

**ANALISIS DAN PEMBUATAN APLIKASI AKUNTANSI BERBASIS  
WEB PADA IIUC**



**ARIF KUNTO WIBISONO  
106091002876**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2011 M/1432 H**

# **ANALISIS DAN PEMBUATAN APLIKASI AKUNTANSI BERBASIS**

## **WEB PADA IIUC**

Oleh:

ARIF KUNTO WIBISONO

106091002876



Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
Sarjana Komputer

Fakultas Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SYARIF HIDAYATULLAH  
JAKARTA  
2011 M/1432 H**

### LAMPIRAN 3

#### DAFTAR PERTANYAAN WAWANCARA

1. Bagaimana pemanfaatan Teknologi Informasi / IT pada proses keuangan di IIUC?
2. Apakah semua karyawan dan manajer sudah memiliki keterampilan dalam menggunakan sarana IT yang tersedia?
3. Bagaimana proses keuangan di IIUC?
4. Apakah ada rencana menggunakan aplikasi akuntansi berbasis *web* untuk menunjang proses keuangan di IIUC ?
5. Apakah sudah mulai diterapkan aplikasi akuntansi berbasis *web* tersebut?
6. Apa *output* yang diharapkan dari aplikasi akuntansi?

Jawaban :

1. Jawab : Masih sangat *manual*. Masih menggunakan *Microsoft Excel* atau *Microsoft Word*
2. Jawab : Sudah
3. Jawab : Karyawan meminta uang kepada bagian keuangan. Bila disetujui, bagian keuangan akan memberikan uang tersebut. Kemudian karyawan akan mendapat bukti transaksi. Bukti transaksi ini akan diberikan kepada bagian administrasi untuk dikumpulkan. Setelah sebulan, bukti – bukti tersebut akan diberikan kepada bagian keuangan untuk dibukukan. Untuk pembayaran mahasiswa, maka mahasiswa harus membayar uang kuliah di bank. Setelah itu, bukti transaksi tersebut akan diserahkan kepada bagian keuangan, untuk selanjutnya dibuat pembukuan.

4. Jawab : Ya, ada.
5. Jawab : Belum
6. Jawab : Dapat memberikan laporan keuangan secara otomatis sehingga manajer dapat melihat keadaan keuangan pada IIUC. Selain itu juga dapat menampilkan keadaan akun – akun yang terdapat di IIUC dalam rentang waktu tertentu, sehingga keadaan keuangan pada IIUC dapat dipantau secara lebih jelas.



## ABSTRAK

**Arif Kunto Wibisono**, Analisis Dan Pembuatan Aplikasi Akuntansi Berbasis Web Pada IIUC, dibimbing oleh **Yusuf Durachman, M.Sc, MIT** dan **Rusdianto Rustam, M.Sc, Dr.**

Saat ini, pemasukan data-data transaksi keuangan yang berlangsung pada IIUC dilakukan secara *manual* oleh staf keuangan. Transaksi keuangan hanya bisa dilakukan pada komputer lokal yang ada pada IIUC saja, sehingga bila staf keuangan atau orang yang bersangkutan sedang berada di luar IIUC, maka ia tidak bisa melakukan transaksi. Ini merupakan suatu hambatan bagi manajer, karena data – data ini sangat penting untuk mengetahui keadaan keuangan saat ini pada IIUC. Melihat permasalahan yang terjadi, maka perlu dibuat aplikasi akuntansi yang dapat membantu pekerjaan bagian keuangan dan serta manajer dalam melihat informasi keuangan. Metode pengembangan sistem yang dipakai dalam penelitian ini adalah *incremental*. Dalam aplikasi ini terdapat logika – logika akuntansi yang digunakan untuk menghitung transaksi – transaksi serta saldo dari rekening yang digunakan. Hasil dari aplikasi ini adalah laporan keuangan yang ditujukan kepada manajer. Pada akhirnya, aplikasi ini dapat membantu pekerjaan bagian keuangan dalam memasukkan transaksi dan manajer dalam melihat laporan keuangan.

*Kata Kunci : Analisa, Perancangan, Aplikasi, Akuntansi, Web.*

*Jumlah Halaman : xviii halaman + 159 Halaman + 85 gambar +23 tabel + 20 lampiran*

*Jumlah Daftar Pustaka : 21 sumber ( tahun 1992 s/d 2010)*

## DAFTAR GAMBAR

	Halaman
Gamba 2.1 Proses Akuntansi.....	13
Gambar 2.2 Penggolongan Rekening.....	14
Gambar 2.3 Struktur Organisasi IIUC.....	35
Gambar 3.1 Paradigma <i>incremental</i> .....	47
Gambar 3.2 Kerangka Berpikir.....	48
Gambar 4.1 <i>Flowchart</i> sistem berjalan untuk permintaan uang dengan <i>Petty Cash</i> .....	52
Gambar 4.2 <i>Flowchart</i> sistem berjalan proses permintaan uang dengan cek.....	53
Gambar 4.3 <i>Flowchart</i> sistem berjalan proses pembayaran mahasiswa di IIUC.....	54
Gambar 4.4 <i>Flowchart</i> sistem berjalan proses laporan keuangan.....	55
Gambar 4.6 Diagram Konteks Aplikasi Akuntansi.....	66
Gambar 4.7 Diagram <i>Level 1</i> Aplikasi Akuntansi.....	68
Gambar 4.8 Diagram <i>Level 2</i> Proses 1.0.....	69
Gambar 4.9 Diagram <i>Level 3</i> Proses 1.3.....	70
Gambar 4.10 Diagram <i>Level 2</i> Proses 2.0.....	70
Gambar 4.11 Diagram <i>Level 3</i> Proses 2.3.....	71
Gambar 4.12 Diagram <i>Level 2</i> Proses 3.0.....	72
Gambar 4.13 Diagram <i>Level 3</i> Proses 3.3.....	73
Gambar 4.14 Diagram <i>Level 2</i> Proses 4.0 .....	73

Gambar 4.15 Diagram <i>Level 3</i> Proses 4.3.....	74
Gambar 4.16 Diagram <i>Level 2</i> Proses 5.0.....	75
Gambar 4.17 Diagram <i>Level 3</i> Proses 5.1.....	76
Gambar 4.18 Diagram <i>Level 3</i> Proses 5.2.....	76
Gambar 4.19 Diagram <i>Level 3</i> Proses 5.3.....	77
Gambar 4.20 Diagram <i>Level 3</i> Proses 5.4.....	78
Gambar 4.21 diagram <i>Level 2</i> Proses 6.0.....	79
Gambar 4.21 Diagram <i>Level 2</i> Proses 7.0.....	80
Gambar 4.23 Diagram <i>Level 2</i> Proses 8.0.....	80
Gambar 4.24 Diagram <i>Level 2</i> Proses 9.0.....	81
Gambar 4.25 <i>Flowchart</i> Halaman Login.....	100
Gambar 4.26 <i>Flowchart</i> Halaman <i>Admin</i> .....	102
Gambar 4.27 <i>Flowchart</i> Halaman <i>User</i> .....	103
Gambar 4.28 <i>Flowchart</i> Halaman Data <i>User</i> .....	103
Gambar 4.29 <i>Flowchart</i> Tambah Data <i>User</i> .....	104
Gambar 4.30 <i>Flowchart</i> Edit Data <i>User</i> .....	105
Gambar 4.31 <i>Flowchart</i> Hapus Data <i>User</i> .....	106
Gambar 4.32 <i>Flowchart</i> Halaman Data Rekening.....	107
Gambar 4.33 <i>Flowchart</i> Tambah Data Rekening .....	108
Gambar 4.34 <i>Flowchart</i> Edit Data Rekening.....	109
Gambar 4.35 <i>Flowchart</i> Hapus Data Rekening.....	110
Gambar 4.36 <i>Flowchart</i> Cari Data Rekening.....	110
Gambar 4.37 <i>Flowchart</i> Data Subklasifikasi Rekening.....	111

Gambar 4.38 <i>Flowchart</i> Tambah Data Subklasifikasi Rekening.....	112
Gambar 4.39 <i>Flowchart</i> Edit Data Subklasifikasi Rekening.....	113
Gambar 4.40 <i>Flowchart</i> Hapus Data Subklasifikasi Rekening.....	114
Gambar 4.41 <i>Flowchart</i> Data Tipe Jurnal.....	114
Gambar 4.42 <i>Flowchart</i> Tambah Tipe Jurnal .....	115
Gambar 4.43 <i>Flowchart</i> Edit Tipe Jurnal.....	116
Gambar 4.44 <i>Flowchart</i> Hapus Tipe Jurnal .....	117
Gambar 4.45 <i>Flowchart</i> Halaman Jurnal Umum.....	118
Gambar 4.46 <i>Flowchart</i> Tambah Baris.....	119
Gambar 4.47 <i>Flowchart</i> Tambah Jurnal Umum.....	119
Gambar 4.48 <i>Flowchart</i> Halaman Jurnal Penyesuaian.....	120
Gambar 4.49 <i>Flowchart</i> Tambah Baris.....	121
Gambar 4.50 <i>Flowchart</i> Tambah Jurnal Penyesuaian.....	121
Gambar 4.51 <i>Flowchart</i> Halaman Jurnal Penutup .....	122
Gambar 4.52 <i>Flowchart</i> Proses Jurnal Penutup.....	123
Gambar 4.53 <i>Flowchart</i> Halaman Data Jurnal.....	123
Gambar 4.54 <i>Flowchart</i> Halaman Data Jurnal Untuk <i>admin</i> .....	124
Gambar 4.55 <i>Flowchart</i> Halaman Data Jurnal Untuk <i>user</i> .....	125
Gambar 4.56 <i>Flowchart</i> Halaman Cari Data Jurnal.....	126
Gambar 4.57 <i>Flowchart</i> Halaman Data Jurnal Berdasarkan Tanggal.....	126
Gambar 4.58 <i>Flowchart</i> Halaman Ekspor Data Jurnal .....	127
Gambar 4.59 <i>Flowchart</i> Halaman Jurnal Koreksi.....	127
Gambar 4.60 <i>Flowchart</i> Halaman Jurnal Koreksi.....	128



Gambar 4.61 <i>Flowchart</i> Halaman Data Buku Besar.....	129
Gambar 4.62 <i>Flowchart</i> Halaman Grafik .....	130
Gambar 4.63 <i>Flowchart</i> Halaman Grafik Harta.....	131
Gambar 4.64 <i>Flowchart</i> Halaman Grafik Kewajiban .....	132
Gambar 4.65 <i>Flowchart</i> Halaman Grafik Modal .....	132
Gambar 4.66 <i>Flowchart</i> Halaman Grafik Pendapatan.....	133
Gambar 4.67 <i>Flowchart</i> Halaman Grafik Beban .....	134
Gambar 4.68 <i>Flowchart</i> Halaman Grafik Laba Rugi.....	135
Gambar 4.69 <i>Flowchart</i> Halaman Laporan Keuangan.....	136
Gambar 4.70 <i>Flowchart</i> Halaman Laporan Laba Rugi.....	137
Gambar 4.71 <i>Flowchart</i> Halaman Neraca Saldo.....	138
Gambar 4.72 <i>Flowchart</i> Halaman Laporan Perubahan Modal.....	139
Gambar 4.73 <i>Flowchart</i> Proses Ekspor Laporan Laba Rugi.....	140
Gambar 4.74 <i>Flowchart</i> Proses Ekspor Neraca Saldo.....	141
Gambar 4.75 <i>Flowchart</i> Proses Ekspor Laporan Perubahan Modal.....	142
Gambar 4.76 <i>Interface</i> Halaman Login.....	143
Gambar 4.77 <i>Interface</i> Halaman Menu Utama .....	143
Gambar 4.78 <i>Interface</i> Halaman Data Rekening.....	144
Gambar 4.79 <i>Interface</i> Jurnal Umum, Jurnal Penyesuaian, dan Jurnal Penutup .....	144
Gambar 4.80 <i>Interface</i> Halaman Jurnal.....	145

## DAFTAR ISI

	Halaman
Halaman Judul .....	i
Persetujuan Pembimbing .....	ii
Halaman Pengesahan .....	iii
Halaman Pernyataan .....	iv
Abstrak .....	v
Kata Pengantar .....	vi
Daftar Isi .....	viii
Daftar Gambar .....	xii
Daftar Tabel .....	xvi
Daftar Lampiran .....	xviii

## BAB I PENDAHULUAN

1.1 Latar Belakang.....	2
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Dan Manfaat .....	4
1.4.1 Tujuan Penulisan.....	4
1.4.2 Manfaat Penulisan.....	4
1.5 Waktu Dan Tempat Pelaksanaan.....	5
1.6 Metodologi Penelitian.....	5
1.6.1 Metode Pengumpulan Data.....	5
1.6.2 Metode Pengembangan Sistem.....	6

1.7 Sistematika Penulisan.....	7
--------------------------------	---

## **BAB II LANDASAN TEORI**

2.1 Perancangan Aplikasi.....	10
2.1.1 Pengertian Aplikasi.....	10
2.1.2 Akuntansi.....	10
2.1.3 <i>Incremental</i> .....	22
2.1.4 <i>Entity Relaitonship Diagram (ERD)</i> .....	24
2.1.5 <i>Data Flow Diagram</i> .....	25
2.1.6 <i>Flowchart</i> .....	26
2.1.7 Studi Sejenis.....	31
2.1.8 IIUC.....	33
2.2 Instalasi Perangkat Lunak .....	36
2.2.1 <i>Web Browser</i> .....	36
2.2.2 PHP.....	36
2.2.3 MySQL .....	37
2.2.4 Apache .....	38
2.2.5 XAMPP .....	39
2.2.6 HTML.....	40
2.2.7 jQuery.....	41
2.2.8 FusionChart Free .....	42
2.2.9 CSS.....	42

## **BAB III METODOLOGI PENELITIAN**

3.1 Metode Pengumpulan Data .....	44
-----------------------------------	----

3.1.1	Studi Lapangan Atau Observasi.....	44
3.1.2	Studi Pustaka.....	45
3.1.3	Wawancara.....	45
3.2	Metode Pengembangan Sistem.....	46
3.2.1	<i>Communication</i> .....	49
3.2.2	<i>Planning</i> .....	49
3.2.3	<i>Modeling</i> .....	49
3.2.4	<i>Construction</i> .....	50
3.2.5	<i>Deployment</i> .....	50
 <b>BAB IV HASIL DAN PEMBAHASAN</b>		
4.1	<i>Communication</i> .....	51
4.2	<i>Planning</i> .....	56
4.3	<i>Modeling</i> .....	57
4.3.1	Analisis Kebutuhan Aplikasi .....	57
4.3.2	Analisis Proses Akuntansi.....	59
4.3.3	<i>Data Flow Diagram</i> .....	66
4.3.4	<i>Normalization</i> .....	84
4.3.5	<i>Entity Relationship Diagram</i> .....	94
4.3.6	<i>Flowchart</i> .....	100
4.3.7	Desain <i>Interface</i> .....	143
4.4	<i>Construction</i> .....	145
4.5	<i>Deployment</i> .....	146
4.5.1	<i>Testing</i> .....	146
4.5.2	<i>Implementation</i> .....	155

## **BAB V KESIMPULAN DAN SARAN**

5.1 Kesimpulan.....156

5.2 Saran.....156

**DAFTAR PUSTAKA .....158**

**LAMPIRAN - LAMPIRAN**



## **BAB I**

### **PENDAHULUAN**

Bab 1 menerangkan tentang mengapa dibuat penelitian ini serta pandangan menyeluruh tentang penelitian ini. Disini akan dijelaskan mengenai alasan dibuatnya penelitian ini, masalah – masalah yang ada dalam teknologi informasi, masalah – masalah yang dihadapi dalam penelitian, pembatasan masalah yang ada dalam penelitian, tujuan yang ingin dicapai dari penelitian ini, manfaat yang akan dicapai dari penelitian ini baik untuk penulis, pihak universitas, dan pihak tempat penelitian, waktu dan tempat pelaksanaan penelitian, metodologi penelitian yang digunakan untuk melakukan penelitian, serta sistematika penulisan dari penelitian ini.

Bab ini terdiri atas subbab :

1. Latar belakang
2. Rumusan masalah
3. Batasan masalah
4. Tujuan dan manfaat
5. Waktu dan tempat pelaksanaan
6. Metodologi penelitian
7. Sistematika penulisan

## 1.1 Latar Belakang

Saat ini kesuksesan organisasi sangat bergantung kepada kemampuan mengumpulkan, memburu, dan mengorganisasi data mengenai operasi – operasi perusahaan secara akurat dan tepat, mampu mengelola data secara efektif, serta menggunakannya untuk analisis dan menuntun aktivitas – aktivitas perusahaan. Jumlah informasi yang telah tersedia telah meledak sedemikian besar serta pandangan data sebagai aset organisasi telah mulai tumbuh. Namun tanpa kemampuan yang memadai dalam pengelolaan data sehingga data mampu memberi informasi yang relevan secara cepat terhadap suatu persoalan, maka jumlah data yang besar justru akan menjadi beban yang harus ditanggung perusahaan. (Hariyanto, 2004 : 3)

Saat ini, pemasukan data-data transaksi keuangan yang berlangsung pada IIUC dilakukan secara *manual* oleh staf keuangan. Transaksi keuangan hanya bisa dilakukan pada komputer lokal yang ada pada IIUC saja, sehingga bila staf keuangan atau orang yang bersangkutan sedang berada di luar IIUC, maka ia tidak bisa melakukan transaksi. Ini merupakan suatu hambatan bagi manajer, karena data – data ini sangat penting untuk mengetahui keadaan keuangan saat ini pada IIUC.

Dengan pembuatan aplikasi baru diharapkan akan mempermudah memproses transaksi keuangan. Para staf keuangan dapat memproses transaksi keuangan secara cepat, manajer dapat melihat keadaan keuangan meskipun tidak berada pada kantor pusat, dan pemeliharaan data dapat dilakukan dengan lebih mudah.

Berdasarkan hal inilah, penulis melakukan penelitian skripsi dengan judul “**Analisis Dan Pembuatan Aplikasi Akuntansi Berbasis Web Pada IIUC**”.

## **1.2 Rumusan Masalah**

Sesuai dengan permasalahan yang diangkat pada latar belakang penulisan, maka penulis merumuskan permasalahan yang ada yaitu :

1. Bagaimana membuat aplikasi akuntansi berbasis *web*.
2. Bagaimana membuat aplikasi akuntansi yang bisa melakukan pemeliharaan data secara lebih teratur.
3. Bagaimana cara penyajian aplikasi akuntansi ini secara mudah dan interaktif.

## **1.3 Batasan Masalah**

Dalam penelitian ini terdapat batasan, yaitu :

1. Pencatatan transaksi keuangan yang ada ke dalam jurnal.
2. Pemrosesan transaksi keuangan yang berlangsung.
3. Pembuatan laporan keuangan dari transaksi – transaksi yang telah berlangsung ke dalam Microsoft Excel.
4. Laporan keuangan yang dibuat adalah neraca saldo, laporan laba rugi, dan laporan perubahan ekuitas.
5. Aplikasi keuangan dibuat untuk lingkungan IIUC.



## **1.4 Tujuan dan Manfaat**

### **1.4.1 Tujuan Penulisan**

Penelitian skripsi ini dilakukan dengan tujuan :

1. Melakukan pembuatan aplikasi akuntansi yang baru untuk mempermudah pekerjaan staf keuangan
2. Melakukan pemrosesan secara otomatis untuk mempermudah pemrosesan transaksi keuangan.
3. Melakukan pemeliharaan data secara lebih teratur.

### **1.4.2 Manfaat Penulisan**

Bagi IIUC :

1. Sebagai aplikasi untuk pencatatan transaksi – transaksi keuangan pembuatan laporan keuangan di IIUC.
2. Sebagai bahan evaluasi dalam implementasi aplikasi akuntansi.

Bagi penulis :

1. Menerapkan ilmu-ilmu yang telah diperoleh selama kuliah.
2. Memahami proses akuntansi secara umum.
3. Sebagai portofolio untuk penulis yang berguna untuk masa yang akan datang.
4. Untuk memenuhi salah satu syarat kelulusan Strata Satu (S1) Program Studi Teknik Informatika Fakultas Sains dan Teknologi.

Bagi Universitas :

1. Sebagai bahan referensi untuk penelitian yang akan datang.
2. Sebagai bahan evaluasi bagi universitas dalam mengembangkan keilmuan, dalam hal ini yang berkaitan dengan aplikasi akuntansi untuk menunjang pengambilan keputusan.

### **1.5 Waktu dan Tempat Pelaksanaan**

Penelitian skripsi dilaksanakan pada :

Tanggal : 3 Juli 2010 – 31 Oktober 2010

Tempat : IIUC

Jl. Melawai Raya no. 8C Kebayoran Baru, Jakarta Selatan

Telp. / Fax : 021-7251885 / 021-7261577

### **1.6 Metodologi Penelitian**

Dalam penelitian ini, penulis menggunakan dua buah metode, yaitu metode pengumpulan data dan metode pengembangan sistem.

#### **1.6.1 Metode Pengumpulan Data**

Dalam metode pengumpulan data, penulis menggunakan tiga metode, yaitu :

1. Observasi : Metode observasi ini biasanya digunakan untuk mengetahui keadaan yang berlangsung secara detail.

2. Interview : Melakukan wawancara langsung dengan orang yang bersangkutan, sehingga akan didapatkan data yang relevan.
3. Studi Pustaka : Mengumpulkan data dengan mencari referensi dari buku-buku serta penelitian sejenis.

### 1.6.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang dipakai oleh penulis yaitu metode *incremental*.

*Incremental* memiliki 5 tahapan, yaitu (Pressman, 2010 : 39) :

#### 1. *Communication*

Bertemu dengan *user* untuk mendefinisikan tujuan – tujuan umum serta syarat – syarat untuk pembuatan perangkat lunak.

#### 2. *Planning*

Membuat jadwal – jadwal yang akan digunakan sebagai acuan untuk membuat perangkat lunak.

#### 3. *Modeling*

Membuat model, yakni dengan menganalisa dan mendesain aplikasi yang akan dibuat.

#### 4. *Construction*

Membuat desain yang sudah direncanakan menjadi kode – kode program serta desain GUI yang nantinya akan menjadi perangkat lunak serta melakukan *testing*.

#### 5. *Deployment*

Aplikasi yang telah jadi kemudian diantarkan kepada *user* untuk digunakan agar didapatkan *feedback* dari *user*.

### 1.7 Sistematika Penulisan

Sistematika penulisan skripsi ini secara keseluruhan terdiri dari lima bab, dimana secara garis besar masing-masing bab membahas hal-hal sebagai berikut:

#### BAB I PENDAHULUAN

Pada bab ini, menerangkan latar belakang, perumusan masalah, pembatasan masalah, tujuan dan manfaat penulisan, waktu dan tempat pelaksanaan dan sistematika penulisan.

#### BAB II LANDASAN TEORI

Bab ini membahas tentang teori-teori dasar yang relevan yang digunakan untuk memecahkan persoalan yang dibahas pada praktek kerja lapangan ini. Membahas pengertian dari akuntansi, ERD, DFD, PHP, MySQL, dan XAMPP.

#### BAB III METODOLOGI PENELITIAN

Menguraikan secara rinci metode penelitian yang digunakan. Metode tersebut meliputi metode pengumpulan data

dan metode pengembangan aplikasi. Disini dijelaskan alasan memilih metode penelitian tersebut.

#### BAB IV HASIL DAN PEMBAHASAN

Menguraikan hasil penelitian. Membahas analisis, perancangan, implementasi dan evaluasi. Analisis tersebut meliputi analisis kebutuhan aplikasi, analisis proses akuntansi, rancangan ERD, DFD, *flowchart*, serta rancangan GUI aplikasi.

#### BAB V PENUTUP

Pada bab ini, terdapat kesimpulan dari apa yang telah diuraikan pada bab sebelumnya serta saran-saran yang diharapkan berguna bagi perkembangan aplikasi ini di masa mendatang.

## BAB II

### LANDASAN TEORI

Bab 2 ini akan menjelaskan secara detail tentang landasan teori yang digunakan dalam penelitian ini. Bagian pertama yaitu perancangan aplikasi. Perancangan aplikasi ini mencakup metode – metode serta dasar – dasar yang digunakan untuk merancang aplikasi, serta pandangan umum mengenai tempat penelitian ini. Metode – metode dan dasar – dasar yang digunakan yaitu pengertian aplikasi, akuntansi, *incremental*, *Entity Relationship Diagram*, *Data Flow Diagram*, *flowchart*, dan studi sejenis. Tempat penelitian ini adalah IIUC.

Penulis memakai ERD untuk *data modeling* untuk memberikan gambaran mengenai perancangan *database*. DFD dipakai sebagai *process model* untuk memberikan gambaran bagaimana data tersebut diproses. *Flowchart* dipakai untuk menjelaskan logika – logika yang berjalan pada aplikasi.

Bagian kedua yaitu instalasi perangkat lunak pendukung. Disini akan dijelaskan mengenai pandangan umum tentang perangkat lunak yang digunakan dalam penelitian ini. Perangkat lunak yang digunakan antara lain *Web Browser*, PHP, MySQL, Apache, XAMPP, jQuery, HTML, FusionCharts Free, dan CSS. Penulis menggunakan XAMPP karena PHP, MySQL, dan Apache sudah termasuk didalamnya dan karena *hosting* yang dipakai berbasis PHP. jQuery dipakai untuk membuat aplikasi lebih dinamis, sedangkan FusionCharts Free dipakai sebagai grafik untuk melihat keadaan keuangan di IIUC dalam rentang waktu tertentu.

## 2.1 Perancangan Aplikasi

### 2.1.1 Pengertian Aplikasi

Aplikasi adalah sebuah program komputer yang digunakan untuk tugas yang spesifik atau khusus, seperti akuntansi, analisis dalam bidang sains atau medis, atau pemrosesan kata. Bagian “aplikasi” mengacu kepada keseluruhan set dari program yang secara kolektif mengimplementasikan sebuah proses bisnis yang spesifik. Program-program individual yang mengimplementasikan bagian dari proses bisnis ini dikenal sebagai “program aplikasi” atau “perangkat lunak aplikasi” (Nader, 1992: 20).

Program aplikasi = program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh – contoh aplikasi ialah program pemroses kata dan *Web Browser*. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi lain yang mendukung. Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app. Secara historis, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. (Febrian, 2007 : 35)

### 2.1.2 Akuntansi

Akuntansi adalah proses pengidentifikasian, pencatatan, dan pengkomunikasian kejadian – kejadian ekonomi suatu organisasi (perusahaan ataupun bukan perusahaan) kepada para pemakai

informasi yang berkepentingan. Akuntansi memberikan jasa yang sangat vital dengan memasok informasi yang dibutuhkan oleh para pengambil keputusan untuk membuat “pilihan – pilihan yang masuk akal di antara alternatif – alternatif penggunaan sumber – sumber daya yang langka dalam menjalankan bisnis dan kegiatan - kegiatan ekonomi”(Simamora, 2000 : 4). Pengertian langka disini adalah sumber daya tersebut istimewa. Sumber daya itu ada dalam perusahaan, namun penggunaannya untuk aktivitas ekonomi tidak dapat langsung dipakai oleh karyawan. Misalkan ketika ingin mengambil dana dari kas untuk pembelian komputer, maka karyawan harus melapor kepada bagian keuangan untuk mendapat izin. Ketika melakukan penjualan barang produksi, maka karyawan harus melapor ke bagian keuangan untuk dicatat pendapatannya.

