) default NULL,
 `terbitan` char(2) default NULL,
 `id_supplier` char(8) default NULL,
 `harga_bandrol` int(11) default NULL,
 `kel_barang` enum('Tabloid','Majalah','Koran')
 default NULL,
 PRIMARY KEY  (`kd_brg`),
 KEY `Idxnama_brg` (`nama_brg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

3. Tabel supplier

   Tabel ini berfungsi untuk menampung data supplier

```
/*Table structure for table `supplier` */
CREATE TABLE `supplier` (
```

```
  `id_supplier` char(8) NOT NULL,
  `nama_supplier` char(30) default NULL,
  `tipe` char(2) default NULL,
  `alamat` char(50) default NULL,
  `kota` char(30) default NULL,
  `contact_person` char(30) default NULL,
  `telepon` char(15) default NULL,
  `fax` char(15) default NULL,
  PRIMARY KEY  (`id_supplier`),
  KEY `Idxnama_supplier` (`nama_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. Tabel pelanggan

   Tabel ini berfungsi untuk menampung data pelanggan

```
/*Table structure for table `pelanggan` */
CREATE TABLE `pelanggan` (
  `id_plg` char(8) NOT NULL,
  `nama_plg` char(30) default NULL,
  `jenis` char(2) default NULL,
  `alamat` char(50) default NULL,
  `contact_person` char(30) default NULL,
  `telepon` char(15) default NULL,
  `fax` char(15) default NULL,
  PRIMARY KEY  (`id_plg`),
  KEY `Idxnama_plg` (`nama_plg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Tabel pembelian

   Tabel ini berfungsi untuk menampung data pembelian

```
/*Table structure for table `pembelian` */
CREATE TABLE `pembelian` (
  `no_faktur` char(20) NOT NULL,
  `tgl_beli` date default NULL,
  `id_supplier` char(8) default NULL,
  `saldo_hutang` int(11) default NULL,
```

```
  `total` int(11) default NULL,
  `jumlah_bayar` int(11) default NULL,
  `no_faktur_suplier` varchar(15) default NULL,
  `total_retur` int(11) default '0',
  PRIMARY KEY  (`no_faktur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. Tabel detail_pembelian

Tabel ini berfungsi untuk menampung data detail_pembelian

```
/*Table structure for table `detail_pembelian` */
CREATE TABLE `detail_pembelian` (
  `no_faktur` char(20) NOT NULL,
  `kd_brg` char(20) NOT NULL,
  `edisi` char(10) default NULL,
  `tgl_terbit` date default NULL,
  `jatah` int(11) default '0',
  `retur` int(11) default '0',
  `harga_bandrol` int(11) default '0',
  `disc` float default '0',
  `harga_bayar` int(11) default '0',
  `sub_total` int(11) default '0',
  PRIMARY KEY  (`no_faktur`,`kd_brg`),
  KEY `Idxkd_brg` (`kd_brg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. Tabel penjualan

Tabel ini berfungsi untuk menampung data penjualan

```
/*Table structure for table `penjualan` */
CREATE TABLE `penjualan` (
  `no_faktur` char(20) NOT NULL,
  `tgl_jual` date default NULL,
  `id_plg` char(8) default NULL,
  `saldo_piutang` int(11) default '0',
  `total` int(11) default '0',
  `jumlah_bayar` int(11) default '0',
  `total_retur` int(11) default '0',
```

```
  PRIMARY KEY  (`no_faktur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

8. Tabel detail_penjualan

Tabel ini berfungsi untuk menampung data detail_penjualan

```
/*Table structure for table `detail_penjualan` */
CREATE TABLE `detail_penjualan` (
  `no_faktur` char(20) NOT NULL,
  `kd_brg` char(20) NOT NULL,
  `edisi` char(10) default NULL,
  `tgl_terbit` date default NULL,
  `eksemplar` int(11) default NULL,
  `harga_jual` int(11) default NULL,
  `sub_total` int(11) default NULL,
  `retur` int(11) default NULL,
  PRIMARY KEY  (`no_faktur`,`kd_brg`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Tabel barang_detail

Tabel ini berfungsi untuk menampung data barang_detail

```
/*Table structure for table `barang_detail` */
CREATE TABLE `barang_detail` (
  `kd_brg` varchar(10) default NULL,
  `edisi` varchar(5) default NULL,
  `jumlah` int(11) default NULL,
  `harga_bandrol` int(11) default NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

10. Tabel returbeli

Tabel ini berfungsi untuk menampung data returbeli

```
/*Table structure for table `returbeli` */
CREATE TABLE `returbeli` (
  `no_retur` char(20) NOT NULL default '',
  `tgl` date default NULL,
```

```
  `id_supplier` char(8) default NULL,
  `total` int(11) default NULL,
  PRIMARY KEY  (`no_retur`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

11.Tabel returbelidetail

Tabel ini berfungsi untuk menampung data returbelidetail

```
/*Table structure for table `returbelidetail` */
CREATE TABLE `returbelidetail` (
  `no_retur` char(20) NOT NULL default '',
  `no_faktur` char(20) default NULL,
  `kd_brg` varchar(10) default NULL,
  `edisi` char(5) default NULL,
  `jumlah` int(11) default NULL,
  `harga` int(11) default NULL,
  `total` int(11) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

12.Tabel returjual

Tabel ini berfungsi untuk menampung data returjual

```
/*Table structure for table `returjual` */
CREATE TABLE `returjual` (
  `no_retur` char(20) NOT NULL default '',
  `tgl` date default NULL,
  `id_plg` char(8) default NULL,
  `total` int(11) default NULL,
  PRIMARY KEY  (`no_retur`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

13.Tabel returjualdetail

Tabel ini berfungsi untuk menampung data returjualdetail

```
/*Table structure for table `returjualdetail` */
CREATE TABLE `returjualdetail` (
```

```
  `no_retur` char(20) NOT NULL default '',
  `no_faktur` char(20) default NULL,
  `kd_brg` varchar(10) default NULL,
  `edisi` char(5) default NULL,
  `jumlah` int(11) default NULL,
  `harga` int(11) default NULL,
  `total` int(11) default NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

14. Tabel buku_hutang

Tabel ini berfungsi untuk menampung data buku_hutang

```
/*Table structure for table `buku_hutang` */
CREATE TABLE `buku_hutang` (
  `no_transaksi` varchar(15) NOT NULL default '',
  `tgl` date default NULL,
  `id_supplier` char(8) default NULL,
  `debit` int(11) default '0',
  `kredit` int(11) default '0',
  `saldo` int(11) default '0',
  `keterangan` enum('Pembayaran Hutang','Pembelian',
   'Retur Pembelian') default 'Pembayaran Hutang',
  PRIMARY KEY  (`no_transaksi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

