

SISTEM INFORMASI AKUNTANSI DENGAN
LAPORAN LABA/ RUGI MENGGUNAKAN METODE
MULTIPLE STEP

SKRIPSI

*Diajukan untuk memenuhi sebagian persyaratan untuk
memperoleh gelar Sarjana Teknik*

UNIVERSITAS BRAWIJAYA



Disusun oleh :

ARIS WIJAYA
NIM. 0210633014-63

DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2009

SISTEM INFORMASI AKUNTANSI DENGAN
LAPORAN LABA/ RUGI MENGGUNAKAN METODE
MULTIPLE STEP

SKRIPSI

UNIVERSITAS BRAWIJAYA

Disusun oleh :

ARIS WIJAYA
NIM. 0210633014

Telah diperiksa dan disetujui oleh
Dosen Pembimbing :

Ir. Muhammad Aswin, MT.
NIP. 19640626 199002 1 001

Ir. Sutrisno, MT.
NIP. 19570325 198701 1 001

SISTEM INFORMASI AKUNTANSI DENGAN
LAPORAN LABA/ RUGI MENGGUNAKAN METODE
MULTIPLE STEP

Disusun oleh :

ARIS WIJAYA
NIM. 0210633014-63

Skripsi ini telah diuji dan dinyatakan lulus pada

Tanggal 14 bulan Agustus 2009

DOSEN PENGUJI

Ir. Heru Nurwasito, M.Kom
NIP. 19650402 199002 1 001

Himawat Aryadita ST., M.Sc
NIP. 19801018 200801 1 003

Suprapto ST.,MT
NIP. 19710727 199603 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

Ir. Heru Nurwasito, M.Kom.
NIP. 19650402 199002 1 001

KATA PENGANTAR

Assalamualaikum Wr. Wb.

Puji syukur kehadirat Tuhan Yang Maha Kuasa atas segala limpahan berkat dan rahmat-Nya yang telah diberikan kepada penulis sehingga dapat menyelesaikan Tugas Akhir dengan judul **SISTEM INFORMASI AKUNTANSI DENGAN LAPORAN LABA/ RUGI MENGGUNAKAN METODE MULTIPLE STEP** sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik di Jurusan Elektro Fakultas Teknik Universitas Brawijaya Malang.

Penyusun menyadari bahwa skripsi ini tidak akan terwujud tanpa bantuan dan dorongan dari banyak pihak, karena itu dalam kesempatan ini penyusun menyampaikan ungkapan terima kasih yang sebesar-besarnya kepada :

1. Allah yang maha esa atas segala nikmat yang telah diberikan.
2. Kedua orang tua tercinta, kakak dan adikku yang telah mendoakan dan menyemangati untuk terus maju dan selalu semangat menuju arah yang lebih baik.
3. Bapak Ir. Heru Nur Wasito , M Kom dan Bapak Rudy Yuwono, ST, MSc selaku Ketua dan sekretaris Jurusan Teknik Elektro yang telah memberikan bantuan sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
4. Bapak Ir. M. Aswin, MT dan Bapak Ir. Sutrisno, MT selaku dosen pembimbing yang telah memberikan bantuan, bimbingan, perhatian, waktu, masukan, serta nasehatnya.
5. Bapak/Ibu Dosen Teknik Elektro yang telah memberikan bimbingan selama masa perkuliahan dan seluruh karyawan Jurusan Teknik Elektro yang telah membantu dalam pengurusan administrasi dan bantuan lainnya.

6. Seluruh rekan-rekan Teknik Elektro Universitas Brawijaya Malang angkatan 2002 yang telah memberikan masukan dan dorongan hingga Tugas Akhir ini dapat diselesaikan.
7. Semua pihak yang telah membantu penyelesaian skripsi ini.

Penulis menyadari sepenuhnya bahwa Tugas Akhir ini masih banyak kekurangan dan jauh dari sempurna, oleh sebab itu, segala kritik dan saran yang membangun akan penulis terima dengan senang hati demi kesempurnaan skripsi ini.

Akhir kata penulis memohon maaf yang sebesar-besarnya kepada semua pihak, apabila selama menempuh pendidikan di **Universitas Brawijaya Malang** terdapat kesalahan-kesalahan yang dilakukan baik disengaja maupun tidak.

Harapan penulis semoga skripsi ini dapat bermanfaat dan berguna bagi kita semua serta pihak yang membutuhkan

Wassalamualaikum Wr. Wb.

Malang, Agustus 2009

Penulis

ABSTRAK

ARIS WIJAYA. 2009: PENGEMBANGAN SISTEM INFORMASI AKUNTANSI DENGAN LAPORAN LABA/RUGI MENGGUNAKAN METODE MULTIPLE STEP. *Sripsi Jurusan Elektro, Fakultas Teknik, Universitas Brawijaya. Dosen Pembimbing: Ir. M. Aswin, MT. dan Ir. Sutrisno, MT.*

Akuntansi merupakan kebutuhan krusial bagi pengembangan suatu badan usaha. Akuntansi akan sangat berguna bagi badan usaha dalam banyak hal, antara lain mengetahui penjualan yang terjadi, keuntungan atau kerugian dan masih banyak informasi lainnya.

Pada skripsi ini dikembangkan sistem informasi akuntansi. Aplikasi sistem informasi akuntansi dengan laporan laba/rugi menggunakan metode multiple step ini bertujuan menghasilkan informasi keuangan yang diperlukan oleh Usaha Bersama(U.B) Miftahur Rizki dan untuk menggantikan pencatatan secara manual.

Aplikasi sistem informasi akuntansi ini dibangun dengan arsitektur *client-server*, dengan menggunakan bahasa pemrograman java dan basis data mysql serta tools pendukung lainnya.

Dengan menggunakan system informasi akuntansi ini diharapkan dapat meningkatkan akurasi dalam pencatatan setiap proses keuangan dan dapat memberikan laporan keuangan secara cepat.

Kata kunci : Sistem Informasi, Akuntansi, Java, MySQL



DAFTAR ISI

LEMBAR PERSETUJUAN	i
KATA PENGANTAR	i
ABSTRAK.....	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	2
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Manfaat	2
1.6 Sistematika Penulisan	3
BAB II DASAR TEORI	5
2.1 Sistem Informasi	5
2.2 Akuntansi	5
2.3 Laporan Keuangan	6
2.3.1 Laporan Neraca	6
2.3.2 Laporan Laba Rugi	7
2.3.3 Laporan Buku Besar	7
2.4 Basis Data	8
2.4.1 Definisi Basis Data	8
2.4.2 Model Bisnis Data Relasional.....	9
2.5 MySQL	9
2.6 JAVA	11
2.7 Aplikasi NetBeans	13
2.8 UML (<i>Unified Modeling Language</i>).....	13
2.9 JDBC	14
BAB III METODE PENELITIAN	19
3.1 Studi Literatur	19
3.2 Perancangan	19
3.3 Implementasi.....	19

3.4 Pengujian.....	20
3.5 Kesimpulan	20
BAB IV PERANCANGAN	21
4.1 Analisa Kebutuhan (<i>Requirements Analysis</i>).....	21
4.1.1 Identifikasi Aktor	21
4.1.2 Daftar Kebutuhan.....	22
4.1.3 Diagram <i>Use Case</i>	25
4.2 Perancangan Perangkat Lunak.....	48
4.2.1 Diagram Klas (<i>Class Diagram</i>)	48
4.2.1.1 Klas Diagram Pada Paket Model	49
4.2.1.2 Klas Diagram Pada Paket finalaplikasi	49
4.2.2 Diagram Sekuensial (<i>Sequence Diagram</i>)	50
4.2.3 Basis Data (<i>Database</i>)	60
4.2.3.1 <i>Entity Relationship Diagram</i> (<i>Diagram ER</i>).....	60
4.2.3.1 Normalisasi Basis Data	62
BAB V IMPLEMENTASI.....	66
5.1 Spesifikasi Sistem	66
5.1.1 Spesifikasi Perangkat Keras (<i>Hardware</i>).....	66
5.1.2 Spesifikasi Perangkat Lunak (<i>Software</i>).....	67
5.2 Implementasi Klas Pada File Program.....	67
5.3 Implementasi Algoritma	68
5.3.1 Algoritma <code>isValid</code>	68
5.3.2 Algoritma proses <code>isDetailValid</code>	69
5.3.3 Algoritma proses <code>simpan</code>	70
5.3.4 Algoritma proses <code>isAlreadyItem</code>	71
5.4 Implementasi Antarmuka Aplikasi Sistem Informasi Akuntansi	72
5.4.1 Implementasi Aplikasi Login.....	72
5.4.2 Implementasi Aplikasi Menu Utama	74
5.4.3 Implementasi Aplikasi Master Supplier.....	75
5.4.4 Implementasi Aplikasi Master Barang.....	77

5.4.5 Implementasi Aplikasi Transaksi Penjualan	78
5.4.6 Implementasi Aplikasi Lihat Transaksi Penjualan	80
5.4.7 Implementasi Aplikasi Transaksi Penjualan Saham	81
BAB VI PENGUJIAN DAN ANALISA SISTEM	83
6.1 Pengujian Unit	84
6.1.1 Pengujian Unit Basis Data	84
6.1.1.1 Pengujian Perancangan	84
6.1.1.1.1 Prosedur Pengujian	85
6.1.1.1.2 Hasil Pengujian	88
6.1.1.1.3 Analisa Pengujian	96
6.1.2 Pengujian Sistem	97
6.2.1 Pengujian Login Form	97
6.2.1.1 Hasil Pengujian	97
6.2.1.2 Analisa Pengujian	98
6.2.2 Pengujian Menu Utama Form	98
6.2.2.1 Analisa Pengujian	106
6.3 Pengujian Waktu Akses <i>Query</i>	106
6.3.1 Tujuan	106
6.3.2 Prosedur Pengujian	107
6.3.3 Hasil Pengujian	107
6.3.4 Analisa Pengujian	109
BAB VII PENUTUP	110
7.1 Kesimpulan	110
7.2 Saran	110
DAFTAR PUSTAKA	112

DAFTAR TABEL

Tabel 4.1	Deskripsi Aktor	22
Tabel 4.2	Deskripsi Daftar Kebutuhan	25
Tabel 4.3	Skenario <i>Use Case</i> Login	27
Tabel 4.4	Skenario <i>Use Case</i> Master Barang	28
Tabel 4.5	Skenario <i>Use Case</i> Master Jenis Barang	28
Tabel 4.6	Skenario <i>Use Case</i> Master Karyawan	29
Tabel 4.7	Skenario <i>Use Case</i> Master Pengguna	30
Tabel 4.8	Skenario <i>Use Case</i> Master Jenis Pembayaran	31
Tabel 4.9	Skenario <i>Use Case</i> Master Jenis Pemasukan	32
Tabel 4.10	Skenario <i>Use Case</i> Master Supplier	33
Tabel 4.11	Skenario <i>Use Case</i> Master Anggota	34
Tabel 4.12	Skenario <i>Use Case</i> Transaksi Pembelian	35
Tabel 4.13	Skenario <i>Use Case</i> Transaksi Penjualan	36
Tabel 4.14	Skenario <i>Use Case</i> Transaksi Penjualan Saham	37
Tabel 4.15	Skenario <i>Use Case</i> Pembayaran	38
Tabel 4.16	Skenario <i>Use Case</i> Pemasukan	39
Tabel 4.17	Skenario <i>Use Case</i> Lihat Transaksi Pembelian	40
Tabel 4.18	Skenario <i>Use Case</i> Lihat Transaksi Penjualan	41
Tabel 4.19	Skenario <i>Use Case</i> Lihat Transaksi Penjualan Saham	42
Tabel 4.20	Skenario <i>Use Case</i> Laporan Buku Besar	43
Tabel 4.21	Skenario <i>Use Case</i> Laporan Neraca	43
Tabel 4.22	Skenario <i>Use Case</i> Laporan Laba Rugi	44
Tabel 4.23	Skenario <i>Use Case</i> Laporan Karyawan	45
Tabel 4.24	Skenario <i>Use Case</i> Laporan Pemegang Saham	46
Tabel 4.25	Skenario <i>Use Case</i> Laporan Persediaan Barang	46
Tabel 4.26	Skenario <i>Use Case</i> Logout	47
Tabel 5.1	Spesifikasi Perangkat Keras	66
Tabel 5.2	Spesifikasi Perangkat Lunak	67

Tabel 5.3	Implementasi Klas Pada Kode Program *.java	67
Tabel 5.4	Penjelasan Tombol-tombol Pada Aplikasi Login	73
Tabel 5.5	Penjelasan Menu-menu Pada Aplikasi Menu Utama	74
Tabel 5.6	Penjelasan Tombol-tombol Pada Aplikasi Master Supplier	76
Tabel 5.7	Penjelasan Tombol-tombol Pada Aplikasi Master Barang	78
Tabel 5.8	Penjelasan Tombol-tombol Pada Aplikasi Transaksi Penjualan	79
Tabel 5.9	Penjelasan Tombol-tombol Pada Aplikasi Lihat Transaksi Penjualan	80
Tabel 5.10	Penjelasan Tombol-tombol Pada Aplikasi Transaksi Penjualan Saham	82
Tabel 6.1	Pengujian Waktu Akses Query	108
Tabel 6.2	Perbandingan Waktu Akses Per Record	109



DAFTAR GAMBAR

Gambar 2.1	Contoh Laporan Neraca	7
Gambar 2.2	Contoh Laporan Laba Rugi	7
Gambar 2.3	Contoh Bentuk Laporan Buku Besar	8
Gambar 2.4	Tipe Driver	15
Gambar 2.5	Tipe Driver	16
Gambar 4.1	Diagram <i>Use Case</i>	26
Gambar 4.2	Hubungan Antar Paket	48
Gambar 4.3	Diagram Klas Pada Paket Model.....	49
Gambar 4.4	Diagram Klas Pada Paket finalaplikasi.....	50
Gambar 4.5	<i>Entity Relationship diagram</i>	61
Gambar 5.1	Algoritma Proses isInputValid	69
Gambar 5.2	Algoritma Proses isInputDetilValid	70
Gambar 5.3	Algoritma Proses Simpan.....	71
Gambar 5.4	Algoritma Proses isAlreadyItem	72
Gambar 5.5	Antarmuka Aplikasi Login.....	73
Gambar 5.6	Antarmuka Aplikasi Menu Utama.....	74
Gambar 5.7	Antarmuka Aplikasi Master Supplier	76
Gambar 5.8	Antarmuka Aplikasi Master Barang	77
Gambar 5.9	Antarmuka Aplikasi Transaksi Penjualan.....	79
Gambar 5.10	Antarmuka Aplikasi Lihat Transaksi Penjualan.....	80
Gambar 5.11	Antarmuka Aplikasi Transaksi Penjualan Saham	81
Gambar 6.1	Diagram Pohon Pengujian Sistem Informasi Akuntansi	84
Gambar 6.2	<i>Conceptual Data Model Object</i>	85
Gambar 6.3	Hasil Dari Pemeriksaan Model Oleh PowerDesigner	86
Gambar 6.4	<i>Physical Data Model Object</i>	87
Gambar 6.5	Hasil dari <i>Generate Database</i> Oleh PowerDesigner.....	88
Gambar 6.6	Tabel-tabel Yang Terbentuk Setelah <i>Generate Database</i>	89
Gambar 6.7	Struktur Tabel Anggota.....	89
Gambar 6.8	Struktur Tabel Barang.....	90
Gambar 6.9	Struktur Tabel detil_beli	90

Gambar 6.10 Struktur Tabel detil_jual	91
Gambar 6.11 Struktur Tabel jenis_brg.....	91
Gambar 6.12 Struktur Tabel jenis_byr.....	92
Gambar 6.13 Struktur Tabel jenis_masuk.....	92
Gambar 6.14 Struktur Tabel jual_saham	93
Gambar 6.15 Struktur Tabel Karyawan	93
Gambar 6.16 Struktur Tabel Operator	94
Gambar 6.17 Struktur Tabel Pemasukan	94
Gambar 6.18 Struktur Tabel Pembayaran.....	95
Gambar 6.19 Struktur Tabel Supplier	95
Gambar 6.20 Struktur Tabel t_beli	96
Gambar 6.21 Struktur Tabel t_jual	96
Gambar 6.22 Memasukkan Nilai Yang Salah.....	97
Gambar 6.23 Pesan Peringatan Ketika Tombol Login Ditekan	97
Gambar 6.24 Pada Saat Tombol Tambah Ditekan	98
Gambar 6.25 Pada Saat Tombol Simpan Ditekan Dan Data Input Valid	99
Gambar 6.26 Pada Saat Tombol Ubah Ditekan	99
Gambar 6.27 Pada Saat Tombol Tambah Ditekan	100
Gambar 6.28 Pada Saat Tombol Simpan Ditekan	101
Gambar 6.29 Pada Saat Tombol Ubah Ditekan	101
Gambar 6.30 Pada Saat Tombol Tambah Ditekan	102
Gambar 6.31 Pada Saat Tombol Simpan Ditekan	103
Gambar 6.32 Pada Saat Tombol Ubah Ditekan	103
Gambar 6.33 Pada Saat Tombol Tambah Ditekan dan Data Tidak Valid	104
Gambar 6.34 Pada Saat Tombol Proses Ditekan dan Input Data Tidak Valid	105
Gambar 6.35 Pada Saat Tombol Proses Ditekan dan Hasil Laporan Buku Besar	105
Gambar 6.36 Pada Saat Tombol Proses Ditekan dan Hasil Laporan Laba Rugi	106
Gambar 6.37 Hasil Pengujian Waktu Akses <i>Query</i> Terhadap Tabel Barang	108

Gambar 6.38 Grafik Rata-rata Waktu Akses *Query*..... 109



BAB I

PENDAHULUAN

1.1 Latar Belakang

Kemajuan teknologi sebagai pendukung pemrosesan data dan informasi telah menjadi kebutuhan pokok perusahaan. Perusahaan jasa, dagang, maupun manufaktur serta semua sektor ekonomi yang ada pada saat ini.

Saat ini banyak sekali badan-badan usaha yang telah mengkomputerisasi tugas-tugas manualnya untuk memecahkan berbagai permasalahan yang timbul akibat perkembangan usahanya. Dalam bidang akuntansi peran komputer telah terbukti dapat meringankan tugas karyawan dalam menyelesaikan berbagai kegiatan rutin di sebuah bidang usaha.

Akuntansi akan sangat berguna bagi badan usaha dalam banyak hal, akuntansi akan membantu pengelola/pemilik untuk mengetahui penjualan yang telah terjadi, biaya yang telah dikeluarkan, keuntungan atau kerugian selama satu periode, jumlah utang atau piutang serta tanggal jatuh temponya, komposisi modal pokok atas barang yang dijual dan masih banyak informasi lainnya yang bisa disediakan oleh sistem akuntansi yang baik.

Dengan mengetahui semua informasi tersebut, pengelola bisa mengevaluasi bisnisnya, mengendalikan keuangannya dan membuat perencanaan untuk masa depan badan usahanya. Tanpa hal tersebut pengelola hanya akan terjebak pada rutinitas penjualan sebanyak-banyaknya dengan harga jual yang rendah. Mereka juga akan mengalami kesulitan membuat perencanaan untuk pengembangan bisnis dimasa mendatang karena tidak tahu posisi keuangan saat ini.

Fungsi utama dalam pembuatan sistem akuntansi ini adalah untuk memberikan kemudahan kepada pengelola U.B Miftahur Rizki dalam

memberikan informasi-informasi akuntansi yang diperlukan oleh pemegang saham secara efisien dan transparan dalam waktu yang singkat, serta mengurangi kesalahan-kesalahan yang terjadi dalam perhitungan secara manual.

1.2 Rumusan Masalah

Mengacu pada permasalahan yang telah diuraikan pada latar belakang, maka rumusan masalah dapat disusun sebagai berikut:

1. Merancang dan membuat sistem informasi akuntansi yang dapat dijadikan alternatif lain dalam pengolahan data dan informasi akuntansi.
2. Merancang basis data kebutuhan sistem informasi akuntansi.
3. Merancang laporan akuntansi.

1.3 Batasan Masalah

Dalam perancangan dan pembuatan sistem informasi akuntansi ini perlu dilakukan pembatasan masalah. Adapun batasan masalah dalam skripsi ini antara lain:

1. Sistem dibuat menggunakan tools pemrograman NetBeans 5.0.
2. Basis data dibuat dengan menggunakan software MySQL Server 5.0.
3. Sistem informasi yang digunakan adalah *Microsoft Windows Vista Ultimate*.
4. Menangani pengolahan data dan informasi akuntansi U.B Miftahur Rizki.

1.4 Tujuan

Tujuan akhir penulisan tugas akhir ini adalah untuk menghasilkan suatu aplikasi yang memiliki performasi yang baik, efektif dan mudah digunakan sehingga dapat dijadikan alternatif dalam memberikan informasi akuntansi.

1.5 Manfaat

Adapun manfaat dari pembuatan sistem informasi akuntansi ini antara lain:

1. Mempermudah pengelola dalam melakukan pencatatan kegiatan akuntansi.
2. Mempermudah dalam pembuatan laporan-laporan yang dibutuhkan.
3. Mempermudah pemegang saham dalam mengevaluasi kinerja U.B

Miftahur Rizki.

1.6 Sistematika Penulisan

Pembahasan tugas akhir ini terdiri dari tujuh bab yang disusun dengan sistematika penulisan sebagai berikut:

BAB I Pendahuluan

Memuat tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.

BAB II Dasar Teori

Memuat tentang dasar-dasar teori yang menunjang laporan tugas akhir ini, seperti Java, NetBeans, MySQL, UML, basis data.

BAB III Metodologi Penelitian

Membahas metode yang digunakan dalam penulisan yang terdiri dari studi literatur, implementasi, pengujian, serta pengambilan kesimpulan.

BAB IV Perancangan

Membahas analisa kebutuhan program yang dibuat sesuai dengan teori yang ada serta membahas perancangan dan implementasi dari program yang dibuat.

BAB V **Implementasi**

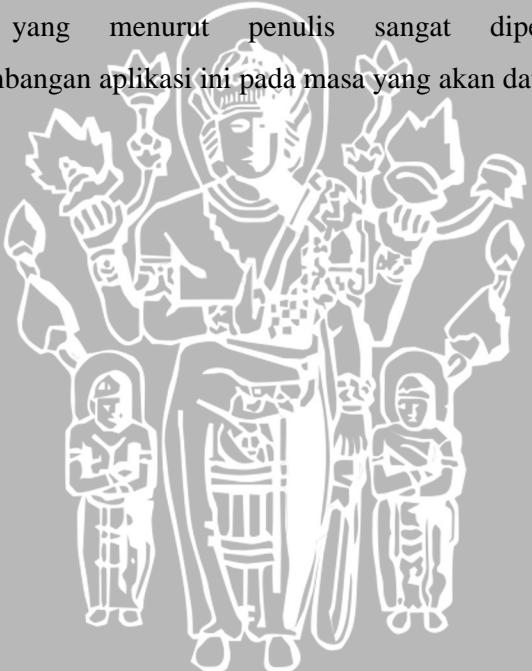
Membahas implementasi program yang dibuat sesuai dengan perancangan yang ada pada bab sebelumnya.

BAB VI **Pengujian dan Analisis**

Membahas pengujian dan analisa aplikasi yang telah dibuat.

BAB VII **Penutup**

Memuat kesimpulan-kesimpulan yang dapat diambil oleh penulis dari hasil perancangan, implementasi, dan pengujian aplikasi serta saran yang menurut penulis sangat diperlukan untuk pengembangan aplikasi ini pada masa yang akan datang.



BAB II

DASAR TEORI

Pada bab ini dijelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi mengenai pengembangan sistem informasi akuntansi dengan laporan laba/rugi menggunakan metode multiple step.

2.1 Sistem Informasi

Sistem informasi adalah kumpulan elemen yang saling berhubungan satu sama lain yang membentuk satu kesatuan untuk mengintegrasikan data, memproses dan menyimpan serta mendistribusikan informasi.

2.2 Akuntansi

Definisi akuntansi dapat dilihat dari 2 (dua) sudut pandang yaitu:

1. Fungsi dan Kegunaan

Akuntansi merupakan aktivitas jasa yang berfungsi memberikan informasi kuantitatif mengenai kesatuan-kesatuan ekonomi terutama yang bersifat keuangan yang bermanfaat dalam pengambilan keputusan.

2. Proses Kegiatan

Akuntansi adalah seni mencatat, mengklasifikasi dan mengikhtisarkan transaksi-transaksi kejadian yang sekurang-kurangnya atau sebagian bersifat keuangan dengan cara menginterpretasikan hasil-hasilnya.

Persamaan dasar akuntansi adalah

$$\boxed{\text{Aktiva} = \text{Hutang} + \text{Modal}}$$

- Aktiva adalah harta yang dimiliki perusahaan yang merupakan sumber ekonomi. Contoh: kas, piutang, gedung dsb.
- Hutang adalah kewajiban yang menjadi beban perusahaan. Contoh : hutang pembelian kredit.
- Modal adalah hak atau klai pemilik atas aktiva perusahaan. Contoh: setoran modal pemilik.

2.3 Laporan Keuangan

Laporan keuangan merupakan hasil akhir dari suatu proses pencatatan, yang merupakan suatu ringkasan dari transaksi-transaksi keuangan yang terjadi selama tahun buku yang bersangkutan.

Agar pembaca laporan keuangan tadi memperoleh gambaran yang jelas, maka laporan keuangan yang disusun harus didasarkan pada prinsip akuntansi yang lazim. Laporan keuangan yang disusun biasanya terdiri dari:

2.3.1 Laporan Neraca

Laporan neraca adalah laporan yang menggambarkan posisi keuangan dari suatu perusahaan yang meliputi aktiva, kewajiban dan ekuitas pada suatu saat tertentu.