Akuntansi dapat didefinisikan sebagai sistem informasi yang menghasilkan laporan kepada pihak – pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi perusahaan.(Warren, *et al.*2006 : 10). Aktivitas ekonomi disini adalah aktivitas perusahaan dalam menggunakan sumber daya perusahaan untuk kelangsungan hidup perusahaan tersebut. Misalkan pembelian peralatan, penggunaan kas, penjualan saham, dan lain – lain. Kondisi perusahaan yang dimaksud disini yaitu keadaan keuangan pada perusahaan tersebut. Dari pencatatan aktivitas – aktivitas ekonomi tersebut, maka dapat



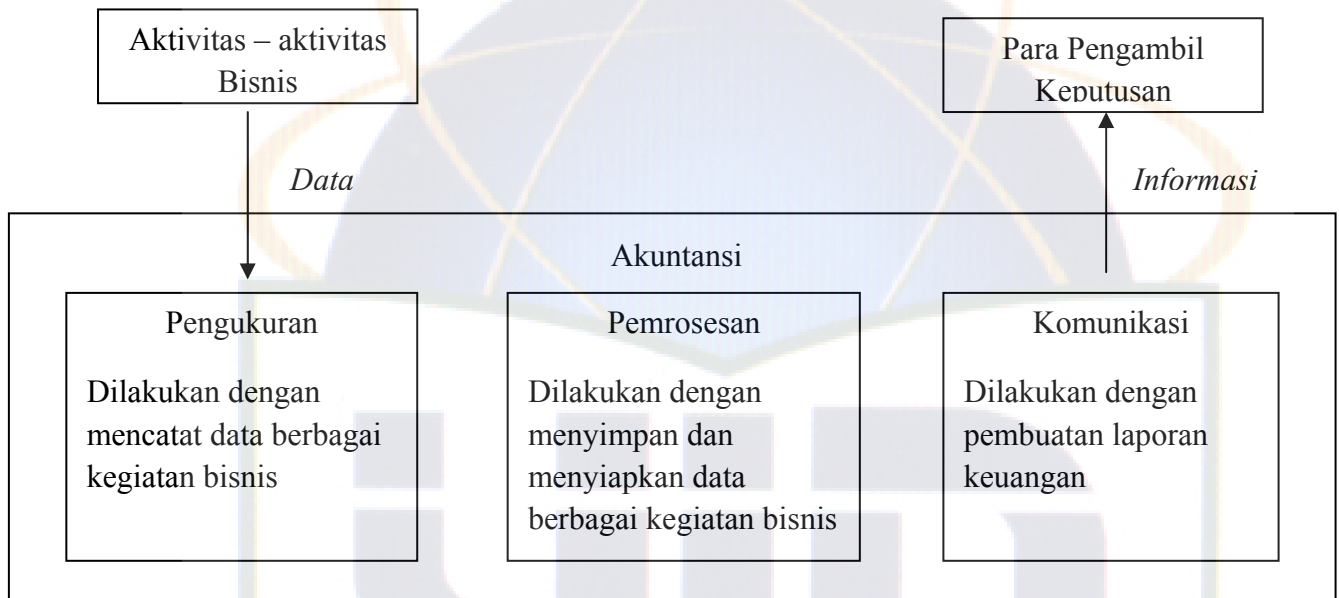
diketahui pendapatan, pengeluaran, laba, rugi, aktiva dan pasiva dalam perusahaan tersebut.

Menurut AAA (*American Accounting Association*) (dalam Kardiman, *et al* .2007 : 2), akuntansi adalah proses pengidentifikasian, pengukuran, dan penyampaian informasi ekonomi yang memungkinkan dilakukannya penilaian dan keputusan yang tepat bagi para pemakai informasi tersebut. Pengertian ini menandakan bahwa akuntansi adalah sebuah sistem informasi, yaitu sebuah sistem yang menghasilkan informasi keuangan yang berdasarkan informasi tersebut dapat dilakukan penilaian dan keputusan yang tepat bagi para pemakainya.

Tujuan akuntansi :

- a) Mengalokasikan sumber - sumber daya langka masyarakat.
- b) Mengelola dan mengarahkan sumber - sumber daya dalam masyarakat.
- c) Melaporkan pertanggungjawaban sumber - sumber daya yang dikendalikan oleh individu maupun organisasi.

Berikut ini adalah gambar dari proses akuntansi.



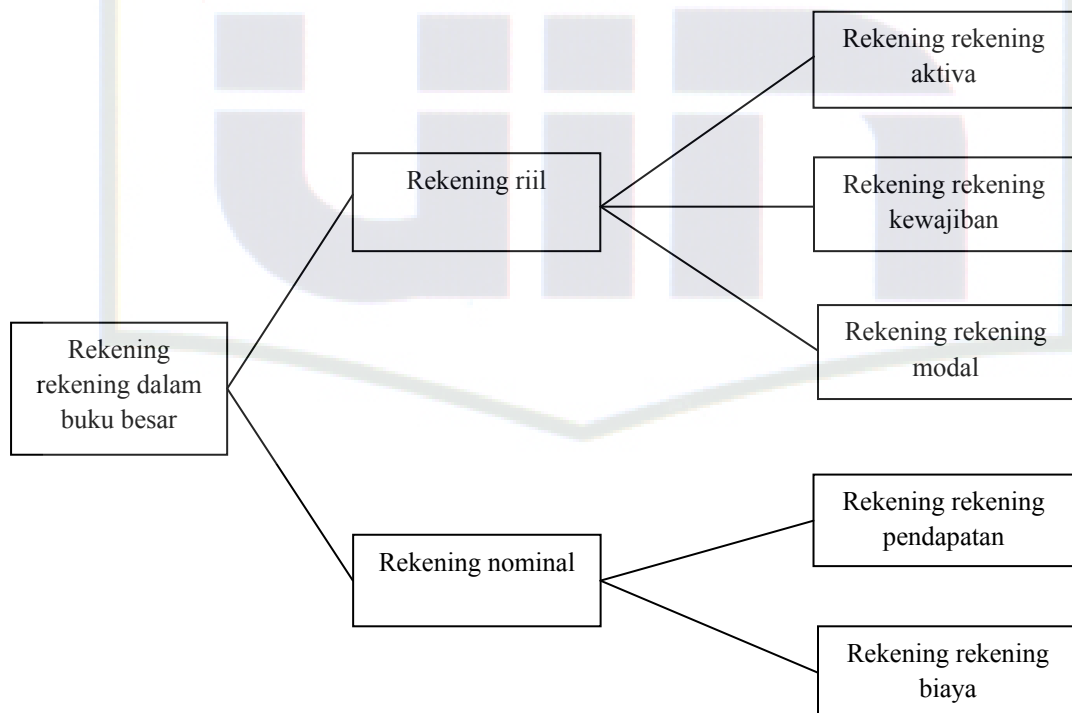
Gambar 2.1 Proses Akuntansi (Simamora, 2000 : 4)

Dalam perusahaan, pasti ada aktivitas – aktivitas bisnis yang dilakukan. Semua aktivitas yang ada kemudian dilakukan pencatatan. Data – data aktivitas tersebut akan dimasukkan ke dalam jurnal dengan menggunakan rekening. Data yang ada di dalam jurnal kemudian disimpan dan dilakukan pemrosesan, yaitu pembuatan buku besar dan pengubahan saldo rekening – rekening terkait. Setelah data tersebut disimpan, kemudian dilakukan pembuatan laporan keuangan. Laporan keuangan inilah yang akan diberikan kepada manajer atau para pengambil keputusan.

#### 2.1.2.1 Rekening

Rekening adalah suatu alat untuk mencatat transaksi – transaksi keuangan yang bersangkutan dengan aktiva,

kewajiban, modal, pendapatan, dan biaya. Tujuan pemakaian rekening adalah untuk mencatat data yang akan menjadi dasar penyusunan laporan – laporan keuangan. (Jusup, 2005 : 64). Semua rekening dapat diklasifikasikan menjadi dua kelompok, yaitu rekening neraca (aset, kewajiban, dan ekuitas pemegang saham) dan rekening laporan laba – rugi (pendapatan dan beban).



Gambar 2.2 Penggolongan Rekening. Sumber (Jusup, 2005 : 65)

Berikut adalah penjelasan dari rekening – rekening diatas :

1. Rekening neraca atau biasa disebut rekening riil yaitu rekening – rekening yang pada akhir periode akan dilaporkan di dalam neraca (Jusup, 2005 : 64). Yang termasuk rekening neraca adalah rekening aktiva,

rekening kewajiban (utang), dan rekening modal / ekuitas.

a. Aktiva

Aktiva (*asset*) adalah sumber daya yang dikuasai oleh perusahaan sebagai akibat dari peristiwa masa lalu dan darinya manfaat ekonomi di masa depan diharapkan akan diraih oleh perusahaan. Aktiva dapat mengambil bermacam – macam bentuk. Aktiva dapat memiliki bentuk fisik yang kasat mata seperti gedung, mesin produksi, atau persediaan barang dagangan. Namun di lain pihak, beberapa aktiva tidak berbentuk fisik atau tidak berwujud; namun dalam bentuk klaim legal ataupun hak. (Simamora, 2000 : 12).

b. Kewajiban

Kewajiban (*liability*) merupakan utang perusahaan masa kini yang timbul dari peristiwa di masa silam, penyelesaiannya diharapkan mengakibatkan arus keluar dari sumber daya perusahaan yang mengandung manfaat ekonomi. (Simamora, 2000 : 13).

### c. Ekuitas

Ekuitas (*equity*) adalah hak residual atas aktiva perusahaan setelah dikurangi semua kewajiban. (Simamora, 2000 : 14). Modal dicantumkan dalam neraca di bawah kewajiban. Modal pada hakikatnya merupakan hak milik perusahaan atas kekayaan (aktiva) perusahaan.

Besarnya hak pemilik sama dengan aktiva bersih perusahaan, yaitu selisih antara aktiva dan kewajiban. (Jusup, 2005 : 23).

2. Rekening laba rugi atau biasa disebut juga rekening nominal, yaitu rekening yang pada akhir periode akan dilaporkan dalam laporan laba rugi. Rekening ini meliputi rekening pendapatan dan rekening biaya. (Jusup, 2005 : 65).

#### a. Pendapatan

Pendapatan (*revenue*) adalah kenaikan aktiva perusahaan atau penurunan kewajiban perusahaan (atau kombinasi dari keduanya) selama periode tertentu yang berasal dari pengiriman barang – barang, penyerahan jasa, atau kegiatan – kegiatan lainnya yang merupakan kegiatan sentral perusahaan. (Simamora, 2000 : 24).

### b. Beban

Beban (*expense*) adalah penurunan manfaat ekonomi selama suatu periode akuntansi dalam bentuk arus keluar atau berkurangnya aktiva atau terjadinya kewajiban yang mengakibatkan penurunan ekuitas yang tidak menyangkut pembagian kepada penanam modal. (Simamora, 2000 : 25).

Tabel berikut memberikan karakteristik rekening :

Tabel 2.1 Tabel karakteristik rekening

(Sumber : Warren, *et al*, 2006 : 71)

	Kenaikan (Saldo Normal)	Penurunan
Akun Neraca		
Aktiva	Debit	Kredit
Kewajiban	Kredit	Debit
Ekuitas Pemilik		
Modal	Kredit	Debit
Penarikan	Debit	Kredit
Akun Laporan Laba Rugi		
Pendapatan	Kredit	Debit
Beban	Debit	Kredit

### 2.1.2.2 Jurnal

Jurnal adalah alat untuk mencatat transaksi perusahaan yang dilakukan secara kronologis (berdasarkan urutan waktu terjadinya) dengan menunjukkan rekening yang harus didebit dan dikredit beserta jumlah rupiahnya masing – masing. Setiap transaksi yang terjadi di dalam perusahaan, sebelum dibukukan ke buku besar, harus dicatat dahulu dalam jurnal. (Jusup, 2005 : 120). Penjurnalan (*journalizing*) adalah pemasukan data transaksi ke dalam jurnal. Setiap entri jurnal memberikan informasi berikut tentang transaksi – transaksi :

- a. Tanggal transaksi bisnis.
- b. Nama rekening – rekening yang didebit dan dikredit
- c. Angka yang didebit dan dikredit ke setiap rekening.
- d. Penjelasan transaksi
- e. Referensi pembukuan. (post. ref. = *Posting Reference*) untuk transaksi.

Dalam penelitian ini, hanya digunakan empat jurnal, yaitu:

#### 1. Ayat Jurnal

Ayat jurnal (*journal entry*) adalah pencatatan transaksi bisnis dalam bentuk debit dan kredit dan

bisa saja mencakup pula penjelasan ringkas transaksi (Simamora, 2000 : 49). Ayat jurnal biasa digunakan dalam jurnal umum.

## 2. Entri Penyesuaian

Entri Penyesuaian adalah entri – entri jurnal yang dibuat pada akhir periode akuntansi atau pada

saat laporan keuangan akan disusun supaya

menghasilkan pengaitan yang tepat antara pendapatan dan beban. (Simamora, 2000 : 75).

Tujuan penyesuaian adalah (Jusup, 2005 : 180):

a. Agar setiap rekening riil, khususnya rekening – rekening aktiva dan rekening- rekening utang, menunjukkan jumlah yang sebenarnya pada akhir periode.

b. Agar setiap rekening nominal (rekening – rekening pendapatan dan rekening – rekening biaya) menunjukkan pendapatan dan biaya yang seharusnya diakui dalam suatu periode.

## 3. Entri Penutup

Jurnal yang dibuat untuk memindahkan saldo – saldo rekening sementara (rekening – rekening nominal dan rekening prive) disebut jurnal penutup (Jusup, 2005 : 276)



Tujuan dibuatnya jurnal penutup adalah (Jusup, 2005 : 277):

- a. Untuk menutup saldo yang terdapat dalam semua rekening sementara.
- b. Agar saldo rekening modal menunjukkan jumlah yang sesuai dengan keadaan pada akhir periode.

Entri penutup biasa digunakan untuk jurnal penutup.

#### 4. Jurnal Koreksi

Bila kesalahan menjurnal sudah terlanjur dibukukan ke buku besar, maka cara koreksinya harus dilakukan dengan membuat jurnal koreksi. (Jusup, 2005 : 145). Untuk menentukan jurnal koreksi, harus diketahui lebih dahulu dua hal berikut ini :

- a. Jurnal yang salah (yang telah dilakukan)
- b. Jurnal yang benar (yang seharusnya dilakukan)

#### 2.1.2.3 Buku Besar

Kumpulan rekening yang digunakan dalam pembukuan suatu perusahaan disebut buku besar atau *ledger*. Buku besar dapat berupa sebuah buku yang halaman – halamannya berfungsi sebagai rekening, atau bisa juga berupa kumpulan kartu. Dalam buku besar biasanya rekening

– rekening disusun dengan urutan tertentu, yaitu rekening – rekening untuk neraca disusun paling depan, dan sesudah itu barulah rekening – rekening yang dicantumkan dalam laporan laba – rugi. (Jusup, 2005 : 64).

#### **2.1.2.4 Laporan Keuangan**

Laporan keuangan (*financial statements*) adalah wahana pokok pelaporan informasi keuangan bertujuan umum kepada orang – orang di luar organisasi bisnis. Laporan keuangan lengkap biasanya mencakup neraca, laporan laba rugi, laporan perubahan posisi keuangan (yang dapat disajikan dalam berbagai cara seperti, misalnya, laporan arus kas atau laporan arus dana), catatan dan laporan lain serta materi penjelasan yang merupakan bagian integral dari laporan keuangan. (Simamora, 2000 : 21). Dalam penelitian ini, dibuat tiga laporan keuangan, yaitu :

##### **1. Laporan Laba Rugi**

Laporan laba rugi (*income statement*) – yang kadangkala disebut laporan pendapatan (*earning statement*) atau laporan operasi (*operations statement*) adalah laporan keuangan resmi yang merangkum kegiatan – kegiatan operasi (pendapatan dan beban) selama periode waktu tertentu, biasanya satu bulan atau satu tahun. (Simamora, 2000 : 22).

## 2. Laporan Ekuitas Pemilik

Laporan ekuitas pemilik (*statement of owner's equity*) menyajikan informasi ikhwal kejadian – kejadian yang menyebabkan perubahan ekuitas pemilik selama suatu periode tertentu. Laporan ini diawali dari ekuitas pemilik pada permulaan periode, kemudian melaporkan kejadian – kejadian yang menyebabkan kenaikan atau penurunan ekuitas pemilik, dan disudahi dengan ekuitas pemilik pada akhir periode. (Simamora, 2000 : 26).

## 3. Neraca

Neraca (*balance sheet*) adalah laporan keuangan yang memperlihatkan jumlah dan sifat aktiva, kewajiban, dan ekuitas pemilik usaha pada saat tertentu. (Simamora, 2000 : 26).

### 2.1.3 Incremental

Ada banyak situasi dimana syarat awal sebuah *software* telah didefinisikan secara jelas, namun lingkungan pengembangannya secara umum tidak selalu bisa linier. Sebagai tambahan, bisa saja ada kebutuhan mendesak untuk memberikan beberapa fungsi dari *software* kepada *user*, kemudian diperbaiki, dan dikembangkan dalam fungsi tersebut dalam pengembangan *software* selanjutnya. Dalam kasus ini, pengembang bisa memilih model proses yang didesain

untuk memproduksi *software* dalam penambahan – penambahan.  
(Pressman, 2010 : 41)

*Incremental* memiliki 5 tahapan, yaitu :

#### 1. *Communication*

Bertemu dengan *user* untuk mendefinisikan tujuan – tujuan umum untuk pembuatan perangkat lunak. Disini, komunikasi memegang peranan yang sangat penting, yaitu untuk mencatat syarat – syarat yang akan dipenuhi oleh pengembang untuk membuat suatu *software* yang diinginkan oleh *user*.

#### 2. *Planning*

Membuat jadwal – jadwal yang akan digunakan sebagai acuan untuk membuat perangkat lunak. Jadwal yang dibuat ini harus dipatuhi oleh pengembang, sehingga tepat waktu dalam memberikan *software* kepada *user*.

#### 3. *Modeling*

Membuat model, yakni dengan menganalisa dan mendesain *software* yang akan digunakan. Syarat – syarat *software* yang diinginkan oleh *user* akan dianalisa serta dibuat desainnya. Desain disini mencakup *Entity Relationship Diagram*, *Data Flow Diagram*, dan *Flowchart*. Tahap ini berfokus kepada pembuatan bagian *software* yang akan terlihat

oleh *user* contohnya (*layout* antarmuka pengguna atau format gambaran *output*).

#### 4. *Construction*

Membuat desain yang sudah direncanakan menjadi kode – kode program serta GUI yang nantinya akan menjadi perangkat lunak. Disini sudah memasuki tahap *coding*, yaitu membuat kode – kode program. *Software* yang akan jadi harus mengandung syarat – syarat *software* yang sudah disepakati oleh *user* dan pengembang. Setelah kode – kode program tersebut selesai, kemudian akan dilakukan *testing*.

#### 5. *Deployment*

*Software* yang sudah jadi kemudian dijalankan dan dievaluasi oleh *user*, yang memberikan umpan balik yang akan digunakan untuk pengembangan lebih lanjut. Disini peran *user* sangat penting, karena akan menentukan pengembangan *software* kearah selanjutnya.

### 2.1.4 *Entity Relationship Diagram (ERD)*

ERD adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. ERD berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur – struktur dan relationship data.

(Ladjamuddin,2005:142). ERD digunakan untuk membuat model data yang dipakai dalam aplikasi, serta relasi antar data tersebut.

### 2.1.5 *Data Flow Diagram (DFD)*

Diagram alir data merupakan model dari sistem untuk menggambarkan pembagian sistem ke modul yang lebih kecil. Salah satu keuntungan menggunakan diagram alir data adalah memudahkan pemakai atau *user* yang kurang menguasai bidang komputer untuk mengerti sistem yang akan dikerjakan (Ladjamuddin, 2005 : 64). DFD dipakai untuk membuat model proses yang berjalan pada aplikasi, serta proses yang mengakses data.

Untuk memudahkan dalam membaca DFD, maka pengembangan DFD disusun berdasarkan tingkatan atau level dari atas ke bawah. Adapun tingkatan dalam DFD :

#### 1. Diagram Konteks

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem ataupun output ke sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks harus ada satu proses.(Ladjamuddin,2005:64)

#### 2. Diagram Nol/Zero (Overview Diagram)

Diagram nol adalah yang menggambarkan proses dan data flow diagram. Diagram nol memberikan pandangan secara menyeluruh mengenai sistem yang ditangani, mengenai tentang fungsi – fungsi utama atau proses yang ada, aliran data, dan *external entity* (Ladjamuddin, 2005:64)

### 3. Diagram Rinci (Level Diagram )

Diagram rinci adalah diagram yang menguraikan proses apa yang ada dalam diagram zero atau diagram level diatasnya. (Ladjamuddin,2005:64).

Untuk memudahkan penggambaran DFD, maka penulis menggunakan Kamus Data. Kamus data sering disebut juga dengan sistem *data dictionary* adalah katalog fakta tentang data dan kebutuhan – kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. Pada tahap analisis, kamus data digunakan sebagai alat komunikasi antara analis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem.

#### 2.1.6 *Flowchart*

*Flowchart* adalah bagan – bagan yang mempunyai arus yang menggambarkan langkah – langkah suatu masalah. *Flowchart* merupakan cara penyajian dari suatu algoritma (Ladjamuddin,

2005:263). *Flowchart* menggambarkan logika – logika yang berjalan pada proses di dalam aplikasi. Ada 2 macam *flowchart* yang menggambarkan proses dengan komputer :

#### 1. Sistem *flowchart*

Bagan yang memperlihatkan urutan proses dalam sistem dengan menunjukkan alat media input, output, serta jenis media penyimpanan dalam proses pengolahan data.

#### 2. Program *flowchart*

Bagan yang memperlihatkan instruksi yang digambarkan dengan simbol tertentu untuk memecahkan masalah dalam suatu program. (Ladjamuddin, 2005: 263)

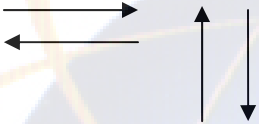
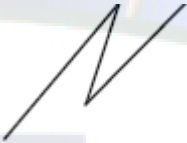

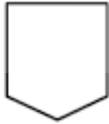
Dengan menggunakan *flowchart* maka kerja dari suatu proses dapat dipantau untuk pengembangan atau perbaikan.

*Flowchart* disusun dengan simbol. Simbol ini dipakai sebagai alat bantu menggambarkan proses di dalam program. Simbol-simbol yang digunakan dapat dibagi menjadi 3 (tiga) kelompok, yaitu :

#### 1. *Flow Direction Symbols* (Simbol Penghubung atau Alur)

Tabel 2.2 Simbol Penghubung (Ladjamudin, 2005 : 266)

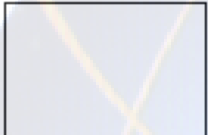


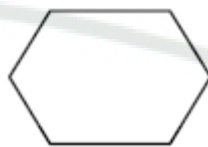

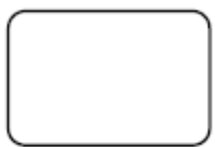



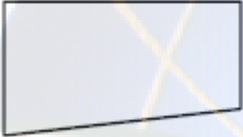
No	Symbol	Nama dan Keterangan
1.		<b>Symbol arus /flow</b> Untuk menyatakan jalannya arus suatu proses
2.		<b>Simbol communication link</b> Untuk menyatakan bahwa ada suatu transisi suatu data / informasi dari satu lokasi ke lokasi lainnya
3.		<b>Simbol connector</b> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman / lembar yang sama
4.		<b>Simbol offline connector</b> Untuk menyatakan sambungan dari satu proses ke proses lainnya dalam halaman / lembar yang berbeda

## 2. Processing Symbols (Simbol Proses)

Simbol yang menunjukkan jenis operasi pengolahan dalam suatu proses / prosedur, simbol-simbol tersebut adalah :

Tabel 2.3 Simbol Proses *Flowchart* (Ladjamudin, 2005 : 267).

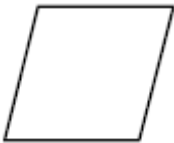

No	Symbol	Nama dan Keterangan
1.		<b>Simbol Offline Connector</b> Untuk menyatakan sambungan dari satu proses ke proses lainnya kedalam halaman/lembar yang berbeda
2.		<b>Simbol Manual</b> Untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer( <i>manual</i> )
3.		<b>Simbol Decision/logika</b> Untuk menunjukan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, ya/tidak
4.		<b>Simbol Predefined Proses</b> Untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
5.		<b>Simbol Terminal</b> Untuk menyatakan permulaan atau akhir suatu program
6.		<b>Simbol Keying Operation</b> Untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard





7.		<b>Simbol off-line storage</b> Untuk menunjukkan bahwa data dalam symbol ini akan disimpan ke suatu media tertentu
8.		<b>Simbol Manual Input</b> Untuk memasukkan data secara manual dengan menggunakan online keyboard

### 3. Input-output Symbols

Simbol yang menunjukkan jenis peralatan yang digunakan sebagai media input atau output, simbol-simbol tersebut adalah :

Tabel 2.4 Simbol input output *flowchart* (Ladjamudin, 2005 : 268).

No	Symbol	Nama dan Keterangan
1.		<b>Simbol Input-output</b> Untuk menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya
2.		<b>Simbol Punched Card</b> Untuk menyatakan input berasal dari kartu atau output ditulis ke kartu

3.		<b>Simbol Magnetic-tape unit</b> Untuk menyatakan input berasal dari pita magnetik atau output disimpan ke pita
4.		<b>Simbol Disk Storage</b> Untuk menyatakan input berasal dari disk atau output disimpan ke disk
5.		<b>Simbol document</b> Untuk mencetak laporan ke printer
6.		<b>Simbol display</b> Untuk menyatakan peralatan output yang digunakan berupa layar (video, komputer)

### 2.1.7 Studi Sejenis

Berdasarkan pengamatan penulis, maka terdapat beberapa studi sejenis yang berhubungan dengan aplikasi akuntansi, yaitu :

1. Literatur yang digunakan sebagai pembanding dalam penelitian ini berjudul “Aplikasi Laporan Keuangan Berbasis Web Untuk Kelurahan Dukuh”. Penulis Wina Setiawati sebagai skripsi untuk lulus dari Universitas Gunadarma tahun 2008. Literatur yang digunakan sebagai pembanding ini merupakan aplikasi keuangan

yang digunakan untuk sebuah kantor pemerintahan, yaitu untuk kelurahan Dukuh. Aplikasi dibuat untuk membuat laporan keuangan secara terkomputerisasi, memudahkan pengerjaannya secara lebih efektif dan efisien. Penulis melakukan analisis dari jalannya pembuatan laporan keuangan yang sedang berjalan. Aplikasi dibuat berbasis *web* agar mudah diakses dan dapat digunakan bersama – sama. Laporan keuangan yang dibuat adalah laporan pemasukan dan laporan pengeluaran. Laporan pemasukan terdiri dari transaksi – transaksi terkait akun pendapatan. Sedangkan laporan pengeluaran terdiri dari transaksi – transaksi terkait akun pengeluaran.

2. Penelitian sejenis yang berjudul “Sistem Informasi Akuntansi Pengeluaran Kas (Studi Kasus : BNI Syariah Fatmawati Jakarta Selatan)”. Penelitian dilakukan oleh Tri Rizki Amalia pada tahun 2010. Penelitian ini bertujuan untuk membuat sistem informasi akuntansi yang dapat membuat jurnal, pencatatan pengeluaran kas, laporan pengeluaran kas, dan grafik pengeluaran kas. Dalam pembuatan sistem, penulis menggunakan metode *waterfall* dan *unified modelling language*. Penulis menggunakan bahasa pemrograman PHP dan *database* MySQL. Laporan pengeluaran kas dibuat dengan akurat, sehingga tidak ada data pengeluaran kas yang salah. Dengan laporan pengeluaran kas, diharapkan *mnajer* dapat mengambil keputusan secara tepat. Kekurangan dari sistem

ini adalah laporan yang dibuat hanya dalam bentuk HTML. Kelebihan dari sistem ini yaitu membuat grafik pengeluaran kas. Grafik akan menampilkan akun – akun apa saja yang memiliki pengeluaran. Dari grafik ini, akan lebih mudah untuk melihat akun – akun apa saja yang memiliki pengeluaran, dari pengeluaran yang terbesar sampai pengeluaran terkecil.

#### 2.1.8 IIUC

*International Islamic University College (IIUC)*, yang dikenal sebagai STMIK Islam Internasional Jakarta adalah lembaga pendidikan swasta yang terdaftar di Kopertis III, Jakarta dan didukung sepenuhnya oleh *International Islamic University Malaysia (IIUM)* atau Universitas Islam Antar Bangsa Malaysia.

Mahasiswa diberikan nilai-nilai luhur pengetahuan dan profesionalisme, dilengkapi dengan pengembangan intelektual dari segi fisik dan emosional. IIUC memberikan pendidikan dan sumber daya manusia yang terampil untuk kemajuan Islam, masyarakat dan negara. IIUC melayani dan memberikan kepedulian untuk mendidik, melatih dan menanamkan nilai-nilai luhur untuk laki-laki dan perempuan muda yang bercita-cita, yang ingin, yang berharap untuk menjadi generasi berikutnya yang profesional.