15. Tabel buku_piutang

Tabel ini berfungsi untuk menampung data buku_piutang

```
/*Table structure for table `buku_piutang` */
CREATE TABLE `buku_piutang` (
  `no_transaksi` varchar(15) NOT NULL default '',
  `tgl` date default NULL,
  `id_plg` char(8) default NULL,
  `debit` int(11) default '0',
  `kredit` int(11) default '0',
  `saldo` int(11) default '0',
  `keterangan` enum('Pembayaran Piutang','Penjualan',
```

```
    'Retur Penjualan') default 'Pembayaran Piutang',
  PRIMARY KEY  (`no_transaksi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

4.1.4 Implementasi Antarmuka

Setelah melakukan inplementasi data dilakukanlah implementasi antar muka yang berguna sebagai tampilan untuk pengguna sistem. Implementasi antar muka dilakukan pada setiap hasil perancangan antar muka yang telah dibuat kedalam bentuk aplikasi pada compiler Delphi 7.

Untuk mengetahui lebih jelas tentang proses yang ada didalam sistem beserta hasil implementasinya dapat dilihat dalam tabel IV.3 dibawah ini.

Tabel IV.3. Implementasi antar muka
Sistem Informasi Penjualan Koran dan Majalah

Menu	Proses ke-	Nama File
Menu Utama: Login	1.0	Ulogin.dcu
		Ulogin.ddp
		Ulogin.dfm
		Ulogin.pas
Menu Admin: Pembukuan Pembelian Penjualan	 2.0 3.0 4.0	Uadmin.dcu
		Uadmin.ddp
		Uadmin.dfm
		Uadmin.pas
Pembuatan Laporan: Lihat Laporan Cetak Laporan	 5.1 5.2	Ulapbrg.dcu
		Ulapbrg.ddp
		Ulapbrg.dfm
		Ulapbrg.pas
		Ulapsupplier.dcu
		Ulapsupplier.ddp
		Ulapsupplier.dfm
		Ulapsupplier.pas
		Ulappelanggan.dcu
		Ulappelanggan.ddp
		Ulappelanggan.dfm
		Ulappelanggan.pas

		Ulappembelian.dcu
		Ulappembelian.ddp
		Ulappembelian.dfm
		Ulappembelian.pas
		Ulappenjualan.dcu
		Ulappenjualan.ddp
		Ulappenjualan.dfm
		Ulappenjualan.pas
		Ulaphutang.dcu
		Ulaphutang.ddp
		Ulaphutang.dfm
		Ulaphutang.pas
		Ulappiutang.dcu
		Ulappiutang.ddp
		Ulappiutang.dfm
		Ulappiutang.pas
Pengolahan Data User: Cari Data	2.1.1	Uuser1.dcu
		Uuser1.ddp
		Uuser1.dfm
		Uuser1.pas
Pengolahan Data User: Tambah Data	2.1.2	Uisiuser.dcu
		Uisiuser.ddp
		Uisiuser.dfm
		Uisiuser.pas
Pengolahan Data User: Hapus Data	2.1.3	Uuser1.dcu
		Uuser1.ddp
		Uuser1.dfm
		Uuser1.pas
Pengolahan Data User: Ubah Data	2.1.4	UEditUser.dcu
		UEditUser.ddp
		UEditUser.dfm
		UEditUser.pas
Pengolahan Data Barang: Cari Data	2.2.1	Ubarang.dcu
		Ubarang.ddp
		Ubarang.dfm
		Ubarang.pas
Pengolahan Data Barang: Tambah Data	2.2.2	Uisibarang.dcu
		Uisibarang.ddp
		Uisibarang.dfm
		Uisibarang.pas
Pengolahan Data Barang:	2.2.3	Ubarang.dcu
		Ubarang.ddp

Hapus Data		Ubarang.dfm
		Ubarang.pas
Pengolahan Data Barang: Ubah Data	2.2.4	Uisibarang.dcu
		Uisibarang.ddp
		Uisibarang.dfm
		Uisibarang.pas
Pengolahan Data Supplier: Cari Data	2.3.1	Usupplier.dcu
		Usupplier.ddp
		Usupplier.dfm
		Usupplier.pas
Pengolahan Data Supplier: Tambah Data	2.3.2	Uisisupplier.dcu
		Uisisupplier.ddp
		Uisisupplier.dfm
		Uisisupplier.pas
Pengolahan Data Supplier: Hapus Data	2.3.4	Usupplier.dcu
		Usupplier.ddp
		Usupplier.dfm
		Usupplier.pas
Pengolahan Data Supplier: Ubah Data	2.3.5	Uisisupplier.dcu
		Uisisupplier.ddp
		Uisisupplier.dfm
		Uisisupplier.pas
Pengolahan Data Pelanggan: Cari Data	2.4.1	Upelanggan.dcu
		Upelanggan.ddp
		Upelanggan.dfm
		Upelanggan.pas
Pengolahan Data Pelanggan: Tambah Data	2.4.2	Uisipelanggan.dcu
		Uisipelanggan.ddp
		Uisipelanggan.dfm
		Uisipelanggan.pas
Pengolahan Data Pelanggan: Hapus Data	2.4.3	Upelanggan.dcu
		Upelanggan.ddp
		Upelanggan.dfm
		Upelanggan.pas
Pengolahan Data Pelanggan: Ubah Data	2.4.4	Uisipelanggan.dcu
		Uisipelanggan.ddp
		Uisipelanggan.dfm
		Uisipelanggan.pas
Pengolahan Data Pembelian: Cari Data	3.3.1	Upembelianutama.dcu
		Upembelianutama.ddp
		Upembelianutama.dfm
		Upembelianutama.pas
Pengolahan Data Pembelian:	3.3.2	Upembelian2.dcu
		Upembelian2.ddp

Tambah Data		Upembelian2.dfm
		Upembelian2.pas
Pengolahan Data Pembelian: Ubah Data	3.3.3	Upembelian2.dcu
		Upembelian2.ddp
		Upembelian2.dfm
		Upembelian2.pas
Pengolahan Data Penjualan: Cari Data	4.3.1	Upenjualan2.dcu
		Upenjualan2.ddp
		Upenjualan2.dfm
		Upenjualan2.pas
Pengolahan Data Penjualan: Tambah Data	4.3.2	Upenjualan.dcu
		Upenjualan.ddp
		Upenjualan.dfm
		Upenjualan.pas

4.1 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.

4.2.1. Metode Pengujian

Pengujian program ini menggunakan metoda *Black Box*. Pengujian *Black Box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *Black Box* dilakukan dengan cara memberikan sejumlah input pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan output yang diinginkan dan sesuai denga fungsi dari program tersebut. Apabila dari input yang diberikan proses menghasilkan output yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi

yang bersangkutan telah benar, tetapi jika output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka akan dilakukan pengujian kembali. Pengujian dan perbaikan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

4.2.2. Rencana Pengujian

Pengujian perangkat lunak Sistem Informasi Penjualan Koran dan Majalah berikut menggunakandata uji berdasarkan data diri masing-masing user. Rancangan selengkapnya dapat dilihat pada tabel IV.4 berikut:

Tabel IV.4. Rencana Pengujian

Kelas uji	Butir Uji	Jenis Pengujian
Login	Verifikasi Username dan Password	*Black Box*
Data User	Cari data user	*Black Box*
	Tambah data user	*Black Box*
	Hapus data user	*Black Box*
	Ubah data user	*Black Box*
Data Barang	Cari data barang	*Black Box*
	Tambah data barang	*Black Box*
	Hapus data barang	*Black Box*
	Ubah data barang	*Black Box*
Data Supplier	Cari data supplier	*Black Box*
	Tambah data supplier	*Black Box*
	Hapus data supplier	*Black Box*
	Ubah data supplier	*Black Box*

Data Pelanggan	Cari data pelanggan	*Black Box*
	Tambah data pelanggan	*Black Box*
	Hapus data pelanggan	*Black Box*
	Ubah data pelanggan	*Black Box*
Data Pembelian	Cari data pembelian	*Black Box*
	Tambah data pembelian	*Black Box*
	Ubah data pembelian	*Black Box*
	Tambah retur pembelian	*Black Box*
	Pembayaran Hutang	*Black Box*
Data Penjualan	Cari data penjualan	*Black Box*
	Tambah data penjualan	*Black Box*
	Tambah retur penjualan	*Black Box*
	Pembayaran piutang	*Black Box*
Laporan	Lihat dan cetak laporan data barang	*Black Box*
	Lihat dan cetak laporan data supplier	*Black Box*
	Lihat dan cetak laporan data pelanggan	*Black Box*
	Lihat dan cetak laporan data pembelian	*Black Box*
	Lihat dan cetak laporan data penjualan	*Black Box*
	Lihat dan cetak laporan data hutang	*Black Box*
	Lihat dan cetak laporan data piutang	*Black Box*

4.2.2.1. Pengujian Login

Tabel IV.5. Verifikasi Username dan Password

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username: 1	Tercantum pada edit box username	Dapat mengisi form login username sesuai yang diharapkan	Diterima
Pasword: 1	Tercantum pada edit box password	Dapat mengisi form login password sesuai yang diharapkan	Diterima
Klik tombol login	Data username dan data password dicari di tabel user	Tombol login dapat berfungsi sesuai yang	Diterima

		diharapkan	
Klik tombol batal	Menutup form login	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data username	Tidak dapat login dan menampilkan pesan "username atau password anda salah"	User tidak dapat login dan menampilkan pesan "username atau password anda salah" sesuai yang diharapkan	Diterima

4.2.2.2. Pengujian Data User

a. Pengujian Cari Data User

Tabel IV.6. Pengujian Cari Data User

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username	Edit box terisi oleh username yang dicari	Edit box pencarian terisi oleh username yang dicari	Diterima
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data User

Tabel IV.7. Pengujian Tambah Data User

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username Password Hak Akses	Username tercantum pada edit box username Password tercantum pada edit box Password Hak akses	Dapat mengisi username, password, dan memilih hak akses sesuai dengan yang diharapkan	Diterima

	tercantum pada combo box hak akses dan dapat memilih salah satu hak akses pembukuan, pembelian, penjualan		
Klik tombol simpan	Data tersimpan pada tabel user	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi user ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username kosong	Data tidak dapat disimpan pada tabel user dan menampilkan pesan “username tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “username tidak boleh kosong” sesuai yang diharapkan	Diterima
Password kosong	Data tidak dapat disimpan pada tabel user dan menampilkan pesan “password tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “password tidak boleh kosong” sesuai yang diharapkan	Diterima
Hak akses kosong	Data tidak dapat disimpan pada tabel user dan menampilkan pesan “Hak akses tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “Hak akses tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Hapus Data User

Tabel IV.8. Pengujian Hapus Data User

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol hapus	Jika menghapus user yang sedang aktif maka akan menampilkan pesan “User sedang aktif tidak dapat dihapus” Jika menghapus user yang tidak sedang aktif maka akan menampilkan pesan “anda yakin akan menghapus data user” jika ya maka data akan dihapus	Memberikan pesan “User sedang aktif tidak dapat dihapus” dan “anda yakin akan menghapus data user” jika ya maka data akan dihapus sesuai dengan yang diharapkan	Diterima

d. Pengujian Ubah Data User

Tabel IV.9. Pengujian Ubah Data User

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih dbgrid	Data yang tertera pada dbgrid keluar pada form ubah data	Username tampil sesuai dengan editbox yang tersedia	Diterima
Klik tombol simpan	Data tersimpan pada tabel user	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi user ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Password kosong	Data tidak dapat disimpan pada	Pengubahan data gagal dan	Diterima

	tabel user dan menampilkan pesan “password tidak boleh kosong”	menampilkan pesan bahwa “password tidak boleh kosong” sesuai yang diharapkan	

4.2.2.3. Pengujian Data Barang

a. Pengujian Cari Data Barang

Tabel IV.10. Pengujian Cari Data Barang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Barang	Edit box terisi oleh kode barang yang dicari	Edit box pencarian terisi oleh kode barang yang dicari	Diterima
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data Barang

Tabel IV.11. Pengujian Tambah Data Barang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Barang	Kode Barang tercantum pada edit box Kode Barang	Dapat mengisi Kode Barang sesuai dengan yang diharapkan	Diterima
Nama barang	Nama barang tercantum pada edit box Password	Dapat mengisi Nama barang sesuai dengan yang diharapkan	Diterima
Terbitan	Dapat memilih salah satu jenis terbitan yang ada pada radio button terbitan	Dapat memilih salah satu jenis terbitan yang ada pada radio button terbitan sesuai yang diharapkan	Diterima
Id supplier	Dapat memilih salah satu supplier yang ada pada combo box id	Dapat memilih salah satu supplier yang ada pada combo box	Diterima

	supplier	id supplier sesuai yang diharapkan	
Harga bandrol	Harga bandrol tercantum pada edit box Harga bandrol	Dapat mengisi Harga bandrol sesuai dengan yang diharapkan	Diterima
Harga jual	Harga jual tercantum pada edit box Harga jual	Dapat mengisi Harga jual sesuai dengan yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada tabel barang	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi barang ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Barang kosong	Data tidak dapat disimpan pada tabel barang dan menampilkan pesan “Kode Barang tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “Kode Barang tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Hapus Data Barang

Tabel IV.12. Pengujian Hapus Data Barang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol hapus	Jika menghapus data barang maka akan menampilkan pesan “apakah anda yakin akan menghapus data barang” jika ya maka data akan dihapus	Memberikan pesan “apakah anda yakin akan menghapus data barang” jika ya maka data akan dihapus sesuai dengan yang diharapkan	Diterima

d. Pengujian Ubah Data Barang

Tabel IV.13. Pengujian Ubah Data Barang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih dbgrid	Data yang tertera pada dbgrid keluar pada form ubah data	Data barang tampil sesuai dengan editbox yang tersedia	Diterima
Klik tombol simpan	Data tersimpan pada tabel barang	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi barang ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Kode Barang kosong	Data tidak dapat disimpan pada tabel barang dan menampilkan pesan “Kode Barang tidak boleh kosong”	Pengubahan data gagal dan menampilkan pesan bahwa “Kode Barang tidak boleh kosong” sesuai yang diharapkan	Diterima

4.2.2.4. Pengujian Data Supplier

a. Pengujian Cari Data Supplier

Tabel IV.14. Pengujian Cari Data Supplier

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id supplier	Edit box terisi oleh Id supplier yang dicari	Edit box pencarian terisi oleh Id supplier yang dicari	Diterima
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data Supplier

Tabel IV.15. Pengujian Tambah Data Supplier

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id supplier	Id supplier tercantum pada edit box Id supplier	Dapat mengisi Id supplier sesuai dengan yang diharapkan	Diterima
Nama supplier	Nama supplier tercantum pada edit box nama supplier	Dapat mengisi Nama supplier sesuai dengan yang diharapkan	Diterima
Tipe supplier	Dapat memilih salah satu tipe supplier yang ada pada radio button tipe supplier	Dapat memilih salah satu tipe supplier yang ada pada radio button tipe supplier sesuai yang diharapkan	Diterima
Alamat	Alamat tercantum pada edit box Alamat	Dapat mengisi Alamat sesuai dengan yang diharapkan	Diterima
Kota	Kota tercantum pada edit box Kota	Dapat mengisi Kota sesuai dengan yang diharapkan	Diterima
Contact person	Contact person tercantum pada edit box Contact person	Dapat mengisi Contact person sesuai dengan yang diharapkan	Diterima
Telepon	Telepon tercantum pada edit box Telepon	Dapat mengisi Telepon sesuai dengan yang diharapkan	Diterima
Fax	Fax tercantum pada edit box Fax	Dapat mengisi Fax sesuai dengan yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada tabel supplier	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi supplier ditutup	Tombol batal dapat berfungsi sesuai yang	Diterima

		diharapkan	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id supplier kosong	Data tidak dapat disimpan pada tabel supplier dan menampilkan pesan “Id supplier tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “Id supplier tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Hapus Data Supplier

Tabel IV.16. Pengujian Hapus Data Supplier

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol hapus	Jika menghapus data supplier maka akan menampilkan pesan “apakah anda yakin akan menghapus data supplier” jika ya maka data akan dihapus	Memberikan pesan “apakah anda yakin akan menghapus data supplier” jika ya maka data akan dihapus sesuai dengan yang diharapkan	Diterima

d. Pengujian Ubah Data Supplier

Tabel IV.17. Pengujian Ubah Data Supplier

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih dbgrid	Data yang tertera pada dbgrid keluar pada form ubah data	Data supplier tampil sesuai dengan editbox yang tersedia	Diterima
Klik tombol simpan	Data tersimpan pada tabel supplier	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi supplier ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id supplier kosong	Data tidak dapat disimpan pada tabel supplier dan menampilkan pesan “Id supplier tidak boleh kosong”	Pengubahan data gagal dan menampilkan pesan bahwa “Id supplier tidak boleh kosong” sesuai yang diharapkan	Diterima

4.2.2.5. Pengujian Data Pelanggan

a. Pengujian Cari Data Pelanggan

Tabel IV.18. Pengujian Cari Data Pelanggan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id pelanggan	Edit box terisi oleh Id pelanggan yang dicari	Edit box pencarian terisi oleh Id pelanggan yang dicari	Diterima
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data Pelanggan

Tabel IV.19. Pengujian Tambah Data Pelanggan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id pelanggan	Id pelanggan tercantum pada edit box Id pelanggan	Dapat mengisi Id pelanggan sesuai dengan yang diharapkan	Diterima
Nama pelanggan	Nama pelanggan tercantum pada edit box nama pelanggan	Dapat mengisi Nama pelanggan sesuai dengan yang diharapkan	Diterima
Jenis pelanggan	Dapat memilih salah satu Jenis pelanggan yang ada pada radio button Jenis	Dapat memilih salah satu Jenis pelanggan yang ada pada radio button Jenis	Diterima

	pelanggan	pelanggan sesuai yang diharapkan	
Alamat	Alamat tercantum pada edit box Alamat	Dapat mengisi Alamat sesuai dengan yang diharapkan	Diterima
Contact person	Contact person tercantum pada edit box Contact person	Dapat mengisi Contact person sesuai dengan yang diharapkan	Diterima
Telepon	Telepon tercantum pada edit box Telepon	Dapat mengisi Telepon sesuai dengan yang diharapkan	Diterima
Fax	Fax tercantum pada edit box Fax	Dapat mengisi Fax sesuai dengan yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada tabel pelanggan	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi pelanggan ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id pelanggan kosong	Data tidak dapat disimpan pada tabel pelanggan dan menampilkan pesan “Id pelanggan tidak boleh kosong”	Pengubahan data gagal dan menampilkan pesan bahwa “Id pelanggan tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Hapus Data Pelanggan

Tabel IV.20. Pengujian Hapus Data Pelanggan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol hapus	Jika menghapus data pelanggan maka akan menampilkan	Memberikan pesan “apakah anda yakin akan menghapus data	Diterima

	pesan “apakah anda yakin akan menghapus data pelanggan” jika ya maka data akan dihapus	pelanggan” jika ya maka data akan dihapus sesuai dengan yang diharapkan	

d. Pengujian Ubah Data Pelanggan

Tabel IV.21. Pengujian Ubah Data Pelanggan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih dbgrid	Data yang tertera pada dbgrid keluar pada form ubah data	Data pelanggan tampil sesuai dengan editbox yang tersedia	Diterima
Klik tombol simpan	Data tersimpan pada tabel pelanggan	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form isi pelanggan ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Id pelanggan kosong	Data tidak dapat disimpan pada tabel pelanggan dan menampilkan pesan “Id pelanggan tidak boleh kosong”	Pengubahan data gagal dan menampilkan pesan bahwa “Id pelanggan tidak boleh kosong” sesuai yang diharapkan	Diterima

4.2.2.6. Pengujian Data Pembelian

a. Pengujian Cari Data Pembelian

Tabel IV.22. Pengujian Cari Data Pembelian

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur	No faktur terisi oleh No faktur	Edit box pencarian terisi	Diterima

	yang dicari	oleh No faktur yang dicari	