TUAN RAKA

NERACA

Per 31 Desember 2002

AKTIVA		HUTANG	
Aktiva Lancar:			
Kas	Rp 356.250.000	Utang Dagang	Rp 50.000.000
Piutang dagang	5.000.000	Utang gaji	2.000.000
Asuransi di bayar dimuka	500.000	Pendapatan ditrm dimuka	600.000
Piutang Pendapatan	500.000		
Aktiva Tetap:			
Kendaraan	150.000.000		
Ak. depr. Kendr	<u>9.000.000</u>		
	<u>Rp 141.000.000</u>		
Peralatan	50.000.000		
	<u>Rp 553.250.000</u>		
			<u>Rp 553.250.000</u>
		MODAL	
		Modal Tn. Raka	500.650.000

Gambar 2.1 Contoh laporan neraca**2.3.2 Laporan Laba Rugi**

Laporan laba rugi adalah laporan mengenai pendapatan, beban, dan laba atau rugi suatu perusahaan dalam suatu periode tertentu.

TUAN RAKA	
LAPORAN LABA RUGI	
<u>untuk periode yang berakhir 31 Desember 2002</u>	
Pendapatan	Rp 12.900.000
Beban-beban:	
- Beban telepon	Rp 1.000.000
- Beban asuransi	Rp 250.000
- Beban depresiasi	Rp 9.000.000
- Beban gaji	<u>Rp 2.000.000 (+)</u>
	<u>Rp 12.250.000 (-)</u>
Laba	<u>Rp 650.000</u>

Gambar 2.2 Contoh laporan laba rugi**2.3.3 Laporan Buku Besar**

Buku Besar adalah buku yang berisi semua akun-akun (kumpulan akun) yang ada dalam laporan keuangan. Buku ini mencatat perubahan-perubahan yang terjadi pada masing-masing akun dan pada akhir periode akan tampak saldo dari akun-akun tersebut. Setiap transaksi yang telah dicatat dalam jurnal akan diposting atau dipindahkan ke Buku Besar secara berkala.

D Kas		K		D Modal		K	
01-Jan	500.000.000	05-Jan	150.000.000			01-Jan	500.000.000
18-Jan	8.000.000	15-Jan	1.000.000				
		20-Jan	750.000				
	508.000.000		151.750.000				
saldo	356.000.000					saldo	500.000.000
D Kendaraan		K		D Beban Telepon		K	
05-Jan	150.000.000			15-Jan	1.000.000		
saldo	150.000.000			saldo	1.000.000		
D Peralatan		K		D Pendapatan		K	
06-Jan	50.000.000					18-Jan	8.000.000
						21-Jan	5.000.000
saldo	50.000.000					Saldo	13.000.000
D Hutang		K		D Beban Asuransi		K	
		06-Jan	50.000.000		20-Jan	750.000	
						saldo	750.000
						saldo	50.000.000
D Piutang		K		D		K	
21-Jan	5.000.000						
saldo	5.000.000						

Gambar 2.3 Contoh bentuk laporan buku besar

2.4 Basis Data

Basis data merupakan komponen terpenting dalam pembangunan sistem informasi, karena menjadi tempat untuk menampung dan mengorganisasikan seluruh data yang ada dalam sistem, sehingga dapat dieksplorasi untuk menyusun informasi-informasi dalam berbagai bentuk. Sedangkan untuk mengelola sistem basis data tersebut digunakan sistem basis data MySQL

2.4.1 Definisi Basis Data

Basis data merupakan himpunan kelompok data yang saling berkaitan. Data tersebut diorganisasikan sedemikian rupa agar tidak terjadi duplikasi yang tidak perlu, sehingga dapat diolah atau dieksplorasi secara cepat dan mudah untuk menghasilkan informasi.

Basis data yang dibentuk diharapkan memiliki sifat-sifat, antara lain:

1. Efisien dan efektif dalam pengorganisasianya, artinya untuk menambah, menyisipkan, atau menghapus data dapat dilakukan dengan mudah dan sederhana.
2. Bebas redundansi, meskipun pada batas-batas tertentu yang dapat ditolelir, redundansi juga diperbolehkan, misalnya untuk mengurangi kompleksitas dalam penulisan program.
3. Fleksibel, artinya basis data dapat diakses dengan mudah, dinamis dan tidak tergantung sepenuhnya pada aplikasi-aplikasi tertentu.
4. Sistem basis data yang dapat diakses secara bersama dalam lingkungan jaringan sehingga mendukung penggunaan bersama dan distribusi data.

2.4.2 Model Bisnis Data Relational

Model basis data relasional sering pula disebut sebagai model relasional atau basis data relasional. Basis data relational berisi kumpulan tabel, dimana setiap table mempunyai nama dan struktur yang unik. Dalam setiap tabel, masing-masing record data diorganisasikan dalam struktur yang sama dan memiliki field kunci yang akan menjadi penghubung antar tabel yang ada dan berkait satu sama lain. Pada model ini, data terorganisir dengan baik dan rapi sehingga dapat dengan mudah dimanipulasi untuk menghasilkan suatu informasi.

Adapun kelebihan model ini antara lain:

1. Kemudahan dalam pembentukan struktur data masing-masing file.
2. Kompleksitas untuk mengaitkan antar tabel tidak terjadi karena hubungan antar tabel ditentukan oleh field kunci yang telah ditetapkan sebagai penghubung antar file.
3. Pemrograman menjadi sederhana, sedangkan tingkat fleksibilitas dalam mengorganisasikan data sangat tinggi.

2.5 MySQL

MySQL merupakan *Relational Database Management System* (DBMS) yang didistribusikan secara gratis, dimana setiap orang bebas untuk menggunakan. Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya dalam query data.

Sebagai database server yang memiliki konsep database modern MySQL memiliki banyak sekali keistimewaan antara lain :

- *Portability*, dimana dapat berjalan stabil pada berbagai sistem operasi.
- *Open Source*.
- *Performance running*, yaitu memiliki kecepatan yang handal.
- *Column Type*, *Command Function*, *Security*.
- *Scalability* yaitu mampu menangani database dalam skala besar.

Perintah-perintah SQL dikelompokkan menjadi 5 macam yaitu :

- *Data Definition Language* (DDL) adalah perintah SQL yang digunakan untuk menjelaskan objek dari database. Dengan kata lain DDL digunakan untuk mendefinisikan kerangka database. Perintahnya adalah :
 - create : untuk membuat/menciptakan objek database.
 - alter : untuk memodifikasi/mengubah objek database.
 - drop : untuk menghapus objek database.
 - objek database yang dimaksud terdiri dari database, tabel, index dan view.
- *Data Manipulation Language* (DML) adalah perintah yang digunakan untuk mengoperasikan atau memanipulasi isi database. SQL menyediakan 4 perintah DML yaitu :
 - select : digunakan untuk mengambil data dari database.
 - delete : digunakan untuk menghapus data dari database
 - insert : untuk menambahkan data pada database.
 - update : untuk memodifikasi data pada database.
- *Security* adalah perintah-perintah yang digunakan untuk menjamin keamanan data. Antara lain terdiri atas :

- *Grant* : memberi akses kepada user tertentu untuk akses ke database.
 - *Revoke* : mencabut hak akses dari user.
 - *Integrity* adalah perintah-perintah yang digunakan untuk menjaga kesatuan data.
 - *Auxiliary* adalah perintah-perintah pelengkap atau tambahan seperti : unload dan rename.
- Contoh : recover table, untuk memperbaiki tabel pada database.

2.6 JAVA

Java pertama kali diluncurkan pada tahun 1995 sebagai bahasa pemrograman umum (general purpose programming language) oleh Sun. Java merupakan bahasa pemrograman yang sederhana semudah C dan seampuh C++, sintaks bahasa yang digunakan dalam Java merupakan pengembangan dari bahasa C/C++.

Java didesain dengan ide utama adalah portabilitas dengan konsep *write once run anywhere*, maka hasil kompilasi Java bukan merupakan *native code* sebagaimana yang dihasilkan oleh bahasa pemrograman umumnya seperti C/C++ ataupun Visual Basic. Hasil kompilasi Java berupa *bytecode*, yang merupakan sekumpulan instruksi yang sangat teroptimasi. Agar bytecode tersebut dapat dieksekusi seperti layaknya program biasa, maka diperlukan suatu program *Java Virtual Machine (JVM)*. Java memiliki kelebihan dibandingkan dengan bahasa pemrograman lainnya.

- Bersifat *portable* dan *platform independent*. Program Java yang ditulis dapat dieksekusi di platform manapun tanpa memerlukan kompilasi ulang (*portable*) asalkan Java Virtual Machine untuk platform tersebut tersedia.
- Memiliki *garbage collection* yang dapat mendealokasi memori secara otomatis. Kita tidak perlu secara eksplisit membebaskan suatu lokasi memori yang dipakai karena ini akan dilakukan secara otomatis oleh Java.

- Menghilangkan pewarisan berganda yang terdapat pada C++. Walaupun kelihatannya lebih sebagai suatu kekurangan namun banyak para ahli yang mengakui bahwa konsep pewarisan berganda lebih banyak mengakibatkan kerugian daripada keuntungan.
- Mengurangi pointer aritmetik. Pengaksesan lokasi memori secara langsung dengan menggunakan pointer memungkinkan program untuk melakukan suatu tindakan yang tidak seharusnya atau tidak boleh dilakukan. Untuk mengurangi dan menghilangkan kemungkinan kesalahan seperti ini, penggunaan pointer pada java telah dibatasi dengan menggunakan reference.
- Memiliki array sejati.
- Mengurangi kerancuan antara pemberian nilai pada statemen kondisional.
Contoh penggunaan tanda '=' dengan '==' pada kondisi if.

Selain itu java juga memiliki karakteristik yang menjadi cirri khas dari java.

- Berorientasi objek, java telah menerapkan konsep pemrograman berorientasi objek yang modern dalam implementasinya.
- Robust, java mendorong pemrograman yang bebas dari kesalahan dengan bersifat *strongly typed* dan memiliki *run-time checking*.
- Portable, program java dapat dieksekusi diplatform manapun selama tersedia java virtual machine untuk platform tersebut.
- Multithreading, java mendukung penggunaan multithreading yang telah terintegrasi langsung dalam bahasa java.
- Dinamis, program java dapat melakukan tindakan yang ditentukan pada saat eksekusi program dan bukan pada saat kompilasi.
- Sederhana, java menggunakan bahasa pemrograman yang sederhana dan mudah dipelajari.
- Terdistribusi, java didesain untuk berjalan pada lingkungan yang terdistribusi seperti halnya internet.
- Aman, aplikasi yang dengan bahasa java dapat dipastikan keamanannya terutama untuk aplikasi internet.

- Netral secara arsitektur, java tidak terikat pada suatu mesin atau system operasi tertentu.
- Interpreted, aplikasi java dapat berjalan pada platform yang berbeda-beda dengan melakukan interpretasi pada bytecode.
- Kinerja tinggi, bytecode java telah sangat teroptimasi sehingga eksekusi program dapat dilakukan secara cepat sekalipun dilakukan dengan cara interpretasi terhadap bytecode.

2.7 Aplikasi NetBeans

NetBeans sebagai IDE ditujukan untuk memudahkan pemrograman Java. Dalam NetBeans, pemrograman dilakukan berbasiskan visual dan event-driven. Seperti IDE lainnya, misalnya Borland Delphi dan Microsoft Visual Studio.

Untuk membuat dialog atau user-interface, kita tidak perlu membuat teks program secara manual baris per baris, tetapi cukup klik pada component-palette. Teks program akan dihasilkan secara otomatis.

NetBeans mencakup compiler atau builder, dan debugger internal. Hal ini sangat memudahkan proses paskaperancangan program. Proses deployment dan/atau tes dapat juga dilakukan dari dalam NetBeans.

NetBeans juga mendukung source-control. Kita dapat memakai internal VCS (version control system) yang telah disediakan (yaitu CVS = concurrent versioning system), ataupun memakai VCS eksternal.

2.8 UML (*Unified Modeling Language*)

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI).

Tujuan utama perancangan UML adalah :

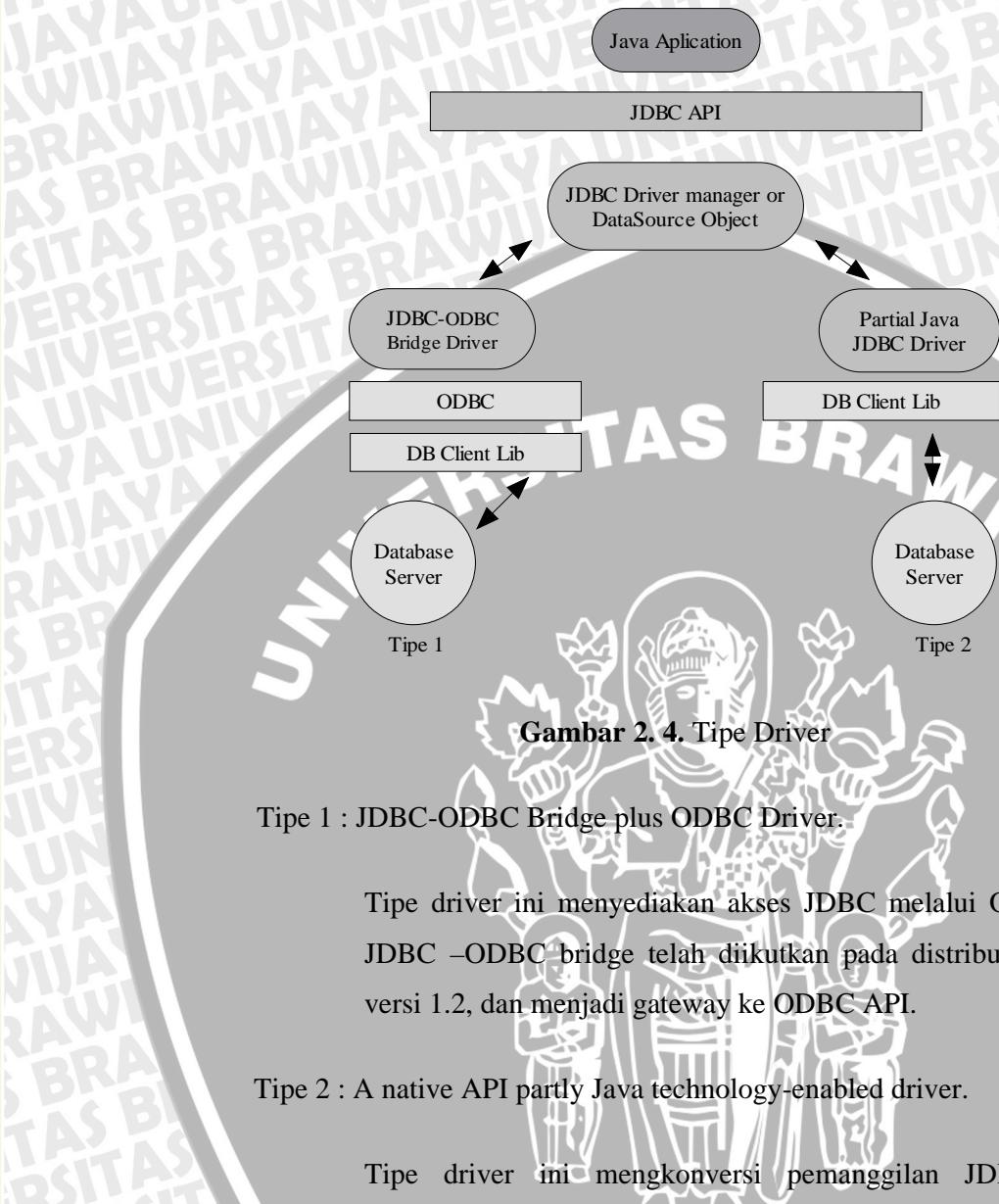
- Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti
- Menyediakan mekanisme perluasan dan spesialisasi untuk memperluas konsep-konsep inti.
- Mendukung spesifikasi independent bahasa pemrograman dan proses pengembangan tertentu.
- Menyediakan basis formal untuk pemahaman bahasa pemodelan.
- Mendorong pertumbuhan pasar kelas berorientasi objek
- Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, framework dan pattern.

2.9 JDBC

JDBC API merupakan kelas Java API untuk mengakses segala data berupa tabel. JDBC memiliki library atau API yang dapat digunakan untuk kebutuhan database, baik itu koneksi maupun pengolahan data. JDBC API memudahkan pengiriman perintah SQL ke sistem database relasional dan mendukung bermacam-macam dialek SQL. Sejak versi ketiga JDBC API telah terdistribusi secara langsung dalam paket yang terdapat pada J2SDK versi 1.4 yaitu pada package java.sql dan javax.sql. kemampuan yang terdapat dalam versi tiga ini diantaranya transaction savepoint, manajemen connection pooling yang lebih mendetail, dan dukungan lebih lengkap terhadap SQL99.

JDBC API terdiri dari dua interface utama yaitu JDBC API untuk penulis aplikasi, dan JDBC Driver API lower-level untuk penulis driver. Teknologi JDBC Driver dikelompokkan dalam empat kategori.

Gambar dibawah ini menggambarkan konektivitas JDBC menggunakan ODBC Driver dan library database klien yang ada.



Gambar 2. 4. Tipe Driver

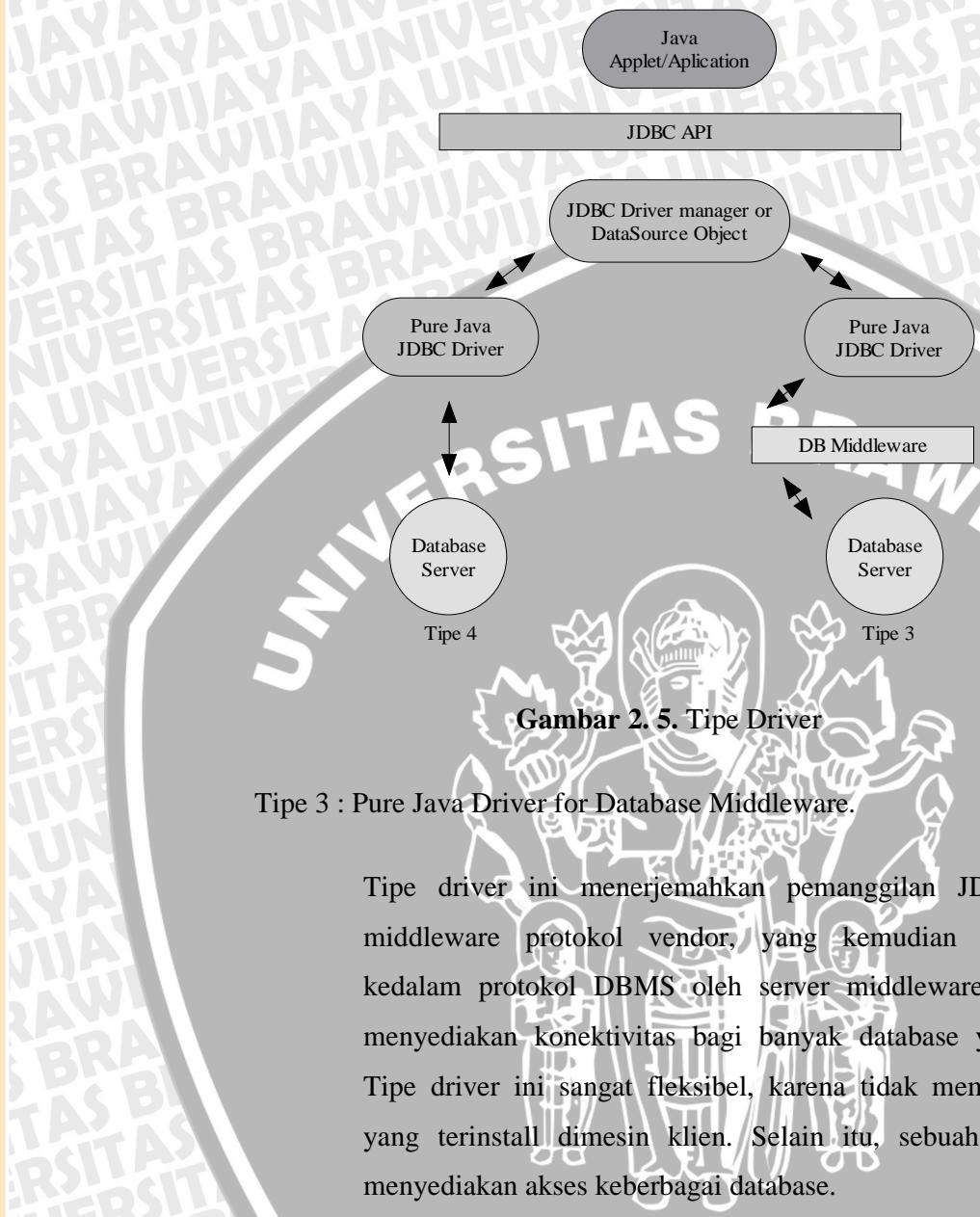
Tipe 1 : JDBC-ODBC Bridge plus ODBC Driver.

Tipe driver ini menyediakan akses JDBC melalui ODBC driver. JDBC –ODBC bridge telah diikutkan pada distribusi JDK sejak versi 1.2, dan menjadi gateway ke ODBC API.

Tipe 2 : A native API partly Java technology-enabled driver.

Tipe driver ini mengkonversi pemanggilan JDBC kedalam pemanggilan klien API untuk Oracle, Sybase, Infomix, DB2, atau DBMS lainnya. Tipe driver ini juga memerlukan kode binary spesifik terhadap system operasi yang di-load kedalam masing-masing mesin klien. Tipe ini memerlukan software tertentu yang diinstall pada mesin klien.

Aplikasi dan applet dapat mengakses database melalui JDBC API menggunakan program Java murni, seperti gambar dibawah ini.



Gambar 2. 5. Tipe Driver

Tipe 3 : Pure Java Driver for Database Middleware.

Tipe driver ini menerjemahkan pemanggilan JDBC kedalam middleware protokol vendor, yang kemudian diterjemahkan kedalam protokol DBMS oleh server middleware. Middleware menyediakan koneksi bagi banyak database yang berbeda. Tipe driver ini sangat fleksibel, karena tidak memerlukan kode yang terinstall dimesin klien. Selain itu, sebuah driver dapat menyediakan akses keberbagai database.

Tipe 4 : Direct-to-Database Pure Java Driver.

Tipe driver ini mengkonversi JDBC secara langsung kedalam protokol jaringan yang digunakan oleh DBMS, memungkinkan suatu panggilan langsung dari mesin klien kepada DBMS server, dan menyediakan suatu solusi praktis untuk akses internet. Driver-driver tipe ini dibangun menggunakan teknologi Java murni sehingga memberikan portabilitas tinggi.

JDBC API memiliki beberapa komponen utama yang berhubungan dengan mekanisme pengolahan data, baik untuk berkomunikasi dengan database server, melakukan koneksi, mengirim perintah SQL, mendapatkan data, menutup koneksi, maupun untuk menangani error yang mungkin terjadi dalam pengolahan data.

- Driver adalah komponen untuk menangani masalah komunikasi dengan database server.
- DriverManager adalah komponen untuk menangani objek driver dimana objek DriverManager juga mengabstraksi detail dari proses kerja objek driver.
- Connection adalah komponen untuk mempresentasikan koneksi secara fisik ke database
- Statement adalah komponen untuk mengirim perintah SQL ke database.
- ResultSet adalah komponen untuk menyimpan data yang didapat dari database setelah perintah SQL dijalankan dengan menggunakan komponen Statement.
- SQLException adalah komponen untuk menangani kesalahan-kesalahan (error) yang mungkin terjadi dalam pengolahan database.

Teknologi JDBC memiliki beberapa keunggulan dalam pembuatan aplikasi berbasis database dibandingkan dengan teknologi lainnya.

- Mempertahankan data yang ada

Dengan teknologi JDBC, aplikasi bisnis dan perusahaan tidak terpaku pada arsitektur tertentu, dan dapat menggunakan database yang sudah ada sebelumnya tanpa harus mengubah atau mengganti isi database yang telah dibuat sebelumnya.

- Menyederhanakan pembuatan aplikasi

Kombinasi antara Java API dan JDBC API menjadikan pembuatan aplikasi lebih mudah, sederhana, dan hemat. JDBC

menyembunyikan tugas-tugas akses data sehingga programmer tidak perlu mengetahui apa yang terjadi dibelakang layar.

- Tidak perlu melakukan konfigurasi pada jaringan komputer

Dengan JDBC API, konfigurasi sistem pada komputer klien tidak perlu dilakukan lagi. Dengan driver yang dituliskan dalam program java, semua informasi yang diperlukan untuk melakukan koneksi dengan database telah didefinisikan oleh JDBC URL atau dengan objek DataSource yang didaftarkan dengan java *Naming and Directory Interface (JNDI)*.

- Akses penuh pada metadata

JDBC API menyediakan akses metadata sehingga memungkinkan untuk mengembangkan sebuah aplikasi yang lebih canggih dan mampu melakukan koneksi database yang lebih baik.

- Tidak perlu diinstall

JDBC driver murni bawaan dari Java, misalnya JDBC-ODBC bridge, tidak perlu diinstall lagi karena secara otomatis sudah terpasang pada mesin klien.

- Koneksi database menggunakan URL

Teknologi JDBC memanfaatkan keunggulan dari standar internet, yaitu *Uniform Resource Locator (URL)*, untuk mengidentifikasi koneksi database. JDBC API juga menyediakan cara yang lebih baik untuk identifikasi dan koneksi database ke sumber data, yaitu dengan menggunakan objek DataSource, dan membuat kode lebih mudah untuk dipindahkan ke platform lain serta lebih mudah dalam pemeliharaan. Objek DataSource juga menyediakan connection pooling dan distributed transaction, yang merupakan hal penting dalam aplikasi database.

BAB III

METODE PENELITIAN

Untuk merealisasikan tujuan penulisan tugas akhir ini, langkah-langkah yang dilakukan adalah :

3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mengkaji hal-hal yang berhubungan dengan teori-teori yang mendukung perencanaan dan perealisasian aplikasi, yaitu:

1. Kajian pustaka mengenai pengembangan perangkat lunak.
2. Kajian pustaka mengenai program aplikasi NetBeans.
3. Kajian pustaka mengenai pemodelan UML.
4. Kajian pustaka mengenai basis data MySQL.
5. Kajian pustaka mengenai Sistem Informasi Akuntansi.