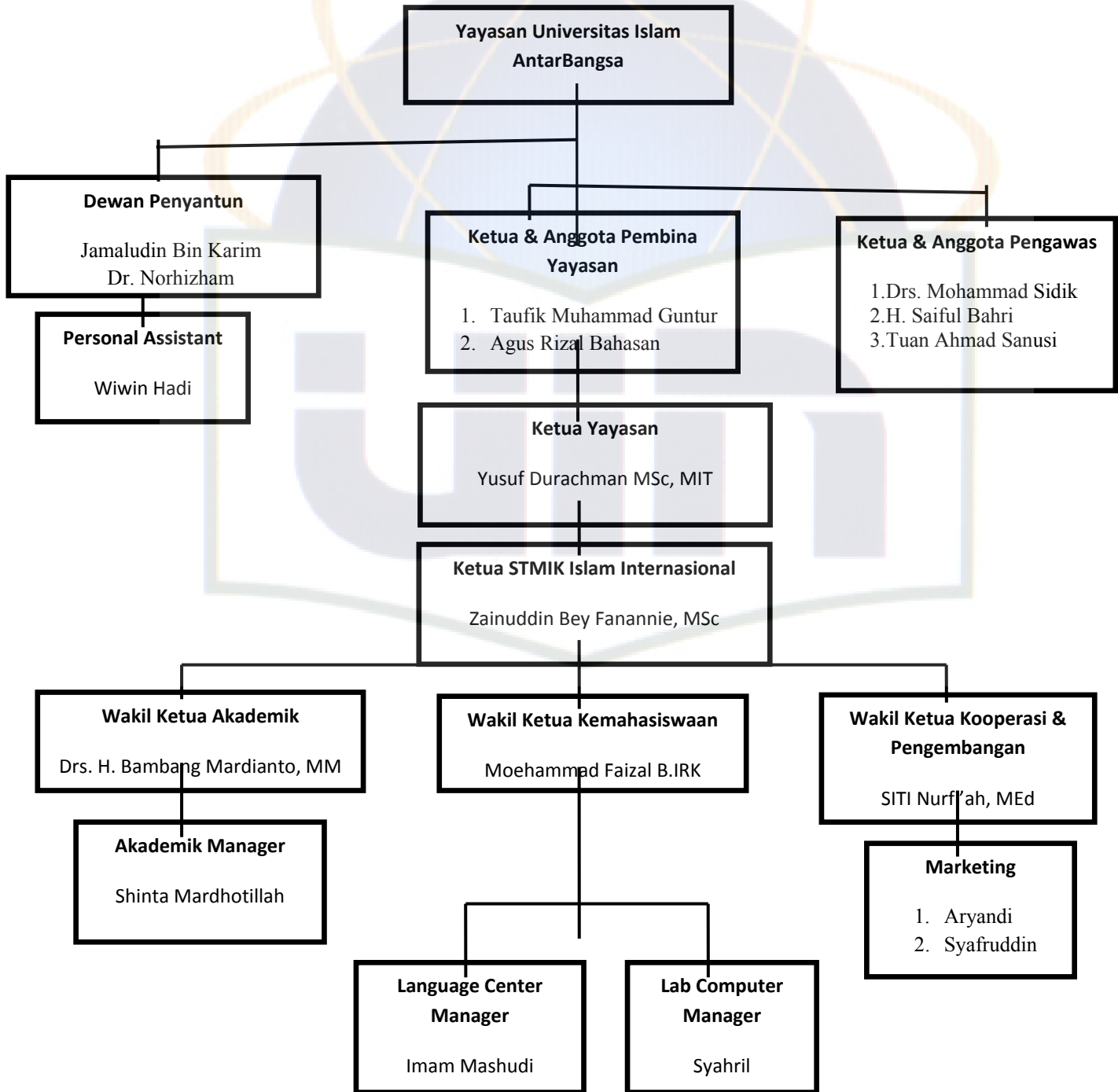
#### **2.1.8.1 Visi**

Menjadi institusi yang terbaik untuk pengembangan profesional Islam global.

#### **2.1.8.2 Misi**

1. Untuk mendirikan sistem pembelajaran yang fleksibel dengan memberikan sesuatu yang berkualitas tinggi dan program-program inovatif yang mencerminkan permintaan saat ini.
2. Untuk melanjutkan dukungan akademis, karir dan pengembangan profesional staf dan mahasiswa.
3. Untuk menyediakan lingkungan belajar yang kondusif untuk meningkatkan pertumbuhan bagi mahasiswa dan staf.
4. Untuk mendapatkan kepercayaan dari para pemangku kepentingan melalui penanaman nilai-nilai Islam inti dan untuk menyeimbangkan keuntungan dan kewajiban sosial tanpa mengorbankan nilai-nilai akademis.
5. Untuk mempertahankan standar kualitas pendidikan yang terbaik.

### 2.1.8.3 Struktur Organisasi



Gambar 2.3 Struktur Organisasi IIUC



## 2.2 Instalasi Perangkat Lunak Pendukung

### 2.2.1 Web Browser

Dikenal juga dengan istilah *browser*, atau peselancar, atau *internet browser*. Adalah suatu program komputer yang menyediakan fasilitas untuk membaca halaman *web* di suatu komputer. Dua program *web browser* yang cukup populer saat ini adalah Microsoft Internet Explorer dan Netscape Navigator. Program *browser* pertama kali adalah Mosaic, yang merupakan suatu *text browser*, yang sekarang telah berkembang ke dalam bentuk *multimedia*. (Febrian, 2007 : 449)

### 2.2.2 PHP

PHP adalah suatu bahasa pemrograman *open source* yang digunakan secara luas terutama untuk pengembangan *web* dan dapat disimpan dalam bentuk HTML.

Untuk menghasilkan sebuah HTML, script yang ditulis menggunakan PHP mempunyai perintah yang lebih singkat dibandingkan bahasa pemrograman lain seperti Perl atau C. *User* hanya perlu memasukkan kode untuk melakukan sesuatu (misalnya menulis suatu kalimat) di antara tag awal dan tag akhir PHP. (Komputer, 2006 : 11)

PHP singkatan dari PHP Hypertext Preprocessor yang digunakan sebagai bahasa *script server-side* dalam pengembangan *web* disisipkan pada dokumen HTML. Penggunaan PHP

memungkinkan *web* dapat dibuat dinamis sehingga *maintenance* situs *web* tersebut menjadi lebih mudah dan efisien. PHP merupakan *software open-source* yang disebar dan dilisensikan secara gratis serta dapat di-*download* secara bebas dari situs resminya, <http://www.php.net>. PHP ditulis menggunakan bahasa C. (Peranginangin, 2006 : 2)

PHP dibuat dan diperkenalkan pertama kali oleh Rasmus Lerdorf pada tahun 1995 menggunakan nama PHP/FI. PHP/FI merupakan akronim dari Personal Home Page /Forms Interpreter. Pada tahun 1997, dikeluarkan PHP/FI versi 2.0. Pada versi inilah pemrogram dapat menempelkan kode terstruktur di dalam tag HTML. Yang menarik, kode PHP juga bisa berkomunikasi dengan *database* dan melakukan perhitungan-perhitungan yang kompleks.

### 2.2.3 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*)

MySQL memiliki dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa digunakan adalah MySQL FreeSoftware yang berada di bawah lisensi GNU/GPL (*General Public License*). Keterangan lengkap dapat dilihat pada <http://www.gnu.org/licenses/>. Selain konsumen juga dapat memiliki

produk MySQL yang sifatnya komersial, biasa disebut dengan MySQL AB. (Nugroho, 2005 : 1)

SQL adalah sebuah bahasa permintaan *database* yang terstruktur. Bahasa SQL dibuat sebagai bahasa yang dapat merelasikan beberapa tabel dalam *database* maupun merelasikan antar *database*. Bahasa SQL ditulis langsung dalam sebuah program *database server* yang melayani permintaan *user*. SQL adalah bahasa permintaan yang digunakan dalam MySQL (Nugroho, 2005 : 5)

MySQL merupakan *database* yang sangat terkenal di kalangan *programmer web* karena MySQL merupakan *database* yang stabil dalam penyimpanan data. Begitu banyak programmer, baik *programmer web* atau *programmer aplikasi database* menggunakan MySQL sebagai media penyimpanan data. (Ibrahim, 2008 : 67)

#### 2.2.4 Apache

*Server HTTP Apache* atau *Server Web/WWW Apache* adalah *web server* yang dapat dijalankan di banyak sistem operasi (Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan menfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas *web/www* ini menggunakan HTTP.

*Web server* adalah suatu program (dan juga mesin yang menjalankan program) yang mengerti protokol HTTP dan dapat

menanggapi permintaan – permintaan dari *web browser* yang menggunakan protokol tersebut. (Febrian, 2007 : 450)

Beberapa dukungan Apache :

1. Kontrol Akses.

Kontrol ini dapat dijalankan berdasarkan nama host atau nomor IP

2. CGI (*Common Gateway Interface*),

3. PHP (*Personal Home Page/PHP Hypertext Processor*);

Apache mendukung PHP dengan menempatkannya sebagai salah satu modulnya (*mod\_php*). Hal ini membuat kinerja PHP menjadi lebih baik

4. SSI (*Server Side Includes*)

5. *Virtual host*,

## 2.2.5 XAMPP

XAMPP merupakan sebuah paket instalasi untuk PHP, Apache, dan MySQL. Dengan menggunakan XAMPP, *user* tidak perlu lagi repot menginstall ketiga *software* itu secara terpisah. XAMPP sama seperti PHP yang sudah sering digunakan oleh *programmer web* dalam membuat aplikasi *web*. (Ibrahim, 2008 : 3)

### 2.2.5.1 Detail Paket pada Xampp

Berikut detail paket instalasi yang disertakan pada XAMPP 1.7.2 yang digunakan pada skripsi ini:

1. Apache 2.2.14 (IPv6 enabled) + OpenSSL 0.9.8l

2. MySQL 5.1.41 + PBXT engine
3. PHP 5.3.1
4. phpMyAdmin 3.2.4
5. Perl 5.10.1
6. FileZilla FTP *Server* 0.9.33
7. Mercury Mail Transport System 4.72

### 2.2.6 HTML

HTML adalah singkatan dari *HyperText Markup Language*. Pemrograman HTML sangat mudah dilakukan tidak sesulit pemrograman murni seperti C++, Delphi atau Java. (Husni, 2007 : 1)

Setiap halaman *web* sebaiknya mempunyai minimal 4 elemen utama, yaitu (Husni, 2007 : 4) :

#### 1. Elemen <html>

*Tag* <html> memulai dan mengakhiri setiap halaman *web*.

Peran utama <html> adalah membungkus semua kode HTML. *Tag* <html> diletakkan pada bagian paling atas *file* halaman *web*, mempunyai penutup </html> yang diletakkan di bagian paling bawah *file*. *Tag* ini tidak boleh dilupakan saat menulis kode HTML.

#### 2. Elemen <head>

Secara garis besar, halaman *web* (HTML) terdiri dari dua bagian, yaitu bagian *head* yang merupakan *header* halaman dan *body* sebagai isinya. Elemen <head> diletakkan antara elemen

`<html>` dan isi dari halaman *web*. Elemen ini berfungsi dibelakang layar, apa yang dikerjakannya tidak terlihat pada halaman *web*.

### 3. Elemen `<title>`

*Tag* ini harus diletakkan di dalam elemen `<head>` untuk memberikan judul dari halaman *web*. Teks yang dijadikan judul ditulis di antara `<title>` dan `</title>`.

### 4. Elemen `<body>`

Elemen `<body>` adalah tempat dimana semua isi dari halaman diletakkan. Isi halaman dapat berupa teks, gambar, suara, tabel, bahkan *video*.

## 2.2.7 jQuery

jQuery adalah Javascript Library, kumpulan kode / fungsi Javascript siap pakai, sehingga mempermudah dan mempercepat *user* dalam membuat kode Javascript (Hakim, 2010 : 5). Beberapa kemampuan yang dimiliki oleh jQuery adalah :

1. Mempermudah akses dan manipulasi elemen tertentu pada dokumen.
2. Mempermudah modifikasi / perubahan tampilan halaman *web*.
3. Mempersingkat *Ajax (Asynchronous Javascript and XML)*
4. Memiliki API (*Application Programming Interface*)
5. Mampu merespon interaksi user dengan halaman *web* dengan lebih cepat.

6. Menyediakan fasilitas untuk membuat animasi sekilas *Flash* dengan mudah.

#### 2.2.8 FusionCharts Free

FusionCharts Free adalah komponen pembuat grafik berbasis Flash yang bebas dan *open-source* yang bisa digunakan untuk membuat grafik. Dibat oleh Macromedia Flash MX, FusionCharts bisa digunakan dengan bahasa pemrograman *web* apapun, seperti PHP, ASP, .NET, JSP, ColdFusion, JavaScript, Ruby on Rails, dan sebagainya untuk memberikan grafik yang interaktif. Menggunakan XML sebagai antarmuka data, FusionCharts membuat fluida yang cantik dari *Flash* untuk membuat grafik yang interaktif, menarik penglihatan dan kompak.

#### 2.2.9 CSS

CSS (*Cascading Style Sheets*) adalah salah satu bahasa pemrograman desain *web* (*style sheet language*) yang mengontrol format tampilan sebuah halaman *web* yang ditulis dengan menggunakan bahasa penanda (*markup language*).

CSS dibuat untuk memisahkan konten utama (biasanya dibuat dengan menggunakan bahasa HTML atau sejenisnya) dengan tampilan dokumen yang meliputi *layout*, warna, dan *font*. Pemisahan ini dapat meningkatkan daya akses konten pada *web*, menyediakan lebih banyak fleksibilitas dan kontrol dalam spesifikasi dari sebuah karakteristik dari sebuah tampilan, memungkinkan untuk membagi

banyak halaman untuk sebuah *formatting* dan mengurangi kerumitan dalam penulisan kode dan struktur dari konten, contohnya teknik *tableless* pada *layout* desain *web* (*layout* tanpa tabel).

Tujuan utama CSS diciptakan adalah untuk membedakan konten dari dokumen dan tampilan dari dokumen, dengan itu pembuatan ataupun pemrograman ulang *web* akan lebih mudah dilakukan. Hal yang termasuk dalam desain *web* diantaranya adalah warna, ukuran, dan *formatting*. Dengan adanya CSS, konten dan desain *web* akan mudah dibedakan, jadi memungkinkan untuk melakukan pengulangan pada tampilan – tampilan tertentu dalam suatu halaman *web*, sehingga akan memudahkan dalam membuat halaman *web* yang banyak, sehingga membuat waktu dalam membuat *web* jauh lebih cepat. (Wismakarma, 2010 : 2).



## **BAB III**

### **METODOLOGI PENELITIAN**

Bab tiga akan menjelaskan tentang metodologi penelitian, yaitu urutan – urutan dalam melakukan penelitian ini. Pertama, bab tiga akan membahas metode pengumpulan data. Metode terdiri dari tiga metode, yaitu observasi, studi pustaka, dan wawancara. Disini akan dijelaskan sasaran – sasaran apa yang harus dipenuhi dalam setiap tahapan.

Bagian kedua dari bab tiga yaitu metode pengembangan sistem. Metode yang dipakai disini adalah metode *incremental*. Dalam bagian kedua ini juga terdapat kerangka berpikir, yaitu urutan – urutan pekerjaan yang dilakukan oleh penulis. Disini akan dijelaskan target – target yang harus dipenuhi dalam setiap tahapan metode *incremental*.

#### **3.1. Metode Pengumpulan Data**

Dalam melakukan pengumpulan data, penulis menggunakan tiga metode, yaitu :

##### **3.1.1 Studi Lapangan atau Observasi**

Observasi merupakan teknik penelusuran fakta dimana analisis sistem berpartisipasi atau melihat seseorang melakukan aktivitas untuk mempelajari sistem (Whitten, 2004 : 245). Observasi yang dilakukan, adalah melakukan tinjauan langsung kelapangan guna

mendapatkan informasi dan fakta pendukung dalam penelitian.

Sasaran dari observasi adalah :

1. Mengetahui sistem keuangan yang telah berjalan pada IIUC.
2. Mengetahui proses akuntansi yang ada di IIUC.

Penelitian dilakukan di IIUC, Jl. Melawai Raya no. 8C Kebayoran Baru, Jakarta Selatan dari tanggal 3 Juli 2010 sampai dengan 30 Oktober 2010.

### 3.1.2 Studi Pustaka

Studi Pustaka, yakni mengumpulkan data primer dan referensi melalui literatur, buku, artikel maupun secara online menggunakan media *internet* untuk mendapatkan referensi yang berhubungan dengan penulisan skripsi ini. Sasaran dari tahapan ini adalah :

1. Mencari literatur – literatur yang berhubungan dengan akuntansi, PHP, MySQL, dan jQuery.
2. Mencari studi sejenis yang berhubungan dengan aplikasi akuntansi.

### 3.1.3 Wawancara

Wawancara merupakan teknik penelusuran fakta dimana analisis sistem mengumpulkan informasi dari individu - individu melalui interaksi *face to face* (Whitten, 2004 : 250).

Pada tahap ini dilakukan wawancara dengan tujuan untuk mengumpulkan data, dan data yang diperoleh adalah melalui tanya

jawab dengan Ibu Shinta Mardhatillah, selaku Manajer Akademik.

Sasaran dari tahapan ini adalah :

1. Mendapatkan informasi tentang sistem keuangan di IIUC.
2. Pemanfaatan teknologi informasi pada bagian keuangan IIUC.
3. Rencana penggunaan aplikasi berbasis *web* untuk menunjang keuangan di IIUC.

Hasil dari wawancara dapat dilihat pada bagian Lampiran. Dari hasil wawancara dan observasi, maka bagian keuangan IIUC membutuhkan :

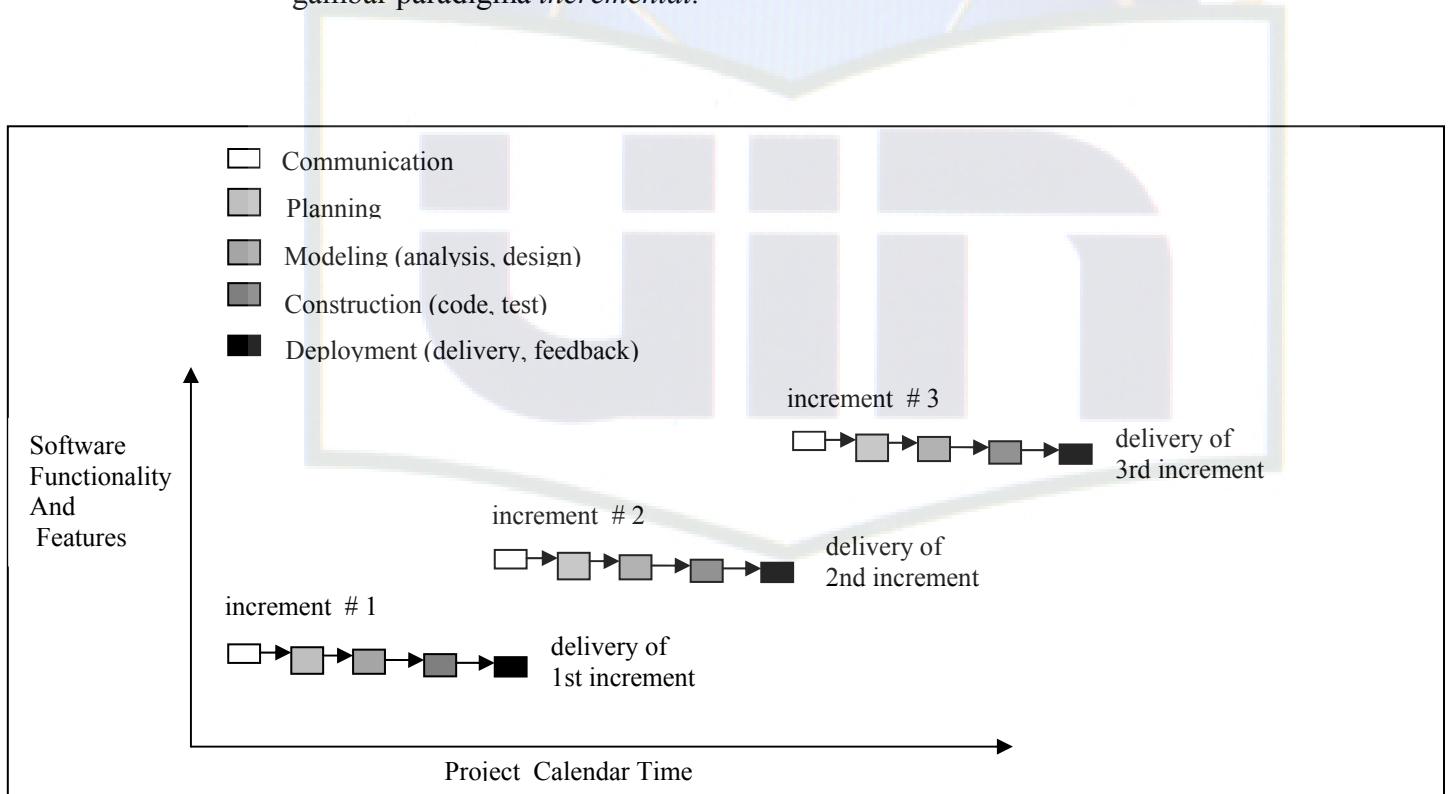
1. Aplikasi akuntansi berbasis *web*.
2. Pembuatan laporan keuangan secara otomatis serta dapat diekspor kedalam Microsoft Excel.
3. Penginformasian transaksi – transaksi yang ada.

### 3.2 Metode Pengembangan Sistem

Dalam penelitian ini, penulis menggunakan metode penelitian *incremental*. Ada banyak situasi dimana syarat awal sebuah *software* telah didefinisikan secara jelas, namun lingkungan pengembangannya secara umum tidak selalu bisa linier. Sebagai tambahan, bisa saja ada kebutuhan mendesak untuk memberikan beberapa fungsi dari *software* kepada *user*, kemudian diperbaiki, dan dikembangkan dalam fungsi tersebut dalam pengembangan *software* selanjutnya. Dalam kasus ini, pengembang bisa

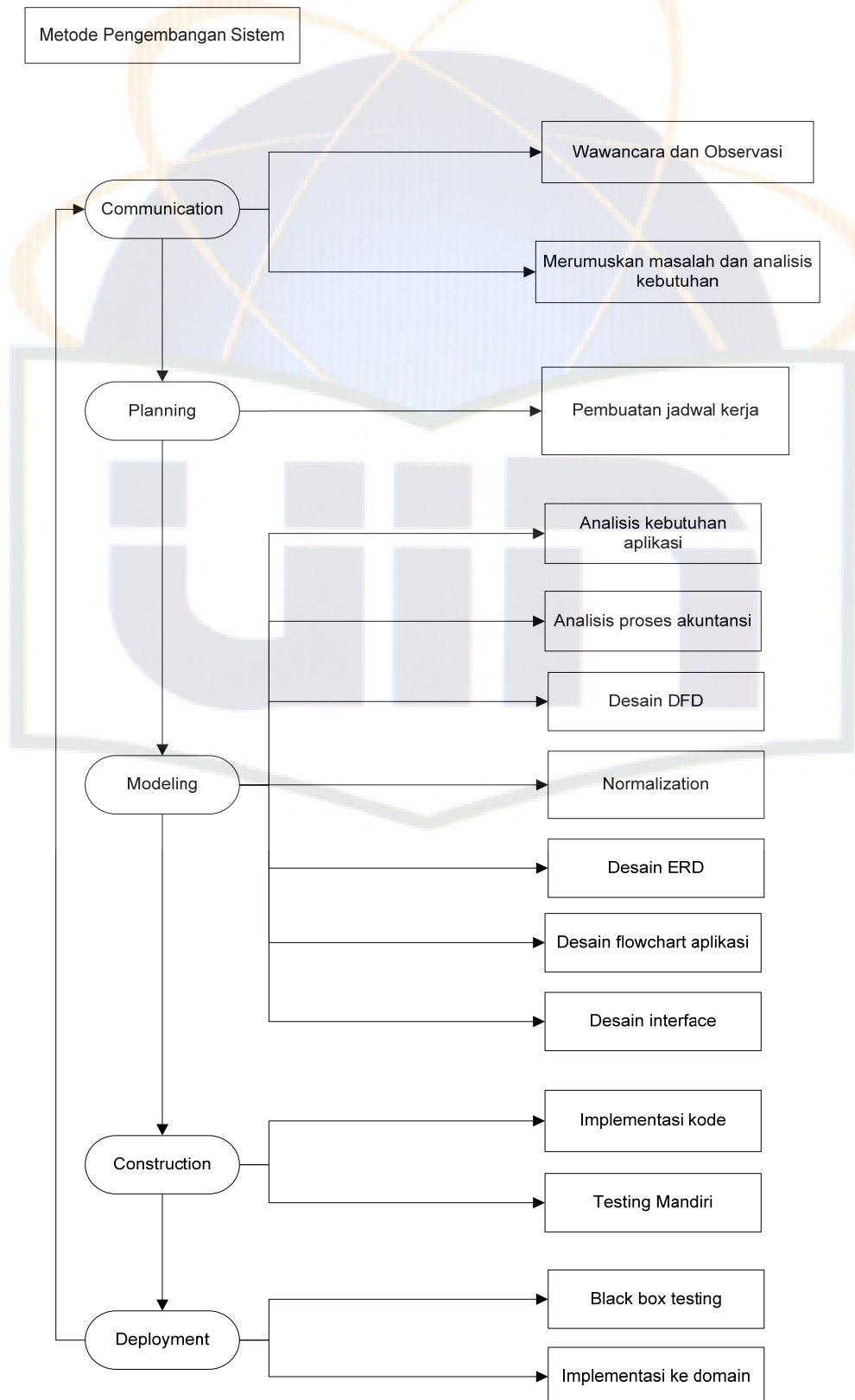
memilih model proses yang didesain untuk memproduksi *software* dalam penambahan – penambahan. (Pressman, 2010 : 41)

Penulis memilih metode ini karena masalah yang dihadapi sudah dimengerti dan ditetapkan dari awal, namun dalam prosesnya terdapat penambahan – penambahan dalam penyempurnaan aplikasi. Berikut adalah gambar paradigma *incremental*.



Gambar 3.1 Paradigma *incremental* (Sumber : Pressman, 2010 : 39)

Untuk lebih jelasnya, metode tersebut akan diimplementasikan ke dalam kerangka berpikir berikut.



Gambar 3.2 Kerangka Berpikir

Berikut adalah penjelasan dari kerangka berpikir ini :

### 3.2.1 *Communication*

Tahap pertama adalah komunikasi, dimana ini sangat penting untuk mendapatkan syarat –syarat kebutuhan dari *user*. Target dari tahapan ini adalah mendapatkan kebutuhan dari *user* yaitu aplikasi akuntansi berbasis *web*, dimana didalam aplikasi tersebut dapat membuat laporan keuangan dan penginformasian transaksi – transaksi yang ada.

### 3.2.2 *Planning*

Disini dibuat jadwal – jadwal perencanaan untuk membuat aplikasi tersebut. Target dari tahapan ini adalah membuat penjadwalan dari tahap *planning* , konstruksi, dan sampai tahap *deployment*.