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data Pembelian

Tabel IV.23. Pengujian Tambah Data Pembelian

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur	No faktur tercantum pada edit box No faktur dan dapat diisi sesuai dengan nomor faktur yang diterima dari supplier	No faktur tercantum pada edit box No faktur dan dapat diisi sesuai dengan nomor faktur yang diterima dari supplier sesuai yang diharapkan	Diterima
Id supplier	Id supplier akan muncul di combo box	Id supplier tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Nama supplier	Jika Id supplier dimasukkan maka nama supplier akan otomatis muncul setelah pemilihan id supplier	nama supplier akan tampil setelah pemilihan id supplier sesuai yang diharapkan	Diterima
Data pembelian	Data pembelian diisi pada dbgrid dengan mencari kode barang maka harga bandrol akan dengan otomatis muncul jika harga bandrol sudah tercantum di data master barang. Harga bayar akan berubah sesuai diskon diisi. Sub total akan	Data pembelian dapat diisi pada dbgrid dengan mencari kode barang maka harga bandrol akan dengan otomatis muncul. Harga bayar akan berubah sesuai diskon diisi. Sub total akan terisi sesuai dengan jatah yang	Diterima

	terisi sesuai dengan jatah yang diambil dikurangi retur dikalikan harga bayar	diambil dikurangi retur dikalikan harga bayar sesuai yang diharapkan	
Total	Total akan terisi berdasarkan jumlah sub total	Total dapat terisi berdasarkan jumlah sub total sesuai yang diharapkan	Diterima
Jumlah bayar	Jumlah bayar tercantum pada edit box Jumlah bayar dan dengan otomatis saldo hutang akan berubah sesuai jumlah bayar dimasukan	Dapat mengisi Jumlah bayar dan dengan otomatis saldo hutang akan berubah sesuai jumlah bayar dimasukan sesuai yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada tabel pembelian	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form tambah pembelian ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur kosong	Data tidak dapat disimpan pada tabel pembelian dan menampilkan pesan “No faktur tidak boleh kosong”	Pennambahan data gagal dan menampilkan pesan bahwa “No faktur tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Ubah Data Pembelian

Tabel IV.24. Pengujian Ubah Data Pembelian

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data pembelian: retur	Retur diisikan pada data pembelian dan otomatis akan	Retur dapat diisikan pada data pembelian dan	Diterima

	mengubah sub total, total dan saldo hutang	otomatis akan mengubah sub total, total dan saldo hutang sesuai yang diharapkan	

d. Pengujian Tambah Retur Pembelian

Tabel IV.25. Pengujian Tambah Retur Pembelian

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No retur	No retur tercantum pada edit box No retur dan terisi secara otomatis	No retur tercantum pada edit box No retur dan terisi secara otomatis sesuai yang diharapkan	Diterima
Id supplier	Id supplier akan muncul di combo box	Id supplier tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Nama supplier	Jika Id supplier dimasukkan maka nama supplier akan otomatis muncul setelah pemilihan id supplier	nama supplier akan tampil setelah pemilihan id supplier sesuai yang diharapkan	Diterima
No faktur	No faktur akan muncul di combo box	No faktur tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Data pembelian	Data pembelian diisi pada dbgrid setelah memilih no faktur yang tersedia	Data pembelian dapat diisi pada dbgrid setelah memilih no faktur yang tersedia sesuai yang diharapkan	Diterima
Klik tombol retur>>	Data pembelian yang akan diretur dipindahkan ke dbgrid retur	Data pembelian yang akan diretur dipindahkan ke dbgrid retur	Diterima

		sesuai yang diharapkan	
Klik tombol <<	Data pembelian yang akan diretur dipindahkan kembali ke dbgrid pembelian	Data pembelian yang akan diretur dipindahkan kembali ke dbgrid pembelian sesuai yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada returbeli dan returbelidetail	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form tambah retur pembelian ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur kosong	Data tidak dapat disimpan pada tabel returbeli dan returbelidetail dan menampilkan pesan “No faktur tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “No faktur tidak boleh kosong” sesuai yang diharapkan	Diterima

e. Pengujian Pembayaran Hutang

Tabel IV.26. Pengujian Pembayaran Hutang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No transaksi	No transaksi tercantum pada edit box No transaksi dan terisi secara otomatis	No transaksi tercantum pada edit box No transaksi dan terisi secara otomatis sesuai yang diharapkan	Diterima
supplier	Id supplier dan nama supplier akan muncul di combo box	Id supplier dan nama supplier tampil di combo box yang tersedia sesuai yang diharapkan	Diterima

Jumlah bayar hutang	Jumlah bayar diisikan sesuai atau lebih kecil dari sisa bayar yang harus dibayarkan	Jumlah bayar diisikan sesuai atau lebih kecil dari sisa bayar yang harus dibayarkan sesuai yang diharapkan	Diterima
Sisa bayar	Jika Id supplier dimasukkan maka sisa bayar akan otomatis muncul setelah pemilihan id supplier	Sisa bayar akan tampil setelah pemilihan id supplier sesuai yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada buku_hutang	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form pembayaran hutang ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No transaksi kosong	Data tidak dapat disimpan pada tabel buku_hutang dan menampilkan pesan “No transaksi tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “No transaksi tidak boleh kosong” sesuai yang diharapkan	Diterima

4.2.2.7. Pengujian Data Penjualan

a. Pengujian Cari Data Penjualan

Tabel IV.27. Pengujian Cari Data Penjualan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur	No faktur terisi oleh No faktur yang dicari	Edit box pencarian terisi oleh No faktur yang dicari	Diterima
Klik tombol cari	Data yang dicari ditunjukan di dbgrid	Data yang keluar sesuai dengan yang diharapkan	Diterima

b. Pengujian Tambah Data Penjualan

Tabel IV.28. Pengujian Tambah Data Penjualan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur	No faktur tercantum pada edit box No faktur dan terisi sesuai dengan bulan, tahun dan nomor urut pembuatan faktur	No faktur tercantum pada edit box No faktur dan terisi sesuai dengan bulan, tahun dan nomor urut pembuatan faktur sesuai yang diharapkan	Diterima
Id pelanggan	Id pelanggan akan muncul di combo box	Id pelanggan tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Nama pelanggan	Jika Id pelanggan dimasukkan maka nama pelanggan akan otomatis muncul setelah pemilihan id pelanggan	nama pelanggan akan tampil setelah pemilihan id pelanggan sesuai yang diharapkan	Diterima
Data penjualan	Data penjualan diisi pada dbgrid dengan mencari kode barang maka harga jual akan dengan otomatis muncul jika harga jual sudah tercantum di data master barang. Sub total akan terisi sesuai dengan eksemplar yang diambil dikalikan harga jual	Data penjualan dapatdiisi pada dbgrid dengan mencari kode barang maka harga jual akan dengan otomatis muncul jika harga jual sudah tercantum di data master barang. Sub total akan terisi sesuai dengan eksemplar yang diambil dikalikan harga jual sesuai yang diharapkan	
Total	Total akan terisi berdasarkan	Total dapat terisi berdasarkan	

	jumlah sub total	jumlah sub total sesuai yang diharapkan	
Jumlah bayar	Jumlah bayar tercantum pada edit box Jumlah bayar dan dengan otomatis saldo piutang akan berubah sesuai jumlah bayar dimasukan	Dapat mengisi Jumlah bayar dan dengan otomatis saldo piutang akan berubah sesuai jumlah bayar dimasukan sesuai yang diharapkan	
Klik tombol simpan	Data tersimpan pada tabel penjualan	Tombol simpan dapat berfungsi sesuai yang diharapkan	
Klik tombol batal	Form tambah penjualan ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur kosong	Data tidak dapat disimpan pada tabel pembelian dan menampilkan pesan “No faktur tidak boleh kosong”	Pennambahan data gagal dan menampilkan pesan bahwa “No faktur tidak boleh kosong” sesuai yang diharapkan	Diterima

c. Pengujian Tambah Retur Penjualan

Tabel IV.29. Pengujian Tambah Retur Penjualan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No retur	No retur tercantum pada edit box No retur dan terisi secara otomatis	No retur tercantum pada edit box No retur dan terisi secara otomatis sesuai yang diharapkan	Diterima
Id pelanggan	Id pelanggan akan muncul di combo box	Id pelanggan tampil di combo box yang tersedia sesuai yang	Diterima

		diharapkan	
Nama pelanggan	Jika Id pelanggan dimasukkan maka nama pelanggan akan otomatis muncul setelah pemilihan id pelanggan	nama pelanggan akan tampil setelah pemilihan id pelanggan sesuai yang diharapkan	Diterima
No faktur	No faktur akan muncul di combo box	No faktur tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Data pembelian	Data pembelian diisi pada dbgrid setelah memilih no faktur yang tersedia	Data pembelian dapat diisi pada dbgrid setelah memilih no faktur yang tersedia sesuai yang diharapkan	Diterima
Klik tombol retur>>	Data penjualan yang akan diretur dipindahkan ke dbgrid retur	Data penjualan yang akan diretur dipindahkan ke dbgrid retur sesuai yang diharapkan	Diterima
Klik tombol <<	Data penjualan yang akan diretur dipindahkan kembali ke dbgrid penjualan	Data penjualan yang akan diretur dipindahkan kembali ke dbgrid penjualan sesuai yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada returjual dan returjualdetail	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form tambah retur penjualan ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No faktur kosong	Data tidak dapat disimpan pada tabel returjual dan	Penambahan data gagal dan menampilkan	Diterima

	returjualdetail dan menampilkan pesan “No faktur tidak boleh kosong”	pesan bahwa “No faktur tidak boleh kosong” sesuai yang diharapkan	

d. Pengujian Pembayaran Piutang

Tabel IV.30. Pengujian Pembayaran Piutang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No transaksi	No transaksi tercantum pada edit box No transaksi dan terisi secara otomatis	No transaksi tercantum pada edit box No transaksi dan terisi secara otomatis sesuai yang diharapkan	Diterima
pelanggan	Id pelanggan dan nama pelanggan akan muncul di combo box	Id pelanggan dan nama pelanggan tampil di combo box yang tersedia sesuai yang diharapkan	Diterima
Jumlah bayar piutang	Jumlah bayar diisikan sesuai atau lebih kecil dari sisa bayar yang harus dibayarkan	Jumlah bayar diisikan sesuai atau lebih kecil dari sisa bayar yang harus dibayarkan sesuai yang diharapkan	Diterima
Sisa bayar	Jika Id pelanggan dimasukkan maka sisa bayar akan otomatis muncul setelah pemilihan id pelanggan	Sisa bayar akan tampil setelah pemilihan id pelanggan sesuai yang diharapkan	Diterima
Klik tombol simpan	Data tersimpan pada buku_piutang	Tombol simpan dapat berfungsi sesuai yang diharapkan	Diterima
Klik tombol batal	Form pembayaran piutang ditutup	Tombol batal dapat berfungsi sesuai yang diharapkan	Diterima

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No transaksi kosong	Data tidak dapat disimpan pada tabel buku_piutang dan menampilkan pesan “No transaksi tidak boleh kosong”	Penambahan data gagal dan menampilkan pesan bahwa “No transaksi tidak boleh kosong” sesuai yang diharapkan	Diterima

4.2.3. Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian alpha dengan kasus uji sample atas perangkat lunak diatas yang menggunakan metode *Black Box* pada tiap tampilan hasil dari pengujian alpha ini dapat dikatakan berhasil atau diterima.

Berdasarkan hasil pengujian ini maka dapat ditarik kesimpulan bahwa perangkat lunak ini telah sesuai dengan yang diharapkan dan kebutuhan user serta bebas error sehingga layak dipergunakan.

4.2 Pengujian Beta

Pengujian Beta merupakan pengujian yang dilakuakn secara objektif dimana diuji secara langsung ke lapangan. Dengan membuat kuesioner mengenai kegunaan sistem yang dibangun, apakah sistem tersebut sudah dapat memberikan apa yang diinginkan oleh pengguna atau belum. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulannya terhadap penilaian penerapan sistem informasi penjualan Koran dan majalah yang dibangun. (Lembaran kuesioner dilampirkan).

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

1. Sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi *supplier* dan pelanggan?

No.	Keterangan
1.	Setuju
2.	Ragu-ragu
3.	Tidak Setuju

2. Sistem yang dibuat mempermudah dan mempercepat mendapatkan informasi data barang?

No.	Keterangan
1.	Setuju
2.	Ragu-ragu
3.	Tidak Setuju

3. Sistem yang dibuat dapat mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan?

No.	Keterangan
1.	Setuju
2.	Ragu-ragu
3.	Tidak Setuju

4. Sistem yang dibuat dapat mempermudah dan mempercepat pembuatan laporan?

No.	Keterangan
1.	Setuju
2.	Ragu-ragu
3.	Tidak Setuju

5. Sistem yang dibuat dapat mempercepat kinerja perusahaan?

No.	Keterangan
1.	Setuju
2.	Ragu-ragu
3.	Tidak Setuju

Berdasarkan data hasil kuesioner, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus: Y = P/Q*100%

Keterangan:
Y = Nilai Persentase
P = Banyaknya jawaban responden tiap soal
Q = Jumlah responden
Berikut adalah hasil yang didapat dari kuesioner:

1. Sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi *supplier* dan pelanggan?

No.	Keterangan	Responden	Persentase (%)
1.	Setuju	8	80%
2.	Ragu-ragu	2	20%
3.	Tidak Setuju	-	-

2. Sistem yang dibuat mempermudah dan mempercepat mendapatkan informasi data barang?

No.	Keterangan	Responden	Persentase (%)
1.	Setuju	8	80%
2.	Ragu-ragu	2	20%
3.	Tidak Setuju	-	-

3. Sistem yang dibuat dapat mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan?

No.	Keterangan	Responden	Persentase (%)
1.	Setuju	8	80%
2.	Ragu-ragu	2	20%
3.	Tidak Setuju	-	-

4. Sistem yang dibuat dapat mempermudah dan mempercepat pembuatan laporan?

No.	Keterangan	Responden	Persentase (%)
1.	Setuju	7	70%
2.	Ragu-ragu	3	30%
3.	Tidak Setuju	-	-

5. Sistem yang dibuat dapat mempercepat kinerja perusahaan?

No.	Keterangan	Responden	Persentase (%)
1.	Setuju	8	80%
2.	Ragu-ragu	2	20%
3.	Tidak Setuju	-	-

4.3.1. Kesipmpulan Hasil Pengujian Beta

Berdasarkan hasil dari kuesioner yang telah diisi maka akan disimpulkan sebagai berikut:

1. Sebanyak 8 orang atau 80% setuju bahwa sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi *supplier* dan pelanggan.
2. Sebanyak 2 orang atau 20% ragu-ragu bahwa sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi *supplier* dan pelanggan.
3. Sebanyak 8 orang atau 80% setuju bahwa sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi data barang.
4. Sebanyak 2 orang atau 20% ragu-ragu bahwa sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi data barang.

5. Sebanyak 8 orang atau 80% setuju bahwa sistem yang dibuat dapat mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan.
6. Sebanyak 2 orang atau 20% ragu-ragu bahwa sistem yang dibuat dapat mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan.
7. Sebanyak 7 orang atau 70% setuju bahwa sistem yang dibuat dapat mempermudah dan mempercepat pembuatan laporan.
8. Sebanyak 3 orang atau 30% ragu-ragu bahwa sistem yang dibuat dapat mempermudah dan mempercepat pembuatan laporan.
9. Sebanyak 8 orang atau 80% setuju bahwa sistem yang dibuat dapat mempercepat kinerja perusahaan.
10. Sebanyak 2 orang atau 20% ragu-ragu bahwa sistem yang dibuat dapat mempercepat kinerja perusahaan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melewati pembangunan Sistem Informasi Penjualan Koran dan Majalah di Maju Terus Agency serta melakukan pengujan, maka dapat dimbil kesimpulan sebagai berikut:

1. Mempermudah dan mempercepat proses kerja karyawan dalam mendapatkan informasi mengenai *supplier*, pelanggan, informasi data barang.
2. Mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan.
3. Mendukung dalam proses pembuatan laporan dan faktur.
4. Mendukung kinerja Maju Terus Agency dengan memberikan fasilitas aplikasi yang lengkap.

5.2 Saran

Sistem Informasi Penjualan Koran dan Majalah di maju Terus Agency ini masih bisa dikembangkan seiring dengan perkembangan spesifikasi kebutuhan pengguna sistem yang harus dipenuhi dalam mencapai tahap yang lebih tinggi dan kinerja sistem yang lebih baik. Berikut adalah saran untuk pengembangan sistem lebih lanjut:

1. Diharapkan adanya laporan transaksi berupa grafik untuk transaksi penjualan.

DAFTAR PUSTAKA

[1] AN.(2003), *Panduan Praktis Pemrograman Delphi 7.0*, Penerbit Andi, Yogyakarta.

[2] Jogiyanto Hartono.(1999), Analisis&Desain Sistem Informasi, Penerbit Andi, Yogyakarta.

[3] Laudon, Kenneth C. (2005), *Sisten Informasi Manajemen Edisi 8*, Penerbit Andi, Yogyakarta, 317-326

[4] Pressman, Roger S. (2005), *Software Engineering Sixth Edition*, McGraw-Hill, New York, 79-80

[5] Sommerville, Ian. (2003), *Software Engineering Jilid 2*, Erlangga, Jakarta, 68-87

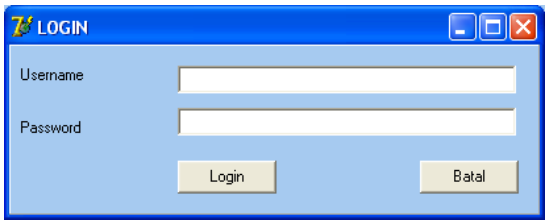
[6] Umi Proboyekti. (16 Januari 2009), Software Process Model I, http://www.lecturer.ukdw.ac.id/othie/sofwareprocess.pdf

LAMPIRAN A

TAMPILAN ANTAR MUKA

Tampilan Antar Muka Sistem Informasi Pendataan

Penyaluran Koran dan Majalah Di Maju Terus Agency



Tampilan Menu Login



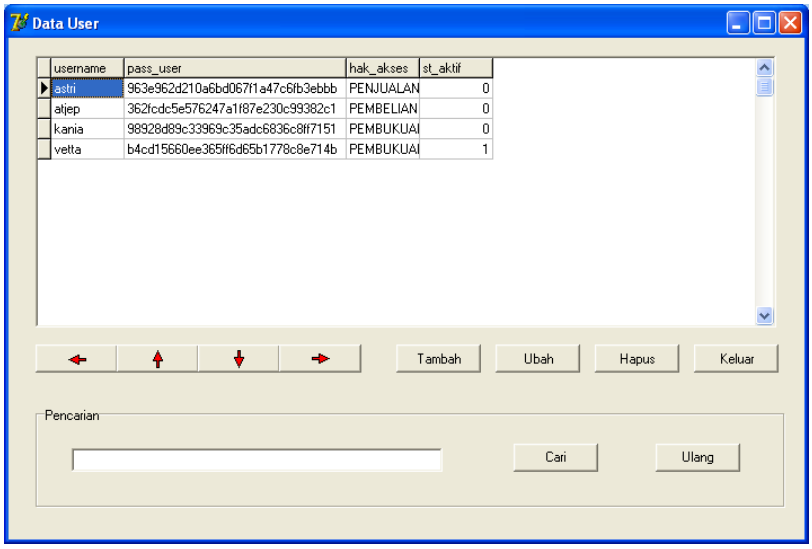
Tampilan Menu Utama Pembukuan (Admin)



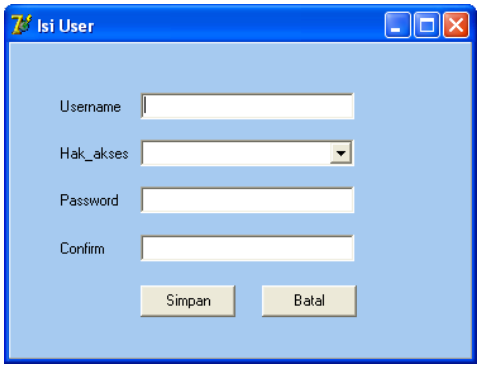
Tampilan Menu Utama Pembelian



Tampilan Menu Utama Penjualan



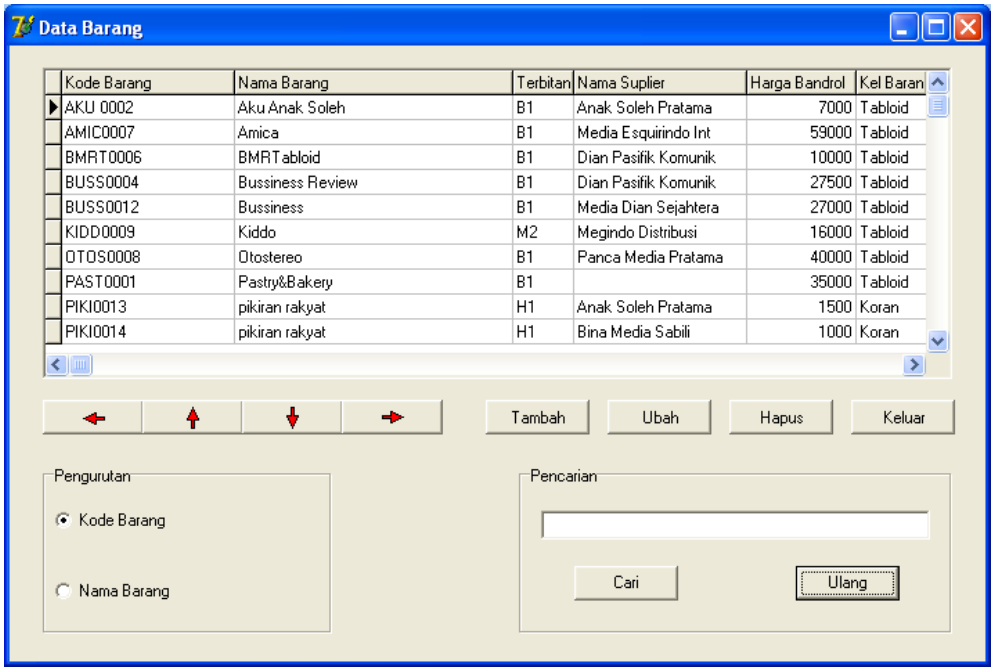
Tampilan Data User



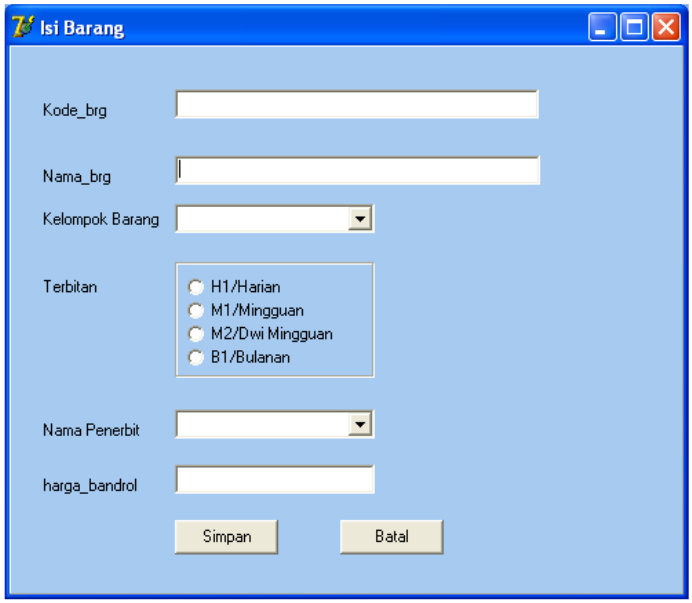
Tampilan Tambah Data User



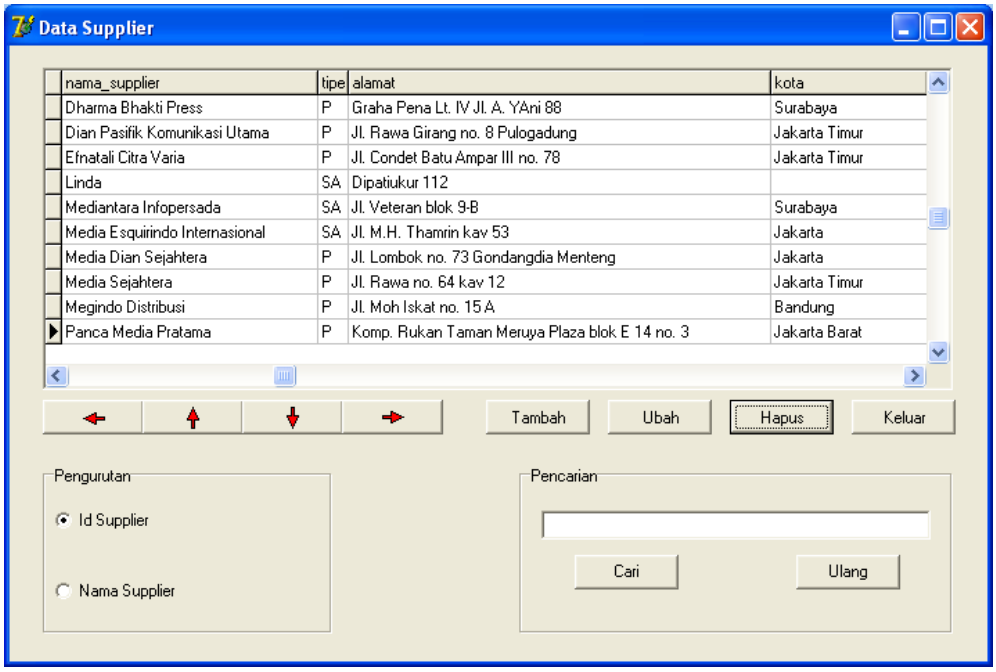
Tampilan Ubah Data User



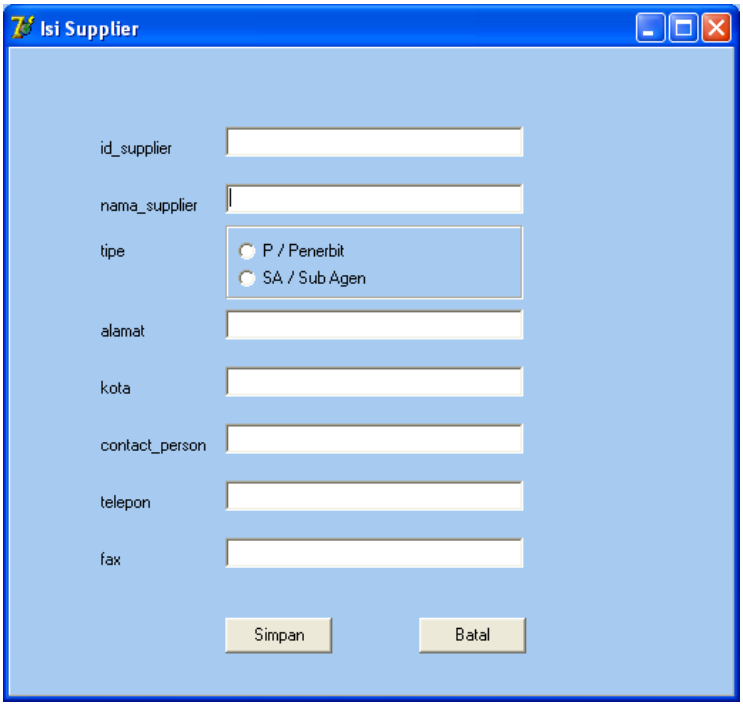
Tampilan Data Barang



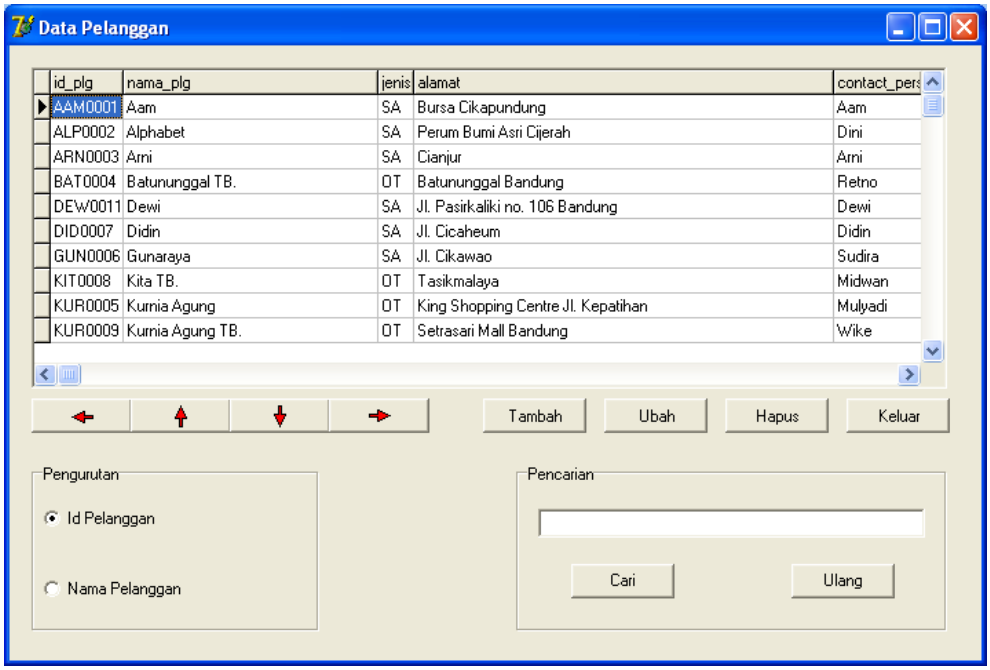
Tampilan Tambah Data Barang



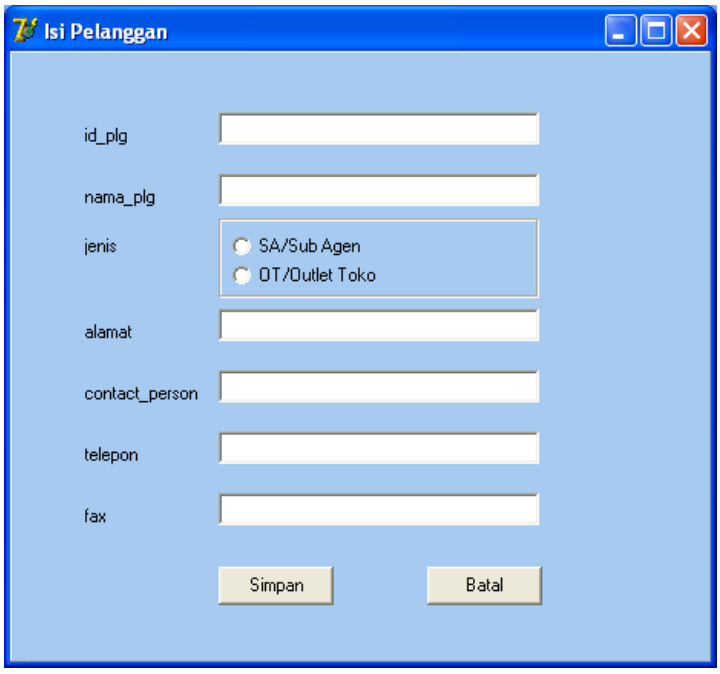
Tampilan Data Supplier



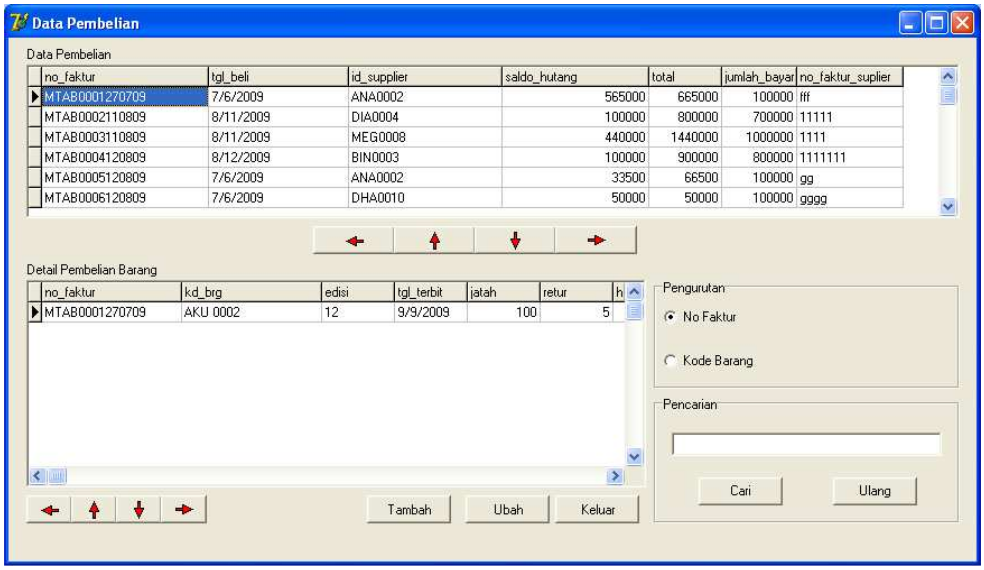
Tampilan Tambah Data Supplier



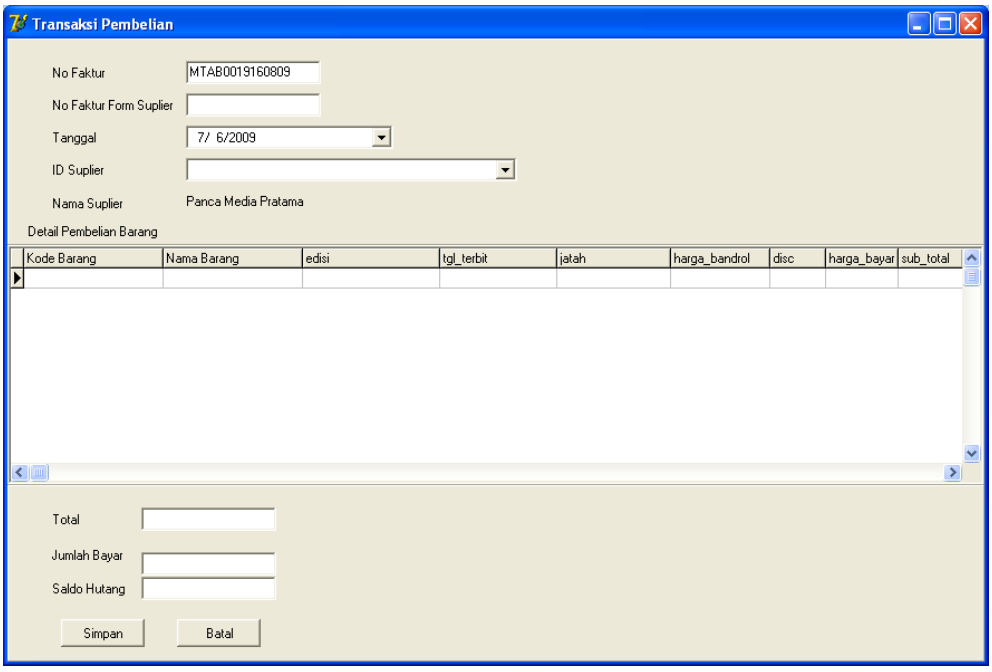
Tampilan Data Pelanggan



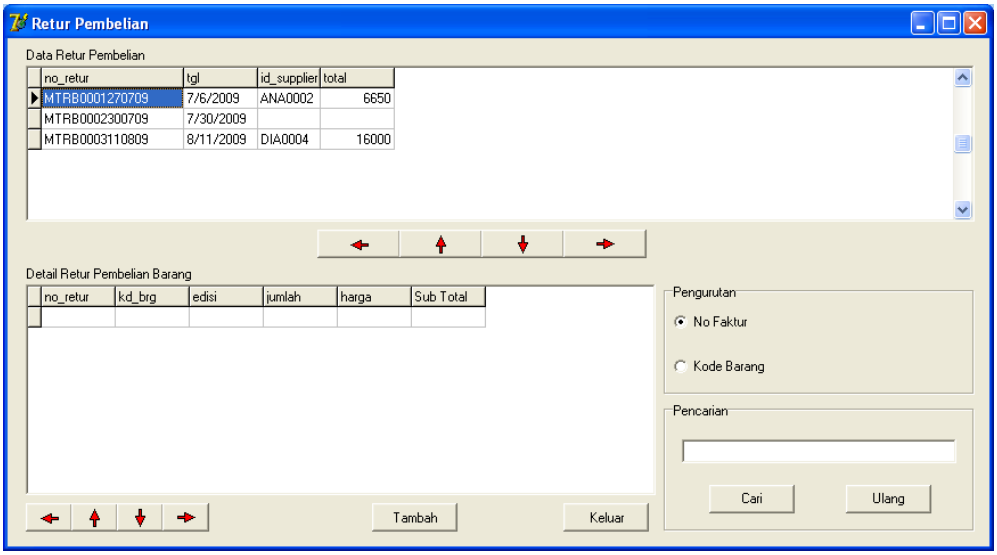
Tampilan Tambah Data Pelanggan



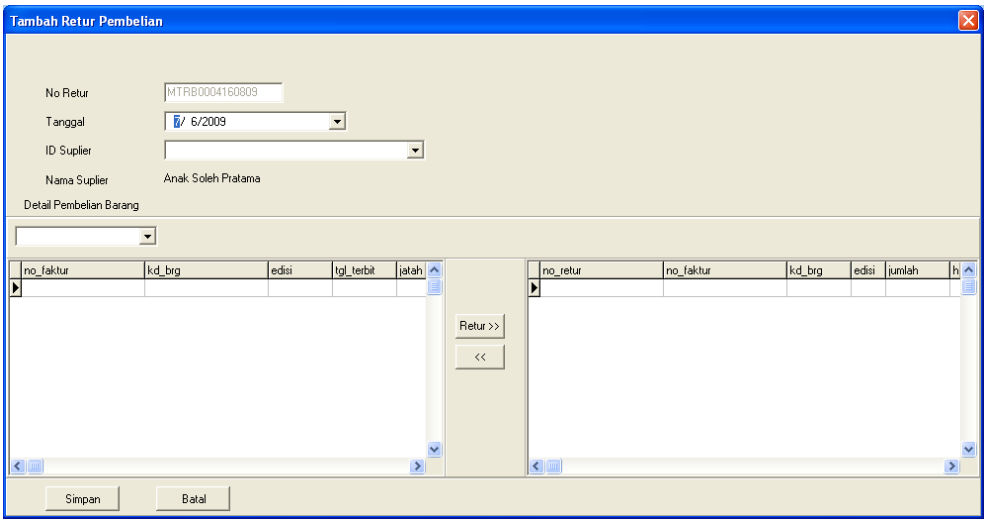
Tampilan Data Pembelian



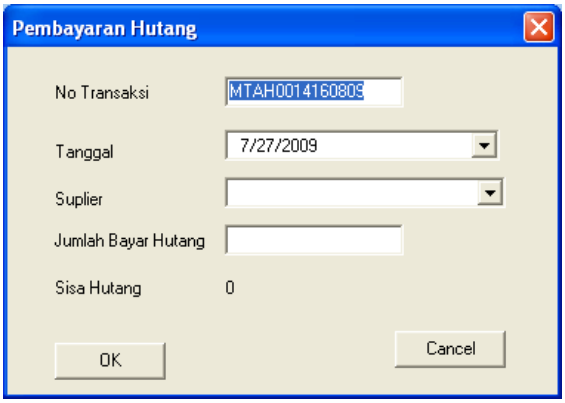
Tampilan Tambah Pembelian



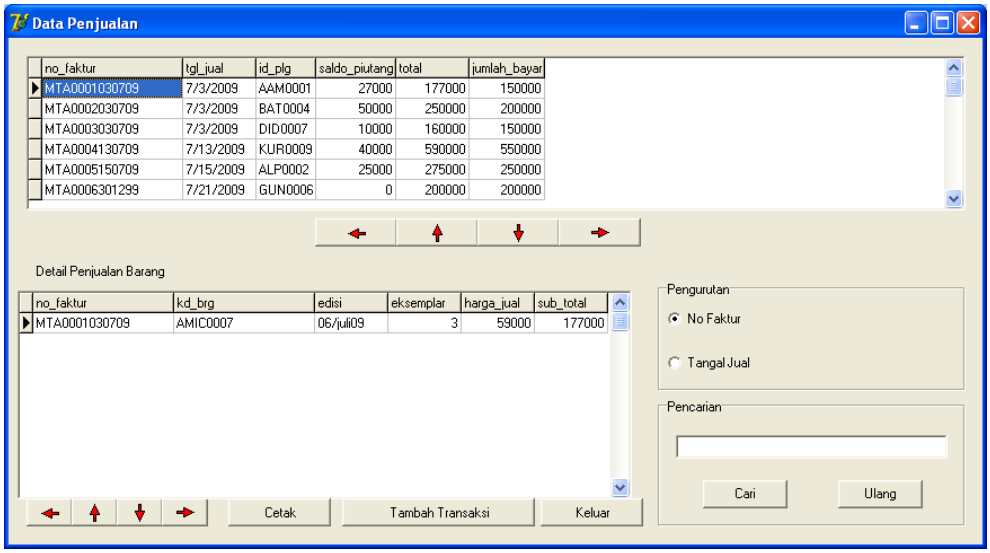
Tampilan Retur Pembelian



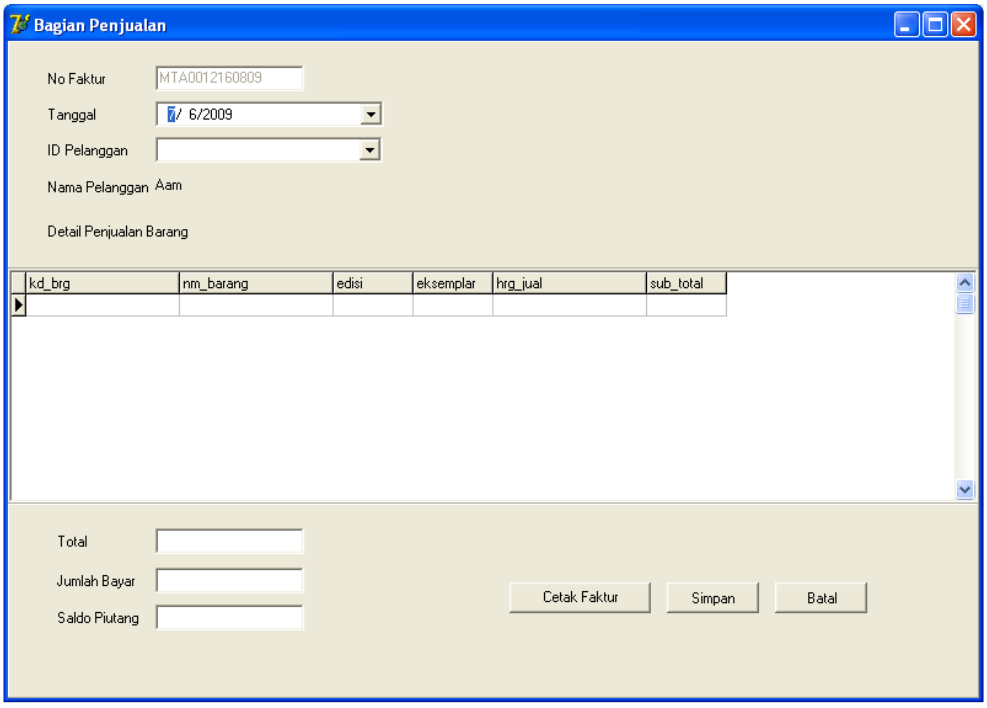
Tampilan Tambah Retur Pembelian



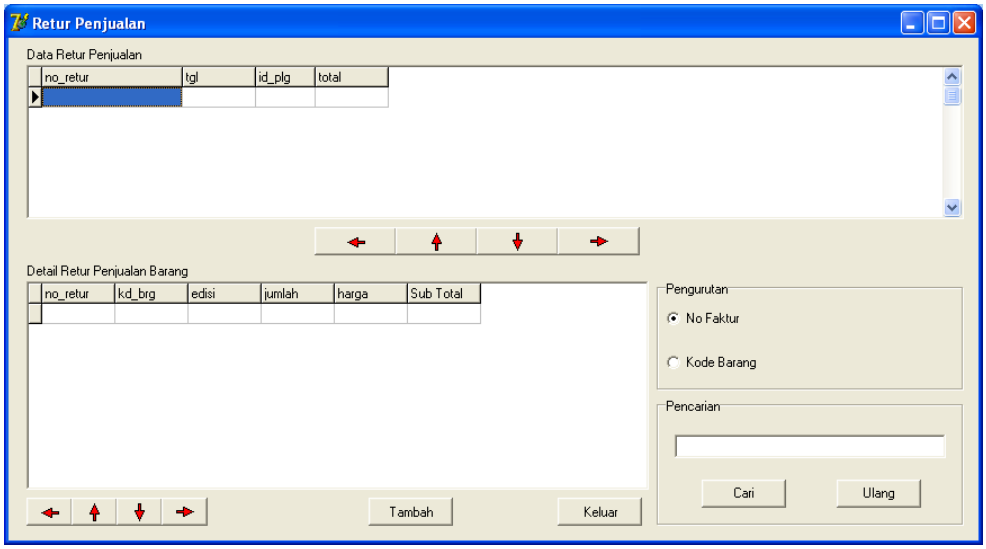
Tampilan Pembayaran Hutang



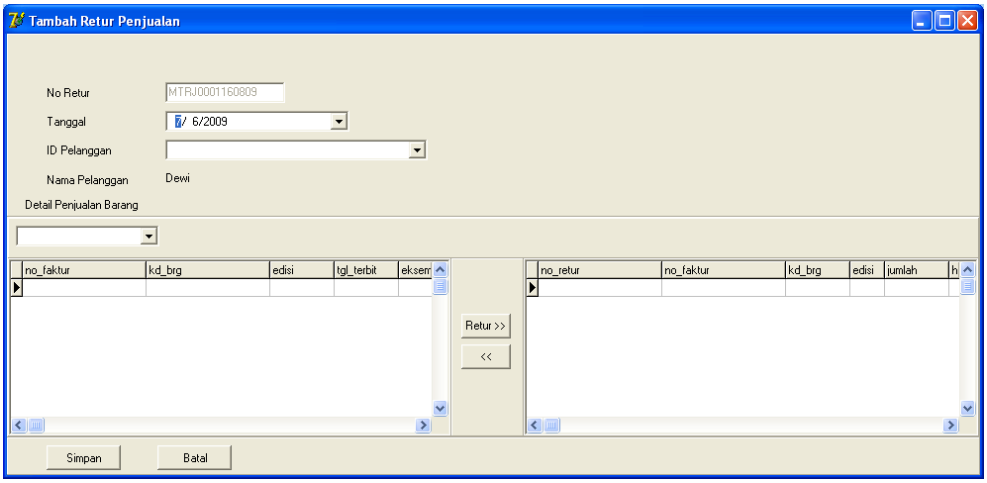
Tampilan Data Penjualan



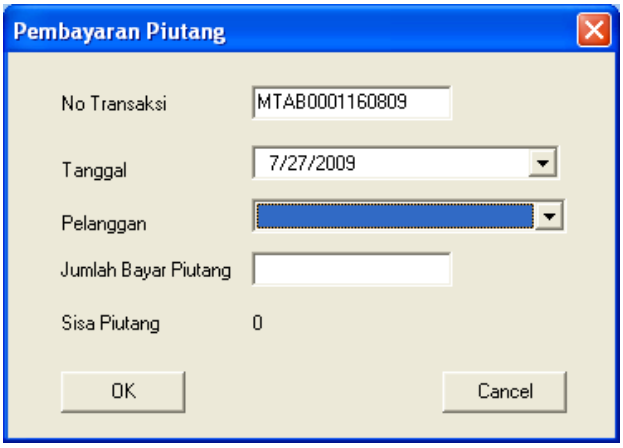
Tampilan Tambah Penjualan



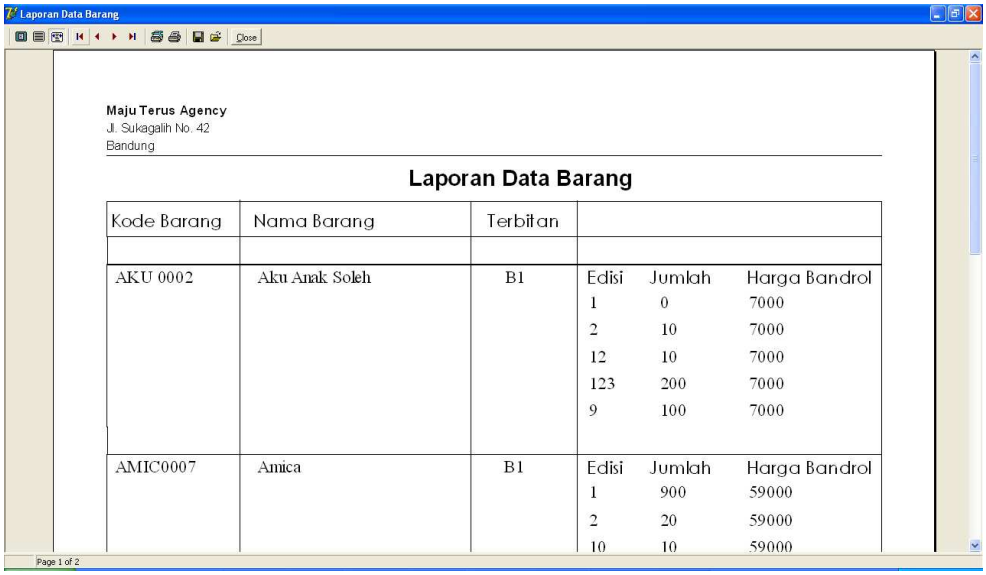
Tampilan Retur Penjualan



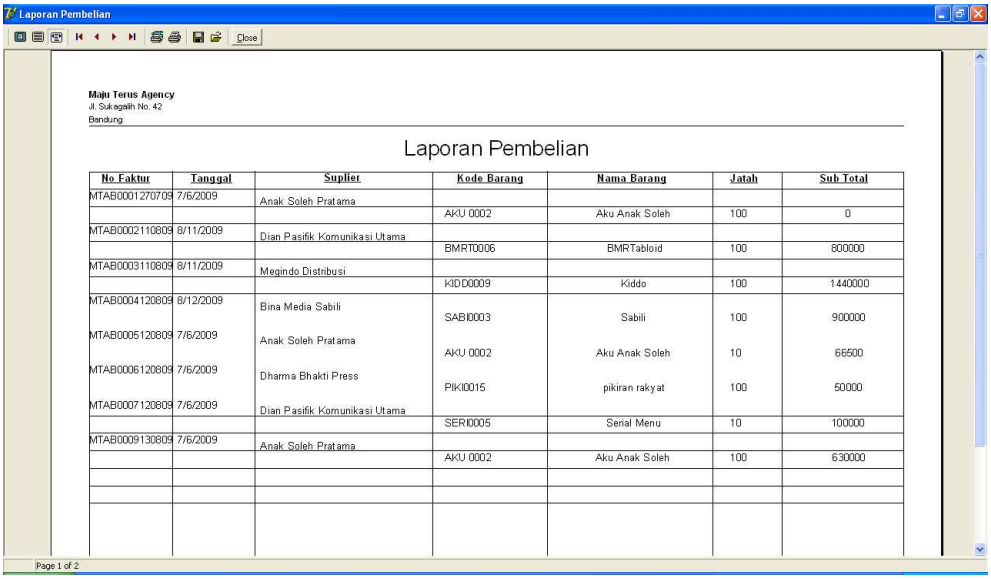
Tampilan Tambah Retur Penjualan



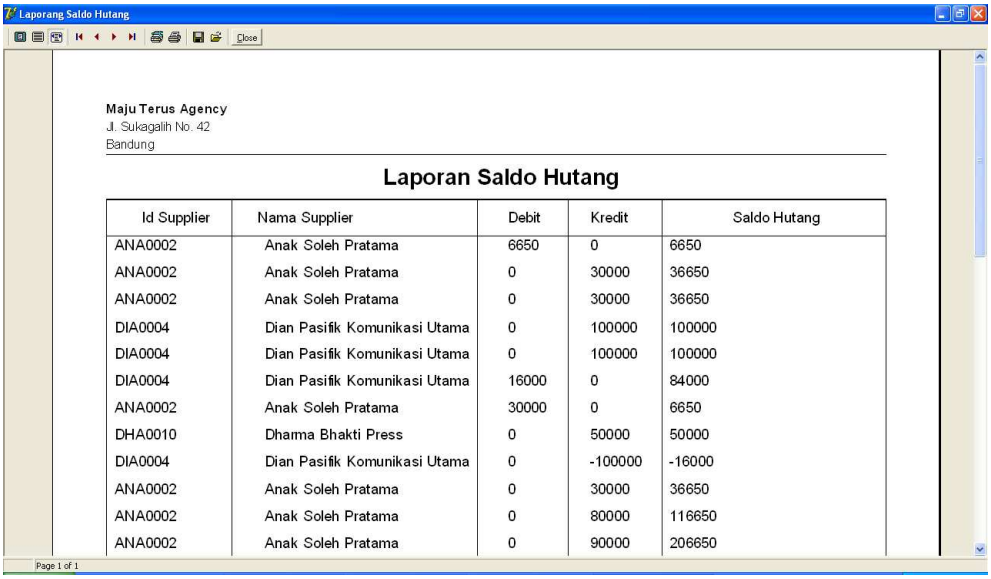
Tampilan Pembayaran Piutang



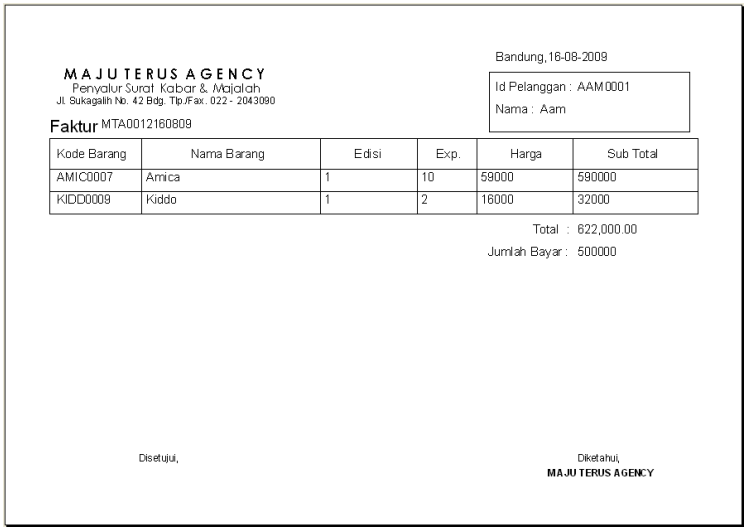
Tampilan Laporan Data Barang



Tampilan Laporan Pembelian



Tampilan Laporan Saldo Hutang



Tampilan Faktur Penjualan

LAMPIRAN B

LISTING PROGRAM

Listing Program
Sistem Informasi Penjualan Koran dan Majalah
di Maju Terus Agency