3.2 Perancangan

Melakukan analisis dan perancangan perangkat lunak sebagai dasar untuk melakukan implementasi. Adapun tahap-tahap dalam perancangan adalah sebagai berikut:

1. Analisis kebutuhan.
2. Perancangan perangkat lunak.

3.3 Implementasi

Melakukan implementasi dari hasil perancangan yang telah didefinisikan sebelumnya kedalam bentuk kode (coding) dengan alat bantu pemrograman NetBeans serta MySQL sebagai sarana untuk penyimpanan basis datanya.

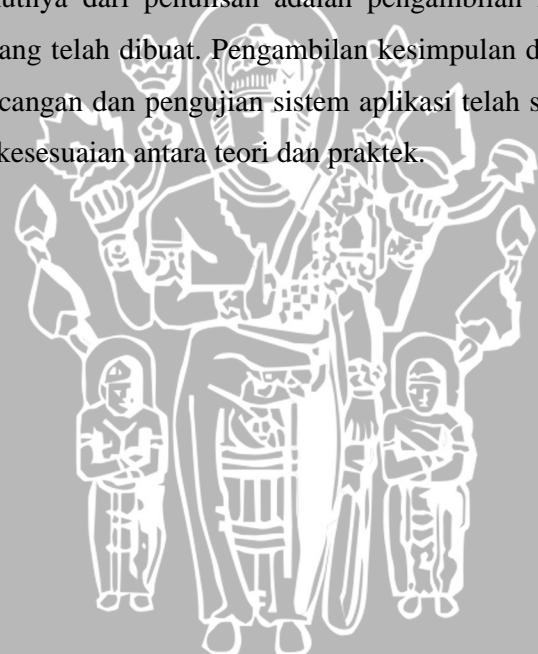
3.4 Pengujian

Melakukan pengujian perangkat lunak untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasi atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi.

Proses pengujian dilakukan dengan menggunakan dua metode pengujian, yaitu pengujian unit (*unit test*) dan pengujian sistem (*system test*).

3.5 Kesimpulan

Tahap berikutnya dari penulisan adalah pengambilan kesimpulan dari keseluruhan sistem yang telah dibuat. Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan dan pengujian sistem aplikasi telah selesai dilakukan dan didasarkan pada kesesuaian antara teori dan praktik.



BAB IV

PERANCANGAN

Bab ini membahas mengenai perancangan perangkat lunak, yang meliputi dua tahap. Pada tahap pertama dilakukan proses analisis kebutuhan (*requirement analysis*), sedangkan tahap yang kedua adalah proses perancangan. Pada tahap analisis kebutuhan digunakan pemodelan dengan diagram *use case*. Pada proses perancangan perangkat lunak digunakan pemodelan dalam bentuk diagram klas (*class diagram*), dan diagram urutan (*sequence diagram*).

4.1 Analisis Kebutuhan (*Requirements analysis*)

Proses analisis ini mengambil acuan dari hasil pengumpulan, pemahaman dan penetapan kebutuhan-kebutuhan (*requirements*) yang ingin didapatkan oleh pengguna, dan harus mampu disediakan oleh sistem perangkat lunak. Pada analisis kebutuhan ini diawali dengan identifikasi aktor-aktor, penjabaran daftar kebutuhan dan kemudian memodelkannya ke dalam suatu diagram *use case*. Analisis kebutuhan ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memecahkan permasalahan yang dihadapi oleh pengguna.

4.1.1 Identifikasi Aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor-aktor yang akan berinteraksi dengan sistem. Tabel 4.1 memperlihatkan dua buah aktor beserta penjelasannya masing-masing yang merupakan hasil dari proses identifikasi aktor.

Aktor	Deskripsi Aktor
User	<i>User</i> merupakan semua aktor pengguna yang belum sah dan belum dapat mengakses semua fungsional sistem,

	hanya <i>login</i> saja yang dapat dilakukan oleh aktor ini.
Administratoristrator	<i>Administrator</i> merupakan aktor pengguna yang sudah terautentifikasi melalui <i>login</i> dan dapat mengakses semua fungsional sistem.
Operator	Operator merupakan aktor pengguna yang sudah terautentifikasi melalui <i>login</i> dan dapat mengakses fungsional sistem secara terbatas.

Tabel 4.1 Deskripsi aktor

Sumber : Analisis kebutuhan

4.1.2 Daftar Kebutuhan

Daftar kebutuhan ini terdiri dari sebuah kolom yang menguraikan kebutuhan yang harus disediakan oleh sistem, dan pada kolom yang lain akan menunjukkan nama *use case* yang akan menyediakan fungsionalitas masing-masing kebutuhan tersebut. Daftar kebutuhan fungsional dan non fungsional sistem yang dikembangkan pada tugas akhir ini ditunjukkan pada Tabel 4.2.

No	Kebutuhan	Aktor	Use Case
1	Sistem harus menyediakan fasilitas login terhadap pengguna dan melakukan proses autentifikasi untuk menjaga keamanan data yang tersimpan dalam sistem.	User	Login
2	Sistem harus dapat digunakan untuk mencatat setiap transaksi penjualan. Dengan memasukkan kode barang maka sistem harus dapat menampilkan nama barang dan harga satuan barang, selain itu sistem juga harus dapat mengkalkulasikan jumlah item dan harga barang yang tercatat serta dapat mencetak struk penjualan dengan nomer transaksi yang berurutan.	Operator, Administrator	Transaksi penjualan

3	Sistem harus dapat digunakan untuk mencatat setiap transaksi pembelian. Dengan memasukkan kode barang maka sistem harus dapat menampilkan nama dan harga barang serta dapat mengkalkulasikan jumlah item dan harga barang yang tercatat. Sistem juga menyediakan fasilitas pencatatan pembayaran secara utang dan supplier yang memasok barang, serta mencetak struk pembelian dengan nomer transaksi yang berurutan.	Operator, Administrator	Transaksi pembelian
4	Sistem harus dapat menampilkan setiap data transaksi penjualan. Dengan memasukkan nomer transaksi penjualan maka sistem akan menampilkan data transaksi penjualan yang sesuai dengan nomer transaksi penjualan	Operator, Administrator	Lihat transaksi penjualan
5	Sistem harus dapat menampilkan setiap data transaksi pembelian. Dengan memasukkan nomer transaksi pembelian maka sistem akan menampilkan data transaksi penjualan yang sesuai dengan nomer transaksi pembelian	Operator, Administrator	Lihat transaksi pembelian
6	Sistem harus dapat digunakan untuk mencatat setiap pembelian saham. Dengan memilih jenis saham yang akan dijual maka sistem akan menampilkan jumlah saham dan jumlah nominal saham yang harus dibayar, Mencatat biodata pembeli saham, serta sistem mencetak transaksi penjualan saham dan mencetak sertifikat saham.	Administrator	Transaksi pembelian saham
7	Sistem harus dapat menampilkan setiap transaksi penjualan dan pembelian secara periode, periode yang dipilih akan ditampilkan dalam bentuk laporan buku besar.	Administrator	Laporan buku besar
8	Sistem harus dapat menampilkan keadaan keuangan secara periode, periode yang dipilih akan ditampilkan dalam bentuk laporan rugi/laba.	Administrator	Laporan rugi/laba
9	Sistem harus dapat menampilkan nilai aktiva, pasiva dan modal secara periode, periode	Administrator	Laporan neraca

	yang dipilih akan ditampilkan dalam bentuk laporan neraca.		
10	Sistem harus dapat menampilkan jumlah persediaan barang yang ada setiap periode, periode yang dipilih akan ditampilkan dalam bentuk laporan persediaan barang.	Administrator	Laporan persediaan barang
11	Sistem harus dapat menampilkan jumlah pemegang saham serta jumlah sisa hasil usaha (SHU) yang didapatkan pemegang saham setiap periode, periode yang dipilih akan ditampilkan dalam bentuk laporan pemegang saham.	Administrator	Laporan pemegang saham
12	Sistem menyediakan fasilitas untuk merubah dan menambah data akun, user memasukkan kode akun baru, menjelaskan deskripsi akun serta mengelompokkan kedalam group.	Administrator	Master akun
13	Sistem menyediakan fasilitas untuk menambah dan merubah data barang. Administrator memasukkan data barang berupa kode barang, nama barang, satuan, harga, akun.	Administrator	Master barang
14	Sistem menyediakan fasilitas untuk menambah dan merubah data supplier. Administrator memasukkan data supplier baru berupa kode supplier, nama, alamat, telepon, kota, akun.	Administrator	Master supplier
15	Sistem harus menyediakan fasilitas untuk dapat mengubah data pemegang saham. Administrator mengubah data pemegang saham berupa nama, alamat, telepon, kota, jumlah saham	Administrator	Master pemegang saham
16	Sistem harus menyediakan fasilitas untuk mengubah data karyawan. Administrator mengubah data karyawan berupa nama, alamat, telepon, kota, bagian, serta gaji.	Administrator	Master karyawan

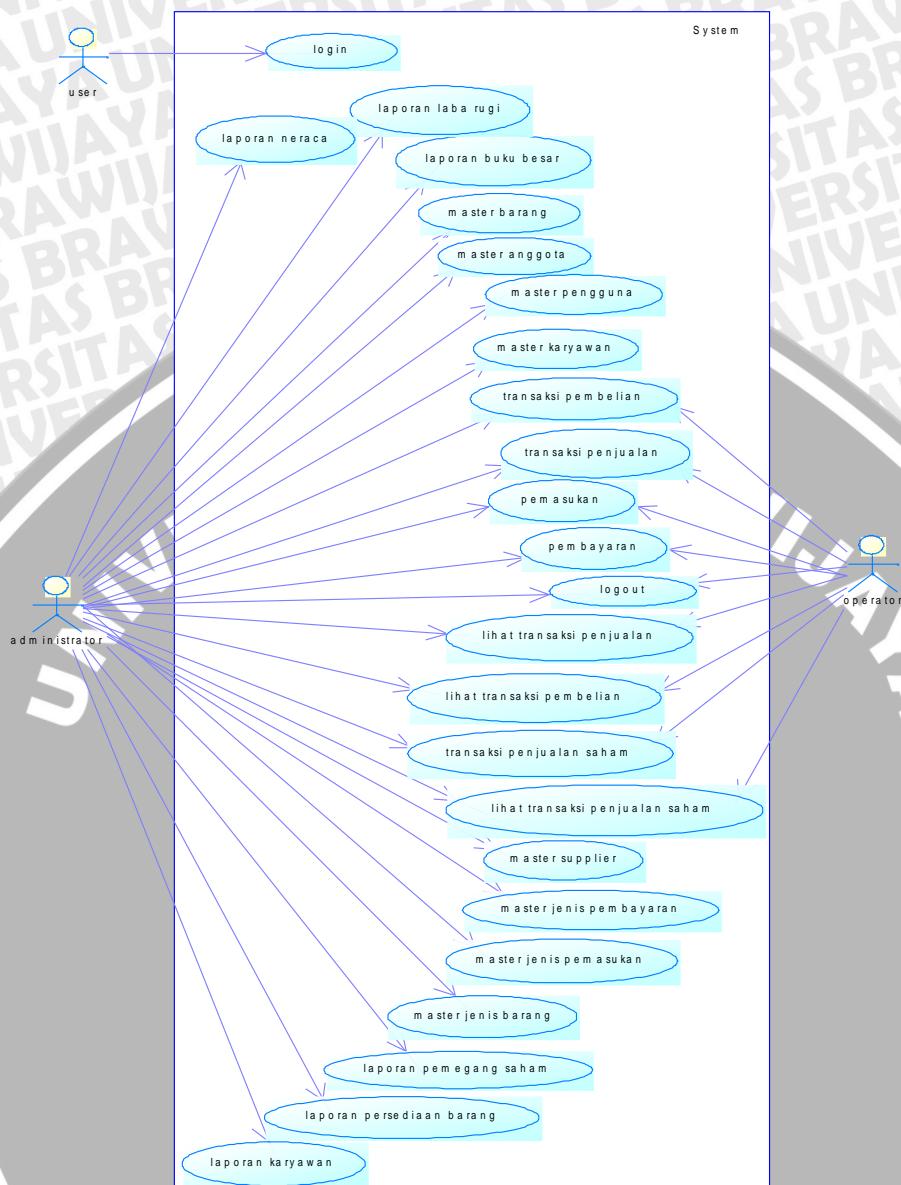
17	Sistem harus menyediakan fasilitas untuk mengubah data account berupa username dan password	Administrator	Master account
18	Sistem menyediakan fasilitas logout bagi user yang telah melakukan login,dalam hal ini administrator dan operator	Operator, Administrator	Logout
19	Sistem dibangun dan dikembangkan dengan bahasa pemrograman Java		
20	Menggunakan basis data MySQL		
21	Metode <i>object oriented programming</i>		

Tabel 4.2 Deskripsi daftar kebutuhan

Sumber : Analisis kebutuhan

4.1.3 **Diagram Use Case**

Kebutuhan-kebutuhan fungsional yang diperlukan oleh pengguna dan harus disediakan oleh sistem akan dimodelkan pada diagram *use case*. *Use case* menspesifikasi perilaku system (atau bagian dari system secara keseluruhan) dan merupakan deskripsi dari sekumpulan aksi-aksi yang diharapkan oleh calon pengguna system/perangkat lunak yang akan kita kembangkan. Gambar 4.1 merupakan diagram *use case* dari sistem ini.



Gambar 4.1 Diagram *use case*

Sumber: Perancangan

Secara lebih mendetail, masing-masing *use case* yang terdapat pada diagram *use case*, dijabarkan dalam skenario *use case*, akan diberikan nama *use case*, aktor yang berhubungan dengan *use case* tersebut, tujuan dari *use case*, deskripsi global tentang *use case*, pra-kondisi yang harus dipenuhi dan post-kondisi yang diharapkan setelah berjalannya fungsional *use case*. Selain itu juga

akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh aktor (aliran utama), serta kejadian alternatif yang akan terjadi jika suatu kondisi tidak bisa terpenuhi (aliran alternatif).

Tabel 4.3 Skenario *use case* login

Use case Name	Login
Use case ID	UC-01
Primary Actor	User
Secondary Actor	N/A
Brief Description	User yang menggunakan aplikasi harus melalui proses autentifikasi, dan hanya valid user yang berhak menggunakan aplikasi ini
Precondition	Menampilkan halaman login
Flow of Events	<ol style="list-style-type: none">1. User memasukan username dan password.2. Sistem melakukan proses autentifikasi.3. Jika benar, maka valid user dapat menggunakan aplikasi.
Post condition	Administrator atau operator object exist
Alternative Flows and Exception	<ol style="list-style-type: none">3. Sistem menampilkan pesan kesalahan jika username dan password tidak terdaftar dalam basis data.2. Sistem menampilkan pesan kesalahan jika penulisan username dan atau password tidak lengkap.1. User menekan tombol batal.2. Sistem akan menampilkan menu utama dalam keadaan tidak aktif.
Nonbehavioral Requirement	N/A

Tabel 4.4 Skenario *use case* master barang

Use case Name	Master barang
Use case ID	UC-02
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data barang
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu master dan menu item barang Sistem menampilkan form master barang Administrator menekan tombol tambah dan melakukan pencatatan data barang. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data barang
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol batal. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. Administrator memilih data barang dan menekan tombol ubah atau popup menu ubah. Sistem menampilkan data barang yang akan diubah. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data barang. Administrator menekan tombol keluar Sistem menutup form master barang.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data barang

Tabel 4.5 Skenario *use case* master jenis barang

Use case Name	Master jenis barang
---------------	---------------------

Use case ID	UC-03
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data jenis barang
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu master dan menu item jenis barang. 2. Sistem menampilkan form master jenis barang. 3. Administrator menekan tombol tambah dan melakukan pencatatan data jenis barang. 4. Administrator menekan tombol simpan. 5. Sistem melakukan proses penyimpanan data jenis barang.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Administrator menekan tombol batal. 5. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. 3. Administrator memilih data jenis barang dan menekan tombol ubah atau popup menu ubah. 4. Sistem menampilkan data jenis barang yang akan diubah. 5. Administrator menekan tombol simpan. 6. Sistem melakukan proses penyimpanan data jenis barang. 7. Administrator menekan tombol keluar. 8. Sistem menutup form master jenis barang.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data jenis barang

Tabel 4.6 Skenario use case master karyawan

Use case Name	Master karyawan
Use case ID	UC-04

Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data karyawan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu master dan menu item karyawan. 2. Sistem menampilkan form karyawan. 3. Administrator menekan tombol tambah dan melakukan pencatatan data karyawan. 4. Administrator menekan tombol simpan. 5. Sistem melakukan proses penyimpanan data karyawan.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Administrator menekan tombol batal. 5. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. 3. Administrator memilih data karyawan dan menekan tombol ubah atau popup menu ubah. 4. Sistem menampilkan data karyawan yang akan diubah. 5. Administrator menekan tombol simpan. 6. Sistem melakukan proses penyimpanan data karyawan. 7. Administrator menekan tombol keluar. 8. Sistem menutup form master karyawan.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data karyawan

Tabel 4.7 Skenario *use case* master pengguna

Use case Name	Master pengguna
Use case ID	UC-05
Primary Actor	Administrator

Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data pengguna
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu master dan menu item operator. 2. Sistem menampilkan form master operator. 3. Administrator menekan tombol tambah dan melakukan pencatatan data pengguna. 4. Administrator menekan tombol simpan. 5. Sistem melakukan proses penyimpanan data pengguna.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Administrator menekan tombol batal. 5. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. 3. Administrator memilih data pengguna dan menekan tombol ubah atau popup menu ubah. 4. Sistem menampilkan data pengguna yang akan diubah. 5. Administrator menekan tombol simpan. 6. Sistem melakukan proses penyimpanan data pengguna. 7. Administrator menekan tombol keluar. 8. Sistem menutup form master operator.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data pengguna

Tabel 4.8 Skenario *use case* master jenis pembayaran

Use case Name	Master jenis pembayaran
Use case ID	UC-06
Primary Actor	Administrator
Secondary Actor	N/A

Brief Description	Administrator melakukan pencatatan data jenis pembayaran
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu master dan menu item jenis pembayaran. Sistem menampilkan form master jenis pembayaran. Administrator menekan tombol tambah dan melakukan pencatatan data jenis pembayaran. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data jenis pembayaran.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol batal. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. Administrator memilih data jenis pembayaran dan menekan tombol ubah atau popup menu ubah. Sistem menampilkan data jenis pembayaran yang akan diubah. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data jenis pembayaran. Administrator menekan tombol keluar. Sistem menutup form master jenis pembayaran.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data jenis pembayaran

Tabel 4.9 Skenario *use case* master jenis pemasukan

Use case Name	Master jenis pemasukan
Use case ID	UC-07
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data jenis pemasukan

Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu master dan menu item jenis pemasukan. 2. Sistem menampilkan form master jenis pemasukan. 3. Administrator menekan tombol tambah dan melakukan pencatatan data jenis pemasukan. 4. Administrator menekan tombol simpan. 5. Sistem melakukan proses penyimpanan data jenis pemasukan.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Administrator menekan tombol batal. 5. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. 3. Administrator memilih data jenis pemasukan dan menekan tombol ubah atau popup menu ubah. 4. Sistem menampilkan data jenis pemasukan yang akan diubah. 5. Administrator menekan tombol simpan. 6. Sistem melakukan proses penyimpanan data jenis pemasukan. 7. Administrator menekan tombol keluar. 8. Sistem menutup form master jenis pemasukan.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data jenis pemasukan

Tabel 4.10 Skenario *use case* master supplier

Use case Name	Master supplier
Use case ID	UC-08
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data supplier
Precondition	Menu utama dalam keadaan aktif

Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu master dan menu item supplier. Sistem menampilkan form master supplier. Administrator menekan tombol tambah dan melakukan pencatatan data supplier. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data supplier.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol batal. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. Administrator memilih data supplier dan menekan tombol ubah atau popup menu ubah. Sistem menampilkan data supplier yang akan diubah. Administrator menekan tombol simpan. Sistem melakukan proses penyimpanan data supplier. Administrator menekan tombol keluar. Sistem menutup form master supplier.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data supplier

Tabel 4.11 Skenario *use case* master anggota

Use case Name	Master anggota
Use case ID	UC-09
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Administrator melakukan pencatatan data anggota
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu master dan menu item anggota. Sistem menampilkan form master anggota. Administrator menekan tombol tambah dan melakukan pencatatan data anggota.

	4. Administrator menekan tombol simpan. 5. Sistem melakukan proses penyimpanan data anggota.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	4. Administrator menekan tombol batal. 5. Sistem membersihkan data dan menonaktifkan tombol ubah dan simpan. 3. Administrator memilih data anggota dan menekan tombol ubah atau popup menu ubah. 4. Sistem menampilkan data anggota yang akan diubah. 5. Administrator menekan tombol simpan. 6. Sistem melakukan proses penyimpanan data anggota. 7. Administrator menekan tombol keluar. 8. Sistem menutup form master anggota.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses pencatatan data anggota

Tabel 4.12 Skenario *use case* transaksi pembelian

Use case Name	Transaksi pembelian
Use case ID	UC-10
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pencatatan data transaksi pembelian
Precondition	Menu utama dalam keadaan aktif
Flow of Events	1. Administrator memilih menu transaksi dan menu item pembelian. 2. Sistem menampilkan form transaksi pembelian. 3. Administrator melakukan pencatatan data transaksi dan menekan tombol tambah. 4. Sistem menampilkan data transaksi dalam daftar pembelian barang. 5. Administrator menekan tombol proses transaksi. 6. Sistem melakukan proses penyimpanan data transaksi

	pembelian.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<p>5. Administrator memilih data transaksi pembelian dan menekan tombol ubah atau popup menu ubah.</p> <p>6. Sistem menampilkan data transaksi pembelian yang akan diubah.</p> <p>7. Administrator menekan tombol proses transaksi.</p> <p>8. Sistem melakukan proses penyimpanan data transaksi pembelian.</p> <p>5. Administrator memilih data transaksi pembelian dan menekan tombol hapus.</p> <p>6. Sistem menghapus data transaksi pembelian.</p> <p>7. Administrator menekan tombol proses transaksi.</p> <p>8. Sistem melakukan proses penyimpanan data transaksi pembelian.</p>
Nonbehavioral Requirement	Operator dapat melakukan proses transaksi pembelian

Tabel 4.13 Skenario *use case* transaksi penjualan

Use case Name	Transaksi penjualan
Use case ID	UC-11
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pencatatan data transaksi penjualan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<p>1. Administrator memilih menu transaksi dan menu item penjualan.</p> <p>2. Sistem menampilkan form transaksi penjualan.</p> <p>3. Administrator melakukan pencatatan data transaksi dan menekan tombol tambah.</p> <p>4. Sistem menampilkan data transaksi dalam daftar</p>

	<p>penjualan barang.</p> <ol style="list-style-type: none"> 5. Administrator menekan tombol proses transaksi. 6. Sistem melakukan proses penyimpanan data transaksi penjualan.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 5. Administrator memilih data transaksi penjualan dan menekan tombol ubah atau popup menu ubah. 6. Sistem menampilkan data transaksi penjualan yang akan diubah. 7. Administrator menekan tombol proses transaksi. 8. Sistem melakukan proses penyimpanan data transaksi penjualan. <ol style="list-style-type: none"> 5. Administrator memilih data transaksi penjualan dan menekan tombol hapus. 6. Sistem menghapus data transaksi penjualan. 7. Administrator menekan tombol proses transaksi. 8. Sistem melakukan proses penyimpanan data transaksi penjualan.
Nonbehavioral Requirement	Operator dapat melakukan proses transaksi penjualan

Tabel 4.14 Skenario *use case* transaksi penjualan saham

Use case Name	Transaksi penjualan saham
Use case ID	UC-12
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pencatatan data transaksi penjualan saham
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu transaksi dan menu item penjualan saham. 2. Sistem menampilkan form transaksi penjualan

	<p>saham.</p> <ol style="list-style-type: none"> 3. Administrator melakukan pencatatan data transaksi dan menekan tombol tambah. 4. Sistem menampilkan data transaksi dalam daftar penjualan saham. 5. Administrator menekan tombol proses transaksi. 6. Sistem melakukan proses penyimpanan data transaksi penjualan saham.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<p>5. Administrator memilih data transaksi penjualan saham dan menekan tombol ubah atau popup menu ubah.</p> <p>6. Sistem menampilkan data transaksi penjualan saham yang akan diubah.</p> <p>7. Administrator menekan tombol proses transaksi.</p> <p>8. Sistem melakukan proses penyimpanan data transaksi penjualan saham.</p> <p>5. Administrator memilih data transaksi penjualan saham dan menekan tombol hapus.</p> <p>6. Sistem menghapus data transaksi penjualan saham.</p> <p>7. Administrator menekan tombol proses transaksi.</p> <p>8. Sistem melakukan proses penyimpanan data transaksi penjualan saham.</p>
Nonbehavioral Requirement	Operator dapat melakukan proses transaksi penjualan saham

Tabel 4.15 Skenario *use case* pembayaran

Use case Name	pembayaran
Use case ID	UC-13
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pencatatan data pembayaran

Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu transaksi dan menu item pembayaran. 2. Sistem menampilkan form transaksi pembayaran. 3. Administrator melakukan pencatatan data transaksi dan menekan tombol tambah. 4. Sistem menampilkan data transaksi dalam daftar pembayaran. 5. Administrator menekan tombol proses transaksi. 6. Sistem melakukan proses penyimpanan data transaksi pembayaran.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> 5. Administrator memilih data pembayaran dan menekan tombol ubah atau popup menu ubah. 6. Sistem menampilkan data pembayaran yang akan diubah. 7. Administrator menekan tombol proses transaksi. 8. Sistem melakukan proses penyimpanan data pembayaran. 5. Administrator memilih data pembayaran dan menekan tombol hapus. 6. Sistem menghapus data pembayaran. 7. Administrator menekan tombol proses transaksi. 8. Sistem melakukan proses penyimpanan data pembayaran.
Nonbehavioral Requirement	Operator dapat melakukan proses transaksi pembayaran

Tabel 4.16 Skenario *use case* pemasukan

Use case Name	pemasukan
Use case ID	UC-14
Primary Actor	Administrator
Secondary Actor	Operator

Brief Description	Administrator atau operator melakukan pencatatan data pemasukan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu transaksi dan menu item pemasukan. Sistem menampilkan form transaksi pemasukan. Administrator melakukan pencatatan data transaksi dan menekan tombol tambah. Sistem menampilkan data transaksi dalam daftar pemasukan. Administrator menekan tombol proses transaksi. Sistem melakukan proses penyimpanan data transaksi pemasukan.
Post condition	Setiap data yang telah tercatat tersimpan dalam basis data
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator memilih data pemasukan dan menekan tombol ubah atau popup menu ubah. Sistem menampilkan data pemasukan yang akan diubah. Administrator menekan tombol proses transaksi. Sistem melakukan proses penyimpanan data pemasukan. Administrator memilih data pemasukan dan menekan tombol hapus. Sistem menghapus data pemasukan. Administrator menekan tombol proses transaksi. Sistem melakukan proses penyimpanan data pemasukan.
Nonbehavioral Requirement	Operator dapat melakukan proses transaksi pemasukan

Tabel 4.17 Skenario *use case* lihat transaksi pembelian

Use case Name	Lihat transaksi pembelian
Use case ID	UC-15
Primary Actor	Administrator

Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pengecekan transaksi pembelian
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu transaksi dan menu item lihat pembelian. 2. Sistem menampilkan form lihat pembelian. 3. Administrator memasukkan nilai nomer pembelian dan menekan tombol proses. 4. Sistem menampilkan data transaksi pembelian dalam daftar transaksi pembelian.
Post condition	Sistem menampilkan data transaksi pembelian
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Sistem menampilkan pesan peringatan apabila data tidak ada dalam basis data 3. Administrator menekan tombol keluar. 4. Sistem akan menutup form lihat pembelian.
Nonbehavioral Requirement	Operator dapat melakukan proses lihat transaksi pembelian

Tabel 4.18 Skenario *use case* lihat transaksi penjualan

Use case Name	Lihat transaksi penjualan
Use case ID	UC-16
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pengecekan transaksi penjualan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu transaksi dan menu item lihat penjualan.