### 3.2.3 *Modeling*

Tahap ini bertujuan untuk mendesain aplikasi yang akan dibuat. Setelah informasi keuangan diformulasikan secara lengkap, kemudian diimplementasikan dengan membuat perancangan aplikasi yang akan dibangun. Target dari tahapan ini, yakni:

1. Menganalisis kebutuhan aplikasi yang akan dibuat.
2. Menganalisis tahapan – tahapan proses akuntansi di IIUC.
3. Merancang *normalization*, *entity relationship diagram*, *data flow diagram*, *kamus data*, dan *flowchart*.
4. Merancang *user interface*

### 3.2.4 *Construction*

Target dari tahap konstruksi adalah :

1. Mengimplementasikan kode – kode program.
2. Menguji aplikasi ini secara mandiri.

### 3.2.5 *Deployment*

Pada tahap ini dilakukan pengujian kepada *user*. Target dari tahapan ini adalah :

1. Bersama – sama dengan *user* untuk melakukan *blackbox testing*. *Black box testing* dilakukan dengan Ibu Shinta Mardhatillah. Hasil pengujian dapat di lihat pada bagian lampiran.
2. Meng-*upload* aplikasi yang telah dites kedalam *hosting*.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Bab empat ini akan menjelaskan tentang pembahasan dari penelitian ini secara detail. Bab ini terdiri dari 5 subbab, yaitu :

1. *Communication*
2. *Planning*
3. *Modeling*
4. *Construction*
5. *Deployment*

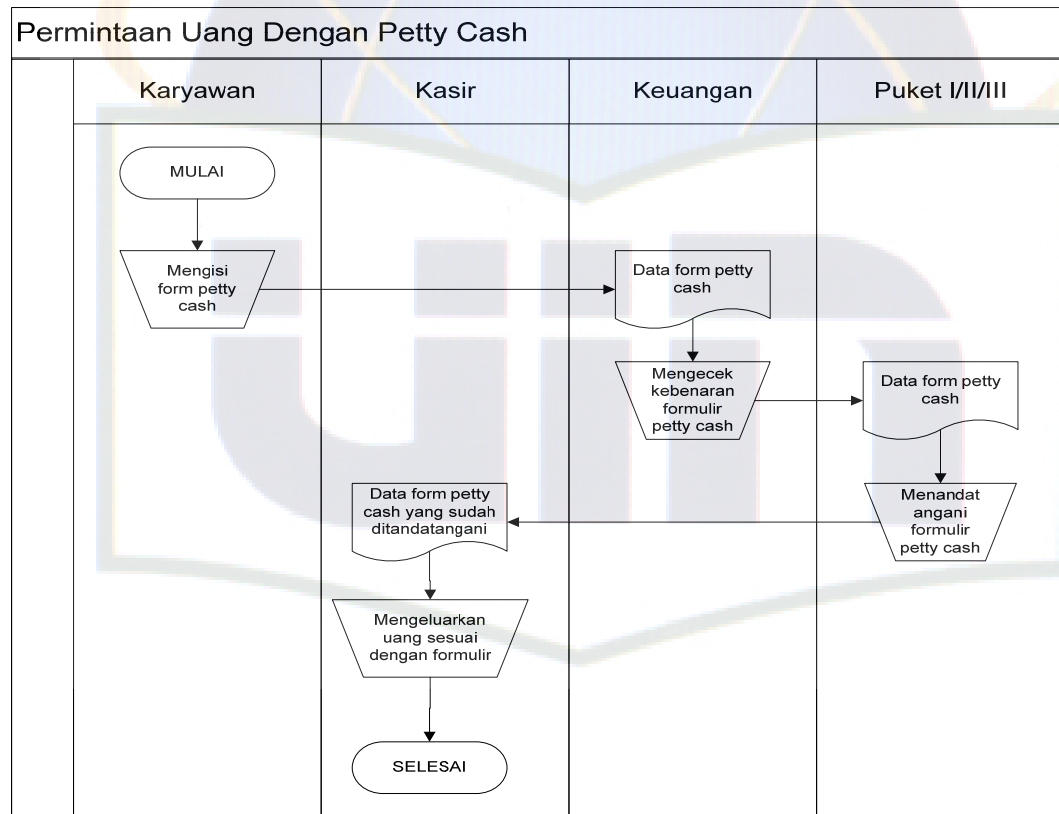
#### **4.1. *Communication***

Melalui wawancara dan observasi yang dilakukan, diketahui bahwa sistem keuangan di IIUC bersifat *manual*, yaitu dengan menggunakan Microsoft Excel. Maka disimpulkan permasalahan yang didapat adalah pembuatan aplikasi akuntansi berbasis *web*. Berangkat dari hasil observasi yang dilakukan kemudian dilakukan pengkajian dan pembatasan masalah yang akan diimplementasikan ke dalam aplikasi akuntansi ini.

Sistem yang telah berjalan di IIUC adalah sistem keuangan secara *manual*, yaitu dengan menggunakan Microsoft Excel. Pertama akan dijelaskan permintaan uang dengan *Petty cash* atau kas kecil. Permintaan



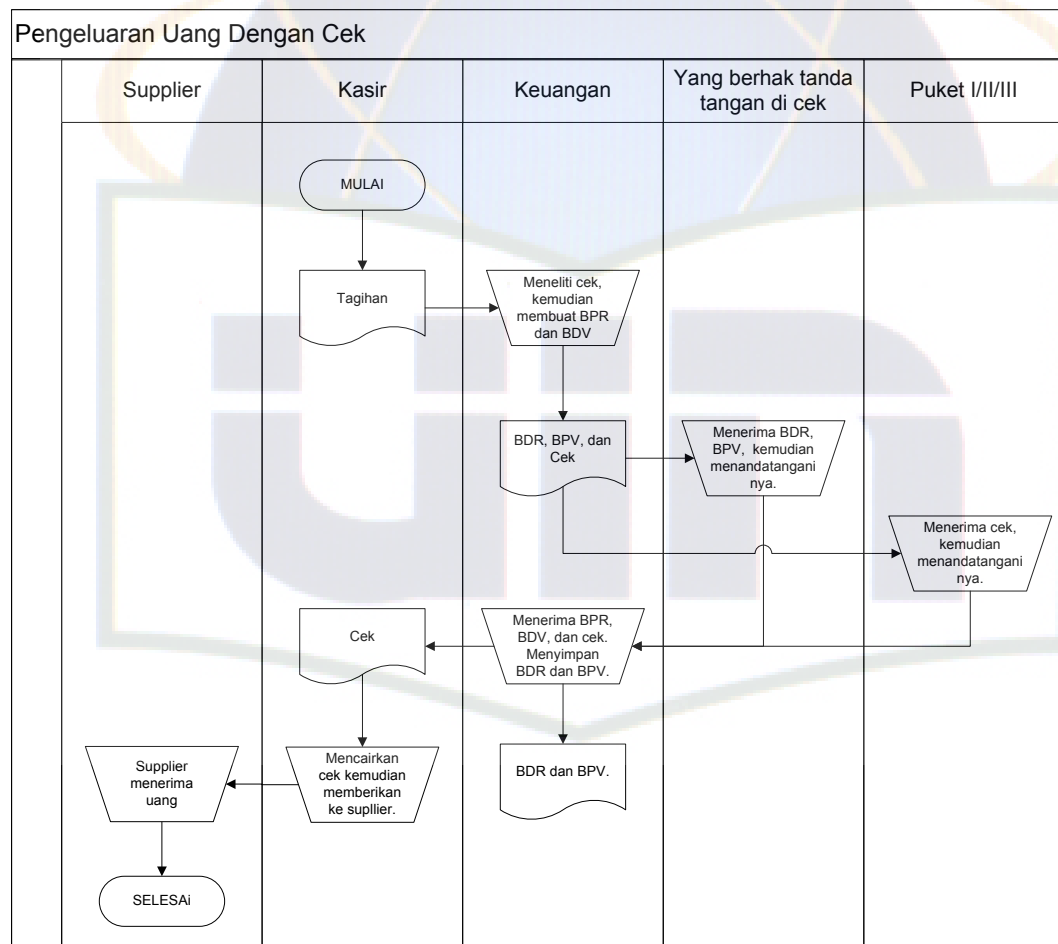
uang kurang dari satu juta rupiah. Karyawan yang ingin meminta uang harus melapor terlebih dahulu kepada bagian keuangan. Kemudian bagian keuangan akan membuat formulir yang harus ditandatangani oleh Puket. Berikut adalah bagan permintaan uang dengan *petty cash*.



Gambar 4.1 *Flowchart* sistem berjalan untuk permintaan uang dengan *petty cash*

Pengeluaran uang lainnya yaitu dengan cek. Untuk memakai cek, maka akan menggunakan dua buah formulir, yaitu BDR (*Bank Disbursement Requisition*) dan BPV (*Bank Payment Voucher*). Kasir akan memberikan ketiga formulir tersebut kepada bagian keuangan. Kemudian bagian keuangan akan menghitung tagihan tersebut. BDR dan BPV akan diberikan kepada yang berhak tanda tangan untuk menandatangani cek,

kemudian ke Puket. Setelah itu akan diberikan kembali kepada bagian keuangan, kemudian ke kasir untuk mencairkan cek. Berikut adalah pengeluaran uang dengan cek.

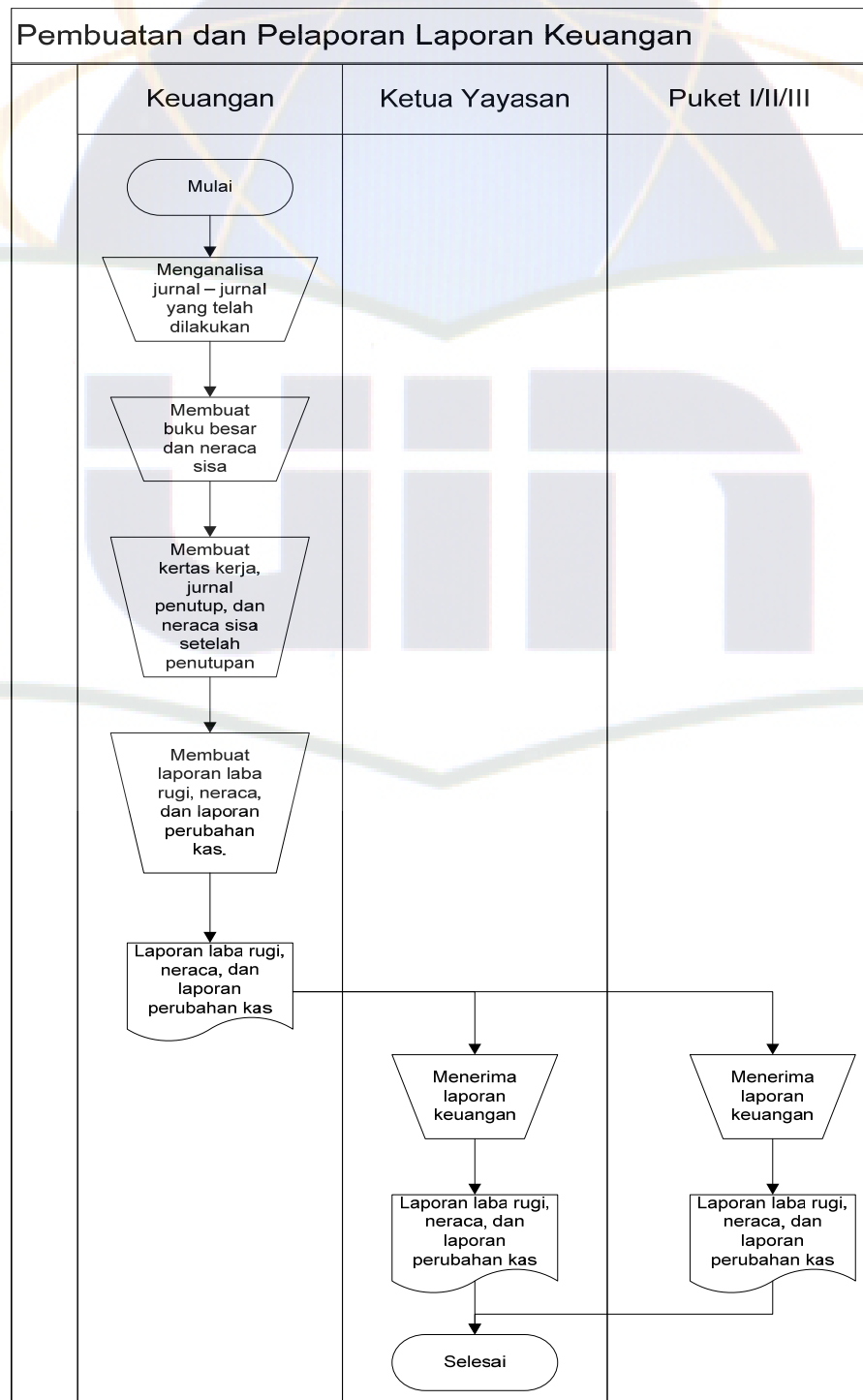


Gambar 4.2 *Flowchart* sistem berjalan proses permintaan uang dengan cek.

Pengeluaran lainnya adalah permintaan uang untuk suatu proyek. *Project Officer* bersama dengan bagian keuangan akan membuat *budget* mengenai proyek tersebut. *Budget* ini akan diberikan kepada Puket untuk disetujui. Bila disetujui, maka keuangan akan membuat BDR, BPV, serta cek. BDR dan BPV akan ditandatangani oleh yang berhak. Cek akan ditandatangani oleh Puket. Setelah itu, BDR dan BPV akan disimpan oleh



laporan ini akan diberikan kepada Puket dan Ketua Yayasan sebagai informasi untuk mengambil keputusan. Berikut adalah bagannya.



Gambar 4.4 *Flowchart* sistem berjalan proses laporan keuangan

Dari wawancara dan observasi ini, maka masalah – masalah yang ada yaitu :

- a. Sistem keuangan masih bersifat *manual*.
- b. Sistem keuangan melalui berbagai administrasi, sehingga membuat manajer sulit untuk mendapatkan informasi keuangan di IIUC.
- c. Ketika manajer ingin mendapatkan data – data tentang keuangan di IIUC, maka bagian keuangan harus bekerja lagi untuk mendata.

Dengan melihat permasalahan yang ada pada sistem yang sedang berjalan di IIUC, maka dibutuhkan suatu aplikasi yang mampu mengatasi masalah waktu dan tenaga. Oleh karena itu, perancangan aplikasi akuntansi berbasis *web* sebagai media penunjang sistem keuangan dianggap perlu di IIUC. Dengan aplikasi akuntansi ini, diharapkan permasalahan yang dihadapi dapat diatasi.

## 4.2 *Planning*

Dalam tahap ini dibuat penjadwalan untuk membuat aplikasi.

Beberapa tahapan kerja yang akan dilakukan antara lain :

1. Analisis kebutuhan aplikasi yang akan dibuat
2. Analisis tahapan proses akuntansi di IIUC
3. Perancangan DFD.
4. *Normalization*
5. Perancangan ERD.
6. Perancangan *flowchart*.

7. Perancangan *interface* serta pengkodean aplikasi.
8. *Testing* kepada *user* untuk mendapatkan *feedback*.

### 4.3 Modeling

Pada tahapan ini, penulis akan merancang sistem guna menyelesaikan masalah yang ada pada sistem keuangan IIUC sebelumnya. Adapun rancangan sistem yang penulis buat meliputi analisis kebutuhan akuntansi, analisis tahapan proses keuangan di IIUC, *normalization*, perancangan ERD, perancangan DFD, perancangan *flowchart*, dan desain *interface*.

#### 4.3.1 Analisis Kebutuhan Aplikasi

Berdasarkan wawancara dan observasi, maka aplikasi yang akan dibuat adalah aplikasi akuntansi berbasis *web*. *Web server* yang digunakan adalah Apache, menggunakan bahasa PHP dan *database* MySQL. Semua ini dapat di-*install* menggunakan XAMPP. Kemudian untuk membuat *chart* menggunakan FusionChart Free dan menggunakan jQuery untuk membuat aplikasi lebih interaktif. Untuk memakai aplikasi ini, maka harus dibuat beberapa program untuk masalah berikut :

1. *User* : berguna untuk mengklasifikasikan *user* yang menggunakan aplikasi ini. Terdapat dua *user*, yaitu *admin* dan *user*. *Admin* dapat mengganti data – data yang ada dan bisa memasukkan

transaksi, sedangkan *user* hanya bisa melihat jurnal, buku besar, laporan keuangan, dan grafik.

2. Rekening : berguna untuk mengklasifikasikan transaksi – transaksi yang ada. Perhitungan transaksi yang ada berdasarkan kelompok rekening ini, yaitu harta, kewajiban, modal, pendapatan, dan beban.

3. Jurnal : berguna untuk memasukkan transaksi tersebut ke dalam *database*. Terdapat empat jurnal, yaitu jurnal umum, jurnal penyesuaian, jurnal penutup, dan jurnal pembalik.

4. Buku besar : berguna untuk mencatat semua transaksi yang berhubungan dengan rekening tertentu. Disini dapat terlihat keadaan perubahan rekening karena transaksi yang menggunakan rekening tersebut.

5. Laporan keuangan : berguna untuk melihat keadaan keuangan dari IIUC dan memproduksinya ke dalam Microsoft Excel. Laporan keuangan yang dibuat adalah laporan laba rugi, neraca saldo, dan laporan perubahan modal.

6. Grafik : berguna untuk melihat saldo rekening tertentu pada rentang waktu tertentu. Program ini menggunakan diagram batang agar menarik dan interaktif.

#### 4.3.2 Analisis Proses Akuntansi

Laporan Keuangan dibuat berdasarkan tahapan – tahapan dalam mengolah informasi keuangan. Tahapan – tahapan itu antara lain :

1. Gambaran proses akuntansi
2. Mekanisme pencatatan transaksi
3. Proses akuntansi
4. Target akhir

##### 4.3.2.1 Gambaran Proses Akuntansi Di IIUC

Pada subbab *communication* telah dijelaskan tentang gambaran proses keuangan. Berikut adalah gambaran umum dari proses akuntansi. Pertama, aplikasi akuntansi mengukur kegiatan – kegiatan usaha dengan mencatat data tentang kegiatan – kegiatan tersebut untuk penggunaannya di masa yang akan datang. Dalam pencatatan ini, transaksi – transaksi diklasifikasikan dan dirangkum. Kedua, data tersebut disimpan hingga kelak dibutuhkan dan kemudian diproses menjadi informasi yang berfaedah. Ketiga, informasi dikomunikasikan, tentunya melalui laporan keuangan, kepada para



manajer. Data perihal kegiatan – kegiatan bisnis merupakan masukan bagi aplikasi akuntansi, dan keluarannya berupa informasi yang berguna bagi para manajer.

#### 4.3.2.2 Mekanisme Pencatatan Transaksi

Berikut akan dipaparkan proses pencatatan transaksi keuangan.

1. *User* harus terlebih dahulu membuat subklasifikasi rekening. *User* akan memasukkan kode subklasifikasi nama klasifikasi, dan kelompok rekening. Ada lima kelompok rekening yang digunakan, yaitu harta, kewajiban, modal, pendapatan, dan beban. Masing – masing kelompok rekening mempunyai perhitungan yang berbeda ketika ditempatkan di debet atau kredit. Setelah itu, *user* dapat membuat rekening, yaitu dengan memasukkan kode rekening, nama rekening, serta subklasifikasi rekening yang akan digunakan.
2. Selanjutnya, *user* akan memasukkan transaksi tersebut kedalam jurnal. Rekening yang sudah dibuat akan digunakan disini. *User* bisa memasukkan rekening lebih dari satu, tapi rekening tersebut tidak bisa muncul dua kali. Selain itu, *user* harus

memasukkan jumlah uang yang digunakan dalam transaksi tersebut. Jumlah uang di kolom debit dan kolom kredit harus sama. Bila tidak, maka transaksi tidak bisa dimasukkan. Terdapat empat jurnal disini, yaitu :

a. Jurnal Umum

Jurnal yang dipakai untuk pencatatan transaksi secara umum.

b. Jurnal Penyesuaian

Jurnal yang dipakai untuk menyesuaikan rekening yang telah berubah nilainya. Contoh akunnya yaitu peralatan dan pendapatan.

c. Jurnal Penutup

Jurnal yang dipakai untuk meng-nolkan rekening pendapatan dan beban, karena akan dimasukkan kedalam ikhtisar laba rugi.

d. Jurnal Koreksi

Jurnal yang dipakai untuk mengoreksi kesalahan yang ada pada transaksi. Jurnal ini membalik posisi rekening yang digunakan pada saat transaksi. Jumlah uang yang digunakan tetap sama saat transaksi.

### 3. Jurnal

Jurnal disini adalah data – data transaksi yang telah berlangsung. *User* bisa melihat rekening – rekening apa saja yang digunakan dalam transaksi tersebut.

### 4. Buku Besar

*User* dapat melihat transaksi – transaksi apa saja yang menggunakan rekening yang dipilih. Disini, *user* juga bisa melihat saldo rekening tersebut selama transaksi berlangsung.

### 5. Laporan Keuangan

Laporan keuangan yang dibuat ada tiga, yaitu laporan laba rugi, neraca saldo, dan laporan perubahan modal. Laporan laba rugi akan memperlihatkan laba atau rugi yang didapat IIUC, neraca saldo menginformasikan aktiva dan pasiva yang dimiliki oleh IIUC, dan laporan perubahan modal memperlihatkan perubahan modal yang terjadi di IIUC karena aktivitas – aktivitas ekonomi.

#### 4.3.2.3 Proses Perhitungan Akuntansi

Setelah *user* memasukkan transaksi tersebut, maka aplikasi akan melakukan perhitungan untuk membuat

buku besar, dan laporan keuangan. Berikut adalah penjelasannya :

#### 1. Jurnal Umum

Perhitungan transaksi – transaksi yang ada berdasarkan kelompok rekening dan jumlah uang yang dipakai. Karakteristik rekening dapat dilihat pada tabel 2.1 Bila rekening harta atau beban berada pada debit, maka saldonya akan ditambah dengan jumlah uang yang digunakan. Bila rekening harta atau beban berada pada kredit, maka saldonya akan dikurangi dengan jumlah uang yang digunakan. Bila rekening kewajiban, modal, dan pendapatan berada pada debit, maka saldonya akan dikurangi. Bila rekening kewajiban, modal, dan pendapatan berada pada kredit, maka saldonya akan ditambah.

#### 2. Jurnal Penyesuaian

Aturan perhitungan pada jurnal penyesuaian sama seperti jurnal umum. Namun, rekening yang digunakan disini adalah rekening yang bisa menyusut atau berkurang nilainya, seperti peralatan, perlengkapan, dan gedung. Berkurangnya nilai rekening tersebut akan

ditampung pada rekening dengan klasifikasi beban.

### 3. Jurnal Penutup

Jurnal penutup berfungsi untuk meng-nolkan rekening pendapatan dan beban. Pada jurnal ini, *user* tidak bisa memasukkan rekening baru, karena sudah ditentukan oleh aplikasi. Aplikasi akan mencari selisih antara jumlah pendapatan dan beban. Bila hasilnya positif, maka IIUC mendapatkan laba. Bila hasilnya negatif, maka IIUC menderita rugi.

### 4. Jurnal Koreksi

Jurnal koreksi berfungsi untuk mengkoreksi jurnal yang telah dimasukkan. Yang dilakukan oleh jurnal koreksi yaitu membalik posisi debit atau kredit rekening pada saat transaksi. Jumlah uang yang digunakan di jurnal koreksi sama dengan jurnal yang akan dikoreksi. Dengan ini, maka saldo dari rekening – rekening yang ada pada jurnal tersebut akan kembali seperti sebelum proses penjurnalan.

### 5. Buku Besar

Buku besar menyimpan semua transaksi yang berhubungan dengan suatu rekening tertentu. Perhitungan dalam buku besar berdasarkan karakteristik dari akun tersebut. Di dalam buku besar akan terlihat jumlah saldo dari rekening tersebut ketika melakukan transaksi.

### 6. Laporan Laba Rugi

Laporan ini adalah untuk menunjukkan laba atau rugi yang didapatkan IIUC. Untuk mendapatkan laba atau rugi adalah dengan mendapatkan selisih antara jumlah pendapatan dan jumlah beban. Didalam laporan ini akan terlihat saldo dari semua rekening pendapatan dan rekening beban.

### 7. Neraca Saldo

Neraca saldo akan memperlihatkan kondisi aktiva dan pasiva yang dimiliki oleh IIUC. Aktiva adalah harta yang dimiliki oleh IIUC, sedangkan pasiva adalah modal dan kewajiban. Semua rekening harta, kewajiban, dan modal akan terlihat disini beserta saldo masing – masing rekening dan jumlah saldo aktiva dan pasiva.

## 8. Laporan Perubahan Modal

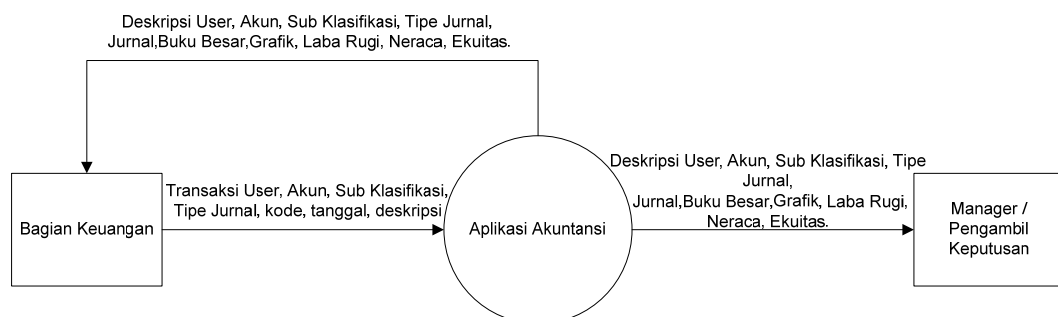
Laporan ini untuk menunjukkan perubahan modal karena aktivitas – aktivitas ekonomi yang berjalan. Disini akan dihitung laba atau rugi yang diterima oleh IIUC dan pengambilan pribadi yang dilakukan. Laba akan menambah modal, sedangkan rugi dan pengambilan pribadi akan mengurangi modal.

### 4.3.2.4 Target akhir

Target akhir yang akan dicapai adalah sebuah aplikasi akuntansi yang bisa membantu bagian keuangan untuk dapat bekerja lebih efisien dan para manajer dalam menerima informasi. Selain itu, para manajer juga dapat dengan mudah untuk mengakses informasi – informasi dalam aplikasi ini.

### 4.3.3 Data Flow Diagram

#### 4.3.3.1. Diagram Konteks Aplikasi Akuntansi

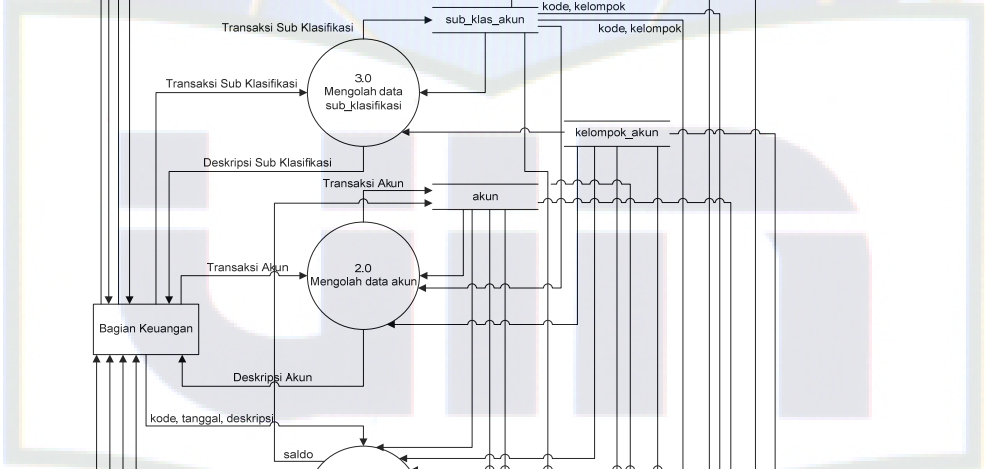


Gambar 4.6 Diagram Konteks Aplikasi Akuntansi

Dalam diagram konteks diatas terdapat dua buah entitas, yaitu bagian keuangan dan *manager* atau pengambil keputusan. Dalam hal ini, yang bisa memasukkan transaksi hanyalah bagian keuangan saja, sedangkan *manager* tidak bisa memasukkan transaksi. Manager hanya bisa menampilkan data – data tentang keuangan serta melihat laporan keuangan. Untuk kamus data dapat dilihat pada subbab 4.3.3.4 pada halaman 82.



The diagram illustrates the data flow for the 'Transaksi Sub Klasifikasi' process. It shows a flow from 'Bagian Keuangan' to '3.0 Mengolah data sub\_klasifikasi' and then to 'sub\_klas\_akun'. Data is also exchanged with 'akun' and '2.0 Mengolah data akun'. The diagram includes labels for data types like 'kode, kelompok', 'kelompok\_akun', 'Transaksi Sub Klasifikasi', 'Deskripsi Sub Klasifikasi', 'Transaksi Akun', 'Deskripsi Akun', and 'saldo'.



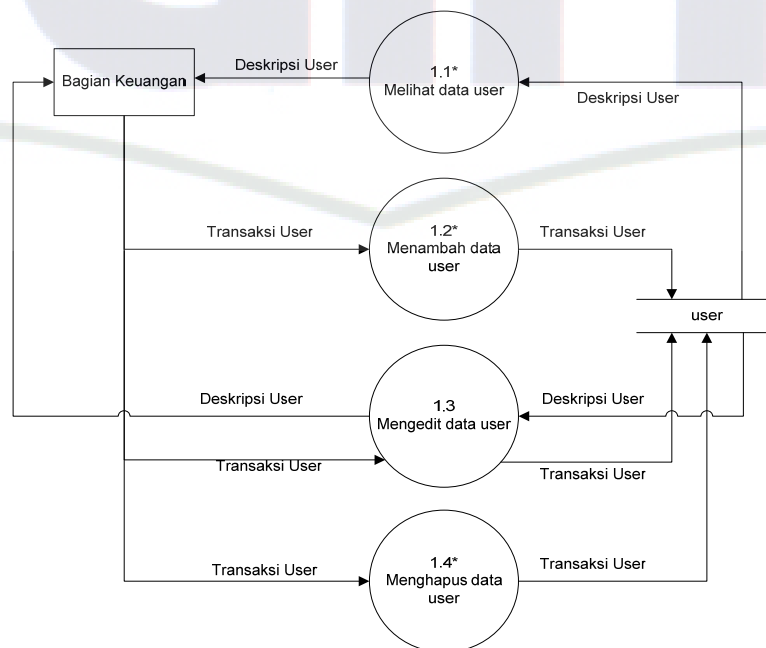
Gambar 4.7 Diagram *Level 1* Aplikasi Akuntansi

Dalam diagram *Level 0* diatas terdapat sembilan proses. Semua proses tersebut dapat dilakukan oleh *user* yang berstatus *admin*. Sedangkan *user* yang berstatus *user* hanya bisa melihat jurnal, buku besar, grafik keuangan, laporan keuangan. Penjelasan mendetail dari setiap proses akan dijelaskan berikut ini.