```
procedure TFadmin.DataBarangClick(Sender:
TObject);
begin
  Application.CreateForm(TFbarang,Fbarang);
  Fbarang.ShowModal;
end;

procedure TFadmin.DataUserClick(Sender:
TObject);
begin
  Application.CreateForm(TFuser1,Fuser1);
  Fuser1.ShowModal;
end;

procedure TFadmin.LDataBarangClick(Sender:
TObject);
begin
  Application.CreateForm(TFlapbrg,Flapbrg);
  Flapbrg.ShowModal;
end;

procedure
TFadmin.LDataSupplierClick(Sender:
TObject);
begin

Application.CreateForm(TFlapsupplier,Flapsup
plier);
  Flapsupplier.ShowModal;
end;

procedure
TFadmin.DataPelanggan2Click(Sender:
TObject);
begin

Application.CreateForm(TFlappelanggan,Flapp
elanggan);
  Flappelanggan.ShowModal;
end;

procedure TFadmin.DataSuplierClick(Sender:
TObject);
begin

Application.CreateForm(TFsupplier,Fsupplier);
  Fsupplier.ShowModal;
end;

procedure
TFadmin.DataPelangganClick(Sender:
TObject);
begin

Application.CreateForm(TFpelanggan,Fpelang
gan);
  Fpelanggan.ShowModal;
end;

procedure TFadmin.Logout1Click(Sender:
TObject);
begin
  if dm.StatusLogin then
  begin
    if MessageDlg('Apakah Anda Akan
Logout?', mtInformation, [mbYes,mbNo],0) =
mrYes then
    begin
    dm.StatusLogin := False;
    dm.UpdateStatusUser;
    HakAkses('NONE');
    end;
  end;
end;

procedure
TFadmin.DataPenjualan1Click(Sender:
TObject);
begin

Application.CreateForm(TFlappenjualan,Flapp
enjualan);
   Flappenjualan.ShowModal;
end;

procedure
TFadmin.DataPembelian1Click(Sender:
TObject);
begin

Application.CreateForm(TFlappembelian,Flap
pembelian);
   Flappembelian.ShowModal;
end;

procedure
TFadmin.DataSaldoPiutang1Click(Sender:
TObject);
begin
```

```
Application.CreateForm(TFlappiutang,Flappiut
ang);
  Flappiutang.ShowModal;
end;

procedure
TFadmin.DataSaldoHutang1Click(Sender:
TObject);
begin

Application.CreateForm(TFlaphutang,Flaphuta
ng);
  Flaphutang.ShowModal;
end;

procedure TFadmin.HakAkses(akses: String);
begin
// procedure menentukan hak akses user
  if akses ='PEMBUKUAN' then
  begin
    MasterData.Visible := true;
    Laporan.Visible := true;
    Transaksi.Visible := false;
    Pembelian.Visible := False;
    Penjualan.Visible := False;
    Retur1.Visible := False;
  end
  else if akses ='PEMBELIAN' then
  begin
    MasterData.Visible := false;
    Laporan.Visible := false;
    Transaksi.Visible := true;
    Penjualan.Visible := False;
    Pembelian.Visible := true;
    Retur1.Visible := True;
    RPembelian.Visible := True;
    RPenjualan.Visible := False;
    Pembayaran.Visible := False;
    Pembayaran.Visible := True;
    Piutang.Visible := False;
    Hutang.Visible := True;
  end
  else if akses ='PENJUALAN' then
  begin
    MasterData.Visible := false;
    Laporan.Visible := false;
    Transaksi.Visible := true;
    Pembelian.Visible := False;
    Penjualan.Visible := true;
    Retur1.Visible := True;
    RPembelian.Visible := False;
    RPenjualan.Visible := True;
    Pembayaran.Visible := False;
    Pembayaran.Visible := True;
    Piutang.Visible := True;
    Hutang.Visible := False;
  end
  else if akses ='NONE' then
  begin
    MasterData.Visible := false;
    Laporan.Visible := false;
    Transaksi.Visible := false;
  end;
end;

procedure TFadmin.PembelianClick(Sender:
TObject);
begin

Application.CreateForm(TFpembelianutama,F
pembelianutama);
  Fpembelianutama.ShowModal;
  dm.Tpembelian.Open;
end;

procedure TFadmin.Login1Click(Sender:
TObject);
begin
  if not dm.StatusLogin then
  begin
    Application.CreateForm(TFlogin, Flogin);
    if  Flogin.ShowModal = mrOk then
    begin
      bar.Panels[0].Text := 'User Login as'+ user;
      HakAkses(akses);
      Flogin.Free;
    end;
  end
  else
    MessageDlg('User Sudah
Login',mtInformation,[mbok],0);
end;

procedure TFadmin.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
  Action := caNone;
  if MessageDlg('Anda Yakin Akan Keluar
Dari Aplikasi?', mtInformation,
[mbYes,mbNo],0) = mrYes then
  begin
    dm.StatusLogin := False;
    if  dm.Quser.Active then
      dm.UpdateStatusUser;
    Action := caFree;
  end;
end;
```

```
procedure TFadmin.Exit1Click(Sender:
TObject);
begin
  if MessageDlg('Anda Yakin Akan Keluar
Dari Aplikasi?', mtInformation,
[mbYes,mbNo],0) = mrYes then
  begin
    dm.Tuser.Open;
    dm.StatusLogin := False;
    dm.UpdateStatusUser;
    Application.Terminate;
  end;
end;

procedure TFadmin.PenjualanClick(Sender:
TObject);
begin

Application.CreateForm(TFpenjualan2,Fpenjua
lan2);
  Fpenjualan2.ShowModal;
end;

procedure TFadmin.Entang1Click(Sender:
TObject);
begin
   Ftentang.Show;
end;

procedure TFadmin.RPembelianClick(Sender:
TObject);
begin

Application.CreateForm(TfrmReturPembelian,
frmReturPembelian);
  frmReturPembelian.ShowModal;
end;

procedure TFadmin.HutangClick(Sender:
TObject);
begin

Application.CreateForm(TfrmBayarHutang,fr
mBayarHutang);
  frmBayarHutang.ShowModal;
end;

procedure TFadmin.RPenjualanClick(Sender:
TObject);
begin

Application.CreateForm(TfrmReturPenjualan,f
rmReturPenjualan);
  frmReturPenjualan.ShowModal;
end;

procedure TFadmin.PiutangClick(Sender:
TObject);
begin

Application.CreateForm(TfrmBayarPiutang,fr
mBayarPiutang);
  frmBayarPiutang.ShowModal;
end;

procedure
TFbarang.SpeedButton1Click(Sender:
TObject);
begin
   dm.Tbarang.first;
end;

procedure
TFbarang.SpeedButton2Click(Sender:
TObject);
begin
   dm.Tbarang.prior;
   if dm.Tbarang.BOF then
       showmessage('anda sudah di awal data');
end;

procedure
TFbarang.SpeedButton3Click(Sender:
TObject);
begin
   dm.Tbarang.next;
   if dm.Tbarang.EOF then
       showmessage('anda sudah di akhir data');
end;

procedure
TFbarang.SpeedButton4Click(Sender:
TObject);
begin
    dm.Tbarang.last;
end;

procedure TFbarang.BitBtn1Click(Sender:
TObject);
begin
  dm.Tbarang.append;

Application.CreateForm(TFisibarang,Fisibaran
g);
       Fisibarang.showmodal;
       if Fisibarang.modalresult=mrok then
             dm.Tbarang.post
       else
             dm.Tbarang.cancel;
end;
```

```
procedure TFbarang.BitBtn2Click(Sender:
TObject);
begin
  dm.Tbarang.edit;

Application.CreateForm(TFisibarang,Fisibaran
g);
       Fisibarang.showmodal;
       if Fisibarang.modalresult=mrok then
             dm.Tbarang.post
       else
             dm.Tbarang.cancel;
end;

procedure TFbarang.BitBtn3Click(Sender:
TObject);
begin
  if messagedlg('anda yakin akan menghapus
data
barang?',mtconfirmation,[mbyes,mbno],0)=mr
Yes then
     dm.Tbarang.delete;
end;

procedure TFbarang.BitBtn6Click(Sender:
TObject);
begin
 close;
end;

procedure
TFbarang.RadioGroup1Click(Sender:
TObject);
begin
      if RadioGroup1.itemindex=0 then
             dm.Tbarang.indexname:="
       else
       if RadioGroup1.itemindex=1 then

dm.Tbarang.indexname:='Idxnama_brg';

end;

procedure TFbarang.Button1Click(Sender:
TObject);
var find:boolean;
begin
       Find:=dm.Tbarang.FindKey([Edit1.text]);
       if Find=false then
             showmessage('File not found');
end;

procedure TFbarang.UlangClick(Sender:
TObject);
begin
   Edit1.Clear;
end;

procedure TFbarang.FormCreate(Sender:
TObject);
begin
  dm.Tbarang.Close;
  dm.Tbarang.Open;
  dm.Tbarang.Filtered := False;
end;

procedure TFbarang.FormClose(Sender:
TObject; var Action: TCloseAction);
begin
  dm.Tbarang.Close;
end;

procedure TFbarang.FormShow(Sender:
TObject);
begin
  dm.Tbarang.Close;
  dm.Tbarang.Open;
end;

procedure
TfrmBayarHutang.Button1Click(Sender:
TObject);
begin
  if DBLookupComboBox1.Text = '' then
  begin
    ShowMessage('Suplier Masih Kosong!!!');
    DBLookupComboBox1.SetFocus;
  end
  else if (DBEdit2.Text = '') or
(dm.TBuku_Hutang.Fields[3].AsInteger = 0 )
then
  begin
    ShowMessage('Jumlah Bayar Tidak Boleh
Kosong atau Nol !!!');
    DBEdit2.SetFocus;
  end
  else
  begin
    try
      dm.TBuku_Hutang.Fields[1].AsDateTime
:= DateTimePicker1.Date;
      dm.TBuku_Hutang.Fields[5].AsInteger :=
saldo - dm.TBuku_Hutang.Fields[3].AsInteger;
      dm.TBuku_Hutang.Fields[6].AsString :=
'Pembayaran Hutang';
      dm.TBuku_Hutang.Post;
    except
    end;
    Close;
```