	<ol style="list-style-type: none"> 2. Sistem menampilkan form lihat penjualan. 3. Administrator memasukkan nilai nomer penjualan dan menekan tombol proses. 4. Sistem menampilkan data transaksi penjualan dalam daftar transaksi penjualan.
Post condition	Sistem menampilkan data transaksi pembelian
Alternative Flows and Exception	<ol style="list-style-type: none"> 4. Sistem menampilkan pesan peringatan apabila data tidak ada dalam basis data 3. Administrator menekan tombol keluar. 4. Sistem akan menutup form lihat pembelian.
Nonbehavioral Requirement	Operator dapat melakukan proses lihat transaksi penjualan

Tabel 4.19 Skenario *use case* lihat transaksi penjualan saham

Use case Name	Lihat transaksi penjualan saham
Use case ID	UC-17
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	Administrator atau operator melakukan pengecekan transaksi penjualan saham
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> 1. Administrator memilih menu transaksi dan menu item lihat penjualan saham. 2. Sistem menampilkan form lihat penjualan saham. 3. Administrator memasukkan nilai nomer saham dan menekan tombol proses. 4. Sistem menampilkan data transaksi penjualan saham dalam daftar penjualan saham.
Post condition	Sistem menampilkan data transaksi penjualan saham
Alternative Flows	<ol style="list-style-type: none"> 4. Sistem menampilkan pesan peringatan apabila data

and Exception	tidak ada dalam basis data 3. Administrator menekan tombol keluar. 4. Sistem akan menutup form lihat penjualan saham.
Nonbehavioral Requirement	Operator dapat melakukan proses lihat transaksi penjualan saham

Tabel 4.20 Skenario *use case* laporan buku besar

Use case Name	Laporan buku besar
Use case ID	UC-18
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan buku besar berdasarkan tahun periode yang diinginkan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	1. Administrator memilih menu laporan dan menu item laporan buku besar . 2. Sistem menampilkan form laporan buku besar. 3. Administrator memilih tahun periode laporan buku besar dan menekan tombol cetak. 4. Sistem mencetak laporan buku besar.
Post condition	Sistem menampilkan laporan buku besar
Alternative Flows and Exception	3. Administrator menekan tombol keluar. 4. Sistem akan menutup form laporan buku besar.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.21 Skenario *use case* laporan neraca

Use case Name	Laporan neraca
---------------	----------------

Use case ID	UC-19
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan neraca berdasarkan tahun periode yang diinginkan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu laporan dan menu item laporan neraca. Sistem menampilkan form laporan neraca. Administrator memilih tahun periode laporan neraca dan menekan tombol cetak. Sistem mencetak laporan neraca.
Post condition	Sistem menampilkan laporan neraca
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol keluar. Sistem akan menutup form laporan neraca.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.22 Skenario *use case* laporan laba rugi

Use case Name	Laporan laba rugi
Use case ID	UC-20
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan laba rugi berdasarkan tahun periode yang diinginkan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu laporan dan menu item

	laporan laba rugi. 2. Sistem menampilkan form laporan laba rugi. 3. Administrator memilih tahun periode laporan laba rugi dan menekan tombol cetak. 4. Sistem mencetak laporan laba rugi.
Post condition	Sistem menampilkan laporan laba rugi
Alternative Flows and Exception	3. Administrator menekan tombol keluar. 4. Sistem akan menutup form laporan laba rugi.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.23 Skenario *use case* laporan karyawan

Use case Name	Laporan pemegang saham
Use case ID	UC-21
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan karyawan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	1. Administrator memilih menu laporan dan menu item laporan karyawan. 2. Sistem menampilkan form laporan laba rugi. 3. Administrator menekan tombol cetak. 4. Sistem mencetak laporan karyawan.
Post condition	Sistem menampilkan laporan karyawan
Alternative Flows and Exception	3. Administrator menekan tombol keluar. 4. Sistem akan menutup form laporan karyawan.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.24 Skenario *use case* laporan pemegang saham

Use case Name	Laporan pemegang saham
Use case ID	UC-22
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan pemegang saham berdasarkan tahun periode yang diinginkan
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu laporan dan menu item laporan pemegang saham. Sistem menampilkan form laporan pemegang saham. Administrator menekan tombol cetak. Sistem mencetak laporan pemegang saham.
Post condition	Sistem menampilkan laporan pemegang saham
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol keluar. Sistem akan menutup form laporan pemegang saham.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.25 Skenario *use case* laporan persediaan barang

Use case Name	Laporan persediaan barang
Use case ID	UC-23
Primary Actor	Administrator
Secondary Actor	N/A
Brief Description	Sistem memberikan daftar laporan persediaan barang berdasarkan periode yang diinginkan

Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu laporan dan menu item laporan persediaan barang. Sistem menampilkan form laporan persediaan barang. Administrator memilih periode laporan persediaan barang dan menekan tombol cetak. Sistem mencetak laporan persediaan barang.
Post condition	Sistem menampilkan laporan persediaan barang
Alternative Flows and Exception	<ol style="list-style-type: none"> Administrator menekan tombol keluar. Sistem akan menutup form laporan persediaan barang.
Nonbehavioral Requirement	Hanya administrator yang dapat melakukan proses ini

Tabel 4.6 Skenario use case logout

Use case Name	Logout
Use case ID	UC-24
Primary Actor	Administrator
Secondary Actor	Operator
Brief Description	User yang telah login, dalam hal ini administrator dan operator akan keluar dari sistem
Precondition	Menu utama dalam keadaan aktif
Flow of Events	<ol style="list-style-type: none"> Administrator memilih menu file dan menu item logout. Sistem akan menonaktifkan fungsi sistem.
Post condition	Sistem bernilai false
Alternative Flows	N/A

and Exception	
Nonbehavioral Requirement	Administrator dan operator dapat melakukan proses ini

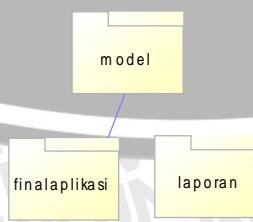
4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak ditujukan untuk memberikan gambaran secara umum mengenai perangkat lunak yang akan dibuat. Hal ini berguna untuk menunjang pembuatan perangkat lunak sehingga kebutuhan perangkat lunak tersebut dapat terpenuhi.

Perancangan perangkat lunak pada tugas akhir ini menggunakan pendekatan desain berorientasi objek yang direpresentasikan dengan menggunakan UML (*Unified Modeling Language*). Pada proses desain dilakukan identifikasi terhadap klas-klas yang dibutuhkan yang dimodelkan dalam *class diagram* dan hubungan interaksi antar elemen (objek) yang telah diidentifikasi, dimodelkan dalam *sequence diagram*.

4.2.1 Diagram klas (*Class Diagram*)

Diagram klas memberikan gambaran pemodelan elemen-elemen klas yang membentuk sebuah sistem perangkat lunak mulai dari atribut-atribut serta method-methodnya dan hubungannya dengan klas-klas lain dalam sebuah sistem. Pada sistem ini terdapat sejumlah klas yang saling membentuk relasi. Klas-klas yang akan membangun sistem ini dikelompokkan menjadi tiga paket seperti ditunjukkan dalam gambar 4.2

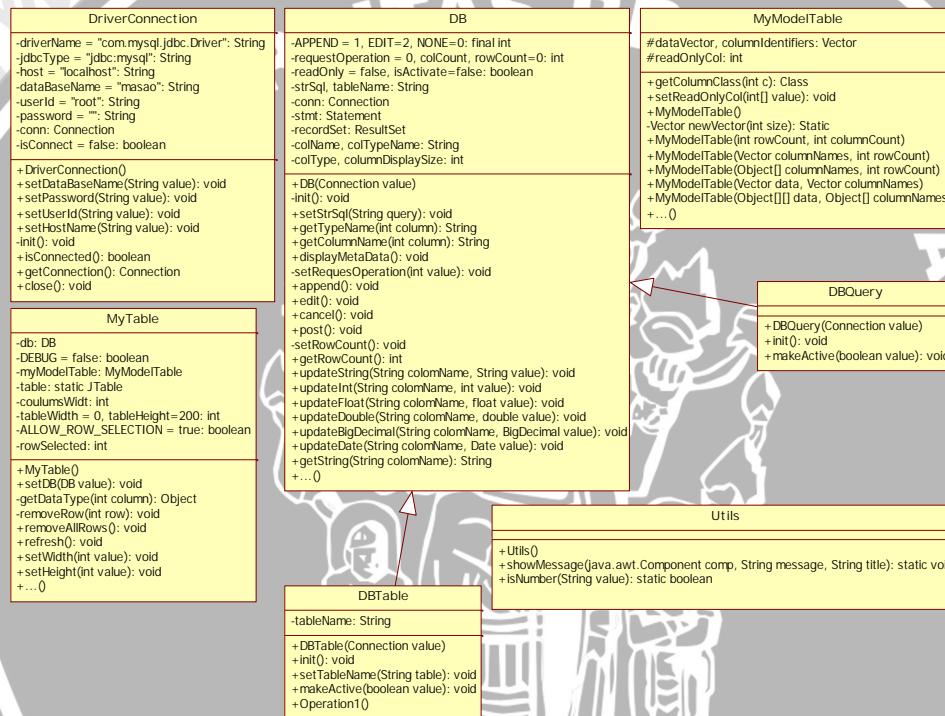


Gambar 4.2 Hubungan antar paket

Sumber: Perancangan

4.2.1.1 Klas Diagram Pada Paket Model

Pada paket model terdapat delapan buah klas, antara lain klas DriverConnection, DB, MyModelTable, MyTable, DBQuery, Utils, DBTable. Atribut dan operasi yang dimiliki oleh masing-masing klas tersebut digambarkan dalam Gambar 4.3.

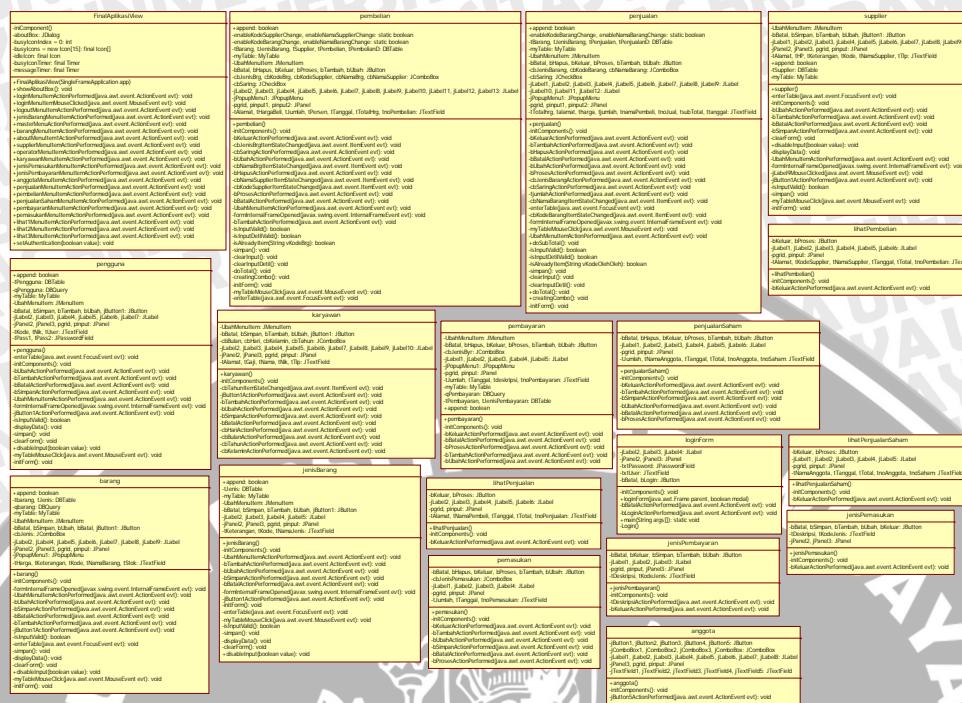


Gambar 4.3 Diagram klas pada paket model

Sumber: Perancangan

4.2.1.2 Klas Diagram Pada Paket finalaplikasi

Pada paket finalaplikasi terdapat delapan belas buah klas, antara lain klas pembelian, penjualan, barang, karyawan, jenisBarang, pengguna, anggota. Atribut dan operasi yang dimiliki oleh masing-masing klas tersebut digambarkan dalam Gambar 4.4.



Gambar 4.4 Diagram klas pada paket finalaplikasi

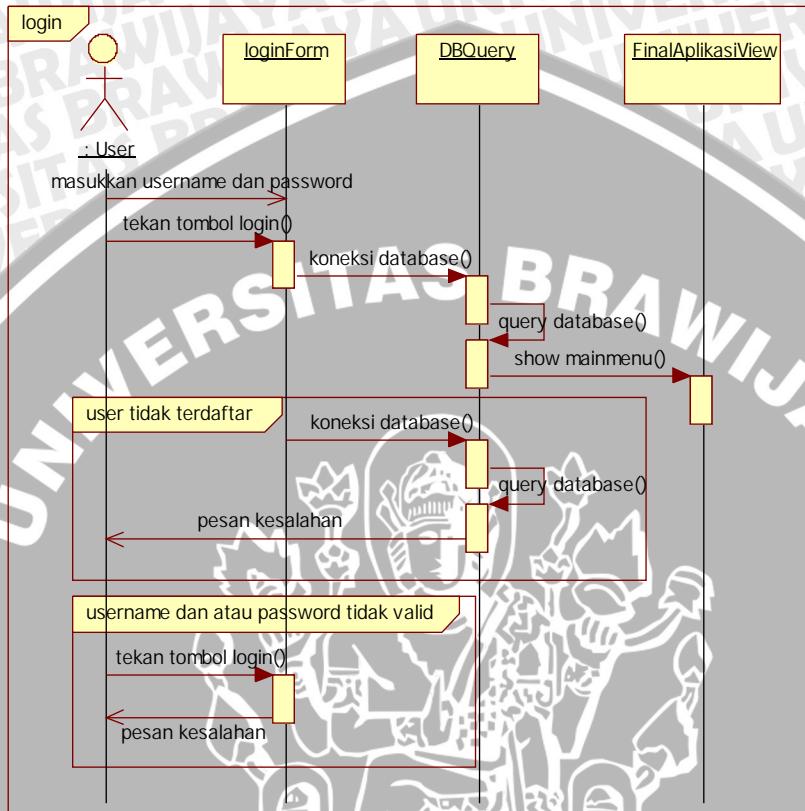
Sumber: Perancangan

4.2.2 Diagram Sekuensial (*Sequence Diagram*)

Pemodelan aliran jalannya proses interaksi antar objek atau klas yang disusun berdasarkan urutan waktu ditunjukkan oleh *sequence diagram*. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Diagram sekuensial (*sequence diagram*) disusun dengan mengambil acuan pada *use case* dan klas-klas yang membentuk fungsionalitas yang digambarkan pada *use case* tersebut.

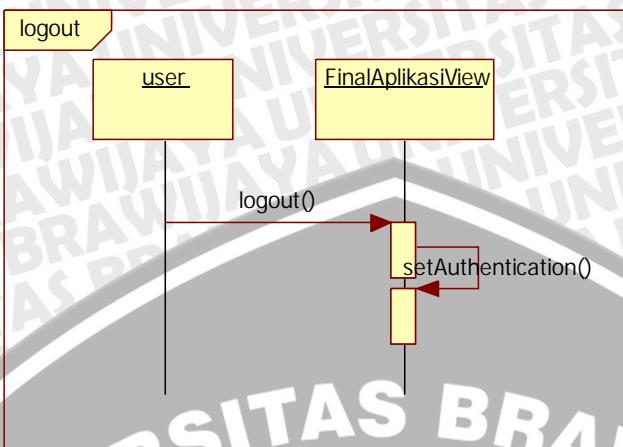
- Diagram sekuensial *use case login*

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang pengguna (*user*) melakukan login aplikasi.



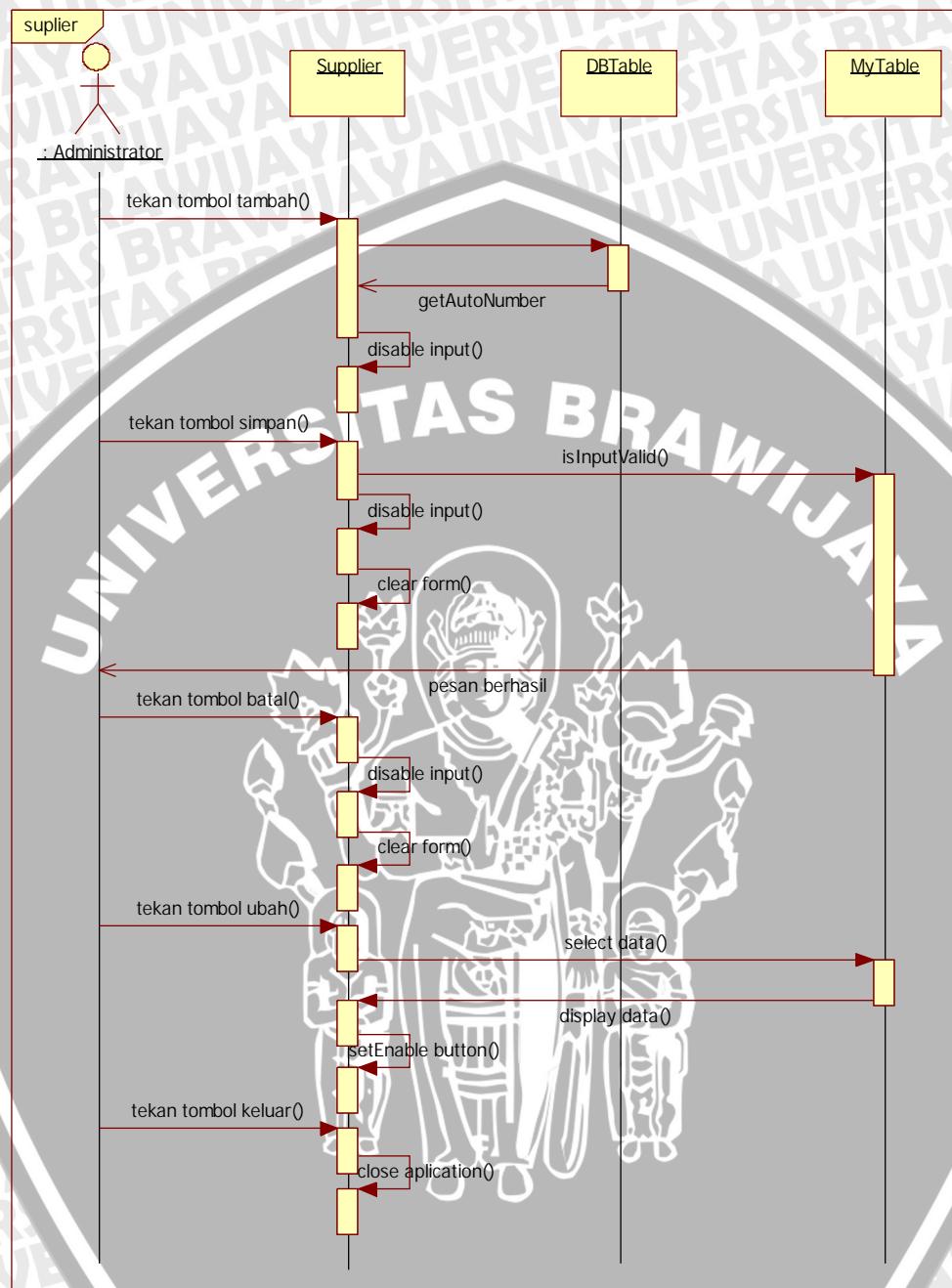
- **Diagram sekuensial *use case logout***

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang pengguna melakukan logout aplikasi.



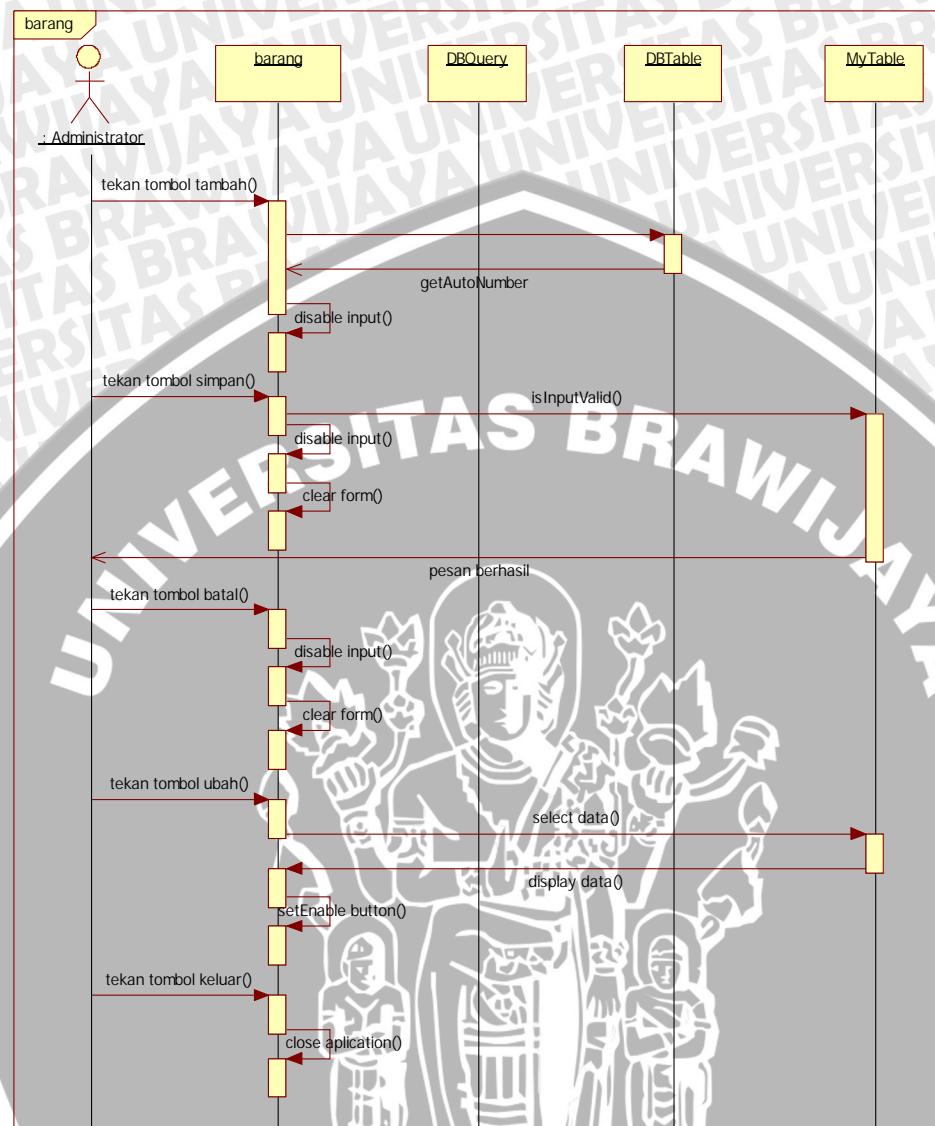
- **Diagram sekuensial *use case master supplier***

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pencatatan data supplier.