#### 4.3.3.3. Diagram *Level 2*

Diagram ini untuk mendetailkan setiap proses yang terjadi pada diagram *Level 2* aplikasi akuntansi.

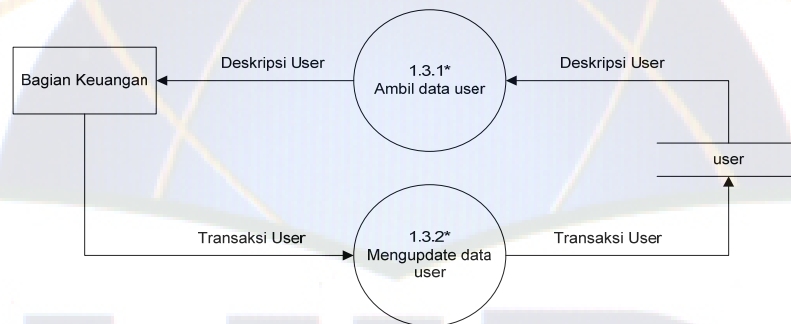
##### 1. Diagram *Level 2* Proses 1.0



Gambar 4.8 Diagram *Level 2* Proses 1.0

Proses 1.0 adalah proses yang berhubungan dengan data *user*. Yang bisa mengakses proses ini adalah hanya *user* yang berstatus *admin*. Didalamnya terdapat empat

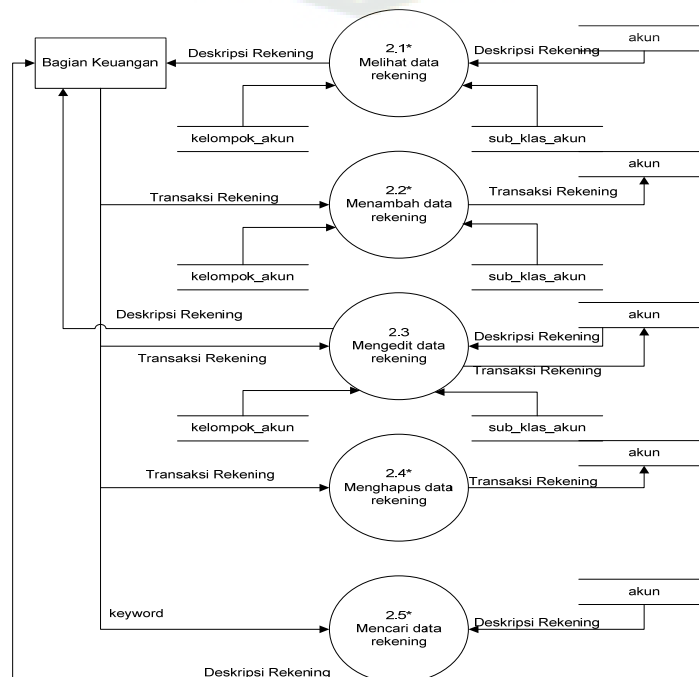
proses, yaitu melihat, menambah, meng-*edit*, dan menghapus data *user*. Berikut adalah detail dari proses *edit user*.



Gambar 4.9 Diagram *Level 3* Proses 1.3

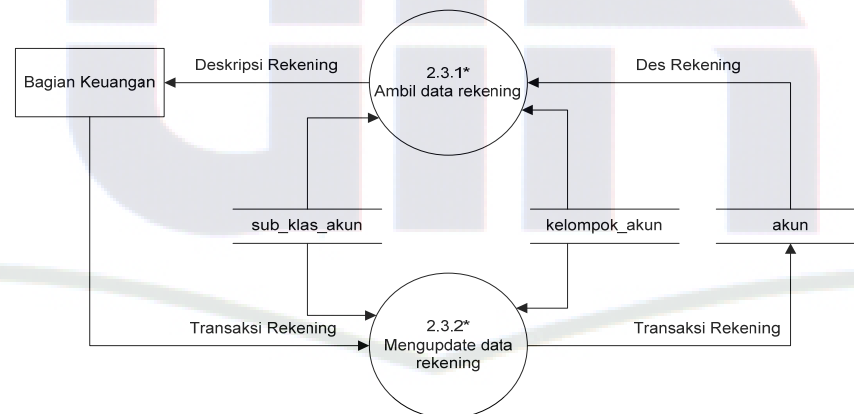
Dalam proses *edit user*, pertama aplikasi akan mengambil data tentang *user* tersebut. Kemudian *admin* akan meng-*edit* data tersebut, dan di-*update*.

## 2. Diagram *Level 2* Proses 2.0



Gambar 4.10 Diagram *Level 2* Proses 2.0

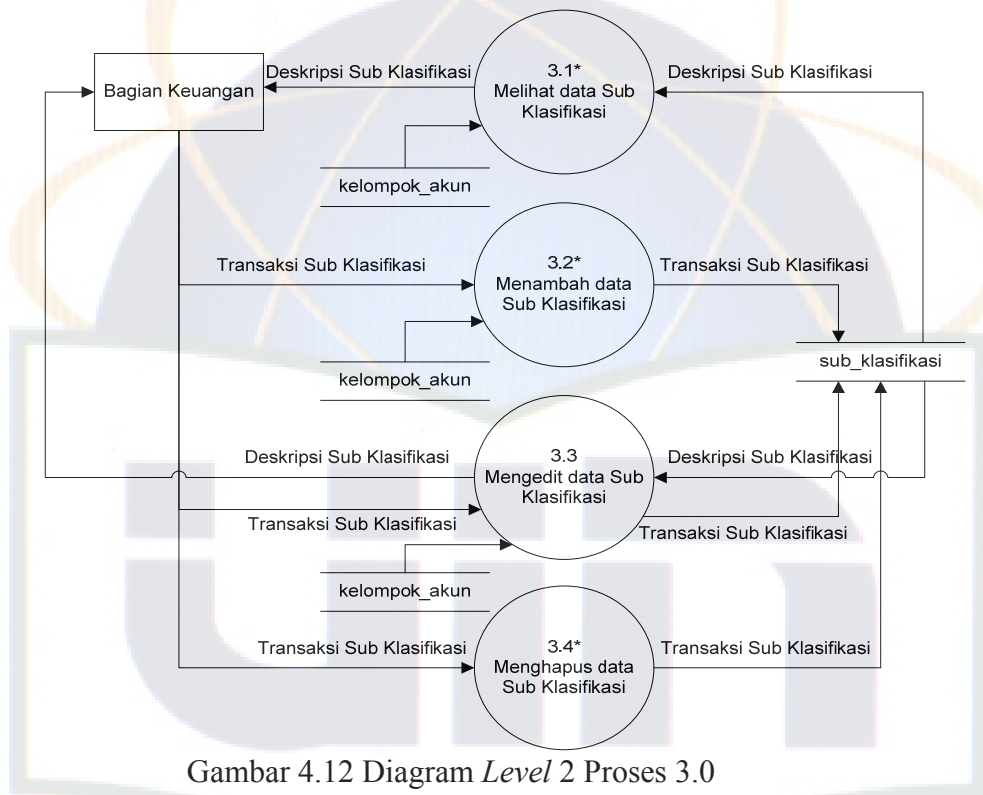
Proses 2.0 adalah proses yang berhubungan dengan rekening. Yang bisa mengakses proses ini hanyalah *user* yang berstatus *admin*. Didalamnya terdapat lima proses, yaitu mencari, melihat, menambah, meng-*edit*, dan menghapus data rekening. Proses ini berhubungan dengan tabel *sub\_klas\_akun* dan *kelompok\_akun* untuk memanipulasi tabel *akun*. Berikut akan digambarkan detail dari proses *edit* rekening.



Gambar 4.11 Diagram *Level 3* Proses 2.3

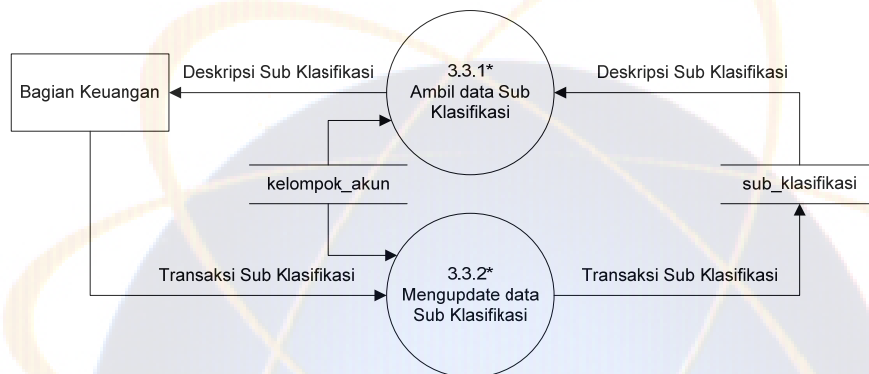
Untuk meng-*edit* rekening, maka aplikasi akan mengambil data tentang rekening tersebut untuk ditampilkan. Setelah itu, *admin* meng-*edit* kemudian meng-*update*.

### 3. Diagram *Level 2* Proses 3.0



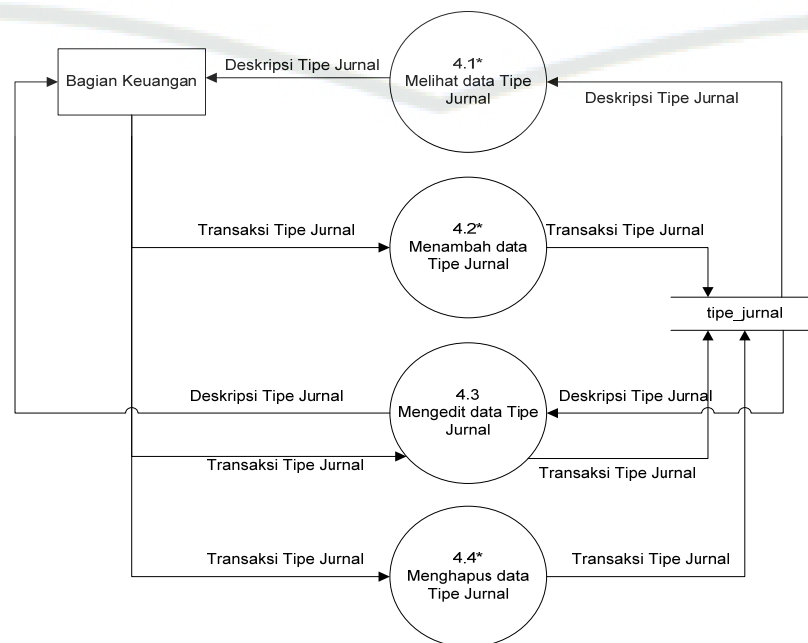
Gambar 4.12 Diagram *Level 2* Proses 3.0

Proses 3.0 adalah proses yang berhubungan dengan sub klasifikasi rekening. Yang bisa mengakses proses ini hanyalah *user* yang berstatus *admin*. Didalamnya terdapat empat proses, yaitu melihat, menambah, meng-*edit*, dan menghapus data sub klasifikasi rekening. Proses ini berhubungan dengan tabel *kelompok\_akun* untuk memanipulasi tabel *sub\_klas\_akun*. Berikut akan digambarkan detail dari proses *edit* sub klasifikasi rekening.

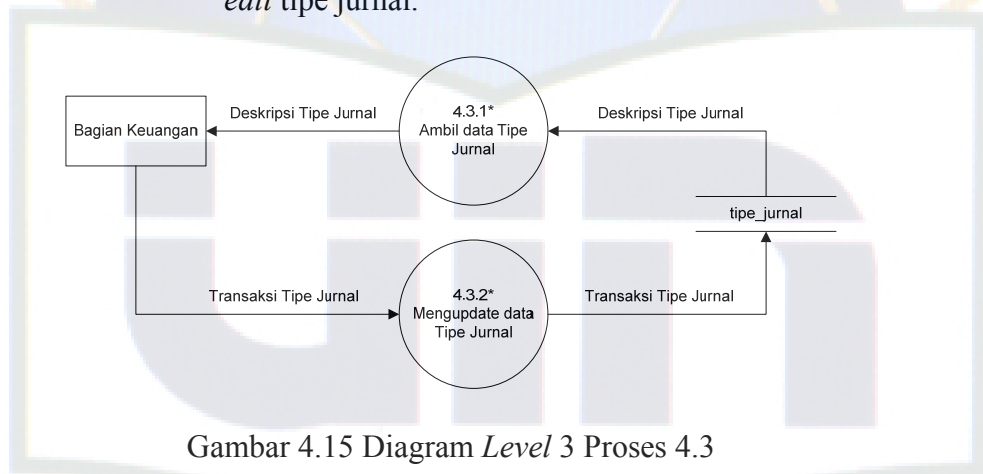
Gambar 4.13 Diagram *Level 3* Proses 3.3

Untuk meng-*edit* sub klasifikasi rekening, maka aplikasi akan mengambil data tentang sub klasifikasi rekening tersebut untuk ditampilkan. Setelah itu, *admin* meng-*edit* kemudian meng-*update*.

#### 4. Diagram *Level 2* Proses 4.0

Gambar 4.14 Diagram *Level 2* Proses 4.0

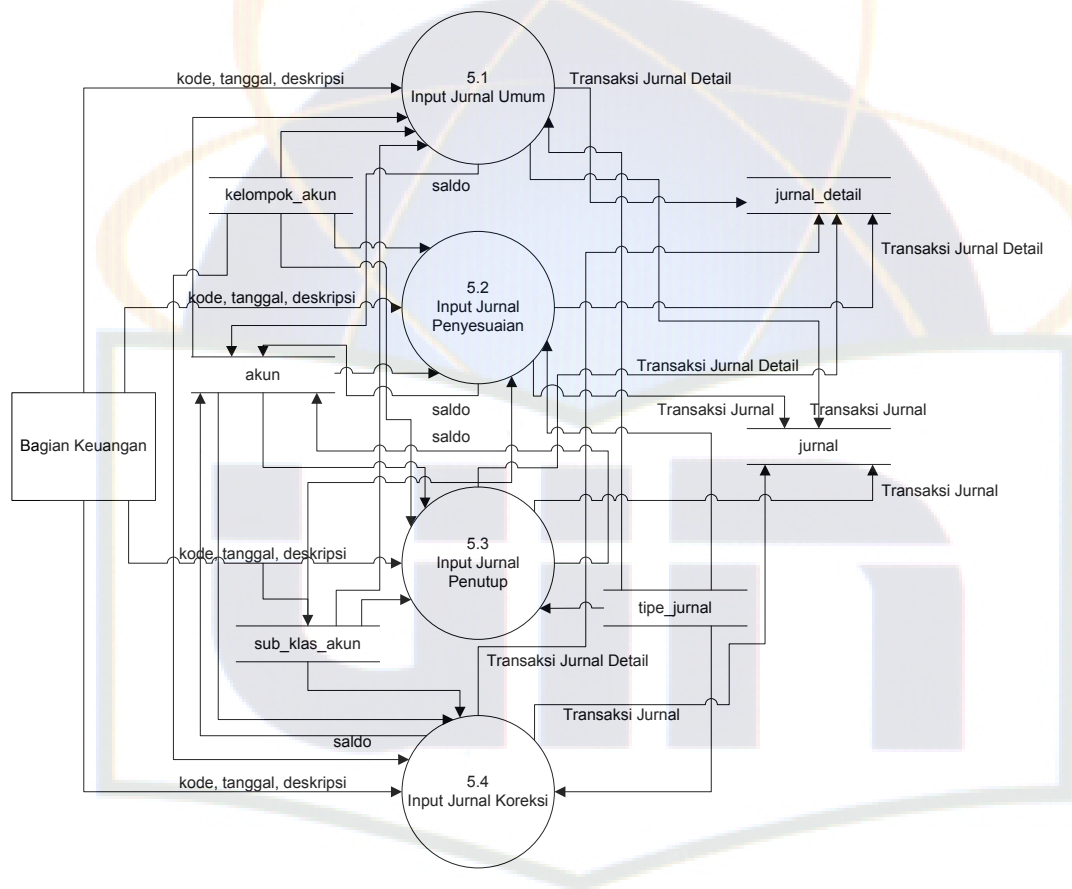
Proses 4.0 adalah proses yang berhubungan dengan tipe jurnal. Yang bisa mengakses proses ini hanyalah *user* yang berstatus *admin*. Didalamnya terdapat empat proses, yaitu melihat, menambah, meng-*edit*, dan menghapus data tipe jurnal. Berikut akan digambarkan detail dari proses *edit* tipe jurnal.



Gambar 4.15 Diagram *Level 3* Proses 4.3

Untuk meng-*edit* tipe jurnal, maka aplikasi akan mengambil data tentang tipe jurnal tersebut untuk ditampilkan. Setelah itu, *admin* meng-*edit* kemudian meng-*update*.

### 5. Diagram *Level 2* Proses 5.0

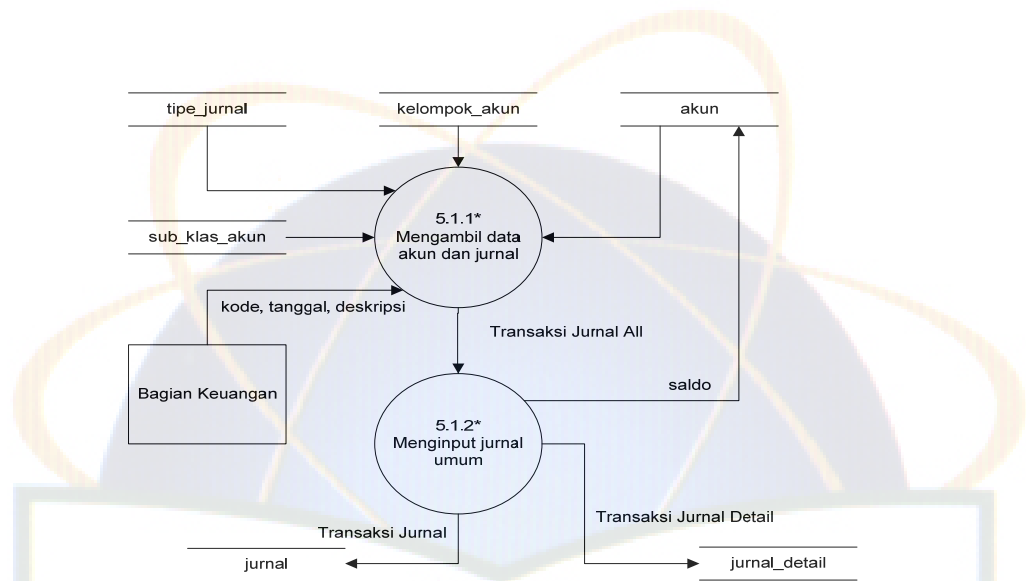


Gambar 4.16 Diagram *Level 2* Proses 5.0

Proses 5.0 adalah proses untuk memasukkan data transaksi melalui jurnal umum, jurnal penyesuaian, jurnal penutup, dan jurnal koreksi. Proses 5.0 berhubungan dengan tabel kelompok\_akun, sub\_klas\_akun, akun, tipe\_jurnal, jurnal, dan jurnal detail. Di dalam proses 5.0 terdapat empat proses lainnya. Berikut akan dijelaskan proses – proses tersebut :

#### a. Diagram *Level 3* Proses 5.1

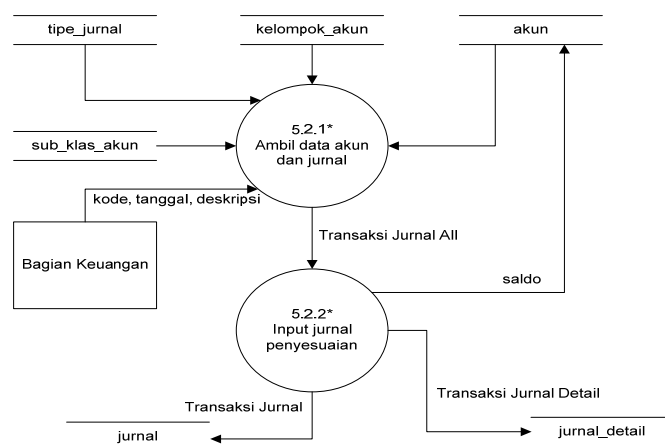




Gambar 4.17 Diagram *Level 3* Proses 5.1

Proses 5.1 adalah proses untuk menambahkan jurnal umum. Yang bisa mengakses proses ini hanya *user* yang berstatus *admin*. Proses ini akan mengambil data dari table akun, sub\_klas\_akun, kelompok\_akun dan tipe jurnal. Proses ini akan memanipulasi tabel akun, jurnal, dan jurnal\_detail.

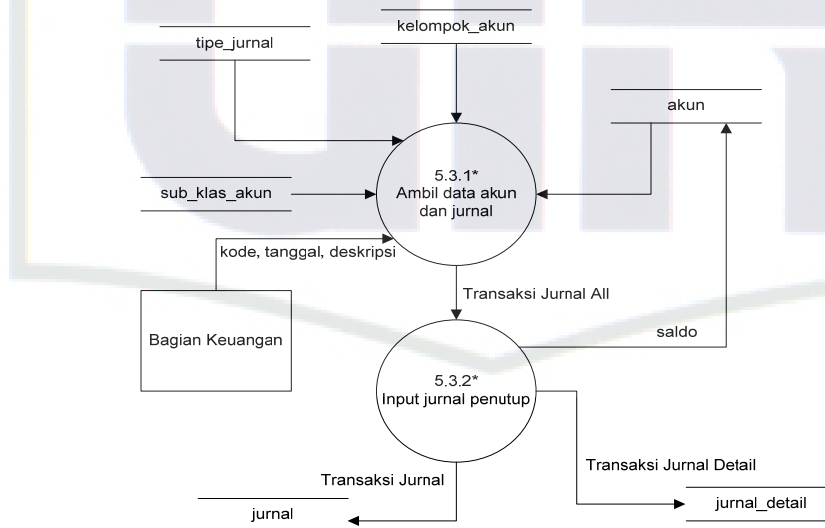
b. Diagram *Level 3* Proses 5.2



Gambar 4.18 Diagram *Level 3* Proses 5.2

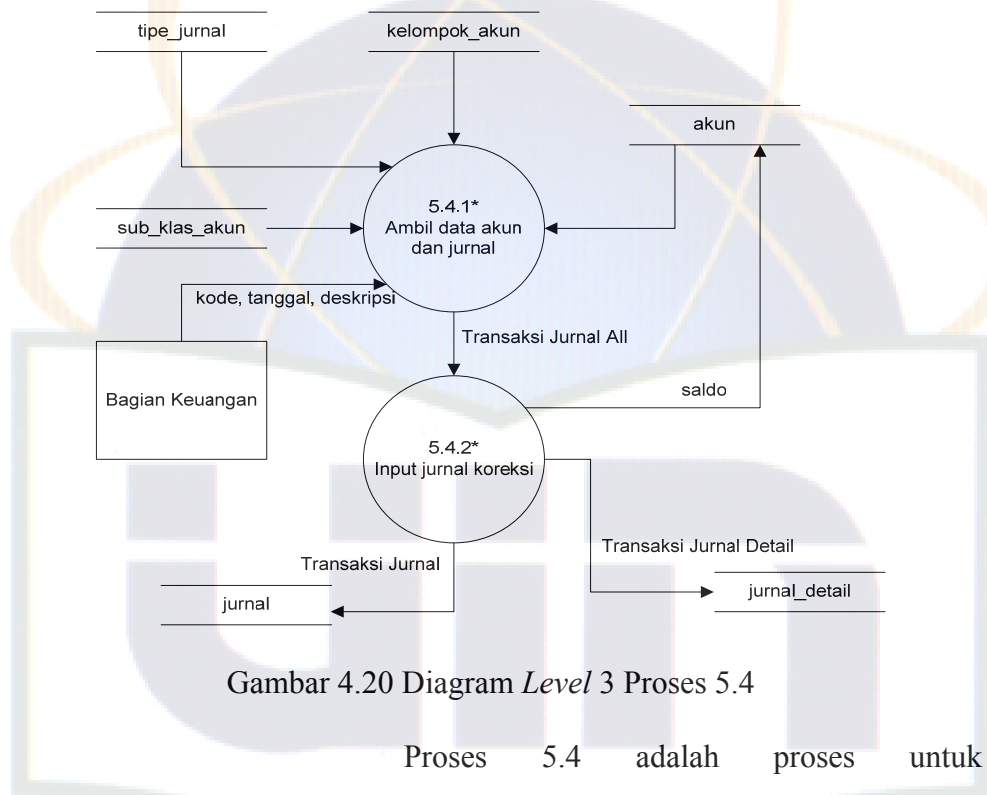
Proses 5.2 adalah proses untuk menambahkan jurnal penyesuaian. Yang bisa mengakses proses ini hanya *user* yang berstatus *admin*. Proses ini akan mengambil data dari table akun, sub\_klas\_akun, kelompok\_akun dan tipe jurnal. Proses ini akan memanipulasi tabel akun, jurnal, dan jurnal\_detail.

c. Diagram *Level 3* Proses 5.3



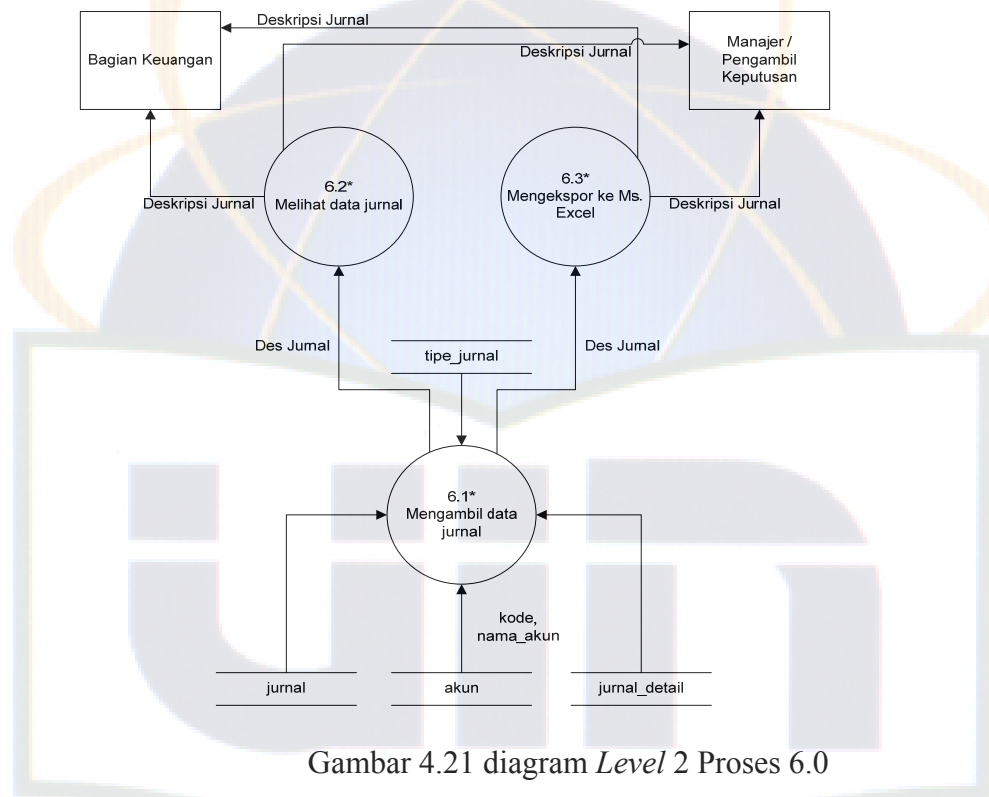
Gambar 4.19 Diagram *Level 3* Proses 5.3

Proses 5.3 adalah proses untuk menambahkan jurnal penutup. Yang bisa mengakses proses ini hanya *user* yang berstatus *admin*. Proses ini akan mengambil data dari table akun, sub\_klas\_akun, kelompok\_akun dan tipe jurnal. Proses ini akan memanipulasi tabel akun, jurnal, dan jurnal\_detail.

d. Diagram *Level 3* Proses 5.4

menambahkan jurnal koreksi. Yang bisa mengakses proses ini hanya *user* yang berstatus *admin*. Proses ini akan mengambil data dari table akun, sub\_klas\_akun, kelompok\_akun dan tipe jurnal. Proses ini akan memanipulasi tabel akun, jurnal, dan jurnal\_detail.