```
  end;
end;

procedure TfrmBayarHutang.saldohutang;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select saldo from buku_hutang
where id_supplier
='+QuotedStr(DBLookupComboBox1.KeyVal
ue)+' order by no_transaksi desc' );
    Open;
    saldo := dm.Q1.Fields[0].AsInteger;
    Label6.Caption := IntToStr(saldo);
  end;
end;

procedure
TfrmBayarHutang.DBLookupComboBox1Clos
eUp(Sender: TObject);
begin
  if DBLookupComboBox1.Text <> '' then
    saldohutang;
end;

procedure
TfrmBayarHutang.FormShow(Sender:
TObject);
begin
  dm.QSuplier.Open;
  dm.TBuku_Hutang.Close;
  dm.TBuku_Hutang.Open;
  dm.TBuku_Hutang.Append;
end;

procedure
TfrmBayarHutang.Button2Click(Sender:
TObject);
begin
  dm.TBuku_Hutang.Cancel;
end;

procedure
TfrmBayarPiutang.FormShow(Sender:
TObject);
begin
  dm.Tpelanggan.Open;
  dm.TBuku_Piutang.Close;
  dm.TBuku_Piutang.Open;
  dm.TBuku_Piutang.Append;
end;

procedure TfrmBayarPiutang.SaldoPiutang;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select saldo from buku_piutang
where id_plg
='+QuotedStr(DBLookupComboBox1.KeyVal
ue)+' order by no_transaksi desc' );
    Open;
    saldo := dm.Q1.Fields[0].AsInteger;
    Label6.Caption := IntToStr(saldo);
  end;
end;

procedure
TfrmBayarPiutang.DBLookupComboBox1Clo
seUp(Sender: TObject);
begin
  if DBLookupComboBox1.Text <> '' then
    SaldoPiutang;
end;

procedure
TfrmBayarPiutang.Button1Click(Sender:
TObject);
begin
  if DBLookupComboBox1.Text = '' then
  begin
    ShowMessage('Pelanggan Masih
Kosong!!!');
    DBLookupComboBox1.SetFocus;
  end
  else if (DBEdit2.Text = '') or
(dm.TBuku_Hutang.Fields[3].AsInteger = 0 )
then
  begin
    ShowMessage('Jumlah Bayar Tidak Boleh
Kosong atau Nol !!!');
    DBEdit2.SetFocus;
  end
  else
  begin
    try
      dm.TBuku_Hutang.Fields[1].AsDateTime
:= DateTimePicker1.Date;
      dm.TBuku_Hutang.Fields[5].AsInteger :=
saldo - dm.TBuku_Hutang.Fields[3].AsInteger;
      dm.TBuku_Piutang.Fields[6].AsString :=
'Pembayaran Piutang';
      dm.TBuku_Hutang.Post;
    except
    end;
    Close;
  end;
```

```
end;

procedure Tdm.DataModuleCreate(Sender:
TObject);
begin
  db.Connected;
end;

procedure
Tdm.Tdetail_pembelianAfterPost(DataSet:
TDataSet);
begin
  try
    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('update detail_pembelian set
harga_bandrol =
'+QuotedStr(dm.Tdetail_pembelianharga_band
rol1.AsString));
      SQL.Add(', harga_bayar = harga_bandrol -
(disc *
'+QuotedStr(dm.Tdetail_pembelianharga_band
rol1.AsString)+' / 100 )');
      SQL.Add(',sub_total = harga_bayar * (jatah
- retur) where kd_brg
='+QuotedStr(dm.Tdetail_pembeliankd_brg.As
String));
      SQL.Add(' and no_faktur =
'+QuotedStr(dm.Tdetail_pembelianno_faktur.A
sString));
      ExecSQL;
    end;
  except
  end;
   dm.Tdetail_pembelian.Refresh;
end;

procedure Tdm.UpdateStatusUser;
begin
  if StatusLogin = False then
  begin
    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('update user set st_aktif = 0
where username =
'+QuotedStr(dm.Quser.Fields[0].AsString));
      ExecSQL;
    end;
  end
  else if StatusLogin  then
  begin
    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('update user set st_aktif = 1
where username =
'+QuotedStr(dm.Quser.Fields[0].AsString));
      ExecSQL;
    end;
  end;
end;

procedure
Tdm.Tdetail_penjualanAfterPost(DataSet:
TDataSet);
begin
  try
    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('update detail_penjualan set
harga_jual =
'+QuotedStr(dm.Tdetail_penjualanhrg_jual.As
String));
      SQL.Add(',sub_total = harga_jual *
eksemplar where kd_brg
='+QuotedStr(dm.Tdetail_penjualankd_brg.As
String));
      SQL.Add(' and no_faktur =
'+QuotedStr(dm.Tdetail_penjualanno_faktur.A
sString));
      ExecSQL;
    end;
  except
  end;
   dm.Tdetail_penjualan.Refresh;
end;

procedure Tdm.TpenjualanAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
  TPenjualan.Fields[1].AsDateTime := Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select substring(no_faktur,4,4)
from penjualan where no_faktur like ' +
QuotedStr('%MTA%')+' order by no_faktur
desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
```

```
   TPenjualan.Fields[0].AsString :=
'MTA0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
//    Kode := DM.Q1.Fields[0].AsString;
   NoFaktur :=
StringReplace(Format('MTA%4d',[(StrToInt(d
m.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
   TPenjualan.Fields[0].AsString := NoFaktur
+
FormatDateTime('ddmmyy',TPenjualan.Fields[
1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure Tdm.TsupplierBeforePost(DataSet:
TDataSet);
var
  GetText, GetTextDb : String;
begin
  if Tsupplier.State in [dsinsert] then
  begin
    GetText :=
copy(Tsupplier.Fields[1].AsString,1,3);
    if GetText = '' then Exit;
    with Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('select id_supplier from supplier
');
      Active := True;
    end;
      GetTextDb :=
UpperCase(GetText)+StringReplace(Format('
%4d',[dm.Q1.RecordCount +1]),'
','0',[rfReplaceAll, rfIgnoreCase]);
UpperCase(GetText)+StringReplace(Format('
%4d',[1]),' ','0',[rfReplaceAll, rfIgnoreCase]);
    DM.Q1.Active := False;
    Tsupplier.Fields[0].AsString := GetTextDb;
  end;
end;

procedure
Tdm.TpelangganBeforePost(DataSet:
TDataSet);
var
  GetText, GetTextDb : String;
begin
  if dm.Tpelanggan.State in [dsinsert] then
  begin
    GetText :=
copy(Tpelanggan.Fields[1].AsString,1,3);
    if GetText = '' then Exit;
    with Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('select id_plg from pelanggan ');
      Active := True;
    end;
      GetTextDb :=
UpperCase(GetText)+StringReplace(Format('
%4d',[dm.Q1.RecordCount + 1]),'
','0',[rfReplaceAll, rfIgnoreCase]);
UpperCase(GetText)+StringReplace(Format('
%4d',[1]),' ','0',[rfReplaceAll, rfIgnoreCase]);
    DM.Q1.Active := False;
    Tpelanggan.Fields[0].AsString :=
GetTextDb;
  end;
end;

procedure Tdm.TbarangBeforePost(DataSet:
TDataSet);
var
  GetText, GetTextDb : String;
begin
  if dm.Tbarang.State in [dsinsert] then
  begin
    GetText :=
copy(Tbarang.Fields[1].AsString,1,4);
    if GetText = '' then Exit;
    with Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('select kd_brg from barang ');
      Active := True;
    end;
UpperCase(GetText)+StringReplace(Format('
%4d',[dm.Q1.RecordCount + 1]),'
','0',[rfReplaceAll, rfIgnoreCase]);
UpperCase(GetText)+StringReplace(Format('
%4d',[1]),' ','0',[rfReplaceAll, rfIgnoreCase]);
    DM.Q1.Active := False;
    Tbarangkd_brg.AsString := GetTextDb;
  end;
end;

procedure
Tdm.TpembelianAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
```

```
  Tpembelian.Fields[1].AsDateTime := Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select substring(no_faktur,5,4)
from pembelian where no_faktur like ' +
QuotedStr('%MTAB%')+' order by no_faktur
desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
    Tpembelian.Fields[0].AsString :=
'MTAB0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
StringReplace(Format('MTAB%4d',[(StrToInt(
dm.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
    Tpembelian.Fields[0].AsString := NoFaktur
+
FormatDateTime('ddmmyy',Tpembelian.Fields
[1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure
Tdm.TRPembelianAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
  TRPembelian.Fields[1].AsDateTime := Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select substring(no_retur,5,4)
from returbeli where no_retur like ' +
QuotedStr('%MTRB%')+' order by no_retur
desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
    TRPembelian.Fields[0].AsString :=
'MTRB0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
StringReplace(Format('MTRB%4d',[(StrToInt(
dm.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
    TRPembelian.Fields[0].AsString :=
NoFaktur +
FormatDateTime('ddmmyy',TRPembelian.Fiel
ds[1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure
Tdm.TBuku_HutangAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
  TBuku_Hutang.Fields[1].AsDateTime :=
Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select
substring(no_transaksi,5,4) from buku_hutang
where no_transaksi like ' +
QuotedStr('%MTAH%')+' order by
no_transaksi desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
    TBuku_Hutang.Fields[0].AsString :=
'MTAH0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
StringReplace(Format('MTAH%4d',[(StrToInt(
dm.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
    TBuku_Hutang.Fields[0].AsString :=
NoFaktur +
FormatDateTime('ddmmyy',TBuku_Hutang.Fie
lds[1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure
Tdm.TRPenjualanAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
  TRPenjualan.Fields[1].AsDateTime := Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select substring(no_retur,5,4)
from returjual where no_retur like ' +
QuotedStr('%MTRJ%')+' order by no_retur
desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
```

```
    TRPenjualan.Fields[0].AsString :=
'MTRJ0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
StringReplace(Format('MTRJ%4d',[(StrToInt(d
m.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
    TRPenjualan.Fields[0].AsString := NoFaktur
+
FormatDateTime('ddmmyy',TRPenjualan.Field
s[1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure
Tdm.TBuku_PiutangAfterInsert(DataSet:
TDataSet);
var NoFaktur : String;
begin
  TBuku_Piutang.Fields[1].AsDateTime :=
Now;
  Q1.Active := False;
  Q1.SQL.Clear;
  Q1.SQL.Add('select
substring(no_transaksi,5,4) from buku_piutang
where no_transaksi like ' +
QuotedStr('%MTAB%')+' order by
no_transaksi desc');
  Q1.Active :=  True;
  if Q1.RecordCount = 0 then
  begin
    TBuku_Piutang.Fields[0].AsString :=
'MTAB0001' +
FormatDateTime('ddmmyy',now);
  end
  else
  begin
StringReplace(Format('MTAB%4d',[(StrToInt(
dm.Q1.Fields[0].AsString)+1)]),'
','0',[rfReplaceAll, rfIgnoreCase]);
    TBuku_Piutang.Fields[0].AsString :=
NoFaktur +
FormatDateTime('ddmmyy',TBuku_Piutang.Fi
elds[1].AsDateTime);
  end;
  Q1.Active := False;
end;

procedure TFEditUser.Button1Click(Sender:
TObject);
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from user where
username = '+QuotedStr(DBEdit1.Text)+' and
pass_user = MD5('+QuotedStr(Edit1.Text)+')');
    Open;
  end;
  if dm.Q1.RecordCount > 0 then
  begin
    if  Length(Edit2.Text) >= 5 then
    begin
      if (Edit2.Text = Edit3.Text) then
        ModalResult := mrOk
      else
        MessageDlg('Confirm Pasword Tidak
Sama Dengan Password
Baru',mtError,[mbok],0);
    end
    else
        MessageDlg('Pasword Baru Minimal 5
Karakter',mtError,[mbok],0);
  end
  else
    MessageDlg('Pasword Lama Anda
Salah',mtError,[mbok],0);
end;

procedure TFisibarang.Button1Click(Sender:
TObject);
var st_duplikat, st_sama : Integer;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select count(*) from barang
where nama_brg =
'+QuotedStr(DBEdit2.Text));
    SQL.Add('and terbitan =
'+QuotedStr(dm.Tbarangterbitan.AsString));
    SQL.Add('and id_supplier =
'+QuotedStr(dm.Tbarangid_supplier.AsString))
;
    SQL.Add('and harga_bandrol =
'+QuotedStr(dm.Tbarangharga_bandrol.AsStrin
g));
    SQL.Add('and kel_barang =
'+QuotedStr(dm.Tbarangkel_barang.AsString))
;
    Open;
    st_duplikat := dm.Q1.Fields[0].AsInteger;
  end;
  with dm.Q1 do
  begin
```

```
    Close;
    SQL.Clear;
    SQL.Add('select count(*) from barang
where nama_brg =
'+QuotedStr(DBEdit2.Text));
    SQL.Add('and kel_barang =
'+QuotedStr(dm.Tbarangkel_barang.AsString))
;
    SQL.Add('and terbitan =
'+QuotedStr(dm.Tbarangterbitan.AsString));
    Open;
    st_sama := dm.Q1.Fields[0].AsInteger;
  end;
  if DBEdit2.Text = '' then
  begin
     ShowMessage('Nama Barang Tidak Boleh
Kosong');
     DBEdit2.SetFocus;
  end
  else if DBComboBox1.Text = '' then
  begin
     ShowMessage('Kelompok Barang Tidak
Boleh Kosong');
     DBComboBox1.SetFocus;
  end
  else if dm.Tbarangterbitan.IsNull then
  begin
     ShowMessage('Terbitan Barang Tidak
Boleh Kosong');
     DBRadioGroup1.SetFocus;
  end
  else if DBLookupComboBox1.Text = '' then
  begin
     ShowMessage('Suplier Barang Tidak Boleh
Kosong');
     DBLookupComboBox1.SetFocus;
  end
  else if DBEdit7.Text = '' then
  begin
     ShowMessage('Harga Bandrol Barang
Tidak Boleh Kosong');
     DBEdit7.SetFocus;
  end
  else if st_duplikat > 0 then
  begin
    ShowMessage('Duplicate Entry !!! ');
  end
  else if st_sama > 0 then
  begin
    ShowMessage('Tipe Barang Sudah Ada !!!
');
  end
  else
    ModalResult := mrOk;
end;

procedure TFisibarang.FormShow(Sender:
TObject);
begin
  dm.QListSuplier.Open;
end;

procedure
TFisipelanggan.Button1Click(Sender:
TObject);
var st_duplikat : Integer;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select count(*) from pelanggan
where nama_plg =
'+QuotedStr(dm.Tpelanggannama_plg.AsStrin
g));
    SQL.Add(' and jenis
='+QuotedStr(dm.Tpelangganjenis.AsString));
    SQL.Add(' and alamat
='+QuotedStr(dm.Tpelangganalamat.AsString))
;
    SQL.Add(' and contact_person =
'+QuotedStr(dm.Tpelanggancontact_person.As
String));
    SQL.Add(' and telepon =
'+QuotedStr(dm.Tpelangganttelepon.AsString));
    SQL.Add(' and fax =
'+QuotedStr(dm.Tpelangganfax.AsString));
    Open;
    st_duplikat := Fields[0].AsInteger;
  end;
  if dm.Tpelanggannama_plg.IsNull then
  begin
    ShowMessage('Nama Pelanggan Tidak
Boleh Kosong');
    DBEdit2.SetFocus;
  end
  else if dm.Tpelangganjenis.IsNull then
  begin
    ShowMessage('Jenis Pelanggan Tidak Boleh
Kosong');
    DBRadioGroup1.SetFocus;
  end
  else if DBEdit4.Text = '' then
  begin
    ShowMessage('Alamat Pelanggan Tidak
Boleh Kosong');
    DBEdit4.SetFocus;
  end
  else if DBEdit5.Text = '' then
  begin
```

```
    ShowMessage('Contact Person Pelanggan
Tidak Boleh Kosong');
    DBEdit5.SetFocus;
  end
  else if DBEdit6.Text = '' then
  begin
    ShowMessage('Telepon Pelanggan Tidak
Boleh Kosong');
    DBEdit6.SetFocus;
  end
  else if st_duplikat > 0 then
  begin
    ShowMessage('Duplikat Entry !!! ');
  end
  else
  begin
    ModalResult := mrOk;
  end;
end;

procedure TFisisupplier.Button1Click(Sender:
TObject);
var st_duplikat : Integer;
begin
{  if dm.Tsupplierid_supplier.IsNull then
  begin
    ShowMessage('Id Suplier Harus DiIsi!!!');
    DBEdit1.SetFocus;
  end}
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select count(*) from supplier
where nama_supplier =
'+QuotedStr(dm.Tsuppliernama_supplier.AsStr
ing));
    SQL.Add(' and tipe
='+QuotedStr(dm.Tsuppliertipe.AsString));
    SQL.Add(' and alamat
='+QuotedStr(dm.Tsupplieralamat.AsString));
    SQL.Add(' and kota =
'+QuotedStr(dm.Tsupplierkota.AsString));
    SQL.Add(' and contact_person =
'+QuotedStr(dm.Tsuppliercontact_person.AsStr
ing));
    SQL.Add(' and telepon =
'+QuotedStr(dm.Tsuppliertelepon.AsString));
    SQL.Add(' and fax =
'+QuotedStr(dm.Tsupplierfax.AsString));
    Open;
    st_duplikat := Fields[0].AsInteger;
  end;

  if DBEdit2.Text = '' then
  begin
    ShowMessage('Nama Suplier Harus
DiIsi!!!');
    DBEdit2.SetFocus;
  end
  else if dm.Tsuppliertipe.IsNull then
  begin
    ShowMessage('Tipe Suplier Harus DiIsi!!!');
    DBRadioGroup1.SetFocus;
  end
  else if DBEdit4.Text = '' then
  begin
    ShowMessage('Alamat Suplier Harus
DiIsi!!!');
    DBEdit4.SetFocus;
  end
  else if DBEdit5.Text = '' then
  begin
    ShowMessage('Kota Suplier Harus DiIsi!!!');
    DBEdit5.SetFocus;
  end
  else if DBEdit6.Text = '' then
  begin
    ShowMessage('Contact Person Suplier
Harus DiIsi!!!');
    DBEdit6.SetFocus;
  end
  else if DBEdit7.Text = '' then
  begin
    ShowMessage('Telepon Suplier Harus
DiIsi!!!');
    DBEdit7.SetFocus;
  end
  else if st_duplikat > 0 then
  begin
    ShowMessage('Duplikat Entry !!! ');
  end
  else
  begin
    ModalResult := mrOk;
  end;
end;

procedure TFisiuser.Button1Click(Sender:
TObject);
begin
  if  DBEdit1.Text = '' then
  begin
     ShowMessage('User Name Tidak Boleh
Kosong');
      DBEdit1.SetFocus;
  end
  else if DBComboBox1.Text = '' then
  begin
```

```
    ShowMessage('Akses Tidak Boleh
Kosong');
    DBComboBox1.SetFocus;
  end
  else if DBEdit2.Text = '' then
  begin
    ShowMessage('Password Tidak Boleh
Kosong');
    DBEdit2.SetFocus;
  end
  else if Edit1.Text = '' then
  begin
    ShowMessage('Confirm Password Tidak
Boleh Kosong');
    Edit1.SetFocus;
  end
  else if Length(DBEdit1.Text) < 5  then
  begin
    ShowMessage('Nama User Minimal 5
Karakter ');
    DBEdit1.SetFocus;
  end
  else if Length(DBEdit2.Text) < 5 then
  begin
    ShowMessage('Password Minimal 5
Karakter');
    DBEdit2.SetFocus;
  end
  else
  begin
    if (DBEdit2.Text = Edit1.Text) then
      ModalResult := mrOk
    else
      MessageDlg('1',mtError,[mbok],0);
  end;
end;

procedure TFlapbrg.Button3Click(Sender:
TObject);
begin
close;
end;

procedure TFlapbrg.Button2Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
  begin
    lapbarang.Qlapbarang.Active := False;
    lapbarang.Qlapbarang.Filtered := False;
    lapbarang.Qlapbarang.Filter := 'kel_barang =
'+QuotedStr(dm.QKel_Barang.Fields[0].AsStri
ng);
    lapbarang.Qlapbarang.Filtered := True;
    lapbarang.Qlapbarang.Active := True;
  end
  else
  begin
    lapbarang.Qlapbarang.Filtered := False;
    lapbarang.Qlapbarang.Active := True;
  end;
  if (Sender as TComponent).Tag = 0 then
    lapbarang.Preview
  else if (Sender as TComponent).Tag = 1 then
    lapbarang.Print;
end;

procedure
TFlapbrg.RadioGroup1Click(Sender: TObject);
begin
      if RadioGroup1.itemindex=0 then
            dm.Tbarang.indexname:=''
       else
       if RadioGroup1.itemindex=1 then

dm.Tbarang.indexname:='Idxnama_brg';

end;

procedure TFlapbrg.FormCreate(Sender:
TObject);
begin
  dm.Tbarang.Close;
  dm.Tbarang.Open;
  dm.TStok.Close;
  dm.TStok.Open;
  dm.QKel_Barang.Close;
  dm.QKel_Barang.Open;
end;

procedure TFlapbrg.FormShow(Sender:
TObject);
begin
  dm.TStok_Master.Close;
  dm.TStok_Master.Open;
  dm.TStok.Close;
  dm.TStok.Open;
end;

procedure
TFlapbrg.DBLookupComboBox1CloseUp(Sen
der: TObject);
begin
  if CheckBox1.Checked then
  begin
    dm.TStok_Master.Filtered := False;
    dm.TStok_Master.Filter := 'kel_barang =
'+QuotedStr(dm.QKel_Barang.Fields[0].AsStri
ng);
    dm.TStok_Master.Filtered := True;
```