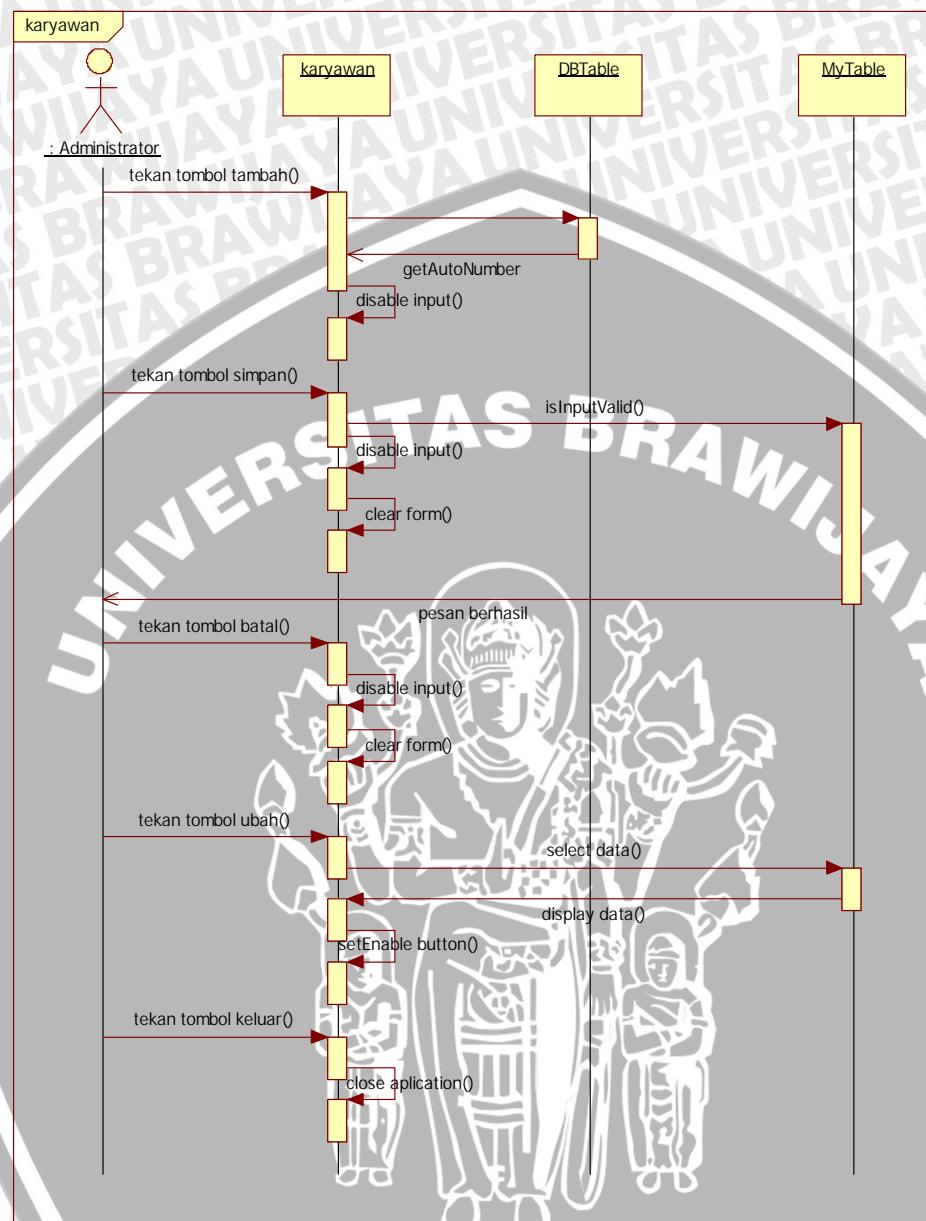
- **Diagram sekuensial *use case master barang***

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pencatatan data barang.



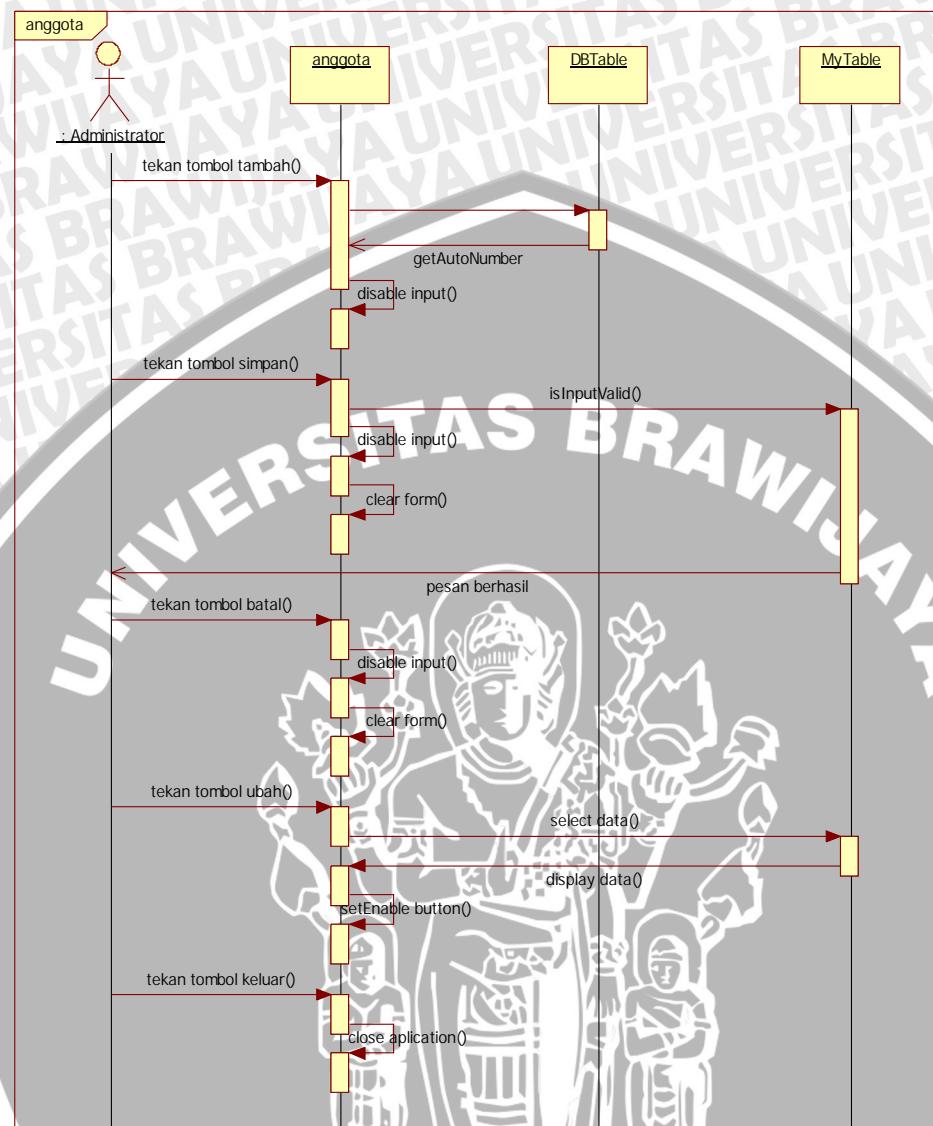
- **Diagram sekuensial *use case master karyawan***

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pencatatan data karyawan.



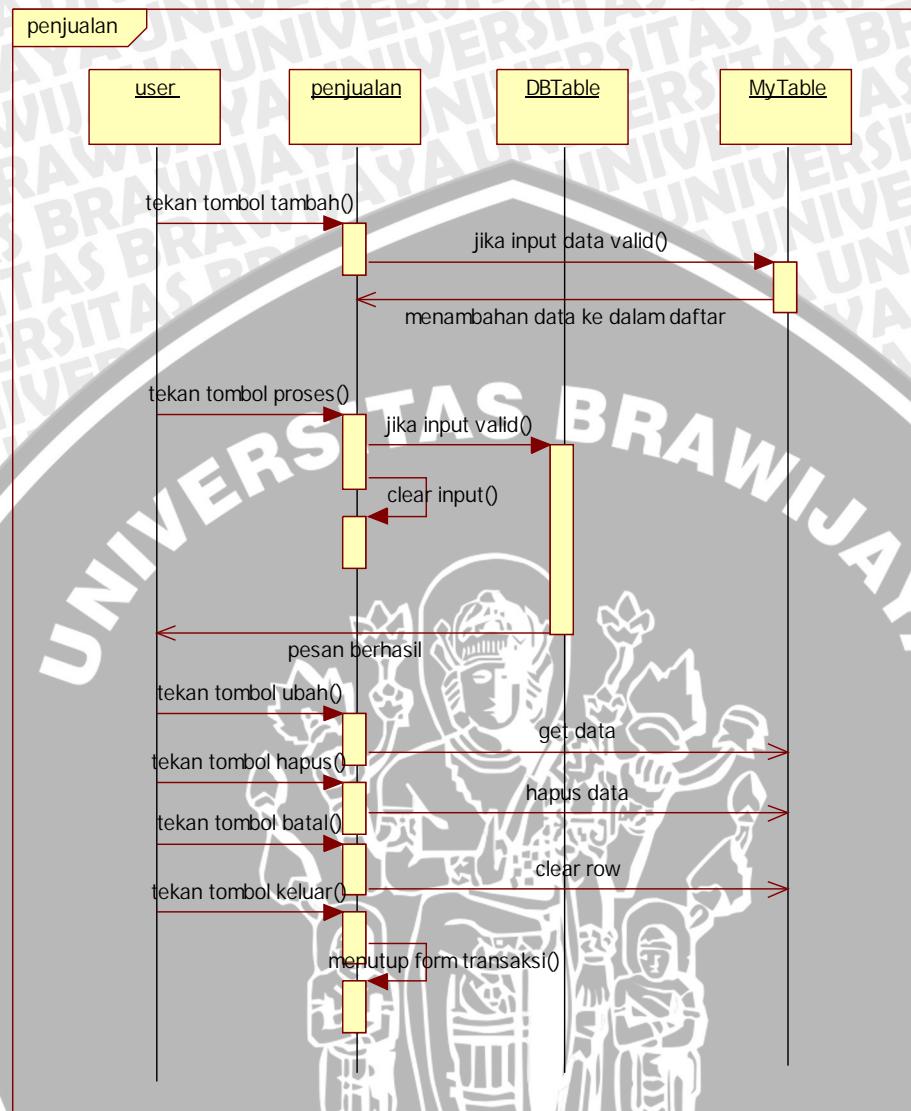
- **Diagram sekuensial *use case master anggota***

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pencatatan data anggota.



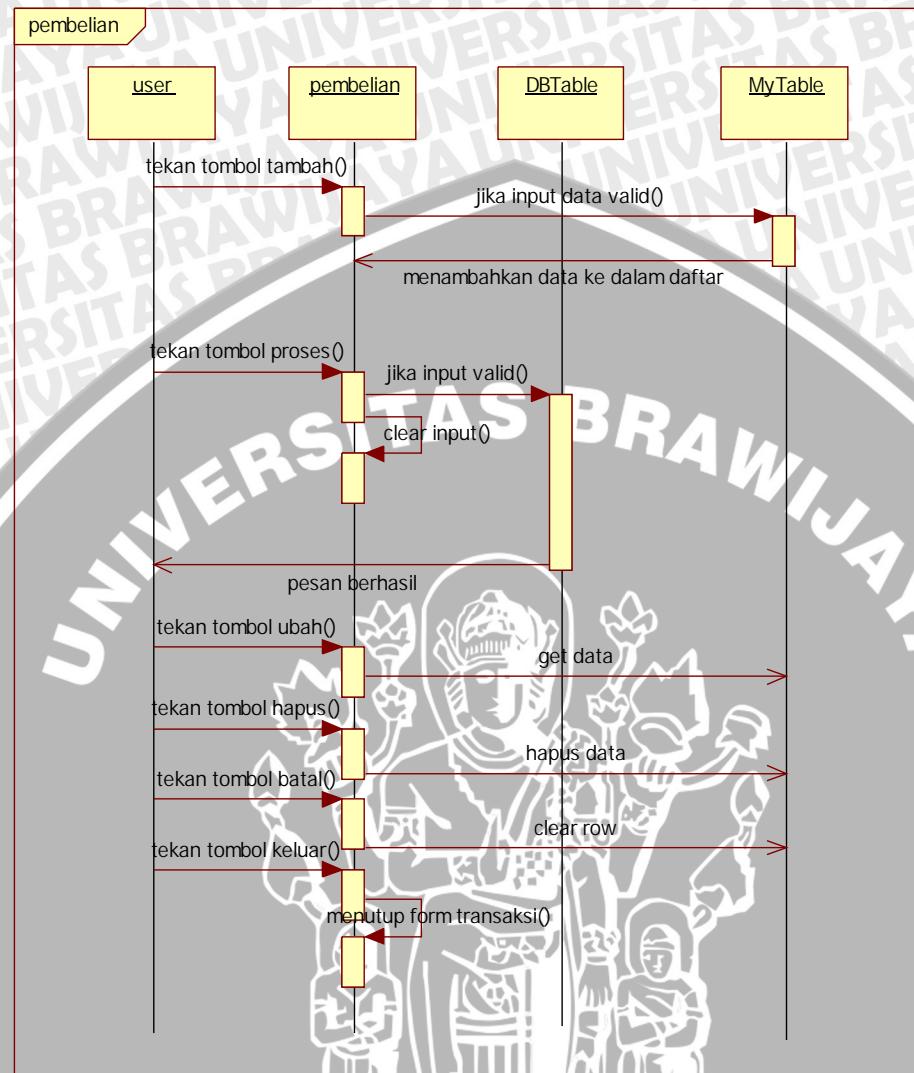
- **Diagram sekuensial *use case* transaksi penjualan**

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan transaksi penjualan.



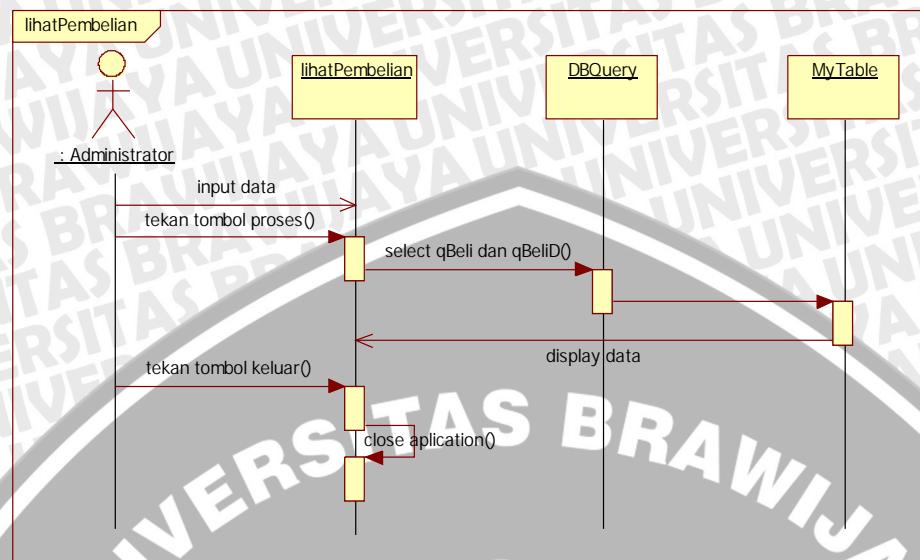
- **Diagram sekuensial *use case* transaksi pembelian**

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan transaksi data pembelian.



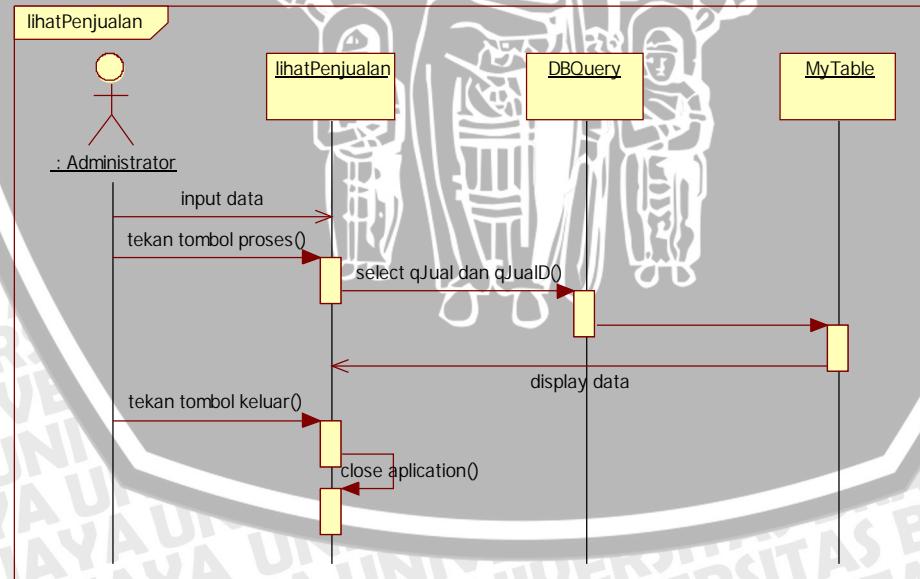
- **Diagram sekuenstrial use case transaksi lihat pembelian**

Diagram sekuenstrial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pengecekan transaksi data pembelian.



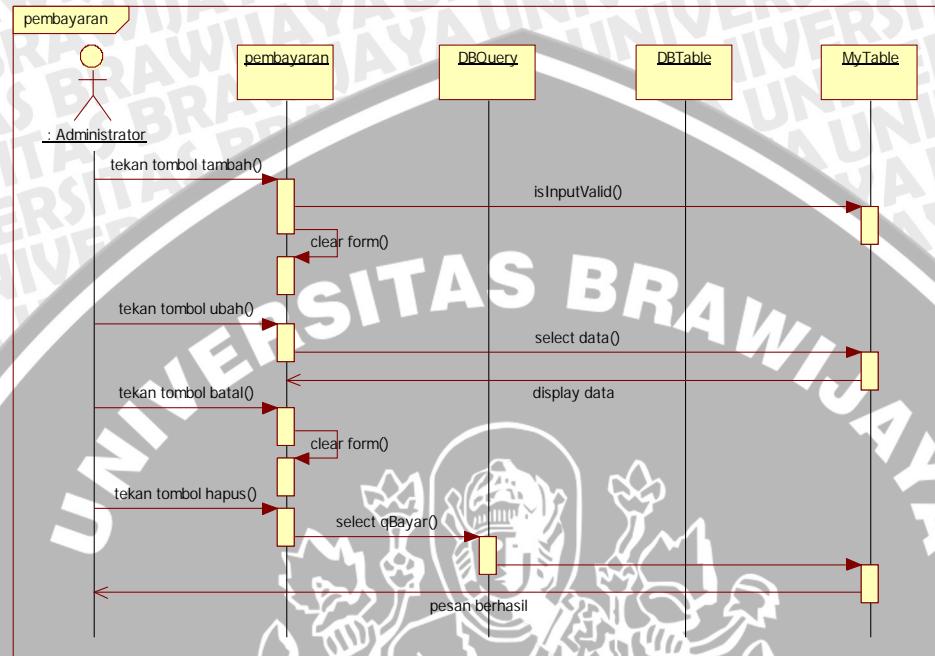
- **Diagram sekuensial *use case* transaksi lihat penjualan**

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan pengecekan transaksi data penjualan.



- **Diagram sekuensial *use case* transaksi pembayaran**

Diagram sekuensial ini menggambarkan interaksi yang terjadi ketika seorang user yang valid melakukan transaksi data pembayaran.

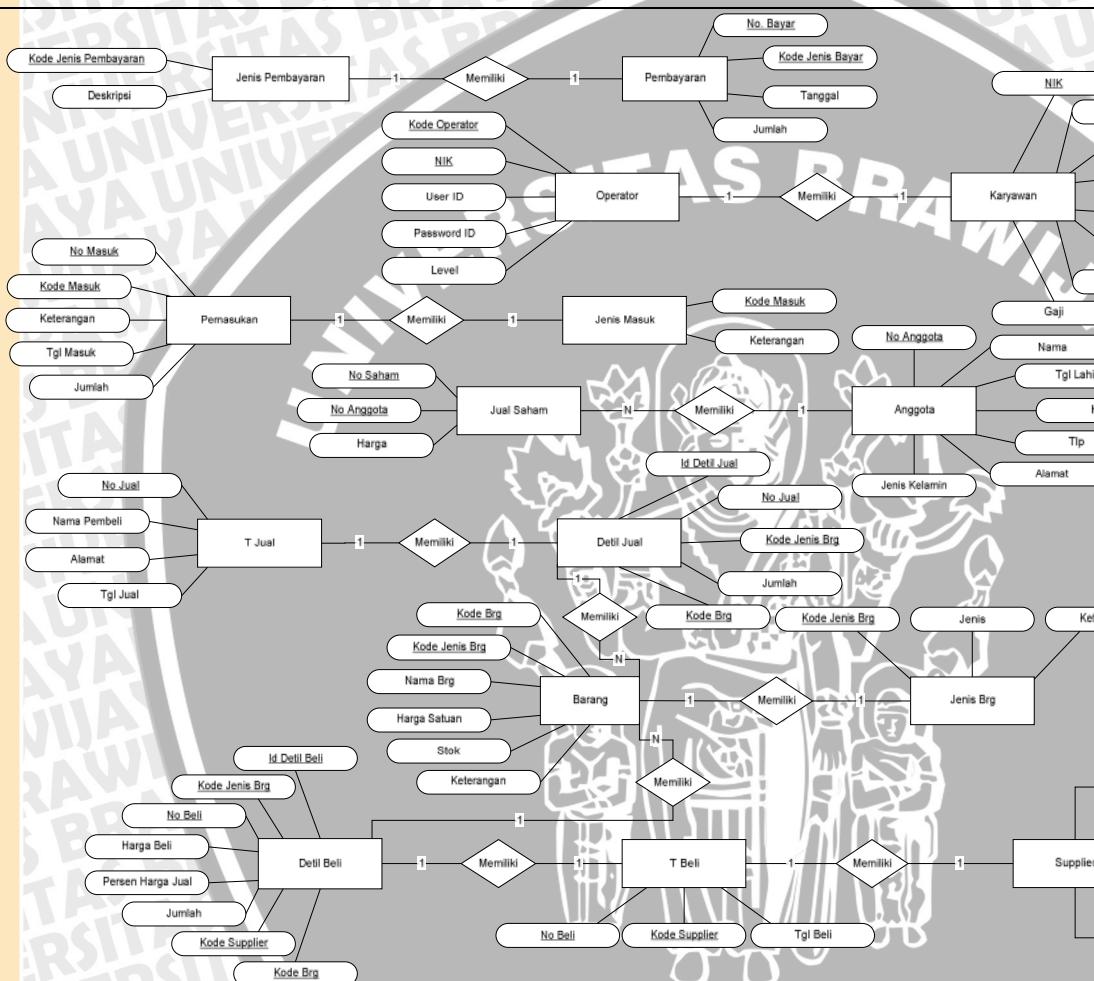


4.2.3 Basis Data (Database)

Perancangan basis data dilakukan agar Sistem Informasi Akuntansi ini mempunyai basis data yang effisien dalam penggunaan ruang penyimpanan, cepat dalam pengaksesan dan mudah dalam pemanipulasi data. Perancangan basis data dilakukan dengan merancang tabel ternormalisasi dan kemudian memodelkannya kedalam bentuk *Entity Relationship Diagram (Diagram ER)*.

4.2.3.1 Entity Relationship Diagram (Diagram ER)

Diagram ER digunakan untuk menggambarkan hubungan entitas antara satu tabel dan lainnya dengan memperlihatkan hubungan antar atribut yang akan dijadikan *key* untuk berelasi antar tabel.



Gambar 4.5 Entity relationship diagram

Sumber: Perancangan

4.2.3.1 Normalisasi Basis Data

Normalisasi adalah proses yang berkaitan dengan model data relational untuk mengorganisasi himpunan data dengan ketergantungan dan keterkaitan yang tinggi atau erat. Hasil dari proses normalisasi adalah himpunan-himpunan data dalam bentuk normal (*normal form*).

- Tabel anggota

Name	Type
no_anggota	varchar
nama	varchar
tgl_lhr	varchar
jenis_kelamin	char
alamat	varchar
tip	varchar
hp	varchar

- Tabel barang

Name	Type
kode_brg	char
kode_jenis_brg	char
nama_brg	varchar
harga_satuan	int
stok	smallint
keterangan	mediumtext

- Tabel karyawan

Name	Type
nik	char
nama_karyawan	varchar
tgl_lhr	varchar
kelamin	char
alamat	varchar
tip	varchar
gaji	smallint

- Tabel supplier

Name	Type
kode_supplier	char
nama	varchar
alamat	varchar
tlp	varchar
hp	varchar
keterangan	mediumtext

- Tabel operator

Name	Type
kode_operator	char
nik	char
user_id	varchar
password_id	varchar

- Tabel pembayaran

Name	Type
no_byr	char
kode_byr	char
keterangan	varchar
tgl	date
jumlah	smallint

- Tabel pemasukan

Name	Type
no_masuk	char
kode_masuk	char
keterangan	varchar
tgl_masuk	date
jumlah	smallint

- Tabel t_beli

Name	Type
► no_beli	char
kode_supplier	char
tgl_beli	date

- Tabel detil_beli

Name	Type
► id_detil_beli	int
no_beli	char
kode_jenis_brg	char
kode_brg	char
harga_beli	double
persen_harga_jual	smallint
jumlah	smallint

- Tabel t_jual

Name	Type
► no_jual	char
nama_pembeli	varchar
alamat	varchar
tgl_jual	date

- Tabel detil_jual

Name	Type
► id_detil_jual	int
no_jual	char
kode_brg	char
jumlah	smallint

- Tabel jenis_masuk

Name	Type
► kode_masuk	char
keterangan	varchar

- Tabel jual_saham

Name	Type
no_saham	int
no_anggota	varchar
harga	smallint

- Tabel jenis_brg

Name	Type
kode_jenis_brg	char
jenis	varchar
keterangan	varchar

- Tabel jenis_byr

Name	Type
kode_byr	char
keterangan	varchar



BAB V

IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi perangkat lunak berdasarkan hasil yang telah didapatkan dari analisis kebutuhan dan proses perancangan perangkat lunak dibuat. Pembahasan terdiri dari penjelasan tentang spesifikasi sistem, batasan-batasan dalam implementasi, implementasi tiap klas pada file program, implementasi algoritma, implementasi antarmuka dan beberapa kendala dalam implementasi.

5.1 Spesifikasi Sistem

Hasil analisis kebutuhan dan perancangan perangkat lunak yang telah diuraikan pada Bab 4 menjadi acuan untuk melakukan implementasi menjadi sebuah sistem Informasi Akuntansi yang dapat berfungsi sesuai dengan kebutuhan. Spesifikasi sistem diimplementasikan pada spesifikasi perangkat keras dan perangkat lunak.