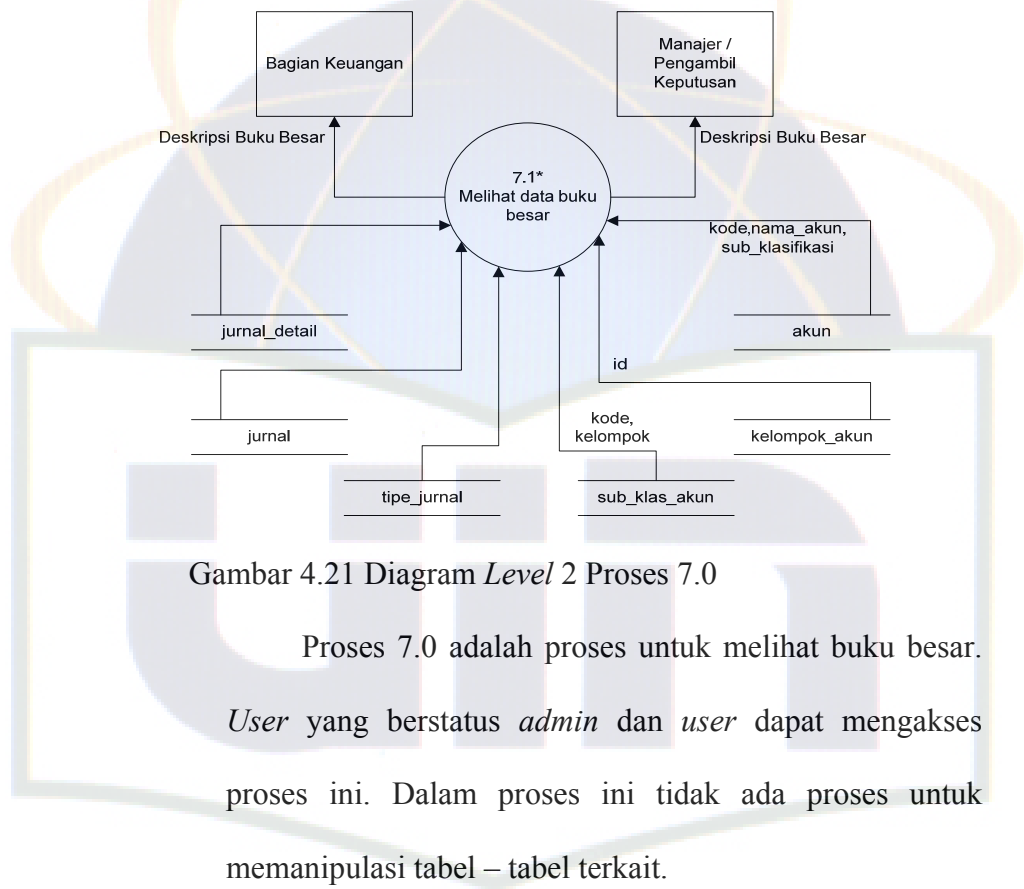
### 6. Diagram *Level 2* Proses 6.0



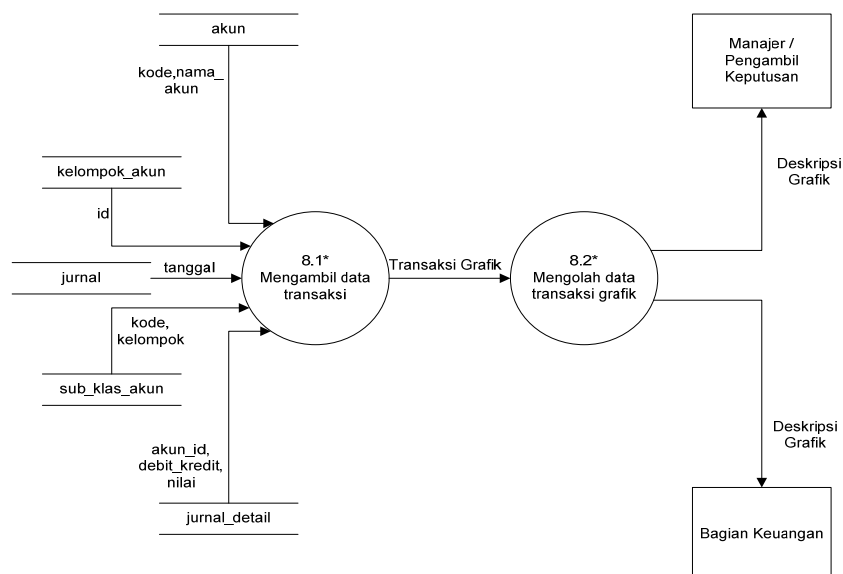
Gambar 4.21 diagram *Level 2* Proses 6.0

Proses 6.0 adalah proses yang berhubungan dengan data jurnal. *User* yang berstatus *admin* dan *user* dapat mengakses proses ini. Didalamnya terdapat proses untuk mengambil data jurnal, melihat data jurnal dan mengekspor data tersebut ke dalam Microsoft Excel. Untuk mengambil, melihat dan mengekspor jurnal hanya bisa dalam rentang waktu satu hari. Dalam proses ini tidak ada proses untuk memanipulasi tabel – tabel terkait.

### 7. Diagram *Level 2* Proses 7.0

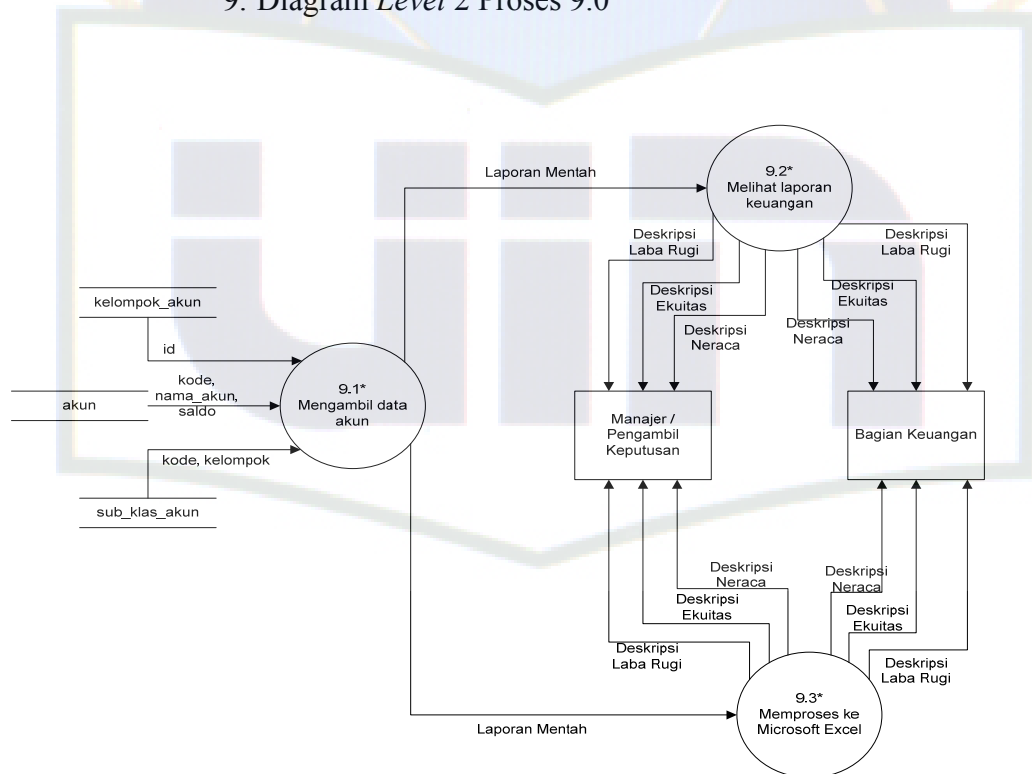


### 8. Diagram *Level 2* Proses 8.0



Proses 8.0 adalah proses yang berhubungan dengan grafik kondisi keuangan pada rentang waktu tertentu. Didalamnya terdapat dua proses, yaitu pengambilan data dan mengolah data grafik. Di proses ini tidak ada proses memanipulasi tabel – tabel terkait.

#### 9. Diagram *Level 2* Proses 9.0



Gambar 4.24 Diagram *Level 2* Proses 9.0

Proses 9.0 adalah proses untuk membuat laporan keuangan. Laporan keuangan yang dibuat yaitu laporan laba rugi, neraca saldo, dan laporan perubahan modal. Laporan – laporan tersebut dapat diekspor ke dalam Microsoft Excel. Dalam proses ini tidak ada proses untuk memanipulasi tabel – tabel terkait.

#### 4.3.3.4 Kamus Data

Berikut ini akan dipaparkan kamus data, yaitu data – data yang mengalir di dalam *Data Flow Diagram* diatas :

1. Transaksi User : username, password, status
2. Deskripsi User : username, status
3. Transaksi Akun : kode akun, nama, sub klasifikasi
4. Des Akun : kode akun, nama akun, saldo, sub klasifikasi.
5. Deskripsi Akun : kode akun, nama akun, saldo, sub klasifikasi, nama kelompok akun.
6. Deskripsi Sub Klasifikasi : kode sub klasifikasi, nama sub klasifikasi, nama kelompok akun
7. Transaksi Sub Klasifikasi : kode sub klasifikasi, nama sub klasifikasi, nama kelompok akun.
8. Deskripsi Tipe Jurnal : id tipe jurnal, nama tipe jurnal.
9. Transaksi Tipe Jurnal : id tipe jurnal, nama tipe jurnal.
10. Transaksi Jurnal : id jurnal, tipe jurnal, tanggal, deskripsi.
11. Transaksi Jurnal Detail : id jurnal detail, id jurnal, item, akun, posisi, nilai.

12. Transaksi Jurnal All : id jurnal, tipe jurnal, tanggal, deskripsi, id jurnal detail, item, akun, posisi, nilai.

13. Des Jurnal : id jurnal, tipe jurnal, tanggal, deskripsi, item, akun, kode akun, debit\_kredit, nilai.

14. Deskripsi Jurnal : id jurnal, tipe jurnal, tanggal, deskripsi, item, akun, kode akun, debit\_kredit, nilai, jumlah

15. Deskripsi Buku Besar : tanggal, deskripsi, tipe jurnal, debit, kredit, DC, saldo.

16. Transaksi Grafik : kode akun, nama akun, id kelompok akun, tanggal jurnal, kode sub klasifikasi akun, kelompok sub klasifikasi akun, debit kredit, nilai.

17. Deskripsi Grafik : nama akun, jumlah nilai.

18. Laporan Mentah : id kelompok akun, kode sub klasifikasi akun, kode kelompok, kode akun, nama akun, saldo.

19. Deskripsi Laba Rugi : nama akun, saldo, jumlah pendapatan, jumlah beban.

20. Deskripsi Neraca : nama aktiva, nama pasiva, saldo, jumlah aktiva, jumlah pasiva.

21. Deskripsi Ekuitas : nama akun, laba, modal akhir.



#### 4.3.4 *Normalization*

Berikut akan dijelaskan proses normalisasi, mulai dari tabel yang belum dinormalisasi sampai 3NF.

##### 1. *Unnormalized Form*

Tahap pertama ini, semua data transaksi akan dicatat. Pada tabel berikut, kolom item, tipe akun, subklasifikasi, kode\_akun, nama\_akun, debit\_kredit, nilai, dan saldo belum memiliki nilai atomik. Ini akan diperbaharui pada bentuk 1NF. Berikut adalah tabel yang belum dinormalisasi tersebut:

Tabel 4.1 Table transaksi *unnormalized*

Kode_Tipe	Tipe Jurnal	Kode_Jurnal	Tanggal	Deskripsi	Item	Kode_Tipe_Akun	Tipe Akun
1	Jurnal umum	12-01	2010-1-12	Setor modal	0	1	Harta
					1	3	Modal
2	Jurnal penyesuaian	13-98	2011-3-2	Peralatan	0	3	Modal
					1	1	Harta

Kode_Sub_Klasifikasi	Subklasifikasi	Kode_Akun	Nama_Akun	Debit_Kredit	Nilai	Saldo
11	Harta	110-10	Kas	1	10000	10000
31	Modal	310-10	Modal	0	10000	10000
32	Saham	320-10	Surat Berharga	1	15000	15000
11	Harta	170-30	PC	0	15000	15000

## 2. *First Normal Form (1NF)*

Syarat pada 1NF adalah semua kolom data harus memiliki nilai atomik. Maka dari itu, kolom item, kode\_tipe\_akun, tipe\_akun, kode\_sub\_klasifikasi, subklasifikasi, kode\_akun, nama\_akun, debit\_kredit, nilai, dan saldo harus diubah nilainya untuk mendapatkan nilai atomik. Untuk mendapatkan nilai atomik itu, maka data kedua yang terdapat pada kolom tersebut akan dipindahkan untuk dibuat baris baru. Berikut ini adalah komposisi tabel yang sudah berada pada 1NF.

Tabel 4.2 Tabel transaksi 1NF

Kode_Tipe	Tipe Jurnal	Kode	Tanggal	Deskripsi	Item	Kode_Tipe_Akun	Tipe Akun	Kode_Sub_Klasifikasi
1	Jurnal umum	12-01	2010-1-12	Setor modal	0	1	Harta	11
1	Jurnal umum	12-01	2010-1-12	Setor modal	1	3	Modal	31
2	Jurnal penyesuaian	13-98	2011-3-2	Pendapatan	0	3	Modal	32
2	Jurnal penyesuaian	13-98	2011-3-2	Pendapatan	1	1	Harta	11

Subklasifikasi	Kode_Akun	Nama_Akun	Debit_Kredit	Nilai	saldo
Harta	110-10	Kas	1	10000	10000
Modal	310-10	Modal	0	10000	10000
Saham	320-10	Surat Berharga	1	15000	15000
Harta	170-30	PC	0	15000	15000

### 3. *Second Normal Form (2NF)*

Persyaratan pada 2NF adalah harus berada dalam 1NF dan menghilangkan dependensi parsial. Data yang bukan *primary key* harus memiliki dependensi fungsional terhadap *primary key*. Berikut adalah *primary key* tersebut beserta kolom yang dipengaruhi :

- a. Kode\_Tipe -> Tipe\_Jurnal
- b. Kode -> Tanggal, Deskripsi, Item, Debit\_Kredit, Kode\_Akun, Nilai
- c. Kode\_Tipe\_Akun -> Tipe\_Akun, Kode\_Sub\_Klasifikasi, Subklasifikasi, Kode\_Akun, Nama\_Akun, Saldo

Maka dari tabel transaksi, akan dipecah dan dibuat tiga tabel baru, yaitu tabel tipe\_jurnal, jurnal, kemudian kelompok\_akun.

Tabel 4.3 Tabel tipe\_jurnal 2NF

id	nama
1	Jurnal umum
2	Jurnal penyesuaian

Tabel 4.4 Tabel jurnal 2NF

id	tipe	Tanggal	deskripsi	item	kode_akun	debit_kredit	nilai
12-01	1	2010-1-12	Setor modal	0	110-10	1	10000
12-01	1	2010-1-12	Setor modal	1	310-10	0	10000
13-98	2	2011-3-2	Pendapatan	0	320-10	1	15000
13-98	2	2011-3-2	Pendapatan	1	170-30	0	15000

Tabel 4.5 Tabel akun 2NF

Kode_Tipe_Akun	Tipe Akun	Kode_Sub_Klasifikasi	Subklasifikasi	Kode_Akun	Nama_Akun	Saldo
1	Harta	11	Harta	110-10	Kas	10000
3	Modal	31	Modal	310-10	Modal	10000
3	Modal	32	Saham	320-10	Surat Berharga	15000
1	Harta	11	Harta	170-30	PC	15000

#### 4. *Third Normal Form (3NF)*

Persyaratan 3NF adalah harus berada dalam 1NF dan 2NF, kemudian penghilangan dependensi transitif. Data yang bukan *primary key* tidak boleh memiliki dependensi transitif dengan *primary key*. Pada tabel tipe\_jurnal, nama berdependensi parsial dengan id, kemudian tidak ada dependensi transitif. Maka tabel tipe\_jurnal sudah berada dalam 3NF.

Pada tabel jurnal, tanggal, deskripsi, item, kode\_akun, debit\_kredit, dan nilai berdependensi parsial dengan kolom id dan berdependensi transitif dengan kolom tipe. Tabel jurnal ini kemudian dipecah menjadi dua, yaitu tabel jurnal dan tabel jurnal\_detail untuk menghilangkan dependensi transitif tersebut.

Pada tabel akun, kolom sub\_klasifikasi berdependensi parsial dengan kolom kode\_subklasifikasi dan berdependensi transitif dengan kolom kode\_tipe\_akun. Kemudian kolom nama\_akun berdependensi parsial dengan kode\_akun dan berdependensi transitif dengan kode\_tipe\_akun. Maka dari itu, dependensi transitif ini harus dihilangkan dengan membuat tabel baru. Ini adalah komposisi dari tabel – tabel tersebut :

a. Tabel tipe\_jurnal

id -> nama.

b. Tabel jurnal



id -> tipe, tanggal, deskripsi.

c. Tabel jurnal\_detail

id -> jurnal\_id, item, akun\_id, debit\_kredit, nilai.

d. Tabel kelompok\_akun

id -> nama\_akun

e. Tabel sub\_klas\_akun

kode -> kelompok, nama.

f. Table akun

kode -> nama\_akun, sub\_klasifikasi, saldo

Berikut adalah tabel – tabel tersebut :

Tabel 4.6 Tabel tipe\_jurnal 3NF

id	Nama
1	Jurnal umum
2	Jurnal penyesuaian

Tabel 4.7 Tabel jurnal 3NF

id	tipe	tanggal	deskripsi
12-01	1	2010-1-12	Setor modal
13-98	2	2011-3-2	Pendapatan

Tabel 4.8 Tabel jurnal\_detail 3NF

id	id_jurnal	item	akun_id	debit_kredit	nilai
1	12-01	0	110-10	1	10000

2	12-01	1	310-10	0	10000
3	13-98	0	320-10	1	15000
4	13-98	1	170-30	0	15000

Tabel 4.9 Tabel kelompok\_akun 3NF

id	nama_akun
1	Harta
3	Modal

Tabel 4.10 Tabel sub\_klas\_akun 3NF

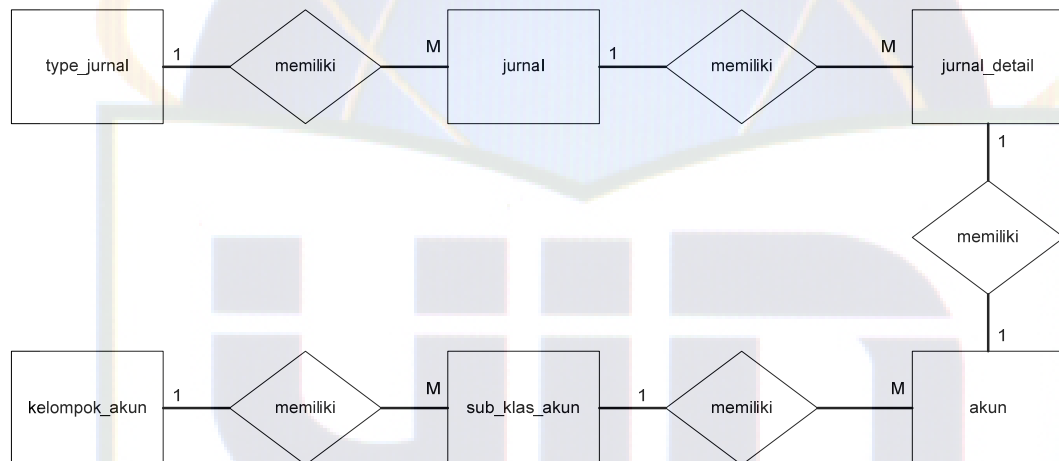
kode	kelompok	nama
11	1	Harta
31	3	Modal
32	3	Saham

Tabel 4.11 Tabel akun 3NF

kode_akun	nama_akun	sub_klasifikasi	saldo
110-10	Kas	11	10000
310-10	Modal	31	10000

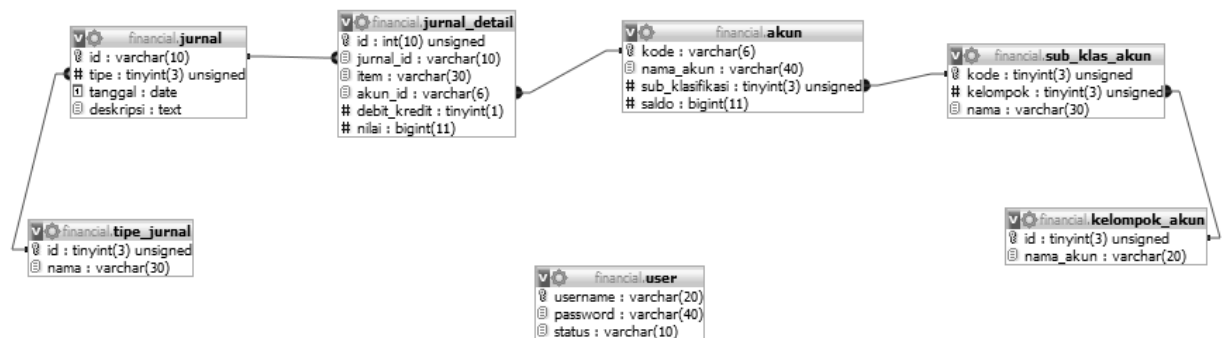
320-10	Surat Berharga	32	15000
170-30	PC	11	15000

#### 4.3.5 Entity Relationship Diagram



Gambar 4.24 Model ERD

Berikut adalah penerjemahan ERD ke LRS



Gambar 4.25 Model Logical Record Structure (LRS)

Seperti terlihat diatas, terdapat tujuh buah tabel. Semua tabel tersebut berhubungan dengan tabel yang lain, kecuali tabel *user*. Tabel *jurnal* memiliki *foreign key* dari tabel *tipe\_jurnal*. Tabel *akun* memiliki *foreign key* dari tabel

sub\_klas\_akun. Tabel sub\_klas\_akun memiliki *foreign key* dari tabel kelompok\_akun. Sedangkan tabel jurnal\_detail memiliki *foreign key* dari tabel jurnal dan tabel akun.

#### 4.3.5.1 Spesifikasi *Database*

Nama *database* : financial

Berikut ini adalah tabel yang terdapat di *database* ini :

##### a. Tabel Kelompok akun

Nama Tabel : kelompok\_akun

*Primary Key* : id

Tabel 4.12 Tabel kelompok akun

Nama <i>Field</i>	Type
id	tinyint(3)
nama_akun	varchar(20)

Tabel kelompok akun terdiri dari id dan nama\_akun. Ini dibuat berdasarkan kelompok rekening dalam akuntansi, yaitu harta, kewajiban, modal, pendapatan, dan pengeluaran. Tabel ini tidak dapat dimanipulasi oleh *user*, karena ini merupakan dasar bagi kegiatan akuntansi.

##### b. Tabel sub\_klas\_akun

Nama Tabel : sub\_klas\_akun

*Primary Key* : kode

*Foreign Key* : kelompok

Tabel 4.13 Tabel sub\_klas\_akun

Nama <i>Field</i>	Type
kode	tinyint(3)
kelompok	tinyint(3)
nama	varchar(20)

Tabel sub\_klas\_akun merupakan tabel subklasifikasi bagi rekening – rekening. Dari lima kelompok rekening tersebut bisa dibuat banyak subklasifikasi rekening. Tabel ini terdiri dari kode, kelompok, dan nama.

c. Tabel akun

Nama Tabel : akun

*Primary Key* : kode

*Foreign Key* : kelompok\_akun\_id

Tabel 4.14 tabel akun

Nama <i>Field</i>	Type
kode	varchar(6)
nama_akun	varchar(40)
sub_klasifikasi	tinyint(3)
saldo	bigint(11)

Tabel akun terdiri dari kode, nama\_akun, sub\_klasifikasi, kelompok\_akun\_id, dan saldo. Tabel akun dibuat sebagai representasi dari akun, yaitu untuk mencatat transaksi serta besarnya transaksi. *Field* sub\_klasifikasi merupakan *foreign key* yang diambil dari tabel sub\_klas\_akun..

d. Table jurnal

Nama Tabel : jurnal

*Primary Key* : id

*Foreign Key* : tipe

Tabel 4.15 tabel jurnal

Nama <i>Field</i>	Type
id	varchar(10)
tipe	tinyint(3) unsigned
tanggal	date
deskripsi	text

Tabel jurnal terdiri dari id, tipe, tanggal, dan deskripsi. Tabel ini akan menyimpan data transaksi secara umum. Detail dari transaksi tersebut akan dicatat pada tabel jurnal\_detail. *Field* tipe adalah *foreign key*

yang diambil dari tabel tipe\_jurnal. *Field* tipe bertipe *unsigned* agar nilai tidak berbentuk negatif.

e. Tabel jurnal detail

Nama Tabel : jurnal\_detail

*Primary Key* : id

*Foreign Key* : rekening\_id dan jurnal\_id

Tabel 4.16 tabel jurnal\_detail

Nama <i>Field</i>	Type
id	int(10)unsigned
jurnal_id	varchar(10)
item	varchar(30)
rekening_id	varchar(6)
debit_kredit	tinyint(1)
nilai	bigint(11)

Tabel jurnal\_detail adalah tabel yang digunakan untuk menyimpan data transaksi secara detail. Tabel ini menyimpan data rekening yang digunakan pada setiap transaksi beserta jumlahnya. *Field* dalam tabel ini yaitu id, jurnal\_id, item, rekening\_id, debit\_kredit, dan nilai. *Foreign key* rekening\_id berasal dari tabel rekening, sedangkan

jurnal\_id berasal dari tabel jurnal. *Field* id bertipe *unsigned* agar nilai tidak berbentuk negatif.

f. Tabel tipe jurnal

Nama Tabel : tipe\_jurnal

*Primary Key* : id

Tabel 4.17 tabel tipe jurnal

Nama <i>Field</i>	Type
id	tinyint(3)unsigned
nama	varchar(30)

Tabel tipe\_jurnal adalah tabel yang digunakan untuk mendefinisikan tipe jurnal. Dalam penelitian ini terdapat empat jurnal, yaitu jurnal umum, jurnal penyesuaian, jurnal penutup, dan jurnal koreksi.

g. Tabel *user*

Nama Tabel : *user*

*Primary Key* : username

Tabel 4.18 tabel tipe jurnal

Nama <i>Field</i>	Type
username	varchar(20)
password	varchar(40)
status	varchar(10)

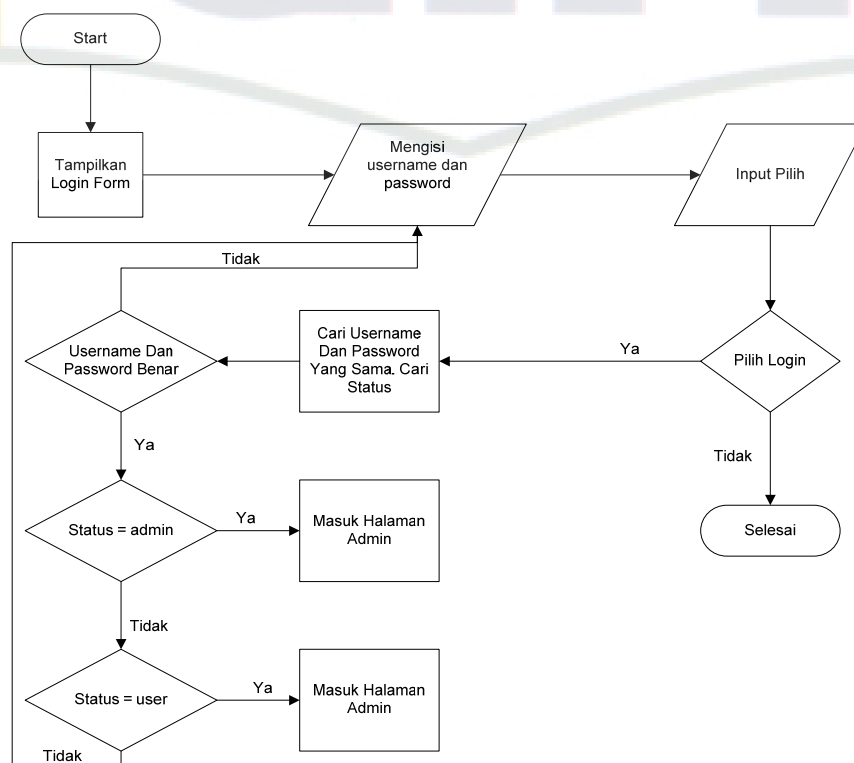


Tabel *user* adalah tabel yang digunakan untuk menyimpan data *user*. Disini terdapat dua tipe *user*, yaitu *admin* dan *user*. *Password* yang dimasukkan kedalam *database* menggunakan enkripsi md5. Status *user* ada dua, yaitu *admin* dan *user*.