```
  end
  else
    dm.TStok_Master.Filtered := false;
end;

procedure TFlapbrg.CheckBox1Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
    DBLookupComboBox1.Visible := True
  else
  begin
    DBLookupComboBox1.Visible := False;
    dm.TStok_Master.Filtered := False;
  end;
end;

procedure TFlaphutang.Button3Click(Sender:
TObject);
begin
  Close;
end;

procedure TFlaphutang.Button2Click(Sender:
TObject);
begin
   laphutang.Qlaphutang.Close;
   laphutang.Qlaphutang.SQL.Clear;

laphutang.Qlaphutang.SQL.Add(dm.Qhutang.S
QL.GetText);
   laphutang.Qlaphutang.Open;
   if CheckBox1.Checked then
   begin
     laphutang.Qlaphutang.Filtered := False;
     laphutang.Qlaphutang.Filter := 'id_supplier
=
'+QuotedStr(dm.Tsupplierid_supplier.AsString
);
     laphutang.Qlaphutang.Filtered := True;
   end
   else
     laphutang.Qlaphutang.Filtered := false;
   if (Sender as TComponent).Tag = 0 then
   begin
     laphutang.Prepare;
     laphutang.Print;
   end
   else
   begin
     laphutang.Prepare;
     laphutang.Preview;
   end;
end;

procedure TFlaphutang.ListTahun;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select distinct year(tgl) from
buku_hutang ');
    Open;
  end;
  dm.Q1.First;
  ComboBox1.Items.Clear;
  ComboBox1.Items.Clear;
  while dm.Q1.Eof  = false  do
  begin

ComboBox1.Items.Add(dm.Q1.Fields[0].AsStr
ing);
    dm.Q1.Next;
  end;
end;

procedure
TFlaphutang.CheckBox1Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
    DBLookupComboBox1.Visible := True
  else
    DBLookupComboBox1.Visible := False;
end;

procedure TFlaphutang.Button4Click(Sender:
TObject);
begin
  with dm.Qhutang do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select a.*, b.nama_supplier from
buku_hutang a join supplier b on a.id_supplier
= b.id_supplier ');
    SQL.Add('where year(a.tgl) =
'+QuotedStr(ComboBox1.Text)+' ');
    Open;
  end;
  if CheckBox1.Checked then
  begin
    dm.Qhutang.Filtered := False;
    dm.Qhutang.Filter := 'id_supplier =
'+QuotedStr(dm.Tsupplierid_supplier.AsString
);
    dm.Qhutang.Filtered := True;
  end
  else
```

```
    dm.Qhutang.Filtered := false;
end;

procedure TFlaphutang.FormCreate(Sender:
TObject);
begin
  ListTahun;
  dm.Tsupplier.Open;
end;

procedure
TFlappelanggan.Button1Click(Sender:
TObject);
begin
   lappelanggan.Qlappelanggan.Active := False;
   lappelanggan.Qlappelanggan.Active := True;
   lappelanggan.Print;
end;

procedure
TFlappelanggan.Button2Click(Sender:
TObject);
begin
   lappelanggan.Qlappelanggan.Active := False;
   lappelanggan.Qlappelanggan.Active := True;
   lappelanggan.Preview;
end;

procedure
TFlappelanggan.Button3Click(Sender:
TObject);
begin
close;
end;

procedure
TFlappelanggan.RadioGroup1Click(Sender:
TObject);
begin
     if RadioGroup1.itemindex=0 then
            dm.Tpelanggan.indexname:="
       else
       if RadioGroup1.itemindex=1 then

dm.Tpelanggan.indexname:='Idxnama_plg';
end;

procedure
TFlappelanggan.FormCreate(Sender: TObject);
begin
  dm.Tpelanggan.Close;
  dm.Tpelanggan.Open;
end;

procedure TFlappelanggan.FormShow(Sender:
TObject);
begin
  dm.Tpelanggan.Close;
  dm.Tpelanggan.Open;
end;

procedure
TFlappembelian.Button2Click(Sender:
TObject);
begin
   lappembelian.Qpembelian.Close;
   lappembelian.Qpembelian.SQL.Clear;

lappembelian.Qpembelian.SQL.Add(dm.Qpem
belian.SQL.GetText);
   lappembelian.Qpembelian.Open;

   lappembelian.QDetail_Pembelian.Close;
   lappembelian.QDetail_Pembelian.SQL.Clear;

lappembelian.QDetail_Pembelian.SQL.Add(d
m.QDetail_Pembelian.SQL.GetText);
   lappembelian.QDetail_Pembelian.Open;
   if (Sender as TComponent).Tag = 0 then
   begin
     lappembelian.Prepare;
     lappembelian.Print;
   end
   else
   begin
     lappembelian.Prepare;
     lappembelian.Preview;
   end;
end;

procedure
TFlappembelian.Button3Click(Sender:
TObject);
begin
close;
end;

procedure TFlappembelian.ReportBulanan;
begin
  with dm.Qpembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
supplier b on  a.id_supplier = b.id_supplier');
    SQL.Add(' where month(tgl_beli) >=
'+QuotedStr(IntToStr(ComboBox1.ItemIndex
+ 1))+' and year(tgl_beli) =
'+QuotedStr(ComboBox2.Text)+' ');
```

```
    Open;
  end;
  with dm.QDetail_Pembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
detail_pembelian b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where month(tgl_beli) >=
'+QuotedStr(IntToStr(ComboBox1.ItemIndex
+ 1))+' and year(tgl_beli) =
'+QuotedStr(ComboBox2.Text)+' ');
    dm.QDetail_Pembelian.DataSource :=
dm.DTpembelian2;
    SQL.Add(' and b.no_faktur =:no_faktur');
    dm.QDetail_Pembelian.DataSource :=
dm.DTpembelian2;
    Open;
  end;
end;

procedure TFlappembelian.ReportHarian;
begin
  with dm.Qpembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
supplier b on  a.id_supplier = b.id_supplier');
    SQL.Add(' where date(tgl_beli) >=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker1.Date))+' and
date(tgl_beli) <=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker2.Date))+' ');
    Open;
  end;
  with dm.QDetail_Pembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
detail_pembelian b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where date(tgl_beli) >=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker1.Date))+' and
date(tgl_beli) <=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker2.Date))+' ');
    SQL.Add(' and b.no_faktur =:no_faktur');
    Open;
  end;
end;

procedure TFlappembelian.ReportTahunan;
begin
  with dm.Qpembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
supplier b on  a.id_supplier = b.id_supplier');
    SQL.Add(' where year(tgl_beli) =
'+QuotedStr(ComboBox3.Text)+' ');
    Open;
  end;
  with dm.QDetail_Pembelian do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from pembelian a join
detail_pembelian b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where year(tgl_beli) =
'+QuotedStr(ComboBox3.Text)+' ');
    SQL.Add(' and b.no_faktur =:no_faktur');
    Open;
  end;
end;

procedure
TFlappembelian.Button4Click(Sender:
TObject);
begin
  if RadioButton1.Checked then
  begin
    ReportHarian;
  end
  else if RadioButton2.Checked then
  begin
    ReportBulanan;
  end
  else if RadioButton3.Checked then
  begin
    ReportTahunan;
  end;
  if CheckBox1.Checked then
  begin
    dm.Qpembelian.Filtered := False;
    dm.Qpembelian.Filter := 'id_supplier =
'+QuotedStr(dm.Tsupplierid_supplier.AsString
);
```

```
    dm.Qpembelian.Filtered := True;
  end
  else
    dm.Qpembelian.Filtered := False;
end;

procedure
TFlappembelian.RadioButton1Click(Sender:
TObject);
begin
  if RadioButton1.Checked then
  begin
    DateTimePicker1.Visible := True;
    DateTimePicker2.Visible := True;
    ComboBox1.Visible := False;
    ComboBox2.Visible := False;
    ComboBox3.Visible := False;
    Label1.Caption := 'Dari';
    Label2.Caption := 'S.D.';
    Label2.Visible := True;
  end
  else if RadioButton2.Checked then
  begin
    DateTimePicker1.Visible := False;
    DateTimePicker2.Visible := False;
    ComboBox1.Visible := True;
    ComboBox2.Visible := True;
    ComboBox3.Visible := False;
    Label2.Visible := True;
    Label1.Caption := 'Bulan';
    Label2.Caption := 'Tahun';
  end
  else if RadioButton3.Checked then
  begin
    DateTimePicker1.Visible := False;
    DateTimePicker2.Visible := False;
    ComboBox1.Visible := False;
    ComboBox2.Visible := False;
    ComboBox3.Visible := True;
    Label1.Caption := 'Tahun';
    Label2.Caption := '';
    Label2.Visible := False;
  end;
end;

procedure
TFlappembelian.FormCreate(Sender: TObject);
begin
   RadioButton1Click(Sender);
   dm.Tsupplier.Open;
end;

procedure
TFlappembelian.CheckBox1Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
    DBLookupComboBox1.Visible := True
  else
    DBLookupComboBox1.Visible := False;
end;

procedure
TFlappembelian.FormActivate(Sender:
TObject);
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select distinct year(tgl_beli) from
pembelian ');
    Open;
  end;
  dm.Q1.First;
  ComboBox2.Items.Clear;
  ComboBox3.Items.Clear;
  while dm.Q1.Eof  = false  do
  begin

ComboBox2.Items.Add(dm.Q1.Fields[0].AsStr
ing);

ComboBox3.Items.Add(dm.Q1.Fields[0].AsStr
ing);
    dm.Q1.Next;
  end;
end;

procedure
TFlappenjualan.Button3Click(Sender:
TObject);
begin
close;
end;

procedure
TFlappenjualan.Button2Click(Sender:
TObject);
begin
   lappenjualan.QPenjualan.Close;
   lappenjualan.QPenjualan.SQL.Clear;

lappenjualan.QPenjualan.SQL.Add(dm.QPenju
alan.SQL.GetText);
   lappenjualan.QPenjualan.Open;

   lappenjualan.QDetail_Penjualan.Close;
   lappenjualan.QDetail_Penjualan.SQL.Clear;
```

```
lappenjualan.QDetail_Penjualan.SQL.Add(dm.
QDetail_Penjualan.SQL.GetText);
  lappenjualan.QDetail_Penjualan.Open;
  if (Sender as TComponent).Tag = 0 then
  begin
    lappenjualan.Prepare;
    lappenjualan.Print;
  end
  else
  begin
    lappenjualan.Prepare;
    lappenjualan.Preview;
  end;

end;

procedure
TFlappenjualan.RadioButton1Click(Sender:
TObject);
begin
 if RadioButton1.Checked then
 begin
   DateTimePicker1.Visible := True;
   DateTimePicker2.Visible := True;
   ComboBox1.Visible := False;
   ComboBox2.Visible := False;
   ComboBox3.Visible := False;
   Label1.Caption := 'Dari';
   Label2.Caption := 'S.D.';
   Label2.Visible := True;
 end
 else if RadioButton2.Checked then
 begin
   DateTimePicker1.Visible := False;
   DateTimePicker2.Visible := False;
   ComboBox1.Visible := True;
   ComboBox2.Visible := True;
   ComboBox3.Visible := False;
   Label2.Visible := True;
   Label1.Caption := 'Bulan';
   Label2.Caption := 'Tahun';
 end
 else if RadioButton3.Checked then
 begin
   DateTimePicker1.Visible := False;
   DateTimePicker2.Visible := False;
   ComboBox1.Visible := False;
   ComboBox2.Visible := False;
   ComboBox3.Visible := True;
   Label1.Caption := 'Tahun';
   Label2.Caption := '';
   Label2.Visible := False;
 end;
end;

procedure TFlappenjualan.ReportBulanan;
begin
  with dm.Qpenjualan do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from penjualan a join
pelanggan b on a.id_plg = b.id_plg ');
    SQL.Add(' where month(tgl_jual) >=
'+QuotedStr(IntToStr(ComboBox1.ItemIndex
+ 1))+' and year(tgl_jual) =
'+QuotedStr(ComboBox2.Text)+' ');
    Open;
  end;
  with dm.QDetail_penjualan do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from penjualan a join
detail_penjualan b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where month(tgl_jual) >=
'+QuotedStr(IntToStr(ComboBox1.ItemIndex
+ 1))+' and year(tgl_jual) =
'+QuotedStr(ComboBox2.Text)+' ');
    SQL.Add(' and b.no_faktur =:no_faktur');
    Open;
  end;
end;

procedure TFlappenjualan.ReportHarian;
begin
  with dm.Qpenjualan do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from penjualan a join
pelanggan b on a.id_plg = b.id_plg ');
    SQL.Add(' where date(tgl_jual) >=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker1.Date))+' and
date(tgl_jual) <=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker2.Date))+' ');
    Open;
  end;
  with dm.QDetail_penjualan do
  begin
    Close;
    SQL.Clear;
```

```
    SQL.Add('select * from penjualan a join
detail_penjualan b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where date(tgl_jual) >=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker1.Date))+' and
date(tgl_jual) <=
'+QuotedStr(FormatDateTime('YYYY-MM-
DD',DateTimePicker2.Date))+' ');
    SQL.Add(' and b.no_faktur =:no_faktur');
    Open;
  end;
end;

procedure TFlappenjualan.ReportTahunan;
begin
  with dm.Qpenjualan do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from penjualan a join
pelanggan b on a.id_plg = b.id_plg ');
    SQL.Add(' where year(tgl_jual) =
'+QuotedStr(ComboBox3.Text)+' ');
    Open;
  end;
  with dm.QDetail_penjualan do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from penjualan a join
detail_penjualan b on a.no_faktur =
b.no_faktur');
    SQL.Add('join barang c on b.kd_brg =
c.kd_brg ');
    SQL.Add(' where year(tgl_jual) =
'+QuotedStr(ComboBox3.Text)+' ');
    SQL.Add(' and b.no_faktur =:no_faktur');
    Open;
  end;
end;

procedure
TFlappenjualan.CheckBox1Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
    DBLookupComboBox1.Visible := True
  else
    DBLookupComboBox1.Visible := False;
end;

procedure
TFlappenjualan.Button4Click(Sender:
TObject);
begin
  if RadioButton1.Checked then
  begin
    ReportHarian;
  end
  else if RadioButton2.Checked then
  begin
    ReportBulanan;
  end
  else if RadioButton3.Checked then
  begin
    ReportTahunan;
  end;
  if CheckBox1.Checked then
  begin
    dm.QPenjualan.Filtered := False;
    dm.QPenjualan.Filter := 'id_plg =
'+QuotedStr(dm.Tpelangganid_plg.AsString);
    dm.QPenjualan.Filtered := True;
  end;
end;

procedure TFlappenjualan.FormCreate(Sender:
TObject);
begin
  RadioButton1Click(Sender);
  ListTahun;
end;

procedure TFlappenjualan.ListTahun;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select distinct year(tgl_jual) from
penjualan ');
    Open;
  end;
  ComboBox2.Items.Clear;
  ComboBox3.Items.Clear;
  while dm.Q1.Eof  = false  do
  begin

ComboBox2.Items.Add(dm.Q1.Fields[0].AsStr
ing);

ComboBox3.Items.Add(dm.Q1.Fields[0].AsStr
ing);
    dm.Q1.Next;
  end;
end;
```

```
procedure TFlappiutang.FormCreate(Sender:
TObject);
begin
  dm.Tpelanggan.Open;
end;

procedure TFlappiutang.ListTahun;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select distinct year(tgl) from
buku_piutang ');
    Open;
  end;
  dm.Q1.First;
  ComboBox1.Items.Clear;
  ComboBox1.Items.Clear;
  while dm.Q1.Eof  = false  do
  begin

ComboBox1.Items.Add(dm.Q1.Fields[0].AsStr
ing);
    dm.Q1.Next;
  end;
end;

procedure TFlappiutang.Button4Click(Sender:
TObject);
begin
  with dm.Qpiutang do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select a.*, b.nama_plg from
buku_piutang a join pelanggan b on a.id_plg =
b.id_plg ');
    SQL.Add('where year(a.tgl) =
'+QuotedStr(ComboBox1.Text));
    Open;
  end;
  if CheckBox1.Checked then
  begin
    dm.Qpiutang.Filtered := False;
    dm.Qpiutang.Filter := 'id_plg =
'+QuotedStr(dm.Tpelangganid_plg.AsString);
    dm.Qpiutang.Filtered := True;
  end
  else
    dm.Qpiutang.Filtered := false;
end;

procedure TFlappiutang.FormActivate(Sender:
TObject);
begin
  ListTahun;
end;

procedure
TFlappiutang.CheckBox1Click(Sender:
TObject);
begin
  if CheckBox1.Checked then
    DBLookupComboBox1.Visible := True
  else
    DBLookupComboBox1.Visible := False;
end;


procedure TFlappiutang.Button2Click(Sender:
TObject);
begin
  lappiutang.Qlappiutang.Close;
  lappiutang.Qlappiutang.SQL.Clear;

lappiutang.Qlappiutang.SQL.Add(dm.Qpiutang
.SQL.GetText);
  lappiutang.Qlappiutang.Open;
  if CheckBox1.Checked then
  begin
    lappiutang.Qlappiutang.Filtered := False;
    lappiutang.Qlappiutang.Filter := 'id_plg =
'+QuotedStr(dm.Tpelangganid_plg.AsString);
    lappiutang.Qlappiutang.Filtered := True;
  end
  else
    lappiutang.Qlappiutang.Filtered := false;
  if (Sender as TComponent).Tag = 0 then
  begin
    lappiutang.Prepare;
    lappiutang.Print;
  end
  else
  begin
    lappiutang.Prepare;
    lappiutang.Preview;
  end;
end;

procedure TFlappiutang.Button3Click(Sender:
TObject);
begin
  Close;
end;

procedure TFlapsupplier.Button3Click(Sender:
TObject);
```

```
begin
close;
end;

procedure TFlapsupplier.Button2Click(Sender:
TObject);
begin
  lapsupplier.Qlapsupplier.Active := False;
  lapsupplier.Qlapsupplier.Active := True;
  lapsupplier.Preview;
end;

procedure TFlapsupplier.Button1Click(Sender:
TObject);
begin
  lapsupplier.Qlapsupplier.Active := False;
  lapsupplier.Qlapsupplier.Active := True;
  lapsupplier.Print;
end;

procedure
TFlapsupplier.RadioGroup1Click(Sender:
TObject);
begin
      if RadioGroup1.itemindex=0 then
            dm.Tsupplier.indexname:="
       else
       if RadioGroup1.itemindex=1 then

dm.Tsupplier.indexname:='Idxnama_supplier';
end;

procedure TFlapsupplier.FormCreate(Sender:
TObject);
begin
  dm.Tsupplier.Close;
  dm.Tsupplier.Open;
end;

procedure TFlapsupplier.FormShow(Sender:
TObject);
begin
  dm.Tsupplier.Close;
  dm.Tsupplier.Open;
end;

procedure TFlogin.loginClick(Sender:
TObject);
begin
  with dm.Quser do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select * from user where
username = '+QuotedStr(Eusername.Text)+'
and pass_user =
MD5('+QuotedStr(Epassword.Text)+')');
    dm.Quser.Open;
  end;
  if dm.Quser.RecordCount > 0 then
  begin
    dm.StatusLogin := True;
    Fadmin.akses :=
dm.Quser.Fields[2].AsString;
    Fadmin.user :=
dm.Quser.Fields[0].AsString;
    Eusername.Text := '';
    Epassword.Text := '';
    dm.UpdateStatusUser;
    ModalResult := mrOk;
  end
  else
    MessageDlg('Password atau username
salah!', mtInformation, [mbOK], 0);
end;


procedure TFlogin.Button1Click(Sender:
TObject);
begin
  ModalResult :=  mrCancel;
end;

procedure
TFlogin.EusernameKeyPress(Sender: TObject;
var Key: Char);
begin
  if key =#13 then
    Epassword.SetFocus;
end;

procedure
TFlogin.EpasswordKeyPress(Sender: TObject;
var Key: Char);
begin
  if key =#13 then
    loginClick(Sender);
end;

procedure
TFpelanggan.SpeedButton1Click(Sender:
TObject);
begin
  dm.Tpelanggan.first;
end;

procedure
TFpelanggan.SpeedButton2Click(Sender:
TObject);
begin
```

```
  dm.Tpelanggan.prior;
  if dm.Tpelanggan.BOF then
      showmessage('anda sudah di awal data');
end;

procedure
TFpelanggan.SpeedButton3Click(Sender:
TObject);
begin
  dm.Tpelanggan.next;
  if dm.Tpelanggan.EOF then
      showmessage('anda sudah di akhir data');
end;

procedure
TFpelanggan.SpeedButton4Click(Sender:
TObject);
begin
  dm.Tpelanggan.last;
end;

procedure TFpelanggan.BitBtn1Click(Sender:
TObject);
begin
  dm.Tpelanggan.append;

Application.CreateForm(TFisipelanggan,Fisipe
langgan);
      Fisipelanggan.showmodal;
      if Fisipelanggan.modalresult=mrok then
            dm.Tpelanggan.post
      else
            dm.Tpelanggan.cancel;
end;

procedure TFpelanggan.BitBtn2Click(Sender:
TObject);
begin
  dm.Tpelanggan.edit;

Application.CreateForm(TFisipelanggan,Fisipe
langgan);
      Fisipelanggan.showmodal;
      if Fisipelanggan.modalresult=mrok then
            dm.Tpelanggan.post
      else
            dm.Tpelanggan.cancel;
end;

procedure TFpelanggan.BitBtn3Click(Sender:
TObject);
begin
 if messagedlg('anda yakin akan menghapus
data
pelanggan?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
     dm.Tpelanggan.delete;
end;

procedure TFpelanggan.BitBtn6Click(Sender:
TObject);
begin
  Close;
end;

procedure
TFpelanggan.RadioGroup1Click(Sender:
TObject);
begin
      if RadioGroup1.itemindex=0 then
            dm.Tpelanggan.indexname:="
       else
       if RadioGroup1.itemindex=1 then

dm.Tpelanggan.indexname:='Idxnama_plg';
end;

procedure TFpelanggan.Button1Click(Sender:
TObject);
var find:boolean;
begin

Find:=dm.Tpelanggan.FindKey([Edit1.text]);
      if Find=false then
            showmessage('File not found');

end;

procedure TFpelanggan.UlangClick(Sender:
TObject);
begin
   Edit1.Clear;
end;

procedure TFpelanggan.FormShow(Sender:
TObject);
begin
  dm.Tpelanggan.Close;
  dm.Tpelanggan.Open;
end;

procedure TFpembelian2.BitBtn1Click(Sender:
TObject);
var saldo : Integer;
begin
  if DBEdit5.Text = '' then
  begin
    ShowMessage('No Faktur Dari Suplier Tidak
Boleh Kosong !!!');
```