5.1.1 Spesifikasi Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk implementasi aplikasi ini ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi perangkat keras

Spesifikasi Perangkat Keras	
Prosesor	Intel Core2Duo CPU T7100 @ 1.8 GHz
Memori (RAM)	2.0 GB DDR2
Hardisk	Hitachi HTS541612J9SA00 ATA Device Kapasitas 120 GB
Mother Board	Acer, inc Prespa
VGA Card	Mobile Intel(R) 965 Express Chipset Family

Sumber : Implementasi

5.1.2 Spesifikasi Perangkat Lunak (*Software*)

Untuk membangun sistem ini dibutuhkan beberapa perangkat lunak yang dapat dilihat pada Tabel 5.2

Tabel 5.2 Spesifikasi perangkat lunak

Spesifikasi Perangkat Lunak	
Sistem Operasi	Windows Vista Ultimate SP1 version 2007
Bahasa pemrograman	Java 2 Standart Edition
Tools Pemrograman	JDK 1.5.0
IDE (<i>Integrated Development Environment</i>)	Netbeans 6.0

Sumber : Implementasi

5.2 Implementasi Klas Pada File Program

Setiap klas yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program *.java. tabel 5.3 menjelaskan mengenai pasangan antara klas dengan *file* program yang digunakan untuk mengimplementasikannya.

Tabel 5.3 Implementasi klas pada kode program *.java

No.	Paket	Nama Klas	Nama <i>File</i> Program
1	finalaplikasi	FinalApplikasiView	FinalApplikasiView.java
2		FinalApplikasiAboutBox	FinalApplikasiAboutBox.java
3		FinalApplikasiApp	FinalApplikasiApp.java
4		anggota	anggota.java
5		barang	barang.java
6		jenisBarang	jenisBarang.java

7	model	jenisPemasukan	jenisPemasukan.java
8		jenisPembayaran	jenisPembayaran.java
9		karyawan	karyawan.java
10		lihatPembelian	lihatPembelian.java
11		lihatPenjualan	lihatPenjualan.java
12		lihatPenjualanSaham	lihatPenjualanSaham.java
13		loginForm	loginForm.java
14		pemasukan	pemasukan.java
15		pembayaran	pembayaran.java
16		pembelian	pembelian.java
17		pengguna	pengguna.java
18		penjualan	penjualan.java
19		penjualanSaham	penjualanSaham.java
20		supplier	supplier.java
21		DB	DB.java
22		DBQuery	DBQuery.java
23		DBTable	DBTable.java
24		DriverConnection	DriverConnection.java
25		MyModelTable	MyModelTable.java
26		MyTable	MyTable.java
27		Utils	Utils.java
28	laporan		

Sumber : Implementasi

5.3 Implementasi Algoritma

Sistem ini mempunyai beberapa proses utama yaitu proses login, koneksi, data transaksi, data master, data laporan

5.3.1 Algoritma isInputValid

Proses isInputValid terdapat dalam klas pembelian dan dilakukan oleh user dengan memasukkan nilai data valid. Sistem akan memproses nilai yang dimasukkan, apabila valid maka user dapat melakukan proses selanjutnya.

```
1 Method isInputValid dari klas pembelian
2
3 Deklarasi
4 TYPE result is boolean
5 TYPE intRow<-myTable.getRowCount()
6
7 Deskripsi
8 IF cbKodeSupplier.getSelectedIndex()<0 THEN
9     cbKodeNamaSupplier.getSelectedIndex()<0 THEN
10    intRow<1
11
12    cbKodeSupplier.requestFocus()
13    result<-false
14
15 ELSE IF tTotalHrg.getText().equals("") THEN
16     ! Utils.isNumber(tTotalHrg.getText())
17
18    cbKodeBrg.requestFocus()
19    result<-false
20
21 ELSE result<-true
22
23 IF result<-false
24
25     Utils.showMessageDialog("Data tidak sah","Informasi")
26     return<-result
```

Gambar 5.1 Algoritma proses isInputValid

Sumber: Implementasi

5.3.2 Algoritma proses isInputDetilValid

Algoritma proses isInputDetilValid terdapat dalam klas pembelian, proses ini untuk mengecek data pembelian dan menambahkan kedalam daftar pembelian barang dan memastikan data belum ada dalam daftar pembelian barang.

```
1 Method isInputDetilValid dari klas pembelian
2
3 Deklarasi
4 TYPE result is boolean
5 TYPE strNote<--" " is String
6 BEGIN
7 Deskripsi
8 IF tPerson.getText().equals("") THEN tJumlah.getText().equals("")
9
10    cbKodeBrg.requestFocus()
11    result<-false
12
13 ELSE result<-true
14
15 IF append==true AND result==true
16     result<-isAlreadyItem(cbKodeBrg.getSelectedItem().toString())
17     strNote<-"Barang tersebut sudah ada"
18
19 IF result==false
20     Utils.showMessage(this, "Data tidak sah " + strNote, "Informasi")
21
22 return result
23 END
```

Gambar 5.2 Algoritma proses isInputDetilValid

Sumber: Implementasi

5.3.3 Algoritma proses simpan

Algoritma proses simpan terdapat dalam klas pembelian, proses ini untuk menyimpan data pembelian kedalam basis data.

```
1 Method isAlreadyItem dari klas pembelian
2
3 Deklarasi
4 TYPE totalRecord=myTable.getRowCount() is int
5 TYPE result is boolean
6
7 BEGIN
8 Deskripsi
9 TRY
10    :Pembelian.append()
11    :Pembelian.updateString("no_beli",String.valueOf(tnoPembelian.getText()))
12    :Pembelian.updateString("kode_supplier",String.valueOf(cbKodeSupplier.getSelectedItem()))
13    Create Object Date
14    Create Object SimpleDateFormat("yyy-mm-dd")
15    String dateString1= fmt.format(current)
16    java.sql.Date dt
17    :Pembelian.updateDate("tgl_beli",dt)
18
19    :Pembelian.post()
20
21    int totalRecord<-myTable.getRowCount()
22    for (int i<-0; i<totalRecord; i++)
23       :PembelianD.append()
24       :PembelianD.updateString("no_pembelian", String.valueOf(tnoPembelian.getText()))
25       :PembelianD.updateString("kode_brg", (myTable.getData(i, 0).toString()))
26       :PembelianD.updateDouble("harga_beli", Double.parseDouble(myTable.getData(i, 2).toString()))
27       :PembelianD.updateInt("jumlah", Integer.parseInt(myTable.getData(i, 3).toString()))
28       :PembelianD.updateInt("persen_hrg_jual", integer.parseInt(myTable.getData(i, 5).toString()))
29       :PembelianD.post()
30
31 catch(SQLException sqle)
32     System.err.println(sale)
33
34 END
```

Gambar 5.3 Algoritma proses simpan

Sumber: Implementasi

5.3.4 Algoritma proses isAlreadyItem

Algoritma proses isAlreadyItem terdapat dalam klas pembelian, proses ini untuk pengecekan data pembelian sebelum ditampung kedalam daftar pembelian barang

```
1 Method isAlreadyItem dari klas pembelian
2
3 Deklarasi
4 TYPE totalRecord=myTable.getRowCount() is int
5 TYPE result is boolean
6
7 BEGIN
8 Deskripsi
9 result<-false
10 FOR int i<0; i<totalRecord; i++
11     IF vKodeBrg==myTable.getData(i, 0).toString()
12         result<-true
13
14 return ! result
15 END
```

Gambar 5.4 Algoritma proses isAlreadyItem

Sumber: Implementasi

5.4 Implementasi Antarmuka Aplikasi Sistem Informasi Akuntansi

Pada aplikasi ini, antarmuka dibuat sesuai dengan kebutuhan fungsional yang harus disediakan oleh sistem. Tujuan utama pengembangan antarmuka adalah untuk memberikan kemudahan bagi pengguna (*user*) dalam menggunakan aplikasi yang telah dibangun.

5.4.1 Implementasi Aplikasi Login

Antarmuka login ini akan ditampilkan ketika pengguna ingin menggunakan aplikasi, sebelum dapat menggunakan semua fasilitas aplikasi pengguna diharuskan untuk melakukan login terlebih dahulu, seperti yang sudah dijelaskan dalam skenario *use case* Login pada analisis kebutuhan. Antarmuka dari login aplikasi ini ditunjukkan pada Gambar 5.1.



Gambar 5.1 Antarmuka aplikasi login

Sumber: Implementasi

Gambar 5.1 merupakan gambar hasil implementasi antarmuka untuk login. Penjelasan masing-masing bagian dari antarmuka login diuraikan pada Tabel 5.4

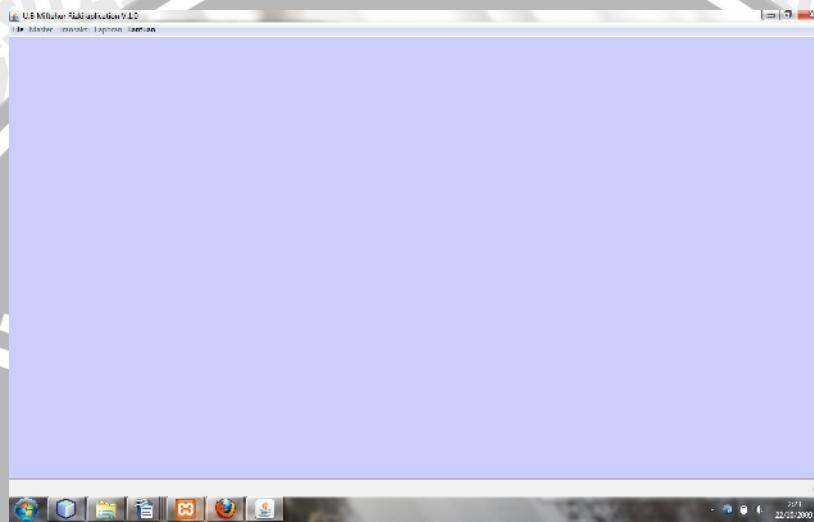
Tabel 5.4 Penjelasan tombol-tombol pada aplikasi login

No.	Label Tombol	Penjelasan
1	Login	Tombol ini digunakan untuk masuk ke dalam menu utama aplikasi, sebelum dapat masuk ke dalam menu utama dan sebelum dapat menggunakan aplikasi, maka pengguna diharuskan untuk mengisi nilai masukan <i>username</i> dan <i>password</i>
2	Batal	Tombol ini digunakan untuk membatalkan menu login dan masuk kedalam menu utama dalam keadaan tidak aktif

Sumber: Implementasi

5.4.2 Implementasi Aplikasi Menu Utama

Antarmuka menu utama aplikasi ini akan ditampilkan ketika pengguna sudah melakukan login dengan benar, pada menu utama ini akan ditampilkan beberapa menu yang berhubungan dengan fasilitas dari sistem. Antarmuka menu utama aplikasi ini ditunjukkan pada Gambar 5.2.



Gambar 5.2 Antarmuka aplikasi menu utama

Sumber: Implementasi

Gambar 5.2 merupakan gambar hasil implementasi antarmuka untuk menu utama. Penjelasan masing-masing bagian dari antarmuka menu utama diuraikan pada Tabel 5.5.

Tabel 5.5 Penjelasan menu-menu pada aplikasi menu utama

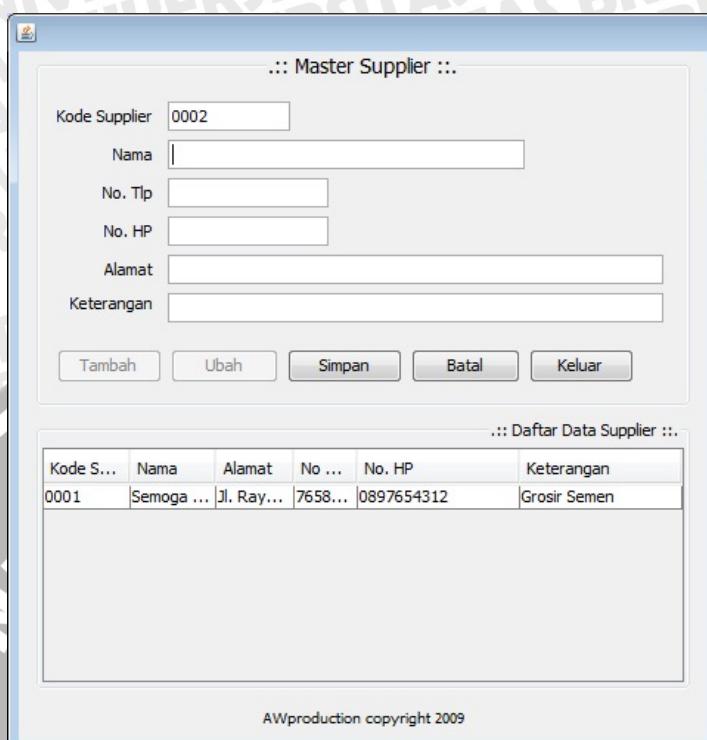
No.	Label Menu	Penjelasan
1	File	Menu ini digunakan untuk proses login dan logout
2	Master	Menu ini digunakan untuk menampilkan menu-menu yang berhubungan dengan master data. Antara lain barang, supplier, karyawan, operator, jenis barang, jenis pemasukan, jenis

		pembayaran.
3	Transaksi	Menu ini digunakan untuk menampilkan menu-menu yang berhubungan dengan transaksi data. Antara lain pembelian, penjualan, lihat pembelian, lihat penjualan, pemasukan, pembayaran, penjualan saham, lihat penjualan saham.
4	Laporan	Menu ini digunakan untuk menampilkan menu-menu yang berhubungan dengan laporan. Antara lain buku besar, neraca, laba rugi, karyawan, pemegang saham, persediaan barang.
5	Bantuan	Menu ini untuk menampilkan menu About dan Help.

Sumber: Implementasi

5.4.3 Implementasi Aplikasi Master Supplier

Antarmuka menu master supplier aplikasi ini akan ditampilkan ketika pengguna ingin mencatat data supplier baru, pada menu master supplier ini juga digunakan untuk mengubah, menyimpan dan membatalkan pencatatan data supplier. Antarmuka menu master supplier aplikasi ini ditunjukkan pada Gambar 5.3.



Gambar 5.3 Antarmuka aplikasi master supplier

Sumber: Implementasi

Gambar 5.3 merupakan gambar hasil implementasi antarmuka untuk menu utama. Penjelasan masing-masing bagian dari antarmuka menu utama diuraikan pada Tabel 5.6.

Tabel 5.6 Penjelasan tombol-tombol pada aplikasi master supplier

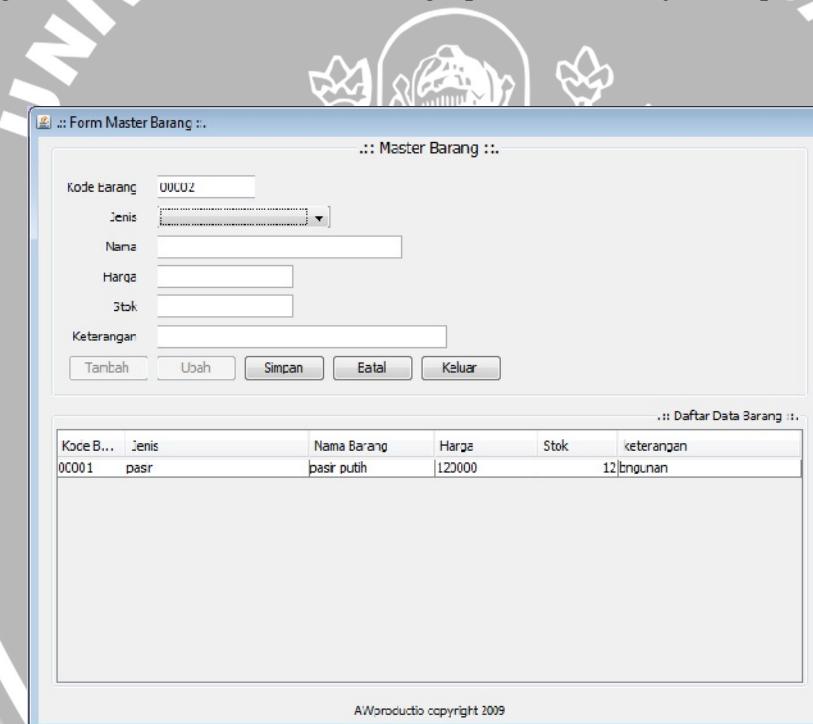
No.	Label Tombol	Penjelasan
1	Tambah	Tombol ini digunakan untuk mengaktifkan form mencatat data supplier baru.
2	Ubah	Tombol ini digunakan untuk mengubah data supplier yang telah ada.
3	Simpan	Tombol ini digunakan untuk menyimpan data supplier yang telah ditambahkan atau diubah.
4	Batal	Tombol ini untuk membatalkan pencatatan data supplier dan menonaktifkan form mencatat

		supplier.
5	Keluar	Tombol ini untuk menutup aplikasi master supplier dan kembali ke menu utama.

Sumber: Implementasi

5.4.4 Implementasi Aplikasi Master Barang

Antarmuka menu master barang aplikasi ini akan ditampilkan ketika pengguna ingin mencatat data barang baru, pada menu master barang ini juga digunakan untuk mengubah, menyimpan dan membatalkan pencatatan data barang. Antarmuka menu master barang aplikasi ini ditunjukkan pada Gambar 5.4.



Gambar 5.4 Antarmuka aplikasi master barang

Sumber: Implementasi

Gambar 5.4 merupakan gambar hasil implementasi antarmuka aplikasi master barang. Penjelasan masing-masing bagian dari antarmuka aplikasi master barang diuraikan pada Tabel 5.7.

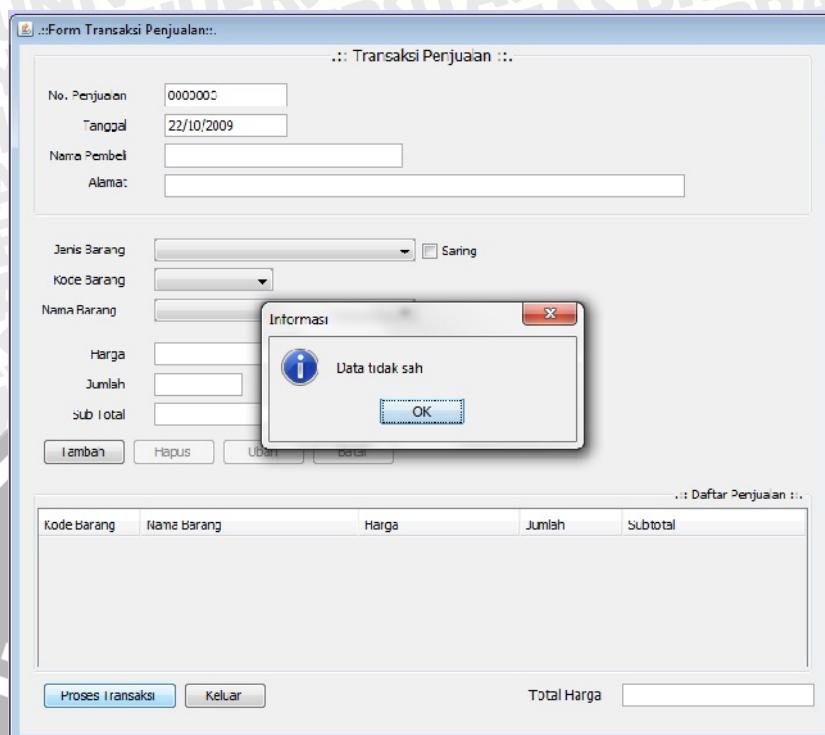
Tabel 5.7 Penjelasan tombol-tombol pada aplikasi master barang

No.	Label Tombol	Penjelasan
1	Tambah	Tombol ini digunakan untuk mengaktifkan form mencatat data barang baru.
2	Ubah	Tombol ini digunakan untuk mengubah data barang yang telah ada.
3	Simpan	Tombol ini digunakan untuk menyimpan data barang yang telah ditambahkan atau diubah.
4	Batal	Tombol ini untuk membatalkan pencatatan data barang dan menonaktifkan form mencatat barang.
5	Keluar	Tombol ini untuk menutup aplikasi master barang dan kembali ke menu utama.

Sumber: Implementasi

5.4.5 Implementasi Aplikasi Transaksi Penjualan

Antarmuka menu transaksi penjualan aplikasi ini akan ditampilkan ketika pengguna melakukan pencatatan transaksi penjualan. Antarmuka menu transaksi penjualan aplikasi ini ditunjukkan pada Gambar 5.5.



Gambar 5.5 Antarmuka aplikasi transaksi penjualan

Sumber: Implementasi

Gambar 5.5 merupakan gambar hasil implementasi antarmuka aplikasi transaksi penjualan. Penjelasan masing-masing bagian dari antarmuka aplikasi transaksi penjualan diuraikan pada Tabel 5.8.

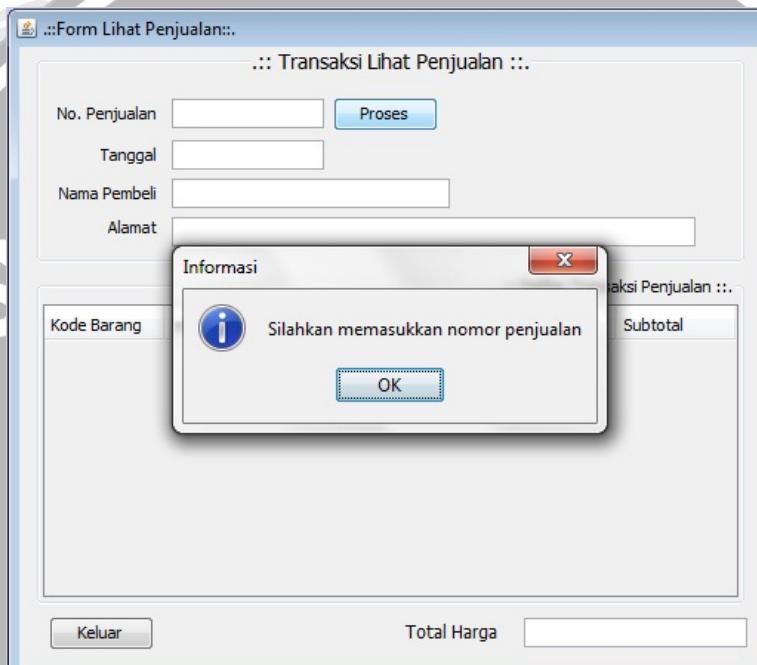
Tabel 5.8 Penjelasan tombol-tombol pada aplikasi master barang

No.	Label Tombol	Penjelasan
1	Tambah	Tombol ini digunakan untuk menampung data yang telah diisikan kedalam daftar penjualan barang
2	Keluar	Tombol ini untuk menutup aplikasi transaksi penjualan dan kembali ke menu utama.

Sumber: Implementasi

5.4.6 Implementasi Aplikasi Lihat Transaksi Penjualan

Antarmuka menu transaksi lihat penjualan aplikasi ini akan ditampilkan ketika pengguna melakukan pengecekan data transaksi penjualan. Antarmuka menu lihat transaksi penjualan aplikasi ini ditunjukkan pada Gambar 5.6.



Gambar 5.6 Antarmuka aplikasi lihat transaksi penjualan

Sumber: Implementasi

Gambar 5.6 merupakan gambar hasil implementasi antarmuka aplikasi lihat transaksi penjualan. Penjelasan masing-masing bagian dari antarmuka aplikasi lihat transaksi penjualan diuraikan pada Tabel 5.9.

Tabel 5.9 Penjelasan tombol-tombol pada aplikasi lihat transaksi penjualan

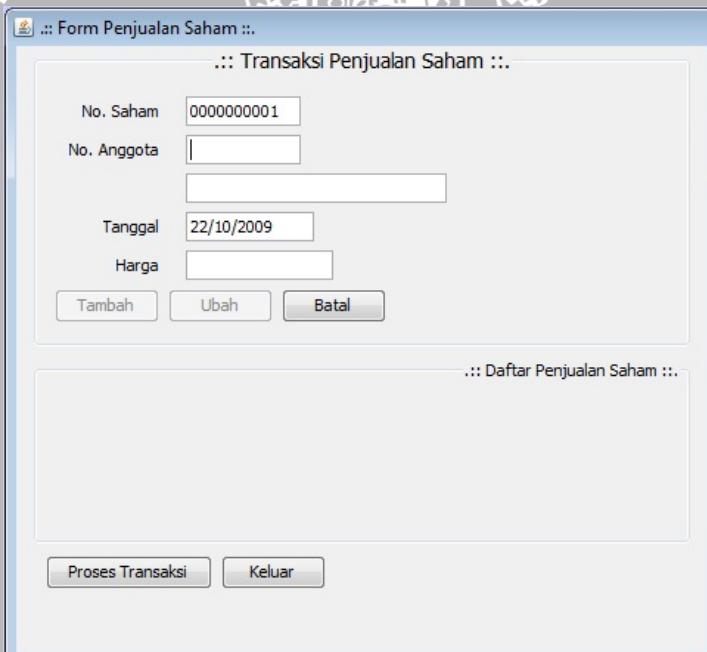
No.	Label Tombol	Penjelasan
1	Proses	Tombol ini digunakan untuk mencari data transaksi yang telah tercatat dalam basis data dan

		menampilkan ke dalam daftar transaksi penjualan.
2	Keluar	Tombol ini untuk menutup aplikasi lihat transaksi penjualan dan kembali ke menu utama.

Sumber: Implementasi

5.4.7 Implementasi Aplikasi Transaksi Penjualan Saham

Antarmuka menu transaksi penjualan saham aplikasi ini akan ditampilkan ketika pengguna melakukan pencatatan transaksi penjualan saham. Antarmuka menu transaksi penjualan saham aplikasi ini ditunjukkan pada Gambar 5.7.



Gambar 5.7 Antarmuka aplikasi transaksi penjualan saham

Sumber: Implementasi

Gambar 5.7 merupakan gambar hasil implementasi antarmuka aplikasi transaksi penjualan saham. Penjelasan masing-masing bagian dari antarmuka aplikasi transaksi penjualan saham diuraikan pada Tabel 5.10.