#### 4.3.6 Flowchart

Program aplikasi akuntansi ini menggunakan *flowchart* untuk membantu dalam pembuatan program. Berikut ini merupakan gambar *flowchart* yang digunakan :

##### 1. Flowchart Login Admin

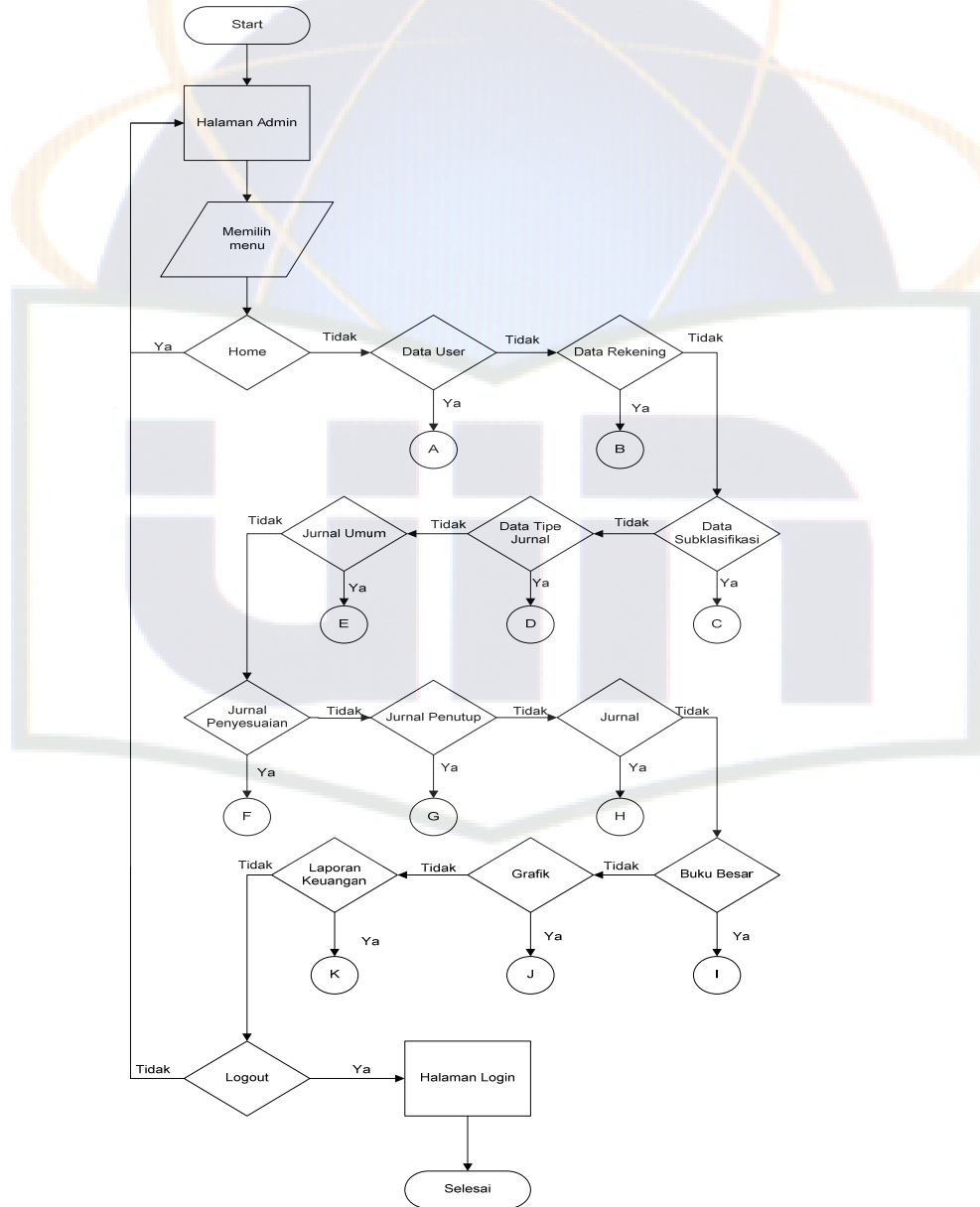


Gambar 4.25 Flowchart Halaman Login

Pertama kali masuk, *user* akan melihat halaman *login*. *User* harus memasukkan *username* dan *password* untuk bisa masuk. Bila *username* dan *password* benar, maka aplikasi akan mengecek status dari *user*. Bila status adalah *admin*, maka akan masuk ke halaman utama *admin*. Bila status adalah *user*, maka akan masuk ke halaman utama *user*.



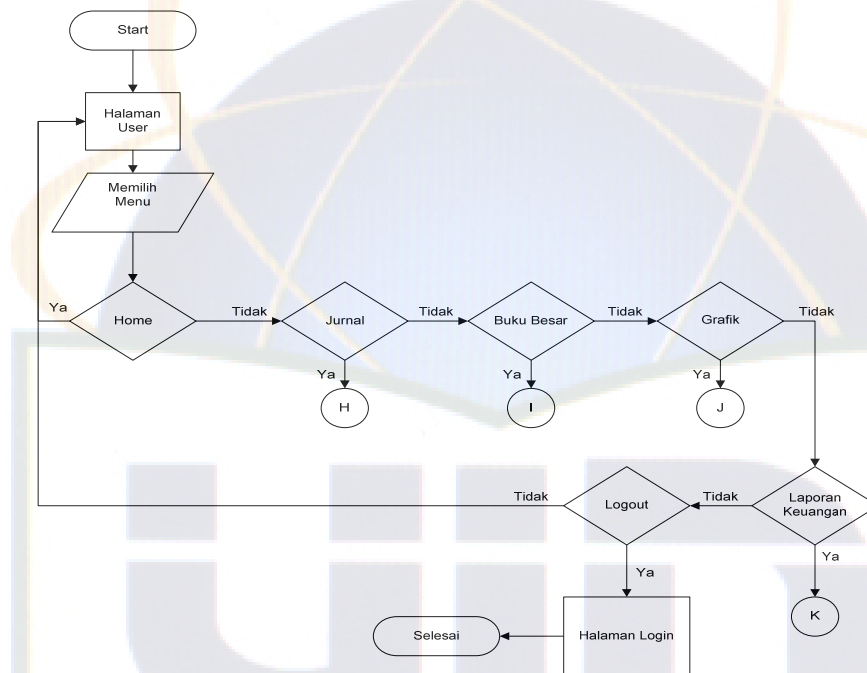
## 2. Flowchart Halaman Admin



Gambar 4.26 Flowchart Halaman Admin

Setelah *admin* masuk, maka *admin* akan melihat sembilan menu.

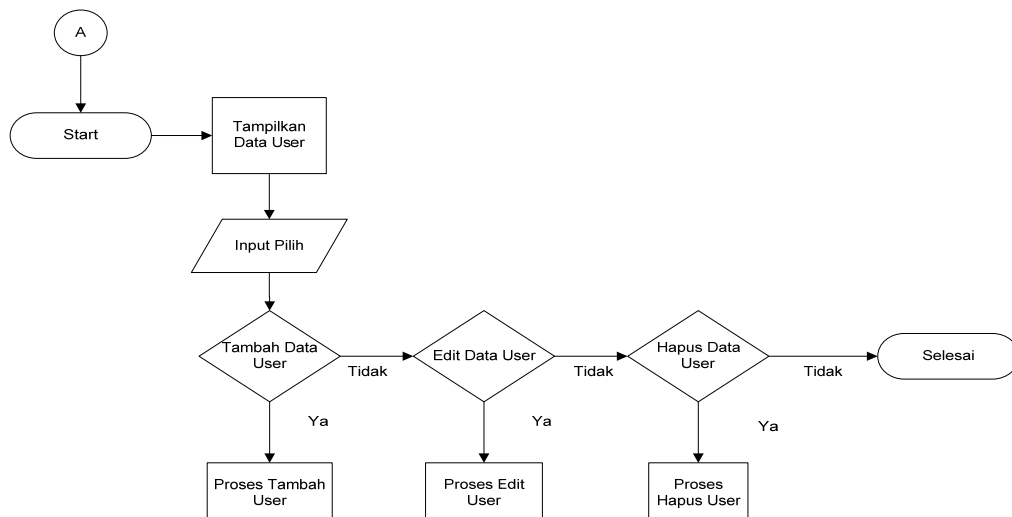
### 3. Flowchart Halaman User



Gambar 4.27 Flowchart Halaman User

Setelah *user* masuk, maka *user* akan melihat empat menu.

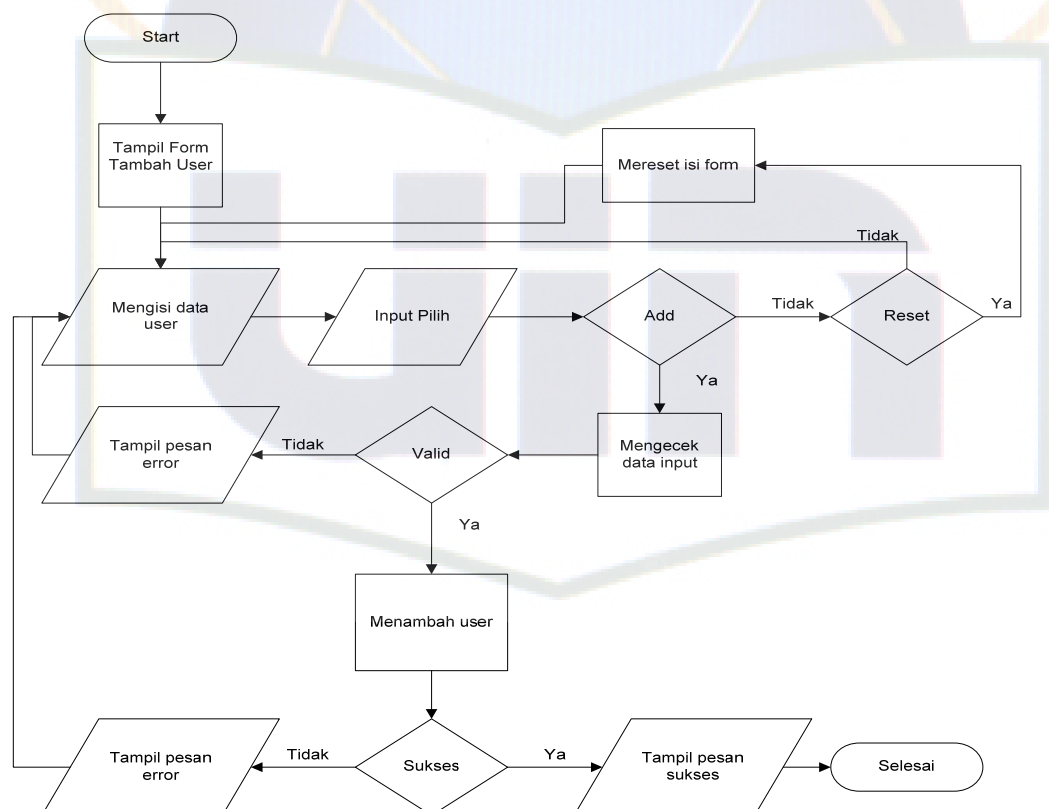
### 4. Flowchart Data User



Gambar 4.28 Flowchart Halaman Data User

Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan menampilkan data semua *user*. Kemudian *admin* bisa meng-*edit* data *user* ataupun menambahkan *user* baru. Berikut adalah detail dari proses – proses data *user* :

a. Proses tambah *user*



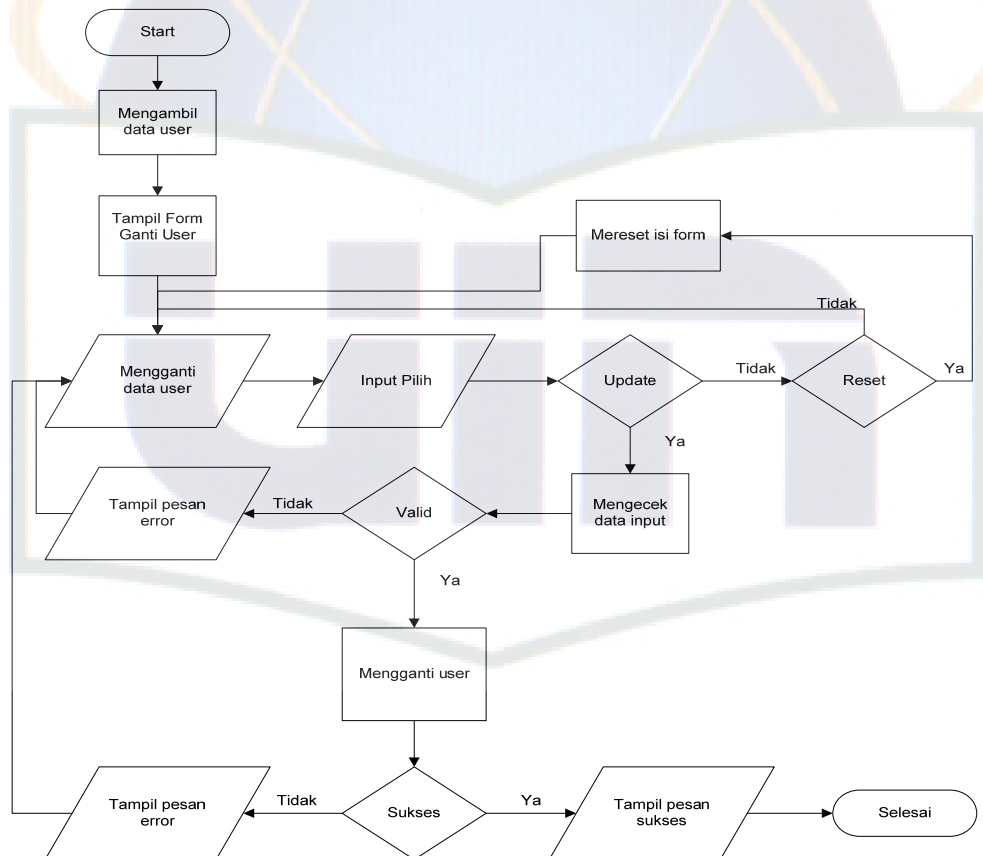
Gambar 4.29 Flowchart Tambah Data User

User harus mengisi *form* yang telah disediakan.

Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *add* atau *reset*. Ketika memilih *add*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses tambah gagal. Bila *input* benar, maka

aplikasi akan menambah *user*. Ketika *user* memilih *reset*, maka *form* isian akan kosong dan *user* akan mengisi *form* kembali.

b. Proses *edit user*

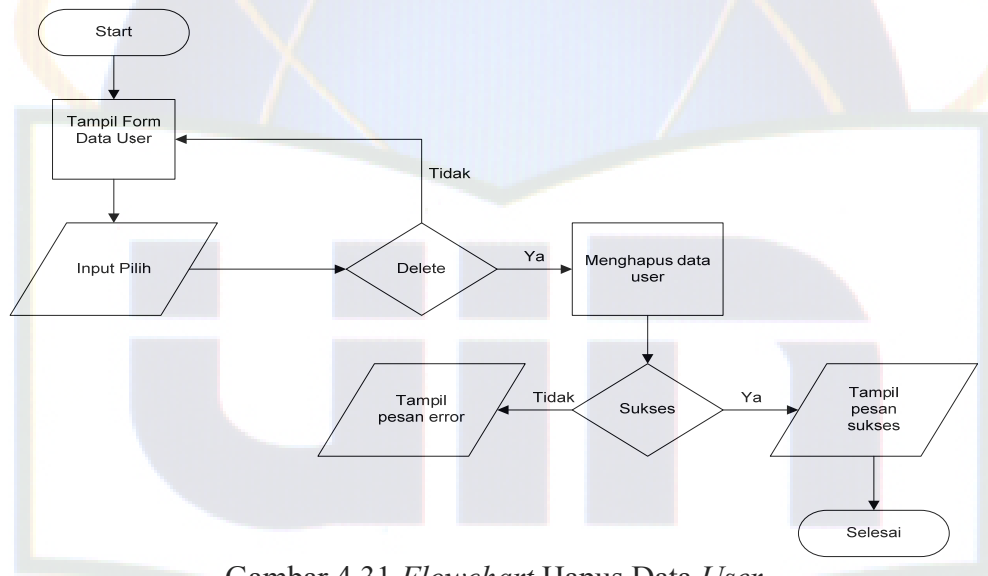


Gambar 4.30 Flowchart Edit Data User

User harus mengisi *form* yang telah disediakan yang berisi data *user*. Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *update* atau *reset*. Ketika memilih *update*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi menampilkan pesan *error* dan proses ganti

gagal. Bila *input* benar, maka aplikasi akan mengganti *user*. Ketika *user* memilih *reset*, maka *form* isian akan berisi data awal dan *user* akan mengisi *form* kembali.

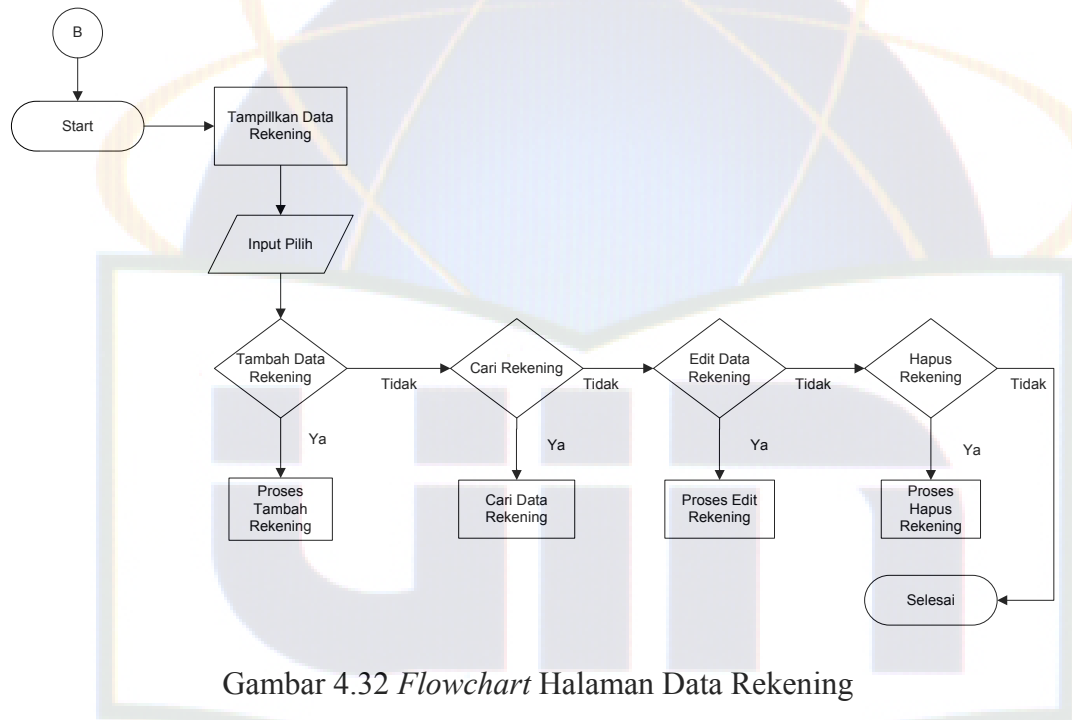
c. Proses hapus *user*



Gambar 4.31 *Flowchart* Hapus Data *User*

Ketika *user* memilih untuk menghapus data *user*, maka aplikasi akan langsung menghapusnya.

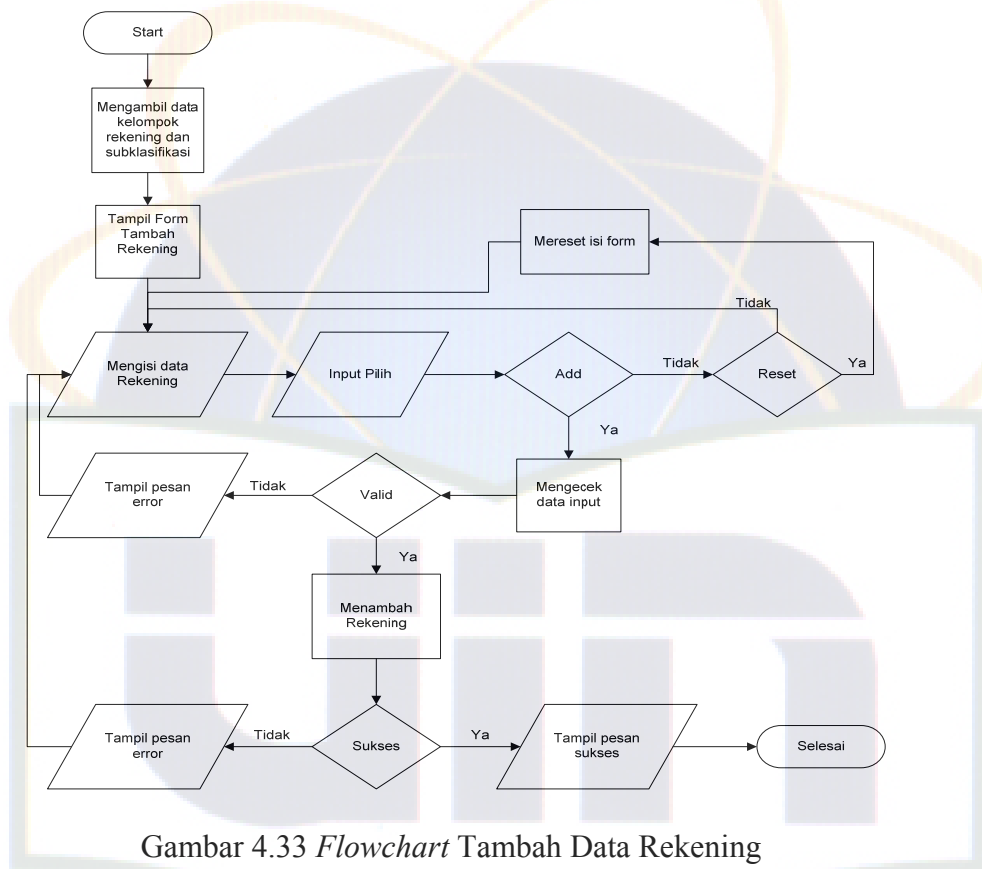
## 5. Flowchart Data Rekening



Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan menampilkan data semua rekening. Kemudian *admin* bisa meng-*edit* data rekening ataupun menambahkan rekening baru, serta mencari rekening. Berikut adalah detail dari proses – proses data rekening :

- a. Proses tambah data rekening

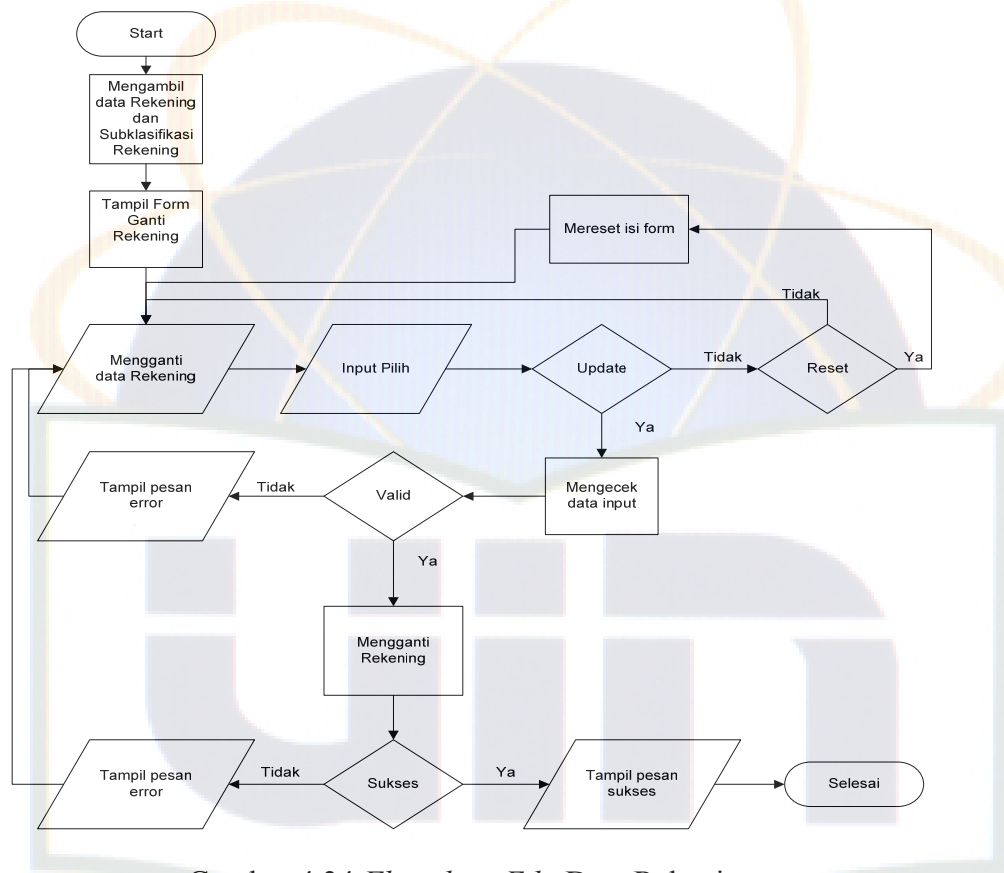




*User* harus mengisi *form* yang telah disediakan.

Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *add* atau *reset*. Ketika memilih *add*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses tambah gagal. Bila *input* benar, maka aplikasi akan menambah rekening. Ketika *user* memilih *reset*, maka *form* isian akan kosong dan *user* akan mengisi *form* kembali.

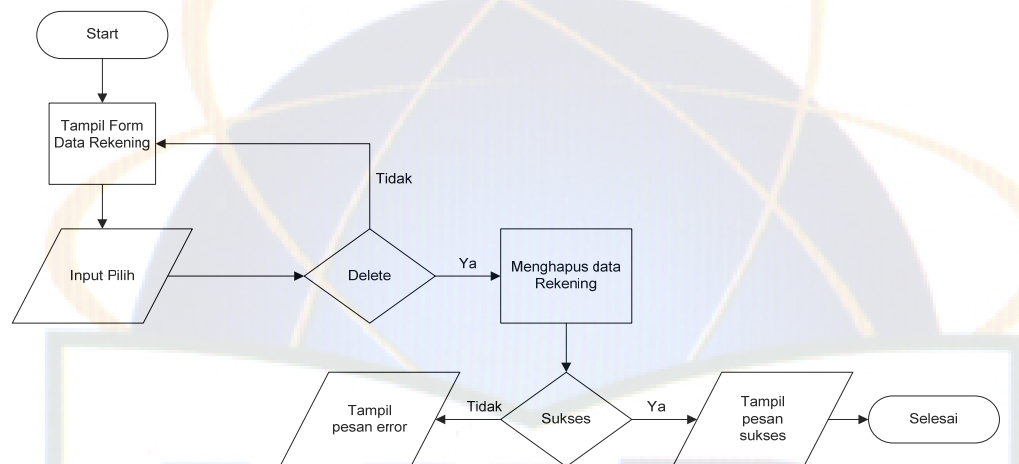
b. Proses *edit* data rekening



Gambar 4.34 Flowchart Edit Data Rekening

User harus mengisi *form* yang telah disediakan yang berisi data rekening. Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *update* atau *reset*. Ketika memilih *update*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses ganti gagal. Bila *input* benar, maka aplikasi akan mengganti rekening. Ketika *user* memilih *reset*, maka *form* isian akan berisi data awal dan *user* akan mengisi *form* kembali.