```
   DBEdit5.SetFocus;
  end
  else if DBLookupComboBox1.Text = '' then
  begin
   ShowMessage(' Suplier Tidak Boleh Kosong
!!!');
   DBLookupComboBox1.SetFocus;
  end
  else if dm.Tdetail_pembelian.RecordCount =
0 then
  begin
   ShowMessage(' Data Barang Yang Dibeli
Masih Kosong !!!');
   DBLookupComboBox1.SetFocus;
  end
  else
  begin
   dm.Tdetail_pembelian.First;
   while dm.Tdetail_pembelian.EOF=false
   do
   begin
    with dm.Q2 do
    begin
     Close;
     SQL.Clear;
     SQL.Add('select * from barang_detail
where kd_brg
='+QuotedStr(dm.Tdetail_pembeliankd_brg.As
String));
     SQL.Add(' and edisi
='+QuotedStr(dm.Tdetail_pembelianedisi.AsSt
ring));
     Open;
    end;
    if dm.Q2.RecordCount > 0 then
    begin
     with dm.Q1 do
     begin
      Close;
      SQL.Clear;
      SQL.Add('update barang_detail set
jumlah = jumlah + :jumlah ');
      SQL.Add(' where kd_brg =
'+QuotedStr(dm.Tdetail_pembeliankd_brg.AsS
tring)+' and edisi
='+QuotedStr(dm.Tdetail_pembelianedisi.AsSt
ring));
      Params[0].AsString :=
dm.Tdetail_pembelianjatah.AsString;
      ExecSQL;
     end;
    end
    else
    begin
     with dm.Q1 do
     begin
      Close;
      SQL.Clear;
      SQL.Add('insert into barang_detail
values('+QuotedStr(dm.Tdetail_pembeliankd_b
rg.AsString)+','+dm.Tdetail_pembelianedisi.As
String+' ');
      SQL.Add(',
'+QuotedStr(dm.Tdetail_pembelianjatah.AsStri
ng)+','+QuotedStr(dm.Tdetail_pembelianharga
_bandrol.AsString)+') ');
      ExecSQL;
     end;
    end;
     dm.Tdetail_pembelian.Next;
   end;
   dm.Tpembelian.Edit;
   dm.Tpembeliantgl_beli.AsDateTime :=
DateTimePicker1.Date;
   dm.Tpembelian.Post;
   with dm.Q1 do
   begin
    Close;
    SQL.Clear;
    SQL.Add('select saldo from buku_hutang
where id_supplier
='+QuotedStr(dm.Tpembelianid_supplier.AsStr
ing)+' order by no_transaksi desc');
    Open;
    saldo := dm.Q1.Fields[0].AsInteger;
   end;
   dm.TBuku_Hutang.Open;
   dm.TBuku_Hutang.Append;
   dm.TBuku_Hutang.Fields[2].AsString :=
dm.Tpembelianid_supplier.AsString;
   dm.TBuku_Hutang.Fields[4].AsInteger :=
dm.Tpembeliansaldo_hutang.AsInteger;
   dm.TBuku_Hutang.Fields[5].AsInteger :=
saldo +
dm.Tpembeliansaldo_hutang.AsInteger;
   dm.TBuku_Hutang.Fields[6].AsString :=
'Pembelian';
   dm.TBuku_Hutang.Post;
   Close;
  end;
end;

procedure
TFpembelian2.FormActivate(Sender:
TObject);
begin
//  ListBarang;
end;
```

```
procedure TFpembelian2.BitBtn2Click(Sender:
TObject);
begin
  FormDestroy(Sender);
  Close;
end;

procedure TFpembelian2.DBEdit3Exit(Sender:
TObject);
begin
  if dm.Tpembelianjumlah_bayar.AsInteger <
dm.Tpembeliantotal.AsInteger then
  begin
    try
      dm.Tpembelian.Edit;
      dm.Tpembeliansaldo_hutang.AsInteger :=
dm.Tpembeliantotal.AsInteger -
dm.Tpembelianjumlah_bayar.AsInteger;
      dm.Tpembelian.Post;
    except
    end;
  end
  else
  begin
    try
      dm.Tpembelian.Edit;
      dm.Tpembeliansaldo_hutang.AsInteger :=
0;
      dm.Tpembelian.Post;
    except
    end;
  end;
end;

procedure TFpembelian2.FormDestroy(Sender:
TObject);
var i : Integer;
begin
 try
 if messagedlg('Batal
Transaksi?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
 begin
   if stEdit then
   begin

   end
   else
   begin
     if dm.Tdetail_pembelian.RecordCount > 0
then
     begin
       for i := 0 to
dm.Tdetail_pembelian.RecordCount - 1 do
       begin
         dm.Tdetail_pembelian.Delete;
       end;
     end;
     dm.Tpembelian.Delete;
   end;
 end;
 except
 end;
end;

procedure TFpembelian2.DBGrid2Exit(Sender:
TObject);
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select sum(sub_total) from
detail_pembelian where no_faktur =
'+QuotedStr(dm.Tpembelianno_faktur.AsStrin
g));
    Open;
  end;
  dm.Tpembelian.Edit;
  dm.Tpembeliantotal.AsInteger :=
dm.Q1.Fields[0].AsInteger;
  dm.Tpembelian.Post;
end;

procedure
TFpembelian2.DBLookupComboBox1Exit(Se
nder: TObject);
begin
  ListBarang;
end;

procedure TFpembelian2.ListBarang;
begin
  dm.Tbarang.Open;
  if dm.Tsuppliertipe.AsString = 'P' then
  begin
    dm.Tbarang.Filter := 'id_supplier =
'+QuotedStr(dm.Tsupplierid_supplier.AsString
);
    dm.Tbarang.Filtered := True;
  end
  else if dm.Tsuppliertipe.AsString = 'SA' then
  begin
    dm.Tbarang.Filtered := False;
  end;
  DBGrid2.Columns.Items[0].PickList.Clear;
  dm.Tbarang.First;
  while dm.Tbarang.EOF=false
  do
  begin
```

```
DBGrid2.Columns.Items[0].PickList.Add(dm.
Tbarang.Fields[0].AsString);
    dm.Tbarang.Next;
  end;
end;

procedure TFpembelian2.FormCreate(Sender:
TObject);
begin
  dm.Tsupplier.Open;
end;

procedure
TFpembelianutama.SpeedButton1Click(Sender
: TObject);
begin
  dm.Tpembelian.first;
end;

procedure
TFpembelianutama.SpeedButton2Click(Sender
: TObject);
begin
  dm.Tpembelian.prior;
   if dm.Tpembelian.BOF then
       showmessage('anda sudah di awal data');
end;

procedure
TFpembelianutama.SpeedButton3Click(Sender
: TObject);
begin
 dm.Tpembelian.next;
   if dm.Tpembelian.EOF then
       showmessage('anda sudah di akhir data');
end;

procedure
TFpembelianutama.SpeedButton4Click(Sender
: TObject);
begin
 dm.Tpembelian.last;
end;

procedure
TFpembelianutama.SpeedButton5Click(Sender
: TObject);
begin
   dm.Tdetail_pembelian.first;
end;

procedure
TFpembelianutama.SpeedButton6Click(Sender
: TObject);
begin
  dm.Tdetail_pembelian.prior;
   if dm.Tdetail_pembelian.BOF then
       showmessage('anda sudah di awal data');
end;

procedure
TFpembelianutama.SpeedButton7Click(Sender
: TObject);
begin
    dm.Tdetail_pembelian.next;
   if dm.Tdetail_pembelian.EOF then
       showmessage('anda sudah di akhir data');
end;

procedure
TFpembelianutama.SpeedButton8Click(Sender
: TObject);
begin
   dm.Tdetail_pembelian.last;
end;

procedure
TFpembelianutama.BitBtn1Click(Sender:
TObject);
begin
  dm.Tpembelian.Open;
  dm.Tdetail_pembelian.Open;
  dm.Tpembelian.Append;

Application.CreateForm(TFpembelian2,Fpemb
elian2);
  Fpembelian2.stEdit := False;
    Fpembelian2.ShowModal;
  end;

procedure
TFpembelianutama.BitBtn6Click(Sender:
TObject);
begin
   Close;
end;

procedure
TFpembelianutama.RadioGroup1Click(Sender:
TObject);
begin
      if RadioGroup1.itemindex=0 then
            dm.Tdetail_pembelian.indexname:="
       else
       if RadioGroup1.itemindex=1 then

dm.Tdetail_pembelian.indexname:='Idxkd_brg'
;
```

```
end;

procedure
TFpembelianutama.Button1Click(Sender:
TObject);
var find:boolean;
begin

Find:=dm.Tpembelian.FindKey([Edit1.text]);
      if Find=false then
            showmessage('File not found');
end;

procedure
TFpembelianutama.BitBtn2Click(Sender:
TObject);
begin
  dm.Tpembelian.Edit;
  dm.Tdetail_pembelian.Edit;

Application.CreateForm(TFpembelian2,Fpemb
elian2);
  Fpembelian2.stEdit := True;
  Fpembelian2.ShowModal;
end;

procedure
TFpembelianutama.FormShow(Sender:
TObject);
begin
  dm.Tpembelian.Open;
  dm.Tdetail_pembelian.Open;
end;

procedure
TFpembelianutama.UlangClick(Sender:
TObject);
begin
  Edit1.Clear;
end;

procedure TFpenjualan.Logout1Click(Sender:
TObject);
begin
  dm.StatusLogin := False;
  dm.UpdateStatusUser;
  close;
end;

procedure TFpenjualan.DBEdit1Exit(Sender:
TObject);
begin
  if not dm.Tpenjualanno_faktur.IsNull then
    dm.Tpenjualan.Post
  else
    begin
      MessageDlg('No Faktur Tidak Boleh
Kosong !!!',mtWarning,[mbok],0);
      DBEdit1.SetFocus;
    end;
end;

procedure TFpenjualan.FormActivate(Sender:
TObject);
begin
  dm.Tpelanggan.Close;
  dm.Tpelanggan.Open;
  dm.Tbarang.Open;
  ListBarang;
end;

procedure TFpenjualan.DBGrid2Exit(Sender:
TObject);
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('select sum(sub_total) from
detail_penjualan where no_faktur =
'+QuotedStr(dm.Tpenjualanno_faktur.AsString
));
    Open;
  end;
  dm.Tpenjualan.Edit;
  dm.Tpenjualantotal.AsInteger :=
dm.Q1.Fields[0].AsInteger;
  dm.Tpenjualan.Post;
end;

procedure TFpenjualan.DBEdit3Exit(Sender:
TObject);
begin
  if dm.Tpenjualanjumlah_bayar.AsInteger <
dm.Tpenjualantotal.AsInteger then
  begin
    try
      dm.Tpenjualan.Edit;
      dm.Tpenjualansaldo_piutang.AsInteger :=
dm.Tpenjualantotal.AsInteger -
dm.Tpenjualanjumlah_bayar.AsInteger;
      dm.Tpenjualan.Post;
    except
    end;
  end
  else
  begin
    try
      dm.Tpenjualan.Edit;
```

```
    dm.Tpenjualansaldo_piutang.AsInteger :=
0;
    dm.Tpenjualan.Post;
   except
   end;
  end;
end;

procedure TFpenjualan.BitBtn1Click(Sender:
TObject);
var saldo : Integer;
begin
  if DBLookupComboBox1.Text = '' then
  begin
    ShowMessage('Pelanggan Tidak Boleh
Kosong');
    DBLookupComboBox1.SetFocus;
  end
  else if dm.Tdetail_penjualan.RecordCount = 0
then
  begin
    ShowMessage('Data Barang Yang Dijual
Masih Kosong!!!' );
  end
  else
  begin
    dm.Tdetail_penjualan.First;
    while dm.Tdetail_penjualan.EOF=false
    do
    begin
    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('update barang_detail set jumlah
= jumlah - :jumlah ');
      SQL.Add(' where kd_brg =
'+QuotedStr(dm.Tdetail_penjualankd_brg.AsSt
ring)+' and edisi
='+QuotedStr(dm.Tdetail_penjualanedisi.AsStr
ing));
      Params[0].AsString :=
dm.Tdetail_penjualaneksemplar.AsString;
      ExecSQL;
    end;
      dm.Tdetail_penjualan.Next;
    end;

    dm.Tpenjualan.Edit;
    dm.Tpenjualantgl_jual.AsDateTime :=
DateTimePicker1.Date;
    dm.Tpenjualan.Post;

    with dm.Q1 do
    begin
      Close;
      SQL.Clear;
      SQL.Add('select saldo from buku_piutang
where id_plg
='+QuotedStr(dm.Tpenjualanid_plg.AsString)+
' order by no_transaksi desc');
      Open;
      saldo := dm.Q1.Fields[0].AsInteger;
    end;
    dm.TBuku_Piutang.Open;
    dm.TBuku_Piutang.Append;
    dm.TBuku_Piutang.Fields[2].AsString :=
dm.Tpenjualanid_plg.AsString;
    dm.TBuku_Piutang.Fields[3].AsInteger :=
dm.Tpenjualansaldo_piutang.AsInteger;
    dm.TBuku_Piutang.Fields[5].AsInteger :=
saldo - dm.Tpenjualansaldo_piutang.AsInteger;
    dm.TBuku_Piutang.Fields[6].AsString :=
'Penjualan';
    dm.TBuku_Piutang.Post;

    Close;
  end;
end;

procedure TFpenjualan.BitBtn2Click(Sender:
TObject);
begin
  FormDestroy(Sender);
  Close;
end;


procedure TFpenjualan.FormClose(Sender:
TObject; var Action: TCloseAction);
//var i : Integer;
begin
{ Action := caNone;
 try
 if messagedlg('Batal
Transaksi?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
 begin
   if Fpenjualan2.StEdit  then
   begin

   end
   else
   begin
     if dm.Tdetail_penjualan.RecordCount > 0
then
     begin
       for i := 0 to
dm.Tdetail_penjualan.RecordCount - 1 do
       begin
```

```
        dm.Tdetail_penjualan.Delete;
       end;
     end;
     try
       dm.Tpenjualan.Delete;
     except end;
   end;
   Action := caFree;
 end;
 except
 end;}
// FormDestroy(Sender);
end;

procedure TFpenjualan.Button1Click(Sender:
TObject);
begin
  fakturpenjualan.Qpenjualan.Close;
  fakturpenjualan.Qpenjualan.SQL.Clear;
  fakturpenjualan.Qpenjualan.SQL.Add('select
* from penjualan a');
  fakturpenjualan.Qpenjualan.SQL.Add('join
pelanggan b on a.id_plg = b.id_plg');
  fakturpenjualan.Qpenjualan.SQL.Add('join
detail_penjualan c on a.no_faktur =
c.no_faktur');
  fakturpenjualan.Qpenjualan.SQL.Add('join
barang d on c.kd_brg  = d.kd_brg');
  fakturpenjualan.Qpenjualan.SQL.Add(' where
a.no_faktur =
'+QuotedStr(dm.Tpenjualanno_faktur.AsString
));
  fakturpenjualan.Qpenjualan.Open;
  fakturpenjualan.Prepare;
  fakturpenjualan.Preview;
end;

procedure TFpenjualan.FormDestroy(Sender:
TObject);
var i : Integer;
begin
 try
 if messagedlg('Batal
Transaksi?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
 begin
   if Fpenjualan2.StEdit  then
   begin

   end
   else
   begin
     if dm.Tdetail_penjualan.RecordCount > 0
then
     begin
       for i := 0 to
dm.Tdetail_penjualan.RecordCount - 1 do
       begin
         dm.Tdetail_penjualan.Delete;
       end;
     end;
     try
       dm.Tpenjualan.Delete;
     except end;
   end;
 end;
 except
 end;
end;

procedure TFpenjualan.ListBarang;
begin
  dm.QBarang.Open;
  DBGrid2.Columns.Items[0].PickList.Clear;
  dm.Tbarang.First;
  while dm.Tbarang.EOF=false
  do
  begin

DBGrid2.Columns.Items[0].PickList.Add(dm.
Tbarang.Fields[0].AsString);
    dm.Tbarang.Next;
  end;
end;

procedure
TFpenjualan.DBGrid2ColExit(Sender:
TObject);
begin
  DBGrid2.Columns.Items[2].PickList.Clear;
  dm.QBarang.Close;
  dm.QBarang.Params[0].AsString :=
dm.Tdetail_penjualankd_brg.AsString;
  dm.QBarang.Open;
  dm.Qbarang.First;
  while dm.Qbarang.EOF=false
  do
  begin

DBGrid2.Columns.Items[2].PickList.Add(dm.
QBarang.Fields[1].AsString);
    dm.Qbarang.Next;
  end;
  if
(not(dm.Tdetail_penjualankd_brg.IsNull))and
(not (dm.Tdetail_penjualanedisi.IsNull)) then
  begin
    with dm.Q1 do
    begin
      Close;
```

```
    SQL.Clear;
    SQL.Add('select jumlah from barang_detail
where kd_brg
='+QuotedStr(dm.Tdetail_penjualankd_brg.As
String)+' and edisi
='+QuotedStr(dm.Tdetail_penjualanedisi.AsStr
ing));
    Open;
   end;
   if dm.Q1.RecordCount > 0 then
   begin
    if
dm.Tdetail_penjualaneksemplar.AsInteger >
dm.Q1.Fields[0].AsInteger then
    begin
     ShowMessage('Jumlah Melebihi Stok
Barang !!! Jumlah Maksimum
'+dm.Q1.Fields[0].AsString);
      dm.Tdetail_penjualaneksemplar.AsInteger
:= dm.Q1.Fields[0].AsInteger;
    end;
   end
   else
   begin
    ShowMessage('Data Barang
'+dm.Tdetail_penjualannm_barang.AsString+'
edisi '+dm.Tdetail_penjualanedisi.AsString+'
tidak ada');
   end;
  end;
end;

procedure TFpenjualan2.FormCreate(Sender:
TObject);
begin
  dm.Tpenjualan.Open;
  dm.Tdetail_penjualan.Open;
end;

procedure TFpenjualan2.BitBtn1Click(Sender:
TObject);
begin
  dm.Tpenjualan.Open;
  dm.Tdetail_penjualan.Open;
  dm.Tpenjualan.Append;
  stEdit := False;

Application.CreateForm(TFpenjualan,Fpenjual
an);
  Fpenjualan.ShowModal;
end;

procedure TFpenjualan2.BitBtn6Click(Sender:
TObject);
begin
  Close;
end;

procedure TFpenjualan2.BitBtn2Click(Sender:
TObject);
begin
  fakturpenjualan.Qpenjualan.Close;
  fakturpenjualan.Qpenjualan.SQL.Clear;
  fakturpenjualan.Qpenjualan.SQL.Add('select
* from penjualan a');
  fakturpenjualan.Qpenjualan.SQL.Add('join
pelanggan b on a.id_plg = b.id_plg');
  fakturpenjualan.Qpenjualan.SQL.Add('join
detail_penjualan c on a.no_faktur =
c.no_faktur');
  fakturpenjualan.Qpenjualan.SQL.Add('join
barang d on c.kd_brg  = d.kd_brg');
  fakturpenjualan.Qpenjualan.SQL.Add(' where
a.no_faktur =
'+QuotedStr(dm.Tpenjualanno_faktur.AsString
));
  fakturpenjualan.Qpenjualan.Open;

  fakturpenjualan.Prepare;
  fakturpenjualan.Preview;
end;

procedure
TfrmReturPembelian.BitBtn1Click(Sender:
TObject);
begin
  dm.TRPembelian.Open;
  dm.TRDetail_Pembelian.Open;
  dm.TRPembelian.Append;

Application.CreateForm(TfrmReturPembelian2
,frmReturPembelian2);
    frmReturPembelian2.ShowModal;
end;

procedure
TfrmReturPembelian.BitBtn6Click(Sender:
TObject);
begin
  Close;
end;

procedure
TfrmReturPembelian.SpeedButton6Click(Send
er: TObject);
begin
  dm.TRDetail_Pembelian.prior;
   if dm.TRDetail_Pembelian.BOF then
       showmessage('anda sudah di awal data');
end;
```

```
procedure
TfrmReturPembelian.SpeedButton5Click(Send
er: TObject);
begin
  dm.TRDetail_Pembelian.First;
end;

procedure
TfrmReturPembelian.SpeedButton8Click(Send
er: TObject);
begin
  dm.TRDetail_Pembelian.Last;
end;

procedure
TfrmReturPembelian.SpeedButton7Click(Send
er: TObject);
begin
    dm.TRDetail_Pembelian.next;
   if dm.TRDetail_Pembelian.EOF then
        showmessage('anda sudah di akhir data');
end;

procedure
TfrmReturPembelian.SpeedButton1Click(Send
er: TObject);
begin
  dm.TRPembelian.First;
end;

procedure
TfrmReturPembelian.SpeedButton4Click(Send
er: TObject);
begin
  dm.TRPembelian.Last;
end;

procedure
TfrmReturPembelian.SpeedButton2Click(Send
er: TObject);
begin
  dm.TRPembelian.Prior;
   if dm.TRPembelian.BOF then
        showmessage('anda sudah di awal data');
end;

procedure
TfrmReturPembelian.SpeedButton3Click(Send
er: TObject);
begin
  dm.TRPembelian.Next;
  if dm.TRPembelian.Eof then
        showmessage('anda sudah di akhir data');
end;

procedure
TfrmReturPembelian.FormCreate(Sender:
TObject);
begin
  dm.TRPembelian.Open;
end;

procedure
TfrmReturPembelian2.FormCreate(Sender:
TObject);
begin
  dm.Tpembelian.Open;
  dm.TRPembelian.Open;
  dm.TRDetail_Pembelian.Open;
  dm.Tsupplier.Open;
  dm.QLPembelian.Close;
  dm.QLPembelian.Open;
end;

procedure
TfrmReturPembelian2.DBLookupComboBox2
CloseUp(Sender: TObject);
begin
  dm.QLPembelian.Close;
  dm.QLPembelian.Params[0].AsString :=
DBLookupComboBox2.KeyValue;
  dm.QLPembelian.Open;
end;

procedure
TfrmReturPembelian2.DBLookupComboBox1
CloseUp(Sender: TObject);
begin
  dm.Tpembelian.Filtered := True;
  dm.Tpembelian.Filter := 'id_supplier =
'+QuotedStr(DBLookupComboBox1.KeyValue
);
  dm.Tpembelian.Filtered := True;
end;

procedure
TfrmReturPembelian2.Button1Click(Sender:
TObject);
var  sisa : Integer;
begin
  if frmInputJumlah.ShowModal = mrOk then
  begin
    sisa := dm.QLPembelian.Fields[4].AsInteger
- dm.QLPembelian.Fields[5].AsInteger;
     if StrToInt (frmInputJumlah.Edit1.Text)  <
sisa then
     begin
       with dm.Q2 do
       begin
```