Tabel 5.10 Penjelasan tombol-tombol pada aplikasi transaksi penjualan saham

No.	Label Tombol	Penjelasan
1	Tambah	Tombol ini digunakan untuk menambah data transaksi kedalam dan menampilkan ke dalam daftar transaksi penjualan saham.
2	Ubah	Tombol ini untuk mengubah data penjualan saham
3	Batal	Tombol ini digunakan untuk membatalkan pencatatan data transaksi
4	Proses transaksi	Tombol ini digunakan untuk proses penyimpanan data transaksi kedalam basis data
5	Keluar	Tombol ini digunakan untuk menutup form transaksi penjualan saham.

Sumber: Implementasi

BAB VI

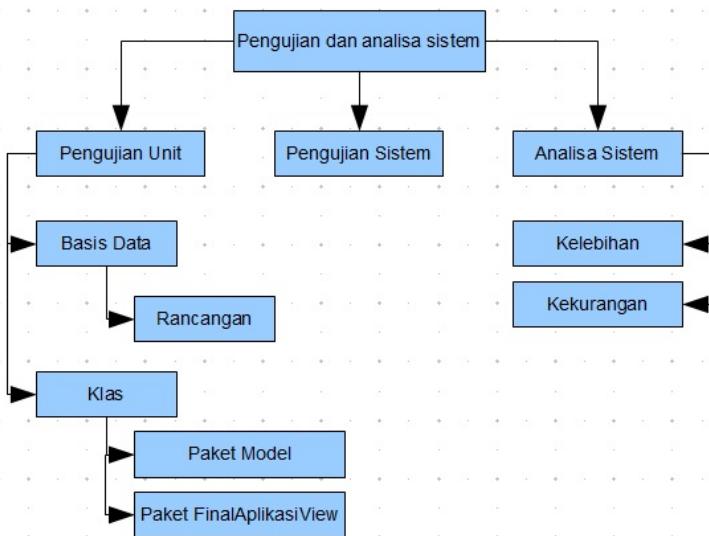
PENGUJIAN DAN ANALISA SISTEM

Pada bab ini membahas pengujian terhadap sistem informasi akuntansi dengan laporan laba/ rugi menggunakan metode multiple step. Proses pengujian dilakukan dengan menggunakan dua metode pengujian, yaitu Pengujian *unit (unit test)*, dan pengujian sistem (*system test*). Pada bagian terakhir bab ini dicantumkan hasil analisa dari seluruh pengujian yang dilakukan terhadap sistem informasi yang akan dibuat.

Pada pengujian unit rancangan basis data dilakukan dengan menggunakan aplikasi Sybase PowerDesigner 12. PowerDesigner dipilih sebagai *tools* pengujian karena Sybase PowerDesigner saat ini memegang 33,7 % mayoritas pasaran perangkat lunak (*software*) yang digunakan untuk mendesain basis data. (www.sybase.com).

Pada pengujian unit aplikasi dilakukan pada login form dan menu utama, dalam hal ini mengacu pada bab implementasi. Pemilihan bagian pengujian ini dilakukan karena dua unit komponen initersebut merupakan komponen utama dari sistem informasi akuntansi.

Pengujian yang dilakukan dapat digambarkan dengan diagram pohon seperti pada gambar 6.1 dibawah ini.



Gambar 6.1 Diagram pohon pengujian sistem informasi akuntansi

Sumber: Pengujian

6.1 Pengujian Unit

Pada pengujian unit diuji fungsionalitas dan kinerja dari masing-masing unit yang ada pada sistem yang digunakan. Pada aplikasi sistem informasi akuntansi ini terdapat dua unit utama yang akan di uji, yaitu unit basis data , unit sistem informasi.

6.1.1 Pengujian Unit Basis Data

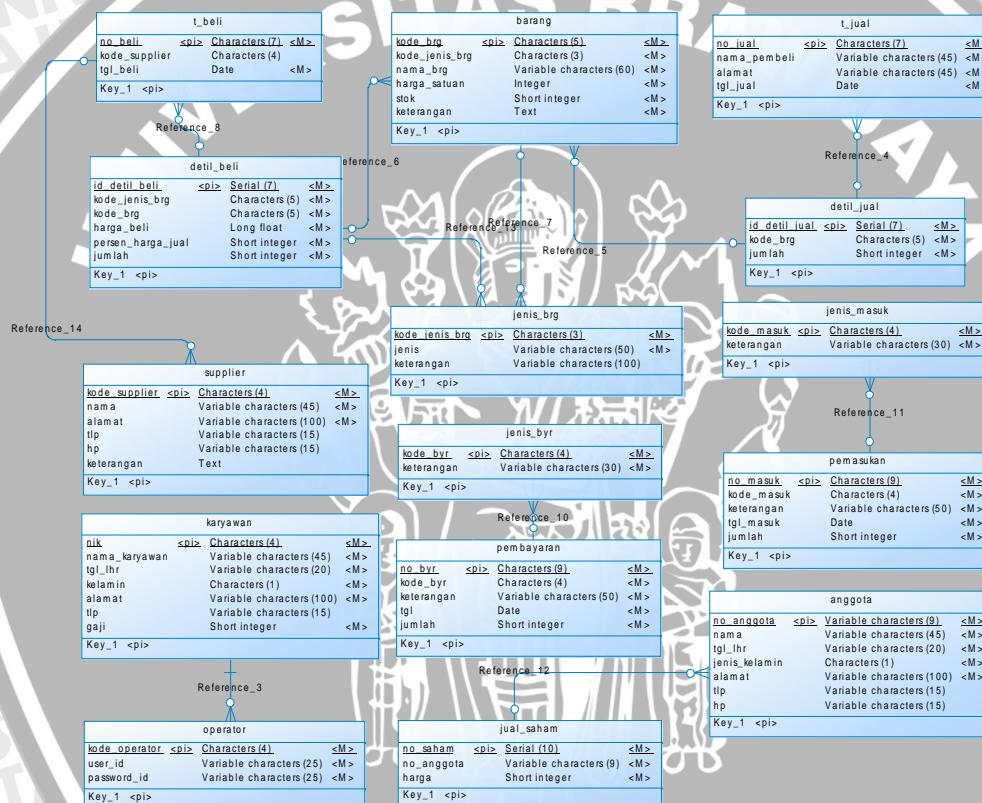
Pada unit basis data akan diuji rancangan diagram ER dari basis data yang dibuat sebelumnya. Untuk pengujian rancangan digunakan aplikasi *Sybase PowerDesigner 12*.

6.1.1.1 Pengujian Rancangan

Pengujian dilakukan untuk mengetahui dan memeriksa validitas rancangan basis data. Pengujian dilakukan dengan menggunakan *Sybase PowerDesigner 12*.

6.1.1.1 Prosedur Pengujian

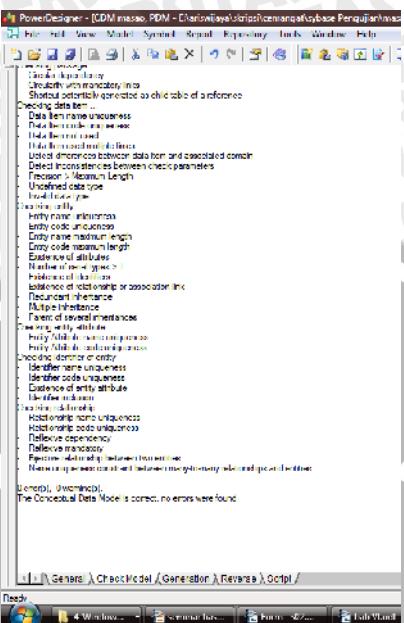
- Membuat basis data dengan menggunakan MySQL Server 5.0.
- Melakukan konfigurasi ODBC untuk basis data yang telah dibuat.
- Menjalankan MySQL Server 5.0.
- Menjalankan aplikasi Sybase PowerDesigner 12.5.
- Menggambarkan diagram ER pada area kerja *conceptual data model* (CDM). Gambar 6.2 menunjukkan ER diagram pada CDM objek.



Gambar 6.2 Conceptual Data Model Object

Sumber: Pengujian

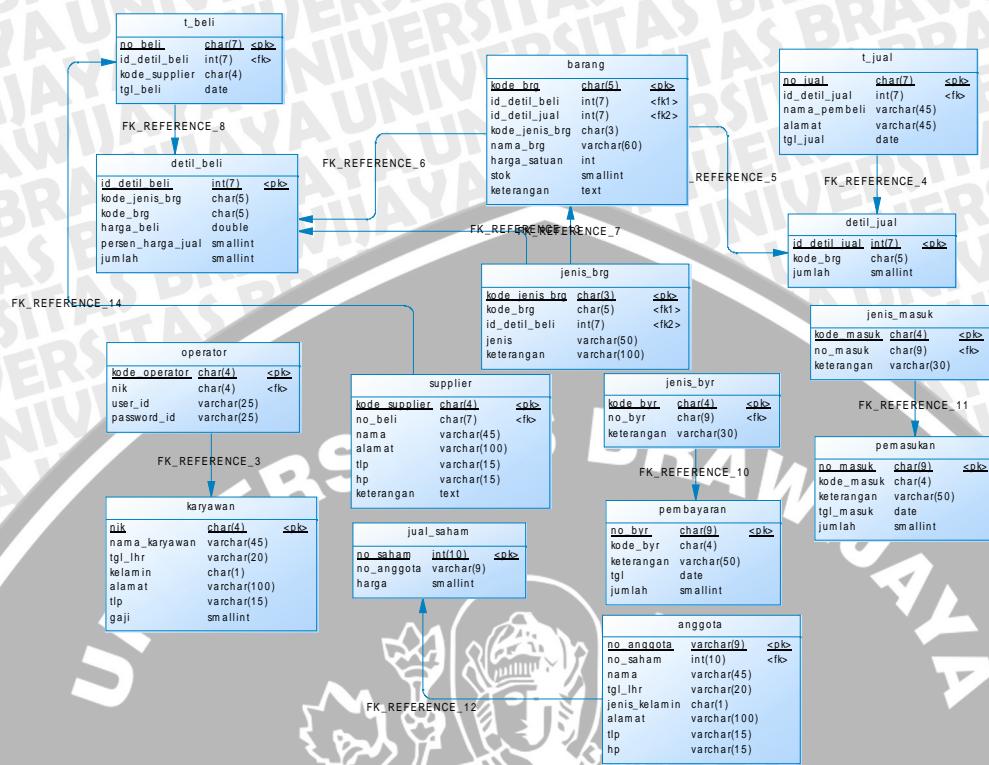
- Memeriksa diagram ER tersebut dengan menekan tombol *check model* pada *toolbar*. Gambar dari hasil *check model* diagram ER ditunjukkan pada gambar 6.3.



Gambar 6.3 Hasil dari pemeriksaan model oleh PowerDesigner

Sumber: Pengujian

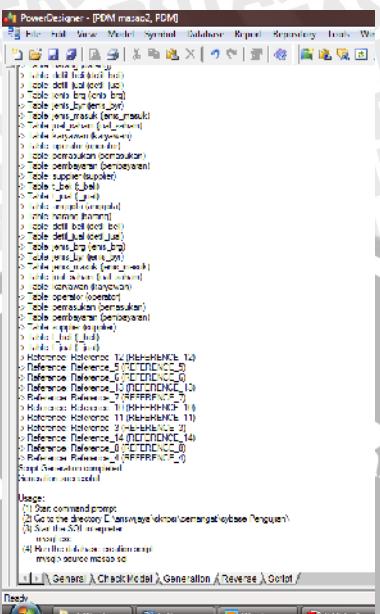
- Mengubah diagram ER pada *Conceptual Data Model Object* menjadi *Physical Data Model (PDM) Object*, dengan cara melakukan proses *generate* dengan menekan tombol *generate physical data model* pada *toolbar*. Gambar PDM object ditunjukkan dalam gambar 6.4.



Gambar 6.4 Physical Data Model Object.

Sumber: Pengujian

- Mengubah *physical data model object* menjadi basis data, dengan cara menekan tombol *generate database* pada *toolbar*. Hasil *generate database* ditunjukkan dalam gambar 6.5 berikut.

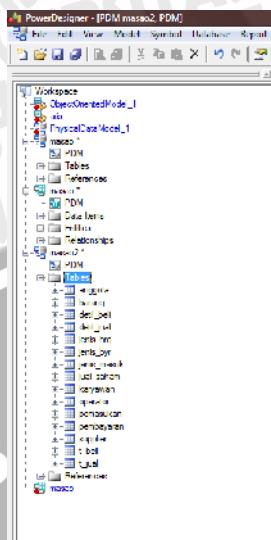


Gambar 6.5 Hasil dari generate database oleh PowerDesigner

Sumber: Pengujian

6.1.1.1.2 Hasil Pengujian

- Gambar 6.6 menunjukkan tabel-tabel yang terbentuk setelah hasil generate database dari PDM Object.



Gambar 6.6 Tabel-tabel yang terbentuk setelah generate database

Sumber: Pengujian

- Melihat struktur dari masing-masing tabel yang telah dibentuk oleh Sybase PowerDesigner 12.5.
- Gambar struktur tabel anggota yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.7.

Name	Type	Data Type	Length	Decimals	PK	U	C
id_anggota	integer	recordID	2	0			
no_anggota	string	int(10)	10	0			
nama	string	varchar(45)	45	0			
alamat	string	varchar(45)	45	0			
id_jabatan	integer	recordID	2	0			
id_kemahasiswaan	integer	recordID	2	0			
id_pemerkasaan	integer	recordID	2	0			
id_pemuda	integer	recordID	2	0			
id_pendidikan	integer	recordID	2	0			
id_pendidikan_bantuan	integer	recordID	2	0			

Gambar 6.7 Struktur tabel anggota

Sumber: Pengujian

- Gambar struktur tabel barang yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.8.

The screenshot shows the 'Table Properties' dialog for the 'barang' table. The 'General' tab is selected. The table has 10 columns: id_barang (int, primary key), kode_barang (varchar(20)), nama_barang (varchar(50)), satuan (varchar(20)), harga_beli (decimal(10,2)), harga_jual (decimal(10,2)), diskon (decimal(5,2)), stok_awal (int), stok_saat_ini (int), and keterangan (text). Primary key constraints are defined for 'id_barang' and 'kode_barang'. Foreign key constraints are defined for 'satuan' (referencing 'satuan' in 'satuan'), 'harga_beli' (referencing 'harga_beli' in 'detil_beli'), 'harga_jual' (referencing 'harga_jual' in 'detil_beli'), and 'diskon' (referencing 'diskon' in 'detil_beli').

Name	Type	Null	Key	Default	Length	Precision	Decimals	Extra
<i>id_barang</i>	int	NO	PK		10			
<i>kode_barang</i>	varchar(20)	NO	PK		20			
<i>nama_barang</i>	varchar(50)	NO			50			
<i>satuan</i>	varchar(20)	NO			20			
<i>harga_beli</i>	decimal(10,2)	NO			10	2		
<i>harga_jual</i>	decimal(10,2)	NO			10	2		
<i>diskon</i>	decimal(5,2)	NO			5	2		
<i>stok_awal</i>	int	NO			10			
<i>stok_saat_ini</i>	int	NO			10			
<i>keterangan</i>	text	NO						

Gambar 6.8 Struktur tabel barang

Sumber: Pengujian

- Gambar struktur tabel detil_beli yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.9.

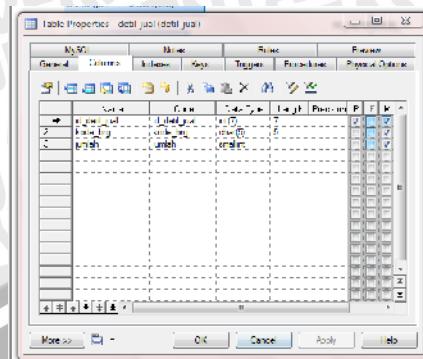
The screenshot shows the 'Table Properties' dialog for the 'detil_beli' table. The 'General' tab is selected. The table has 8 columns: id_detil_beli (int, primary key), id_barang (int), id_transaksi (int), jumlah (int), harga_beli (decimal(10,2)), harga_jual (decimal(10,2)), diskon (decimal(5,2)), and keterangan (text). Primary key constraints are defined for 'id_detil_beli'. Foreign key constraints are defined for 'id_barang' (referencing 'id_barang' in 'barang') and 'id_transaksi' (referencing 'id_transaksi' in 'transaksi').

Name	Type	Null	Key	Default	Length	Precision	Decimals	Extra
<i>id_detil_beli</i>	int	NO	PK		10			
<i>id_barang</i>	int	NO			10			
<i>id_transaksi</i>	int	NO			10			
<i>jumlah</i>	int	NO			10			
<i>harga_beli</i>	decimal(10,2)	NO			10	2		
<i>harga_jual</i>	decimal(10,2)	NO			10	2		
<i>diskon</i>	decimal(5,2)	NO			5	2		
<i>keterangan</i>	text	NO						

Gambar 6.9 Struktur tabel detil_beli

Sumber: Pengujian

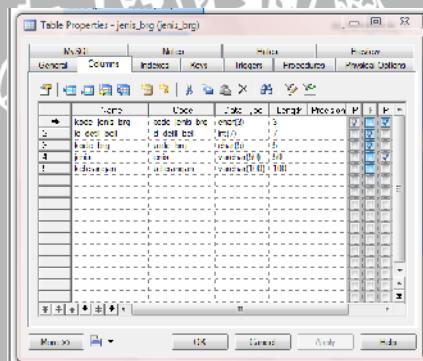
- Gambar struktur tabel detil_jual yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.10.



Gambar 6.10 Struktur tabel detil_jual

Sumber: Pengujian

- Gambar struktur tabel jenis_brg yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.11.



Gambar 6.11 Struktur tabel jenis_brg

Sumber: Pengujian

- Gambar struktur tabel jenis_byr yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.12.

The screenshot shows the 'Table Properties - jenis_byr [jenis_byr]' dialog in MySQL Workbench. The 'General' tab is selected. The table structure is as follows:

Name	Type	Collation	Length	Decimal
id_jenis	int(11)	latin1_swedish_ci	4	0
jenis_byr	varchar(30)	latin1_swedish_ci	30	0

Gambar 6.12 Struktur tabel jenis_byr

Sumber: Pengujian

- Gambar struktur tabel jenis_masuk yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.13.

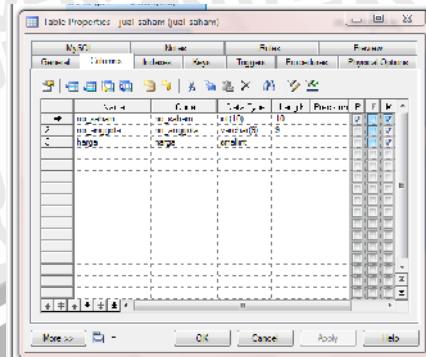
The screenshot shows the 'Table Properties - jenis_masuk [jenis_masuk]' dialog in MySQL Workbench. The 'General' tab is selected. The table structure is as follows:

Name	Type	Collation	Length	Decimal
kode_masuk	code masuk	latin1_swedish_ci	4	0
ds_masuk	ds masuk	latin1_swedish_ci	4	0
jumlah	jumlah	latin1_swedish_ci	10	0

Gambar 6.13 Struktur tabel jenis_masuk

Sumber: Pengujian

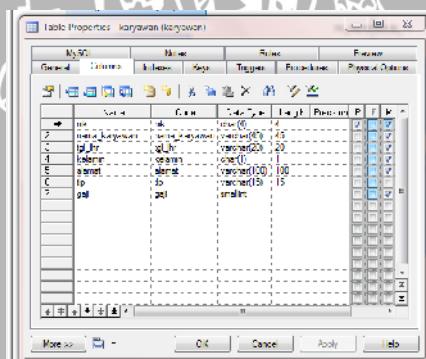
- Gambar struktur tabel jual_saham yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.14.



Gambar 6.14 Struktur tabel jual_saham

Sumber: Pengujian

- Gambar struktur tabel karyawan yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.15.



Gambar 6.15 Struktur tabel karyawan

Sumber: Pengujian

- Gambar struktur tabel operator yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.16.

The screenshot shows the 'Table Properties' dialog box for the 'operator' table. The table has 4 columns: id, name, login, and password. The 'id' column is defined as INT(11) with a primary key constraint. The 'name' column is defined as VARCHAR(25). The 'login' column is defined as VARCHAR(25). The 'password' column is defined as VARCHAR(25). The 'Name' tab is selected.

Name	Type	Length	Key
id	INT(11)	11	PK
name	VARCHAR(25)	25	
login	VARCHAR(25)	25	
password	VARCHAR(25)	25	

Gambar 6.16 Struktur tabel operator

Sumber: Pengujian

- Gambar struktur tabel pemasukan yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.17.

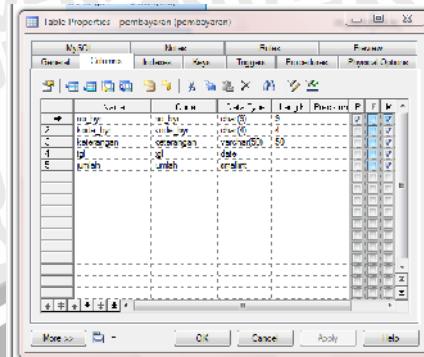
The screenshot shows the 'Table Properties' dialog box for the 'pemasukan' table. The table has 6 columns: id, tgl_pemasukan, jenis, jumlah, date, and created. The 'id' column is defined as INT(11) with a primary key constraint. The 'tgl_pemasukan' column is defined as DATE. The 'jenis' column is defined as VARCHAR(50). The 'jumlah' column is defined as DECIMAL(10,2). The 'date' column is defined as DATETIME. The 'created' column is defined as DATETIME. The 'Name' tab is selected.

Name	Type	Length	Key
id	INT(11)	11	PK
tgl_pemasukan	DATE	10	
jenis	VARCHAR(50)	50	
jumlah	DECIMAL(10,2)	10,2	
date	DATETIME	19	
created	DATETIME	19	

Gambar 6.17 Struktur tabel pemasukan

Sumber: Pengujian

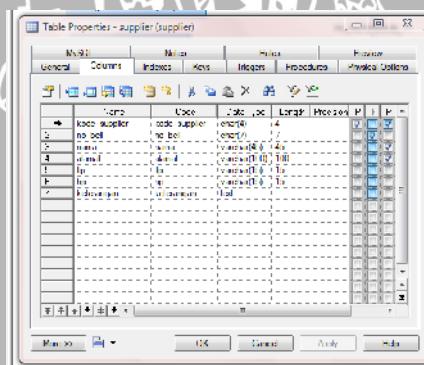
- Gambar struktur tabel pembayaran yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.18.



Gambar 6.18 Struktur tabel pembayaran

Sumber: Pengujian

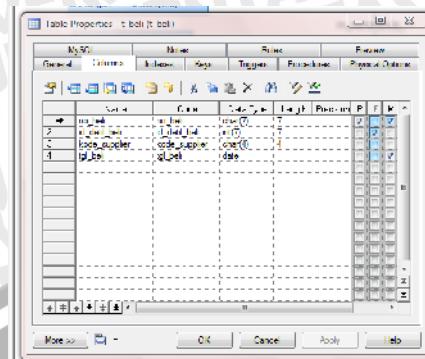
- Gambar struktur tabel supplier yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.19.



Gambar 6.19 Struktur tabel supplier

Sumber: Pengujian

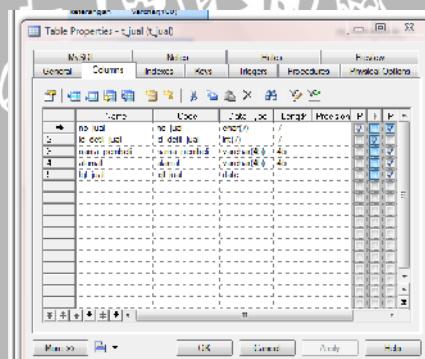
- Gambar struktur tabel t_beli yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.20.



Gambar 6.20 Struktur tabel t_beli

Sumber: Pengujian

- Gambar struktur tabel t_jual yang terbentuk setelah proses *generate database* ditunjukkan dalam Gambar 6.21.



Gambar 6.21 Struktur tabel t_jual

Sumber: Pengujian

6.1.1.3 Analisa Pengujian

Dari hasil pengujian yang dilakukan dengan menggunakan aplikasi sybase power designer dapat dilihat bahwa tabel-tabel yang ada pada basis data dapat terbentuk dengan baik dan sesuai dengan rancangan diagram ER yang telah dibuat sebelumnya.

6.2 Pengujian Sistem

Pada pengujian sistem (*system test*), aplikasi diuji secara keseluruhan, pengujian ini biasa disebut juga *integration test*. Sistem informasi akuntansi ini akan diuji secara terintegrasi

6.2.1 Pengujian Login Form

Pengujian ini dilakukan untuk membatasi jumlah pengguna, sehingga pengguna yang telah terdaftar dalam basis data saja yang dapat menggunakan aplikasi ini.

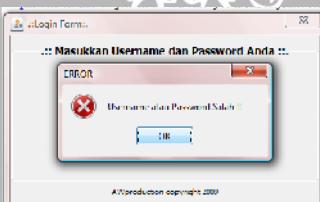


Gambar 6.22 Memasukkan nilai yang salah

Sumber: Pengujian

6.2.1.1 Hasil Pengujian

Hasil pengujian dengan memasukkan nilai yang tidak terdaftar dalam basis data ditunjukkan pada gambar 6.23.



Gambar 6.23 Pesan peringatan ketika tombol login ditekan

Sumber: Pengujian

6.2.1.2 Analisa Pengujian

Dari hasil pengujian dapat diketahui bahwa sistem dapat berjalan dengan baik.

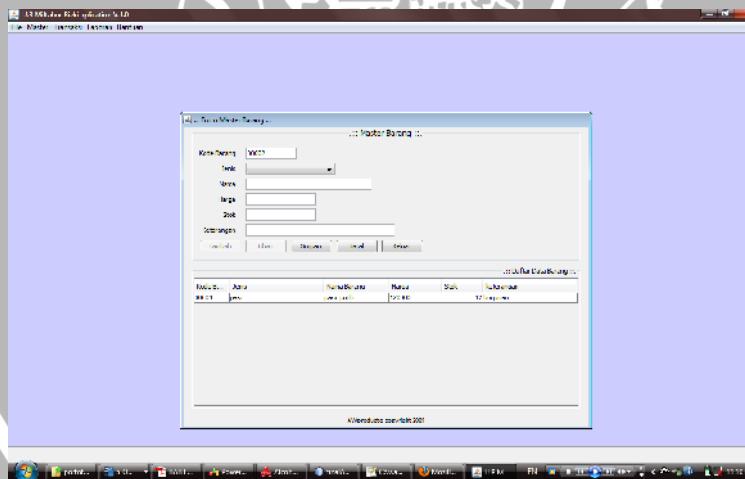
6.2.2 Pengujian Menu Utama Form

Pengujian pada menu utama mengambil beberapa contoh form yang terdapat pada menu utama.

- Master barang

Pengujian pada form sub menu master barang bertujuan untuk mengetahui apakah sistem akan menerima masukan yang diberikan oleh pengguna yang telah melakukan login.

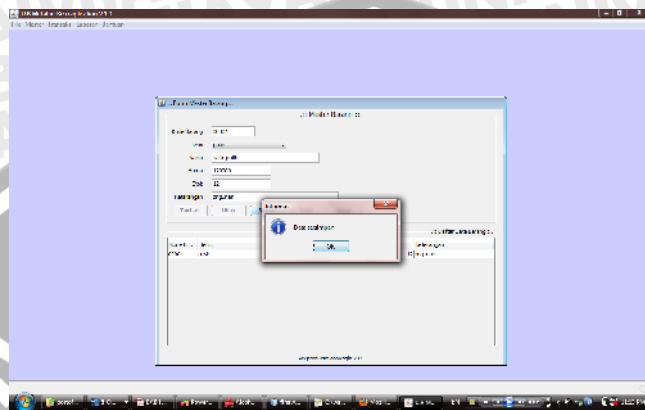
- Pada saat tombol tambah ditekan, field input berubah menjadi keadaan aktif dan siap menerima input data.



Gambar 6.24 Pada saat tombol tambah ditekan

Sumber: Pengujian

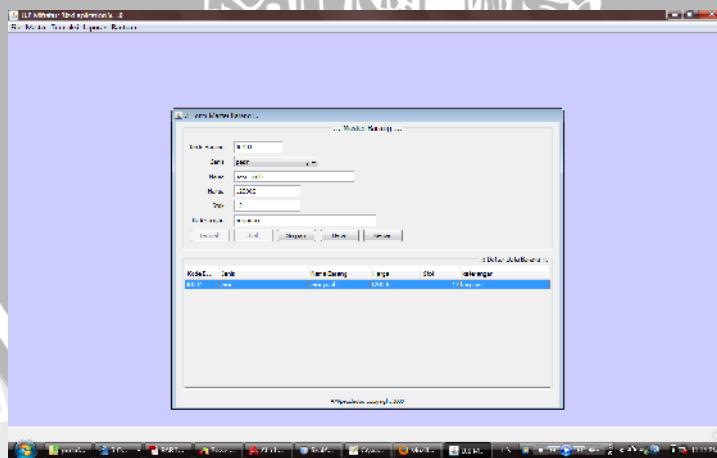
- Pada saat tombol simpan ditekan, data akan tersimpan kedalam basis data apabila field input telah terisi dengan valid. Apabila data input tidak valid, maka akan muncul pesan peringatan.



Gambar 6.25 Pada saat tombol simpan ditekan dan data input valid

Sumber: Pengujian

- Pada saat data barang di pilih maka tombol ubah akan dalam keadaan aktif, bila tombol ubah ditekan maka data barang akan ditampilkan pada field input.



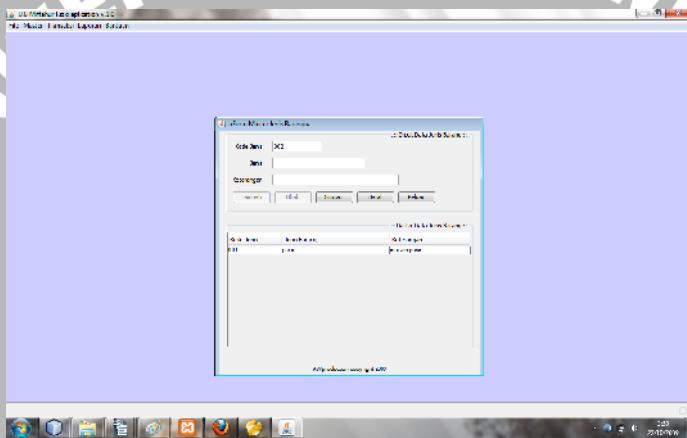
Gambar 6.26 Pada saat tombol ubah ditekan

Sumber: Pengujian

- Pada saat tombol keluar ditekan, maka sistem akan menutup form barang.
- Master jenis barang

Pengujian pada form sub menu master jenis barang bertujuan untuk mengetahui apakah sistem akan menerima masukan yang diberikan oleh pengguna yang telah melakukan login.

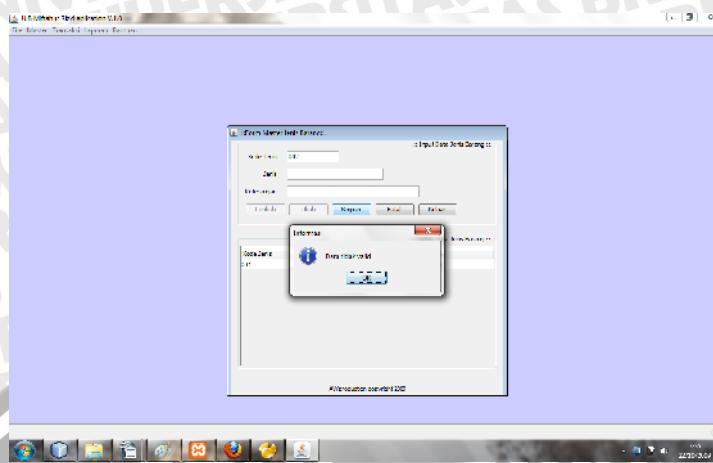
- Pada saat tombol tambah ditekan, field input berubah menjadi keadaan aktif dan siap menerima input data.



Gambar 6.27 Pada saat tombol tambah ditekan

Sumber: Pengujian

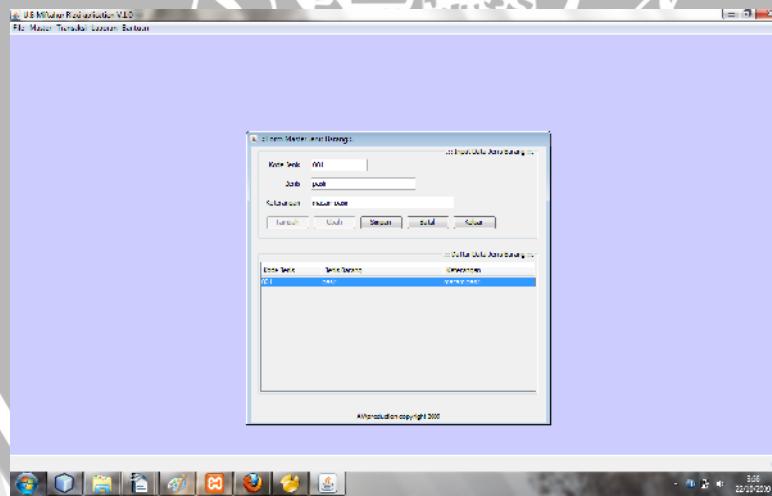
- Pada saat tombol simpan ditekan, data akan tersimpan kedalam basis data apabila field input telah terisi dengan valid. Apabila data input tidak valid, maka akan muncul pesan peringatan.



Gambar 6.28 Pada saat tombol simpan ditekan

Sumber: Pengujian

- Pada saat data jenis barang di pilih maka tombol ubah akan dalam keadaan aktif, bila tombol ubah ditekan maka data barang akan ditampilkan pada field input.



Gambar 6.29 Pada saat tombol ubah ditekan

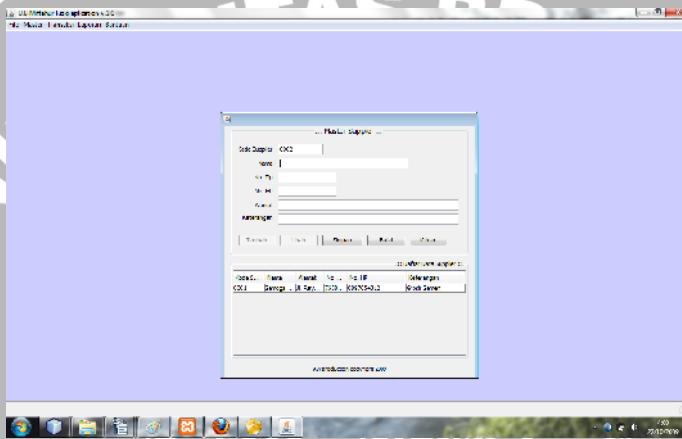
Sumber: Pengujian

- Pada saat tombol keluar ditekan, maka sistem akan menutup form barang.

- Master supplier

Pengujian pada form sub menu master supplier bertujuan untuk mengetahui apakah sistem akan menerima masukan yang diberikan oleh pengguna yang telah melakukan login.

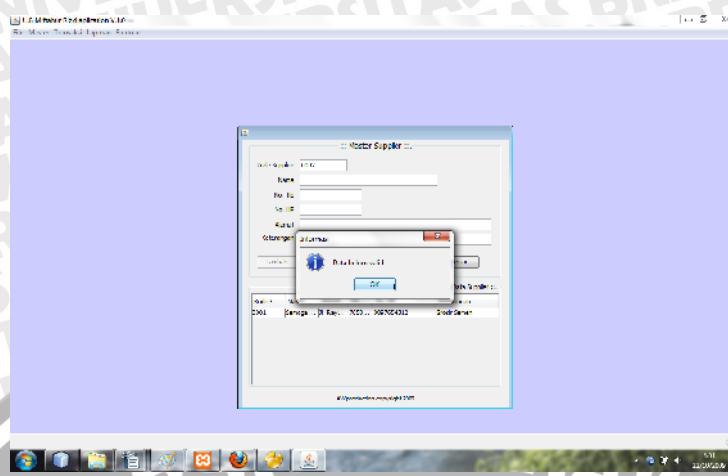
- Pada saat tombol tambah ditekan, field input berubah menjadi keadaan aktif dan siap menerima input data.



Gambar 6.30 Pada saat tombol tambah ditekan

Sumber: Pengujian

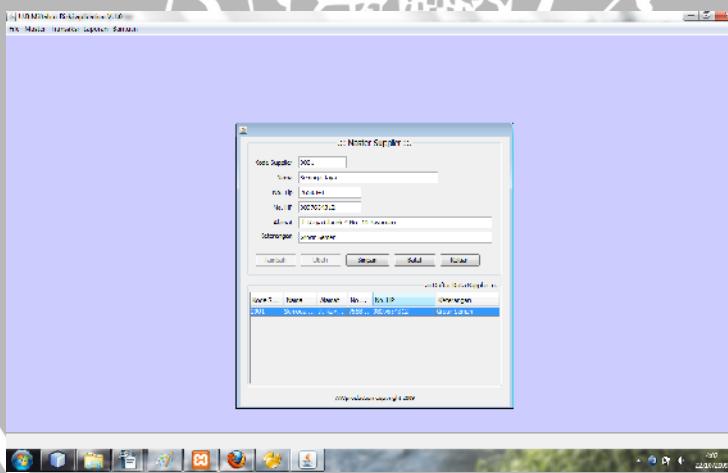
- Pada saat tombol simpan ditekan, data akan tersimpan kedalam basis data apabila field input telah terisi dengan valid. Apabila data input tidak valid, maka akan muncul pesan peringatan.



Gambar 6.31 Pada saat tombol simpan ditekan

Sumber: Pengujian

- Pada saat data supplier di pilih maka tombol ubah akan dalam keadaan aktif, bila tombol ubah ditekan maka data barang akan ditampilkan pada field input.

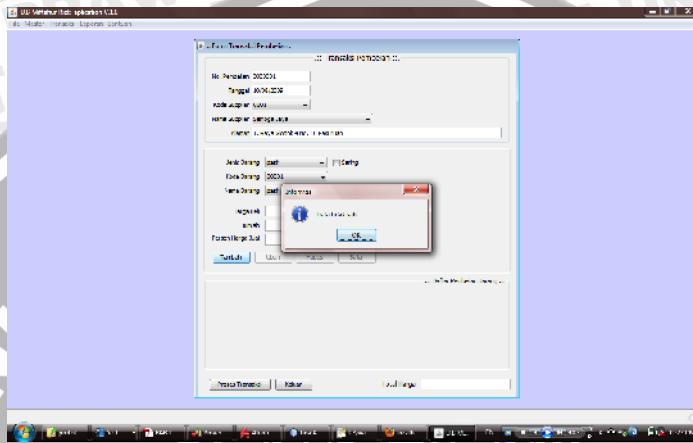


Gambar 6.32 Pada saat tombol ubah ditekan

Sumber: Pengujian

- Pada saat tombol keluar ditekan, maka sistem akan menutup form barang.

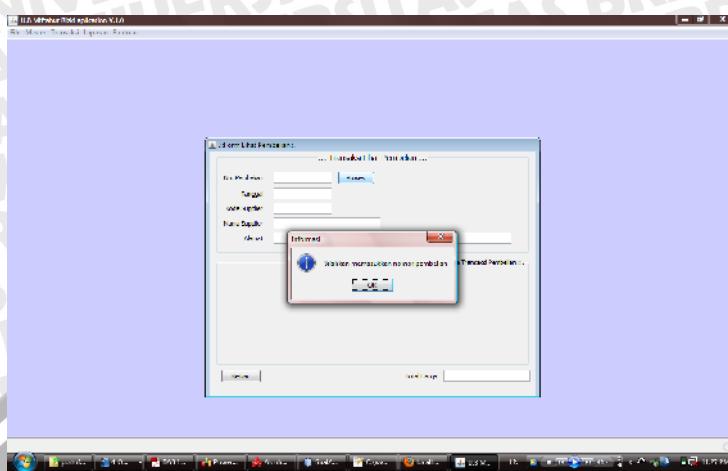
- Transaksi pembelian
 - Pada saat tombol tambah ditekan dan input data valid, maka data yang telah diisi akan ditampilkan kedalam daftar pembelian. Apabila data tidak valid maka akan ditampilkan pesan peringatan.



Gambar 6.33 Pada saat tombol tambah ditekan dan data tidak valid

Sumber: Pengujian

- Apabila data dalam daftar pembelian dipilih maka tombol ubah dan hapus akan aktif.
- Pada saat tombol proses transaksi ditekan maka sistem akan menyimpan data masukan dan mencetak.
- Lihat transaksi pembelian
 - Pada saat tombol proses ditekan maka dan data input valid maka data transaksi pembelian akan ditampilkan. Apabila data tidak valid maka akan ditampilkan pesan peringatan.



Gambar 6.34 Pada saat tombol proses ditekan dan input data tidak valid

Sumber: Pengujian

- Pada saat tombol keluar ditekan, maka sistem akan menutup form transaksi lihat pembelian.
- Laporan buku besar
 - Pada saat tombol proses ditekan dan data valid, maka sistem akan mencetak hasil laporan buku besar.

Laporan Buku Besar					
U.B. Miftahur Rizki					
Tahun Periode 2008					
Tanggal	No. Referensi	Keterangan	Debet	Kredit	Balance
31-12-2007		Saldo awal			30,000,000
15-01-2008	0000001	Bayar listrik	100,000		29,900,000
30-01-2008	0000002	Bayar gaji	3,000,000		26,900,000
02-02-2008	0000003	Pembelian barang	2,000,000		24,900,000
10-02-2008	1000001	Penjualan tunai	3,000,000		27,900,000
		Jumlah	3,000,000	5,100,000	27,900,000

-1-

29-02-2008

Gambar 6.35 Pada saat tombol proses ditekan dan hasil laporan buku besar

Sumber: Pengujian

- Laporan laba rugi
 - Pada saat tombol proses ditekan dan data valid, maka sistem akan mencetak hasil laporan laba rugi.

Laporan Laba Rugi		
U.B. Miftahur Rizki		
Tahun Periode 2008		
Pendapatan usaha		3,000,000
Beban usaha		
Bayar listrik	100,000	
Bayar gaji	3,000,000	
	Total beban usaha	3,100,000
	Rugi usaha	100,000

-1- 29-02-2008

Gambar 6.36 Pada saat tombol proses ditekan dan hasil laporan laba rugi

Sumber: Pengujian

6.2.2.1 Analisa Pengujian

Dari hasil pengujian sistem setiap proses dapat berjalan dengan baik, ditunjukkan dari berfungsinya masing-masing komponen dalam form aplikasi.

6.3 Pengujian Waktu Akses *Query*

6.3.1 Tujuan

- Pengujian dilakukan untuk mengetahui waktu yang dibutuhkan untuk melakukan sebuah *query* pada basis data.
- Pengujian dilakukan untuk mendapatkan perbandingan waktu *query* yang dilakukan terhadap jumlah data yang berbeda pada basis data.

6.3.2 Prosedur Pengujian

- Mengakses query_tes.php, Halaman ini dibuat khusus untuk melakukan pengujian waktu akses *query*.
- Tabel transaksi diberi masukan data sebanyak 100, 200, 400, 600, 800, dan 1000 *data entry* dalam basis data.
- Melakukan perhitungan untuk mendapatkan nilai waktu akses *query* yang dilakukan terhadap jumlah data yang berbeda dalam basis data.

6.3.3 Hasil Pengujian

- Hasil pengujian waktu akses *query* terhadap tabel barang dengan jumlah data sebanyak 100, 200, 400, 600, 800, 1000 *data entry* ditunjukkan pada gambar 6.11.

Loop = 5 Jumlah Data = 100 Query 1 = 0.2395 ms Query 2 = 0.1236 ms Query 3 = 0.1245 ms Query 4 = 0.1236 ms Query 5 = 0.1229 ms Rata - rata waktu query = 0.14682 ms	Loop = 5 Jumlah Data = 200 Query 1 = 0.3814 ms Query 2 = 0.252 ms Query 3 = 0.2538 ms Query 4 = 0.2548 ms Query 5 = 0.248 ms Rata - rata waktu query = 0.278 ms
Loop = 5 Jumlah Data = 400 Query 1 = 0.5464 ms Query 2 = 0.5004 ms Query 3 = 0.4864 ms Query 4 = 0.4992 ms Query 5 = 0.486 ms Rata - rata waktu query = 0.50368 ms	Loop = 5 Jumlah Data = 600 Query 1 = 0.8142 ms Query 2 = 0.7308 ms Query 3 = 0.7254 ms Query 4 = 0.7254 ms Query 5 = 0.726 ms Rata - rata waktu query = 0.74436 ms
Loop = 5 Jumlah Data = 800 Query 1 = 1.072 ms Query 2 = 0.96 ms Query 3 = 0.9528 ms Query 4 = 0.9512 ms Query 5 = 0.952 ms Rata - rata waktu query = 0.9776 ms	Loop = 5 Jumlah Data = 1000 Query 1 = 1.684 ms Query 2 = 1.232 ms Query 3 = 1.212 ms Query 4 = 1.209 ms Query 5 = 1.203 ms Rata - rata waktu query = 1.308 ms

Gambar 6.37 Hasil pengujian waktu akses *query* terhadap tabel barang**Sumber:** Pengujian

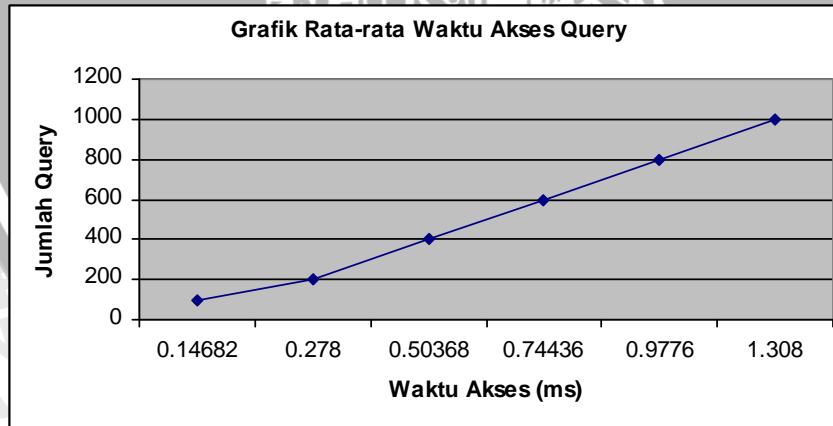
- Rata-rata pengujian waktu akses *query* barang dengan jumlah data Tabel barang sebanyak 100, 300, 500, 700, 900, dan 1100 data *entry* ditunjukkan dalam Tabel 6.22

Tabel 6.22 Tabel Pengujian Waktu Akses *Query*

Pengujian	Jumlah Record	Waktu Akses (ms)
	Tabel Transaksi	
1	100	0.14682
2	200	0.278
3	400	0.50368
4	600	0.74436
5	800	0.9776
6	1000	1.308

Sumber : Pengujian

- Grafik perbandingan waktu akses *query* barang dengan jumlah data yang berbeda diperlihatkan di dalam Gambar 6.12.

**Gambar 6.38** Grafik Rata-Rata Waktu Akses *Query***Sumber:** Pengujian

- Dari pengujian waktu akses *query* dapat diketahui waktu akses per *record* untuk *query* barang, merupakan hasil bagi antara waktu akses dengan jumlah *record*. Perbandingan waktu akses per *record* dari *query* barang ditunjukkan dalam Tabel 6.23.

Tabel 6.23 Tabel Perbandingan Waktu Akses Per *Record*

Pengujian	Jumlah <i>record</i>	Waktu akses (ms)	Waktu akses per <i>record</i> (ms)
1	100	0.14682	0.0014682
2	200	0.278	0.00278
3	400	0.50368	0.0050368
4	600	0.74436	0.0074436
5	800	0.9776	0.009776
6	1000	1.308	0.01308
Rata-rata waktu akses <i>query</i> per <i>record</i> (ms)			0.00659743

Sumber : Pengujian

6.3.4 Analisa Pengujian

Berdasarkan hasil pengujian pada Tabel 6.23 waktu akses *query* untuk *query* transaksi pemesanan dengan jumlah data Tabel pesan sebanyak 100, 200, 400, 600, 800, dan 1000 data *entry*, dapat diketahui bahwa rata-rata waktu akses per *record* sebesar 0.00659743 ms. Dari hasil pengujian dan analisis dapat diketahui bahwa waktu akses rata-rata *query* transaksi pemesanan dengan jumlah data yang ditampilkan semakin banyak, maka waktu akses akan meningkat secara linear.

BAB VII

PENUTUP

7.1 Kesimpulan

Kesimpulan yang didapat dari perancangan, implementasi, dan pengujian aplikasi Sistem informasi akuntansi dengan laporan laba rugi menggunakan metode multiple step antara lain:

- Kebutuhan-kebutuhan (fungsional dan non-fungsional) untuk merancang Sistem Informasi Akuntansi dengan Laporan Laba/Rugi Menggunakan Metode Multiple Step telah dirumuskan di dalam daftar kebutuhan. Daftar kebutuhan ini yang menjadi acuan untuk perancangan yang lebih detail.
- Sistem Informasi Akuntansi dikembangkan dengan menggunakan bahasa pemrograman Java 2 Standart Edition dan basis data MySQL 5.0.45. Menggunakan editor pemrograman NetBeans 5 dengan koneksi basis data menggunakan JDBC.
- Sistem Informasi Akuntansi menggunakan pendekatan berorientasi objek.
- Sistem Informasi Akuntansi dikembangkan pada perangkat komputer menggunakan sistem operasi *windows vista ultimate sp1*.
- Pengujian basis data menggunakan Sybase Power Designer 12.5.
- Sistem dapat berjalan sesuai dengan kebutuhan perancangan.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan “Sistem Informasi Akuntansi dengan Laporan Laba/Rugi Menggunakan Metode Multiple Step ” antara lain :

- Pengembangan dibidang akuntansi yang lebih luas, yaitu analisis arus keuangan.

- Meningkatkan keamanan basis data agar tidak bisa diakses oleh pihak-pihak yang tidak berkepentingan.



DAFTAR PUSTAKA

[APS-05] Nugroho, Adi, 2005, "Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek", Informatika Bandung.

[PGS-07] Sri Hartati, G, Herry Suharto, B, Soesilo Wijono, M, 2007,"Pemrograman GUI Swing Java dengan NetBeans5", Andi Yogyakarta.

[MAD-08] Komputer, Wahana, 2008,"Membuat Aplikasi Database dengan Java dan MySQL",Andi Yogyakarta.

[MAJ-06] Komputer, Wahana, 2006,"Membuat Aplikasi Database dengan Java2",2006,Andi Yogyakarta.

[PBO-04] MM, S.Kom, Indrajani, S.Kom, Martin,2004,"Pemrograman Berorientasi Objek dengan Java", Elex Media Koputindo.

[SIP-08] Supardi, Yuniar, Ir,2008,"Sistem Informasi Penjualan dengan Java",Elex Media Komputindo.

[PLM-08] Nugroho, Bunafit, 2008,"Panduan Lengkap Menguasai Perintah SQL",Mediakita.

[RSB-04] MT,Ir, Hariyanto, Bambang, 2004,"Rekayasa Sistem Berorientasi Objek",Informatika Bandung.

[UML-06] MM,S.Kom, Irwanto, Djon, 2006,"Perancangan Object Oriented Software dengan UML" Andi Yogyakarta.,,

[APD-05] Sulistiawan, Dedhy, 2005,"Akuntansi Perusahaan Dagang Menggunakan Accurate",Elex Media Komputindo.

[MAP-08] Huda, Miftakhul, Subiyantoro,Cuk, Nugroho, Bunafit, 2008,"Membuat Aplikasi Penjualan dengan Java dan MySQL",Elex Media Komputindo.