```
      Close;
      SQL.Clear;
      SQL.Add('select count(*) from
returbelidetail where no_faktur
='+QuotedStr(dm.QLPembelian.Fields[0].AsSt
ring));
      SQL.Add(' and kd_brg =
'+QuotedStr(dm.QLPembelian.Fields[1].AsStri
ng)+' and edisi
='+QuotedStr(dm.QLPembelian.Fields[2].AsSt
ring));
      Open;
    end;
    if dm.Q2.Fields[0].AsInteger > 0 then
    begin
      UpdateRetur;
    end
    else
    begin
      InsertRetur;
    end;
  end
  else
  begin
    ShowMessage('Sisa Barang Yang Dapat Di
Retur ='+IntToStr(sisa));
  end;
 end;
 dm.TRDetail_Pembelian.Open;
end;

procedure TfrmReturPembelian2.InsertRetur;
begin
 dm.TRDetail_Pembelian.Append;
 dm.TRDetail_Pembelian.Fields[1].AsString
:= dm.QLPembelian.Fields[0].AsString;
 dm.TRDetail_Pembelian.Fields[2].AsString
:= dm.QLPembelian.Fields[1].AsString;
 dm.TRDetail_Pembelian.Fields[3].AsString
:= dm.QLPembelian.Fields[2].AsString;
 dm.TRDetail_Pembelian.Fields[4].AsString
:= frmInputJumlah.Edit1.Text;
 dm.TRDetail_Pembelian.Fields[5].AsInteger
:= dm.QLPembelian.Fields[6].AsInteger -
(dm.QLPembelian.Fields[6].AsInteger *
dm.QLPembelian.Fields[7].AsInteger div 100);
 dm.TRDetail_Pembelian.Fields[6].AsInteger
:= dm.TRDetail_Pembelian.Fields[5].AsInteger
* StrToInt(frmInputJumlah.Edit1.Text);
  dm.TRDetail_Pembelian.Post;
end;

procedure TfrmReturPembelian2.UpdateRetur;
begin
 with dm.Q1 do
 begin
   Close;
   SQL.Clear;
   SQL.Add('update returbelidetail set jumlah
=jumlah +:jumlah, total = jumlah * harga
where no_faktur
='+QuotedStr(dm.QLPembelian.Fields[0].AsSt
ring));
   SQL.Add(' and kd_brg
='+QuotedStr(dm.QLPembelian.Fields[1].AsSt
ring)+' and edisi
='+QuotedStr(dm.QLPembelian.Fields[2].AsSt
ring));
   Params[0].AsString :=
frmInputJumlah.Edit1.Text;
   ExecSQL;
 end;
end;

procedure
TfrmReturPembelian2.BitBtn1Click(Sender:
TObject);
var  saldo : Integer;
begin
 if DBLookupComboBox1.Text = '' then
 begin
   ShowMessage('Suplier Masih Kosong!!!');
   DBLookupComboBox1.SetFocus;
 end
 else
 begin
   dm.TRDetail_Pembelian.Refresh;
   dm.TRDetail_Pembelian.First;
   while dm.TRDetail_Pembelian.EOF=false
   do
   begin
     with dm.Q1 do
     begin
       Close;
       SQL.Clear;
       SQL.Add('update detail_pembelian set
retur = retur + :jumlah where no_faktur
='+QuotedStr(dm.QLPembelian.Fields[0].AsSt
ring));
       SQL.Add(' and kd_brg
='+QuotedStr(dm.QLPembelian.Fields[1].AsSt
ring)+' and edisi
='+QuotedStr(dm.QLPembelian.Fields[2].AsSt
ring));
       Params[0].AsString :=
dm.TRDetail_Pembelian.Fields[4].AsString;
       ExecSQL;

       Close;
       SQL.Clear;
```

```
      SQL.Add('update pembelian set
total_retur = total_retur + :total_retur where
no_faktur
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
1].AsString));
  //      SQL.Add(' and kd_brg
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
1].AsString)+' and edisi =
'+QuotedStr(dm.TRDetail_Pembelian.Fields[3]
.AsString));
      Params[0].AsInteger :=
dm.TRDetail_Pembelian.Fields[6].AsInteger;
      ExecSQL;

        Close;
        SQL.Clear;
        SQL.Add('update barang_detail set
jumlah = jumlah - :jumlah ');
        SQL.Add(' where kd_brg
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
2].AsString)+' and edisi =
'+QuotedStr(dm.TRDetail_Pembelian.Fields[3]
.AsString));
        Params[0].AsInteger :=
dm.TRDetail_Pembelian.Fields[6].AsInteger;
        ExecSQL;
     end;
     dm.TRDetail_Pembelian.Next;
   end;
   with dm.Q1 do
   begin
     Close;
     SQL.Clear;
     SQL.Add('select sum(total) from
returbelidetail where no_retur
='+QuotedStr(dm.TRPembelian.Fields[0].AsSt
ring));
     Open;
   end;

   dm.TRPembelian.Edit;
   dm.TRPembeliantgl.AsDateTime :=
DateTimePicker1.Date;
   dm.TRPembeliantotal.AsInteger :=
dm.Q1.Fields[0].AsInteger;
   dm.TRPembelian.Post;

   with dm.Q1 do
   begin
     Close;
     SQL.Clear;
     SQL.Add('select saldo from buku_hutang
where id_supplier
='+QuotedStr(dm.TRPembelian.Fields[2].AsSt
ring)+' order by no_transaksi desc');
     Open;
     saldo := dm.Q1.Fields[0].AsInteger;
   end;
   dm.TBuku_Hutang.Open;
   dm.TBuku_Hutang.Append;
   dm.TBuku_Hutang.Fields[2].AsString :=
dm.TRPembelian.Fields[2].AsString;
   dm.TBuku_Hutang.Fields[3].AsInteger :=
dm.TRPembelian.Fields[3].AsInteger;
   dm.TBuku_Hutang.Fields[5].AsInteger :=
saldo - dm.TRPembelian.Fields[3].AsInteger;
   dm.TBuku_Hutang.Fields[6].AsString :=
'Retur Pembelian';
   dm.TBuku_Hutang.Post;

   Close;
  end;
end;

procedure
TfrmReturPembelian2.BitBtn2Click(Sender:
TObject);
begin
  FormDestroy(Sender);
  Close;
end;

procedure
TfrmReturPembelian2.FormDestroy(Sender:
TObject);
begin
 try
 if messagedlg('Batal
Transaksi?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
 begin
   if dm.TRDetail_Pembelian.RecordCount > 0
then
   begin
     with dm.Q1 do
     begin
       Close;
       SQL.Clear;
       SQL.Add('delete from returbelidetail
where no_retur
='+QuotedStr(dm.TRPembelian.Fields[0].AsSt
ring));
       ExecSQL;
     end;
   end;
   dm.TRPembelian.Delete;
 end;
 except
 end;
end;
```

```
procedure
TfrmReturPembelian2.Button2Click(Sender:
TObject);
begin
 if messagedlg('Hapus Data Retur
?',mtconfirmation,[mbyes,mbno],0)=mrYes
then
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('delete from returbelidetail where
no_faktur
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
1].AsString));
    SQL.Add(' and kd_brg
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
2].AsString));
    SQL.Add(' and edisi
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
3].AsString));
    SQL.Add(' and no_retur
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
0].AsString));
    ExecSQL;
  end;
end;
procedure
TfrmReturPenjualan.BitBtn1Click(Sender:
TObject);
begin
  dm.TRPenjualan.Open;
  dm.TRDetail_Penjualan.Open;
  dm.TRPenjualan.Append;

Application.CreateForm(TfrmReturPenjualan2,
frmReturPenjualan2);
    frmReturPenjualan2.ShowModal;
end;

procedure
TfrmReturPenjualan.BitBtn6Click(Sender:
TObject);
begin
  Close;
end;

procedure
TfrmReturPenjualan.SpeedButton1Click(Sende
r: TObject);
begin
  dm.TRPenjualan.First;
end;

procedure
TfrmReturPenjualan.SpeedButton4Click(Sende
r: TObject);
begin
  dm.TRPenjualan.Last;
end;

procedure
TfrmReturPenjualan.SpeedButton2Click(Sende
r: TObject);
begin
  dm.TRPenjualan.Prior;
   if dm.TRPenjualan.BOF then
       showmessage('anda sudah di awal data');
end;

procedure
TfrmReturPenjualan.SpeedButton3Click(Sende
r: TObject);
begin
  dm.TRPenjualan.Next;
  if dm.TRPenjualan.Eof then
       showmessage('anda sudah di akhir data');
end;

procedure
TfrmReturPenjualan.FormCreate(Sender:
TObject);
begin
  dm.TRPenjualan.Open;
end;

procedure
TfrmReturPenjualan2.DBLookupComboBox1
CloseUp(Sender: TObject);
begin
  dm.Tpenjualan.Filtered := True;
  dm.Tpenjualan.Filter := 'id_plg =
'+QuotedStr(DBLookupComboBox1.KeyValue
);
  dm.Tpenjualan.Filtered := True;
end;

procedure
TfrmReturPenjualan2.DBLookupComboBox2
CloseUp(Sender: TObject);
begin
  dm.QLPenjualan.Close;
  dm.QLPenjualan.Params[0].AsString :=
DBLookupComboBox2.KeyValue;
  dm.QLPenjualan.Open;
end;
```

```
procedure
TfrmReturPenjualan2.Button1Click(Sender:
TObject);
var  sisa : Integer;
begin
  if frmInputJumlah.ShowModal = mrOk then
  begin
   sisa := dm.QLPenjualan.Fields[4].AsInteger -
dm.QLPenjualan.Fields[7].AsInteger;
    if StrToInt (frmInputJumlah.Edit1.Text)  <
sisa then
    begin
      with dm.Q2 do
      begin
        Close;
        SQL.Clear;
        SQL.Add('select count(*) from
returjualdetail where no_faktur
='+QuotedStr(dm.QLPenjualan.Fields[0].AsStr
ing));
        SQL.Add(' and kd_brg =
'+QuotedStr(dm.QLPenjualan.Fields[1].AsStri
ng)+' and edisi
='+QuotedStr(dm.QLPenjualan.Fields[2].AsStr
ing));
        Open;
      end;
      if dm.Q2.Fields[0].AsInteger > 0 then
      begin
        UpdateRetur;
      end
      else
      begin
        InsertRetur;
      end;
    end
    else
    begin
      ShowMessage('Sisa Barang Yang Dapat Di
Retur ='+IntToStr(sisa));
    end;
  end;
end;

procedure TfrmReturPenjualan2.InsertRetur;
begin
  dm.TRDetail_Penjualan.Append;
  dm.TRDetail_Penjualan.Fields[1].AsString :=
dm.QLPenjualan.Fields[0].AsString;
  dm.TRDetail_Penjualan.Fields[2].AsString :=
dm.QLPenjualan.Fields[1].AsString;
  dm.TRDetail_Penjualan.Fields[3].AsString :=
dm.QLPenjualan.Fields[2].AsString;
  dm.TRDetail_Penjualan.Fields[4].AsString :=
frmInputJumlah.Edit1.Text;
  dm.TRDetail_Penjualan.Fields[5].AsInteger
:= dm.QLPenjualan.Fields[5].AsInteger;
  dm.TRDetail_Penjualan.Fields[6].AsInteger
:= dm.TRDetail_Penjualan.Fields[5].AsInteger
* StrToInt(frmInputJumlah.Edit1.Text);
  dm.TRDetail_Penjualan.Post;
end;

procedure TfrmReturPenjualan2.UpdateRetur;
begin
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('update returjualdetail set jumlah
=jumlah +:jumlah, total = jumlah * harga
where no_faktur
='+QuotedStr(dm.QLPenjualan.Fields[0].AsStr
ing));
    SQL.Add(' and kd_brg
='+QuotedStr(dm.QLPenjualan.Fields[1].AsStr
ing)+' and edisi
='+QuotedStr(dm.QLPenjualan.Fields[2].AsStr
ing));
    Params[0].AsString :=
frmInputJumlah.Edit1.Text;
    ExecSQL;
  end;
end;

procedure
TfrmReturPenjualan2.Button2Click(Sender:
TObject);
begin
 if messagedlg('Hapus Data Retur
?',mtconfirmation,[mbyes,mbno],0)=mrYes
then
  with dm.Q1 do
  begin
    Close;
    SQL.Clear;
    SQL.Add('delete from returjualdetail where
no_faktur
='+QuotedStr(dm.TRDetail_Penjualan.Fields[1
].AsString));
    SQL.Add(' and kd_brg
='+QuotedStr(dm.TRDetail_Penjualan.Fields[2
].AsString));
    SQL.Add(' and edisi
='+QuotedStr(dm.TRDetail_Penjualan.Fields[3
].AsString));
    SQL.Add(' and no_retur
='+QuotedStr(dm.TRDetail_Penjualan.Fields[0
].AsString));
    ExecSQL;
```

```
  end;
end;

procedure
TfrmReturPenjualan2.BitBtn2Click(Sender:
TObject);
begin
  FormDestroy(Sender);
  Close;
end;

procedure
TfrmReturPenjualan2.FormDestroy(Sender:
TObject);
begin
 try
 if messagedlg('Batal
Transaksi?',mtconfirmation,[mbyes,mbno],0)=
mrYes then
 begin
   if dm.TRDetail_Penjualan.RecordCount > 0
then
   begin
     with dm.Q1 do
     begin
       Close;
       SQL.Clear;
       SQL.Add('delete from returjualdetail
where no_retur
='+QuotedStr(dm.TRPenjualan.Fields[0].AsStr
ing));
       ExecSQL;
     end;
   end;
   dm.TRPenjualan.Delete;
 end;
 except
 end;
end;

procedure
TfrmReturPenjualan2.FormCreate(Sender:
TObject);
begin
  dm.Tpenjualan.Open;
  dm.TRPenjualan.Open;
  dm.TRDetail_Penjualan.Open;
  dm.Tpelanggan.Open;
  dm.QLPenjualan.Close;
  dm.QLPenjualan.Open;
end;

procedure
TfrmReturPenjualan2.BitBtn1Click(Sender:
TObject);
var  saldo : Integer;
begin
  if DBLookupComboBox1.Text = '' then
  begin
    ShowMessage('Pelanggan Masih
Kosong!!!');
    DBLookupComboBox1.SetFocus;
  end
  else
  begin
    dm.TRDetail_Penjualan.Refresh;
    dm.TRDetail_Penjualan.First;
    while dm.TRDetail_Penjualan.EOF=false
    do
    begin
      with dm.Q1 do
      begin
        Close;
        SQL.Clear;
        SQL.Add('update detail_penjualan set
retur = retur + :jumlah where no_faktur
='+QuotedStr(dm.QLPenjualan.Fields[0].AsStr
ing));
        SQL.Add(' and kd_brg
='+QuotedStr(dm.QLPenjualan.Fields[1].AsStr
ing)+' and edisi
='+QuotedStr(dm.QLPenjualan.Fields[2].AsStr
ing));
        Params[0].AsString :=
dm.TRDetail_Penjualan.Fields[4].AsString;
        ExecSQL;

        Close;
        SQL.Clear;
        SQL.Add('update penjualan set total_retur
= total_retur + :total_retur where no_faktur
='+QuotedStr(dm.TRDetail_Penjualan.Fields[1
].AsString));
  //      SQL.Add(' and kd_brg
='+QuotedStr(dm.TRDetail_Pembelian.Fields[
1].AsString)+' and edisi =
'+QuotedStr(dm.TRDetail_Pembelian.Fields[3]
.AsString));
        Params[0].AsInteger :=
dm.TRDetail_Penjualan.Fields[6].AsInteger;
        ExecSQL;

         Close;
         SQL.Clear;
         SQL.Add('update barang_detail set
jumlah = jumlah + :jumlah ');
         SQL.Add(' where kd_brg
='+QuotedStr(dm.TRDetail_Penjualan.Fields[2
].AsString)+' and edisi =
```

```
'+QuotedStr(dm.TRDetail_Penjualan.Fields[3].
AsString));
        Params[0].AsInteger :=
dm.TRDetail_Penjualan.Fields[6].AsInteger;
        ExecSQL;
     end;
     dm.TRDetail_Penjualan.Next;
   end;
   with dm.Q1 do
   begin
     Close;
     SQL.Clear;
     SQL.Add('select sum(total) from
returjualdetail where no_retur
='+QuotedStr(dm.TRPenjualan.Fields[0].AsStr
ing));
     Open;
   end;

   dm.TRPenjualan.Edit;
   dm.TRPenjualantgl.AsDateTime :=
DateTimePicker1.Date;
   dm.TRPenjualantotal.AsInteger :=
dm.Q1.Fields[0].AsInteger;
   dm.TRPenjualan.Post;

   with dm.Q1 do
   begin
     Close;
     SQL.Clear;
     SQL.Add('select saldo from buku_piutang
where id_plg
='+QuotedStr(dm.TRPenjualan.Fields[2].AsStr
ing)+' order by no_transaksi desc');
     Open;
     saldo := dm.Q1.Fields[0].AsInteger;
   end;
   dm.TBuku_Piutang.Open;
   dm.TBuku_Piutang.Append;
   dm.TBuku_Piutang.Fields[2].AsString :=
dm.TRPenjualan.Fields[2].AsString;
   dm.TBuku_Piutang.Fields[3].AsInteger :=
dm.TRPenjualan.Fields[3].AsInteger;
   dm.TBuku_Piutang.Fields[5].AsInteger :=
saldo - dm.TRPenjualan.Fields[3].AsInteger;
   dm.TBuku_Piutang.Fields[6].AsString :=
'Retur Penjualan';
   dm.TBuku_Piutang.Post;

   Close;
  end;
end;

procedure TFsupplier.BitBtn1Click(Sender:
TObject);
begin
  dm.Tsupplier.append;

Application.CreateForm(TFisisupplier,Fisisupp
lier);
       Fisisupplier.showmodal;
       if Fisisupplier.modalresult=mrok then
            dm.Tsupplier.post
       else
            dm.Tsupplier.cancel;
end;

procedure TFsupplier.BitBtn2Click(Sender:
TObject);
begin
  dm.Tsupplier.edit;

Application.CreateForm(TFisisupplier,Fisisupp
lier);
       Fisisupplier.showmodal;
       if Fisisupplier.modalresult=mrok then
            dm.Tsupplier.post
       else
            dm.Tsupplier.cancel;
end;

procedure TFsupplier.BitBtn3Click(Sender:
TObject);
begin
 if messagedlg('anda yakin akan menghapus
data
supplier?',mtconfirmation,[mbyes,mbno],0)=mr
Yes then
      dm.Tsupplier.delete;
end;

procedure TFsupplier.BitBtn6Click(Sender:
TObject);
begin
   close;
end;

procedure
TFsupplier.RadioGroup1Click(Sender:
TObject);
begin
      if RadioGroup1.itemindex=0 then

dm.Tsupplier.IndexFieldNames:='id_supplier'
       else
       if RadioGroup1.itemindex=1 then

dm.Tsupplier.IndexFieldNames:='nama_suppli
er';
end;
```

```
procedure TFsupplier.Button1Click(Sender:
TObject);
var find:boolean;
begin

Find:=dm.Tsupplier.FindKey([Edit1.text]);
      if Find=false then
            showmessage('File not found');
end;

procedure TFsupplier.UlangClick(Sender:
TObject);
begin
  Edit1.Clear;
end;

procedure TFsupplier.FormShow(Sender:
TObject);
begin
 dm.Tsupplier.Close;
 dm.Tsupplier.Open;
end;

procedure TFuser1.BitBtn1Click(Sender:
TObject);
begin
 dm.Tuser.append;
 Application.CreateForm(TFisiuser,Fisiuser);
 Fisiuser.showmodal;
      if Fisiuser.modalresult=mrok then
      begin
            dm.Tuser.post;
              with dm.Q1 do
              begin
                Close;
                SQL.Clear;
                SQL.Add('update user set
pass_user =
MD5('+QuotedStr(dm.Tuser.Fields[1].AsString
)+') where username =
'+QuotedStr(dm.Tuser.Fields[0].AsString));
                ExecSQL;
              end;
      end
      else
            dm.Tuser.cancel;
end;

procedure TFuser1.BitBtn2Click(Sender:
TObject);
begin
 if (dm.Tuserst_aktif.AsInteger = 0) then
 begin
  try
  dm.Tuser.edit;
  except
  end;
      Application.CreateForm(TFEditUser,
FEditUser);
      FEditUser.showmodal;
      if FEditUser.modalresult=mrok then
      begin
            try
              dm.Tuser.post;
            except end;
            with dm.Q1 do
            begin
              Close;
              SQL.Clear;
              SQL.Add('update user set
pass_user =
MD5('+QuotedStr(FEditUser.Edit2.Text)+')
where username =
'+QuotedStr(dm.Tuser.Fields[0].AsString));
                ExecSQL;
            end;
      end
      else
      try
            dm.Tuser.cancel;
      except end;
  dm.Tuser.Refresh;
 end
 else
   MessageDlg('User Sedang Aktif Tidak Bisa
Di Edit!', mtInformation,[mbok],0);
end;

procedure TFuser1.BitBtn3Click(Sender:
TObject);
begin
 if (dm.Tuserst_aktif.AsInteger = 0) then
 begin
  if messagedlg('anda yakin akan menghapus
data
user?',mtconfirmation,[mbyes,mbno],0)=mrYes
then
   begin
     with dm.Q1 do
     begin
      Close;
      SQL.Clear;
      SQL.Add('delete from user where
username
='+QuotedStr(dm.Tuserusername.AsString)+'
and pass_user
='+QuotedStr(dm.Tuserpass_user.AsString)+'
');
```

```
        SQL.Add( ' and hak_akses
='+QuotedStr(dm.Tuserhak_akses.AsString));
        ExecSQL;
      end;
      dm.Tuser.Refresh;
   end;
 end
 else
   MessageDlg('User Sedang Aktif Tidak Bisa
Di Hapus!', mtInformation,[mbok],0);
end;

procedure TFuser1.BitBtn6Click(Sender:
TObject);
begin
 close;
end;


procedure TFuser1.Button2Click(Sender:
TObject);
var find:boolean;
begin
      Find:=dm.Tuser.FindKey([Edit1.text]);
      if Find=false then
            showmessage('File not found');
end;
end.
```

LAMPIRAN C

HASIL KUESIONER

KUESIONER

(Sistem Informasi pendataan penyaluran koran dan majalah)

Nama :

Jabatan :

Pertanyaan:

1. Sistem yang dibuat dapat mempermudah dan mempercepat mendapatkan informasi *supplier* dan pelanggan?
   - ☐ Setuju
   - ☐ Ragu-ragu
   - ☐ Tidak setuju

2. Sistem yang dibuat mempermudah dan mempercepat mendapatkan informasi data barang?
   - ☐ Setuju
   - ☐ Ragu-ragu
   - ☐ Tidak setuju

3. Sistem yang dibuat dapat mempermudah dan mempercepat dalam pembuatan catatan transaksi pembelian dan penjualan?
   - ☐ Setuju
   - ☐ Ragu-ragu
   - ☐ Tidak setuju

4. Sistem yang dibuat dapat mempermudah dan mempercepat pembuatan laporan?
   - ☐ Setuju
   - ☐ Ragu-ragu
   - ☐ Tidak setuju

5. Sistem yang dibuat dapat mempercepat kinerja perusahaan?
   - ☐ Setuju
   - ☐ Ragu-ragu
   - ☐ Tidak setuju

Mengetahui,

_____

Ket: beri tanda (✓) pada kotak yang tersedia

LAMPIRAN D

DOKUMEN MANUAL DARI SISTEM LAMA

LAMPIRAN E

SURAT PENELITIAN