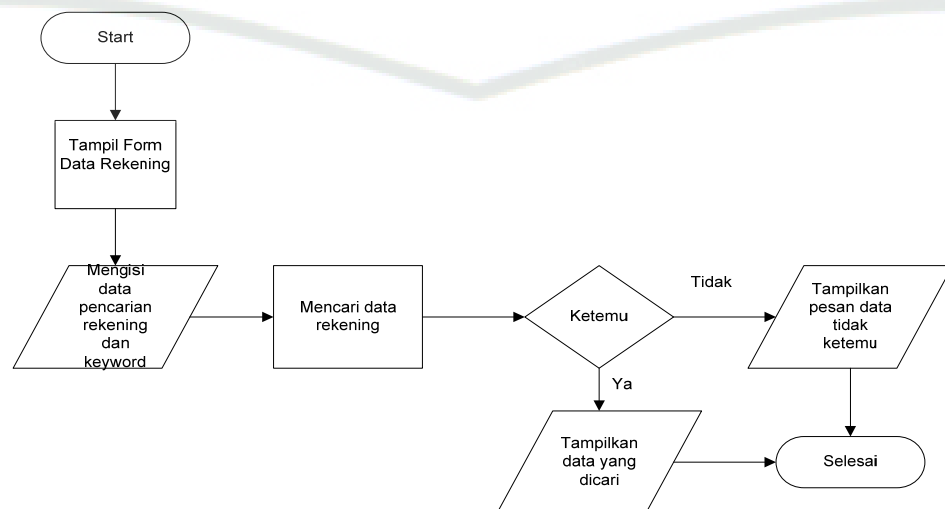
### c. Proses hapus data rekening



Gambar 4.35 *Flowchart* Hapus Data Rekening

Ketika *user* memilih untuk menghapus data rekening, maka aplikasi akan langsung menghapusnya.

### d. Proses cari data rekening

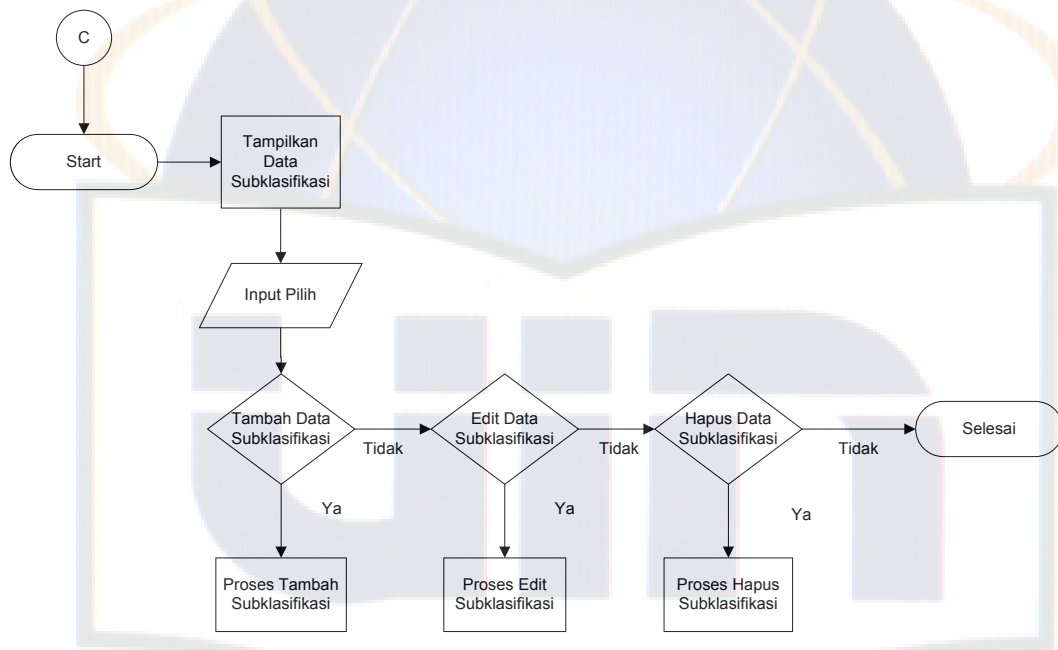


Gambar 4.36 *Flowchart* Cari Data Rekening

Pertama, *user* harus memasukkan *keyword* yang akan dicari beserta kolom mana yang akan digunakan. Bila data yang dicari ditemukan, maka aplikasi akan

menampilkan sejumlah data yang ditemukan. Bila tidak ditemukan, maka akan tampil pesan data kosong.

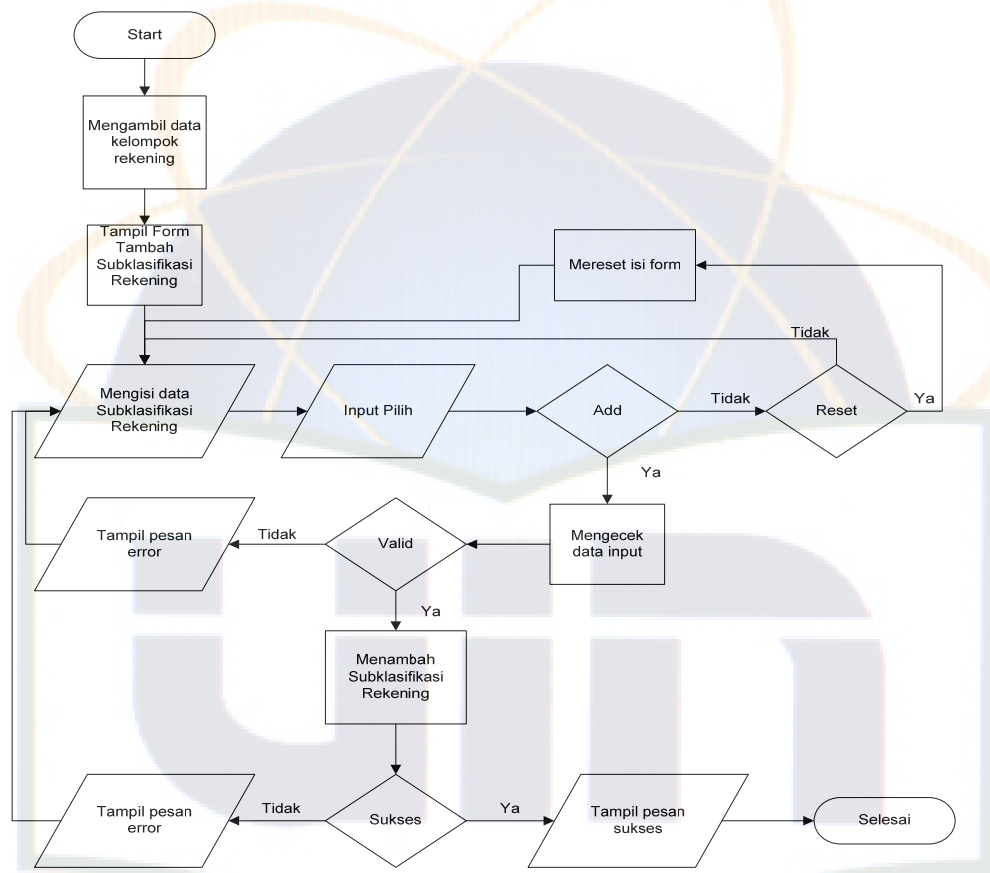
## 6. *Flowchart Data Subklasifikasi Akun*



Gambar 4.37 *Flowchart Data Subklasifikasi Rekening*

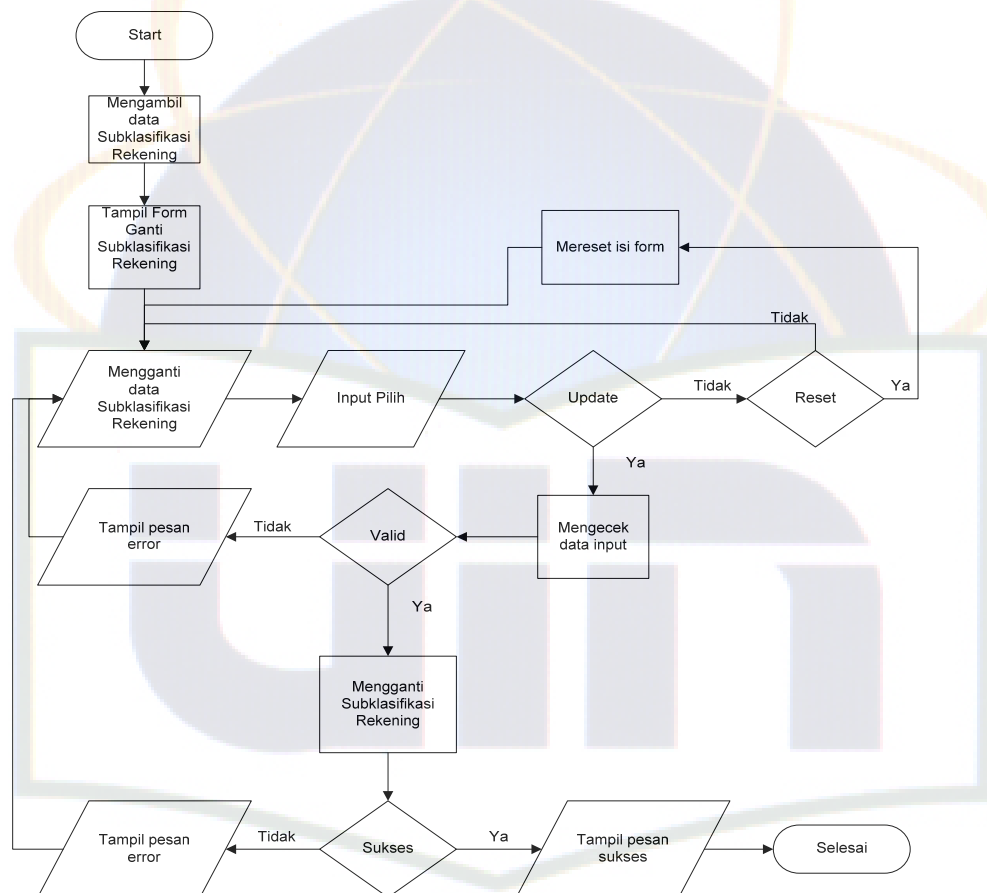
Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan menampilkan data semua subklasifikasi rekening. Kemudian *admin* bisa meng-*edit* data subklasifikasi rekening ataupun menambahkan data subklasifikasi rekening baru. Berikut adalah detail dari proses – proses data subklasifikasi rekening :

### a. Proses tambah data subklasifikasi rekening



Gambar 4.38 *Flowchart* Tambah Data Subklasifikasi Rekening

*User* harus mengisi *form* yang telah disediakan. Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *add* atau *reset*. Ketika memilih *add*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses tambah gagal. Bila *input* benar, maka aplikasi akan menambah subklasifikasi rekening. Ketika *user* memilih *reset*, maka *form* isian akan kosong dan *user* akan mengisi *form* kembali.

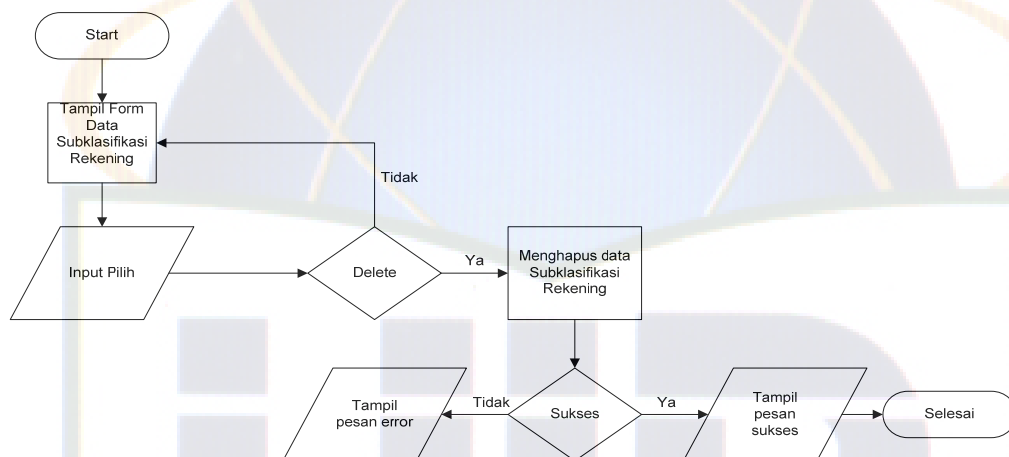
b. Proses *edit* subklasifikasi rekening

Gambar 4.39 Flowchart Edit Data Subklasifikasi Rekening

User harus mengisi *form* yang telah disediakan yang berisi data subklasifikasi rekening. Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *update* atau *reset*. Ketika memilih *update*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses ganti gagal. Bila *input* benar, maka aplikasi akan mengganti subklasifikasi rekening. Ketika *user* memilih

*reset*, maka *form* isian akan berisi data awal dan *user* akan mengisi *form* kembali.

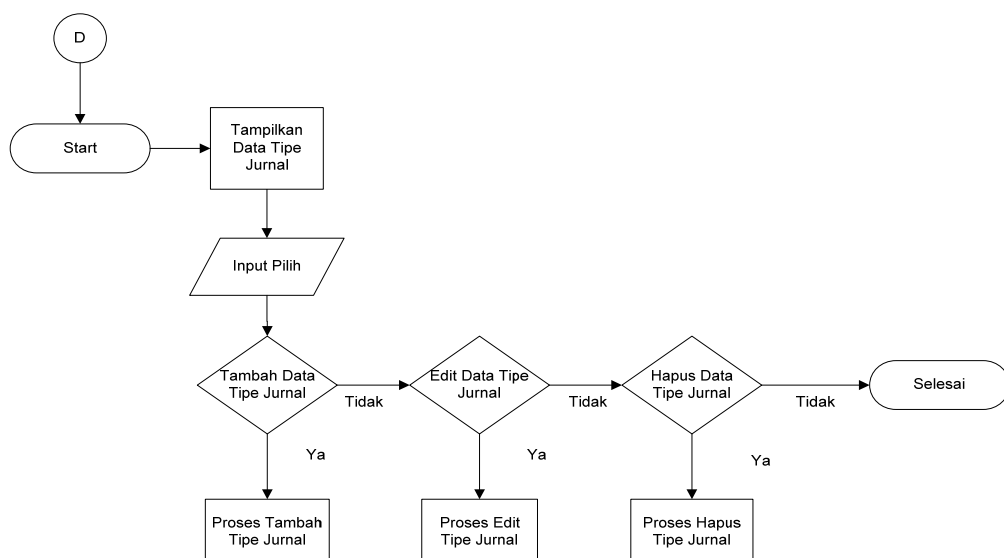
c. Proses hapus subklasifikasi rekening



Gambar 4.40 *Flowchart* Hapus Data Subklasifikasi Rekening

Ketika *user* memilih untuk menghapus data subklasifikasi rekening, maka aplikasi akan langsung menghapusnya.

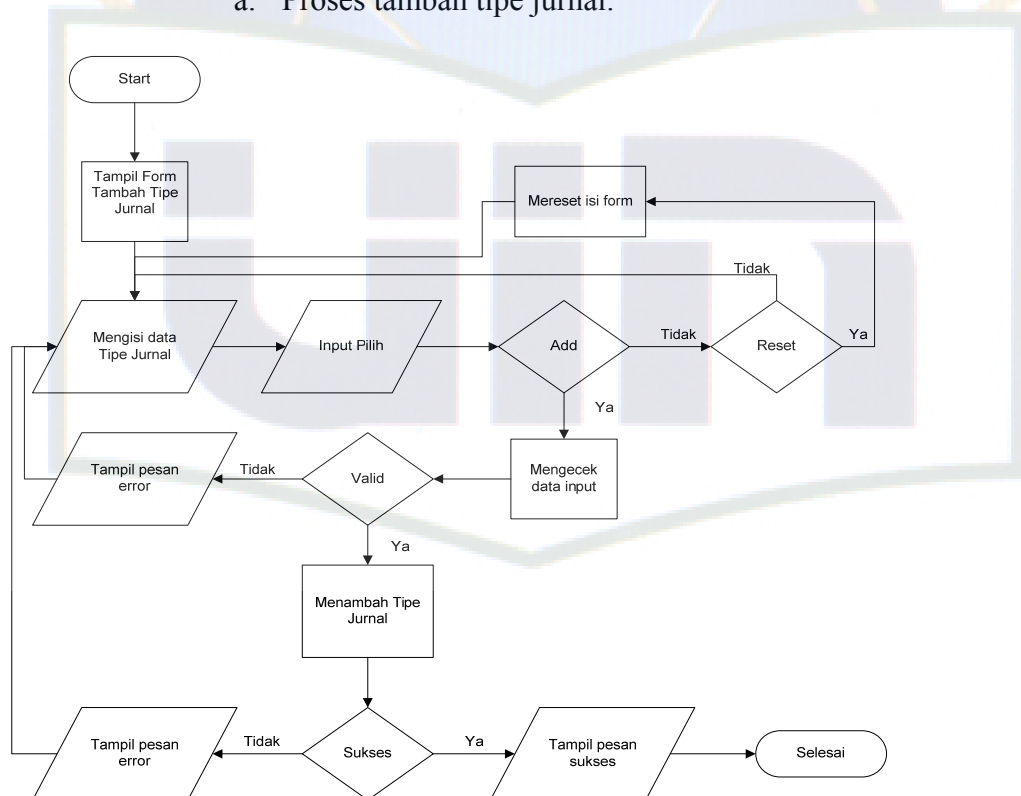
## 7. *Flowchart* Tipe Jurnal



Gambar 4.41 *Flowchart* Data Tipe Jurnal

Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan menampilkan data semua tipe jurnal. Kemudian *admin* bisa meng-*edit* data tipe jurnal ataupun menambahkan data tipe jurnal baru. Berikut adalah detail dari proses – proses data tipe jurnal :

a. Proses tambah tipe jurnal.



Gambar 4.42 *Flowchart* Tambah Tipe Jurnal

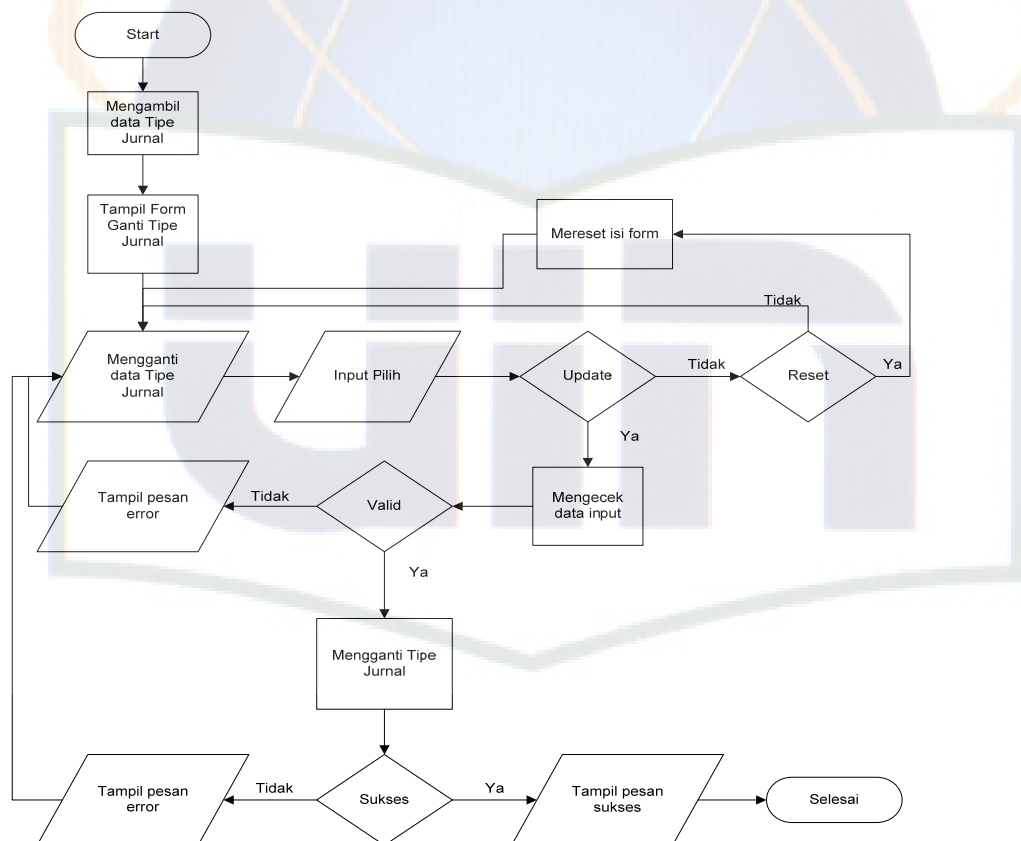
*User* harus mengisi *form* yang telah disediakan.

Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *add* atau *reset*. Ketika memilih *add*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses tambah gagal. Bila *input* benar, maka aplikasi



akan menambah data tipe jurnal. Ketika *user* memilih *reset*, maka *form* isian akan kosong dan *user* akan mengisi *form* kembali.

b. Proses *edit* tipe jurnal

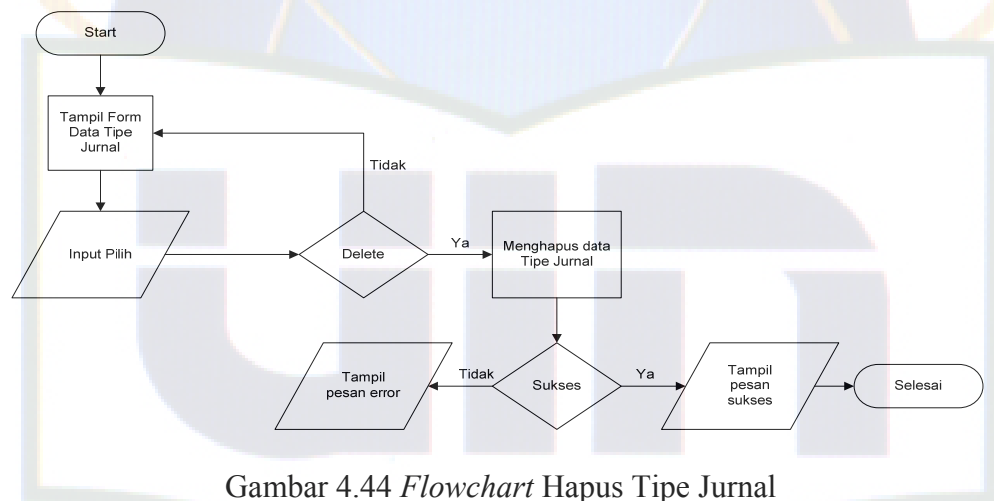


Gambar 4.43 Flowchart Edit Tipe Jurnal

*User* harus mengisi *form* yang telah disediakan yang berisi data tipe jurnal. Kemudian *user* akan dihadapkan pada dua pilihan, yaitu *update* atau *reset*. Ketika memilih *update*, maka aplikasi akan mengecek data yang akan dimasukkan. Bila data ada yang salah, maka aplikasi akan menampilkan pesan *error* dan proses

ganti gagal. Bila *input* benar, maka aplikasi akan mengganti data tipe jurnal. Ketika *user* memilih *reset*, maka *form* isian akan berisi data awal dan *user* akan mengisi *form* kembali.

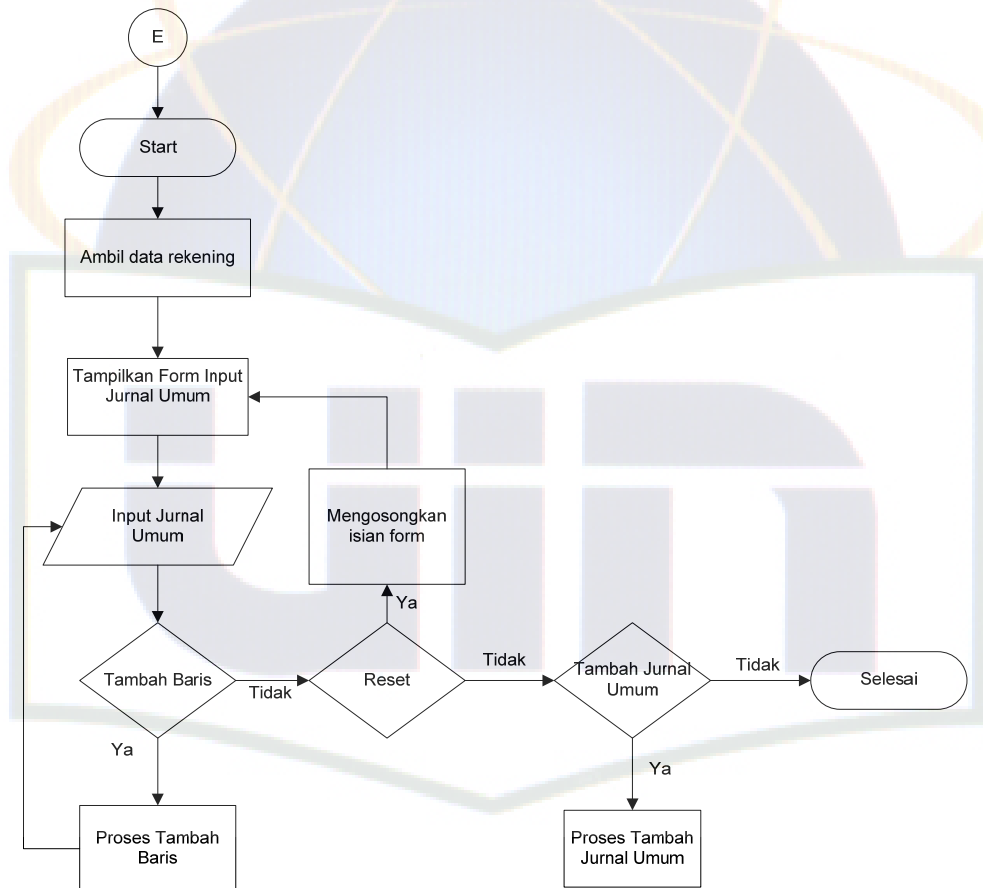
c. Proses hapus tipe jurnal



Gambar 4.44 Flowchart Hapus Tipe Jurnal

Ketika *user* memilih untuk menghapus data tipe jurnal, maka aplikasi akan langsung menghapusnya.

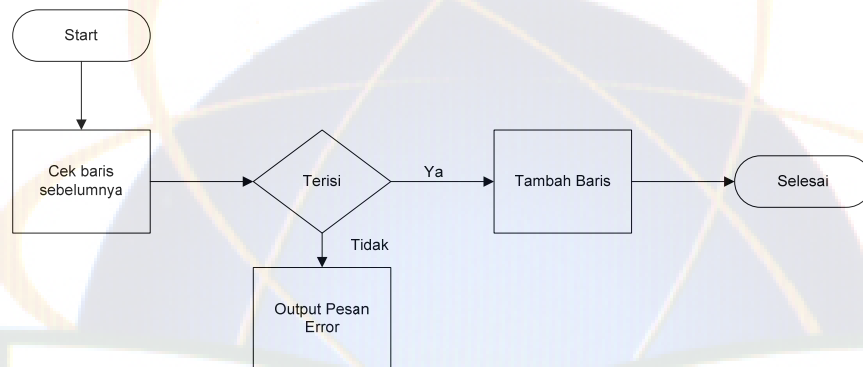
### 8. Flowchart Jurnal Umum



Gambar 4.45 Flowchart Halaman Jurnal Umum

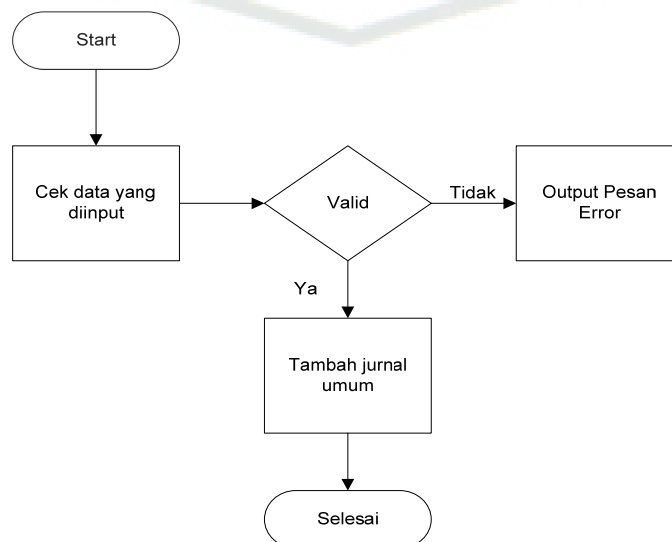
Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan mengambil data rekening – rekening yang ada. Kemudian aplikasi menampilkan *form* untuk mengisi jurnal umum. Disini terdapat dua subproses, yaitu :

## a. Tambah baris

Gambar 4.46 *Flowchart* Tambah Baris

Aplikasi akan mengecek, apakah baris sebelumnya telah diisi semua oleh data. Bila belum, maka tidak bisa menambahkan baris. Bila benar, maka baris akan bertambah

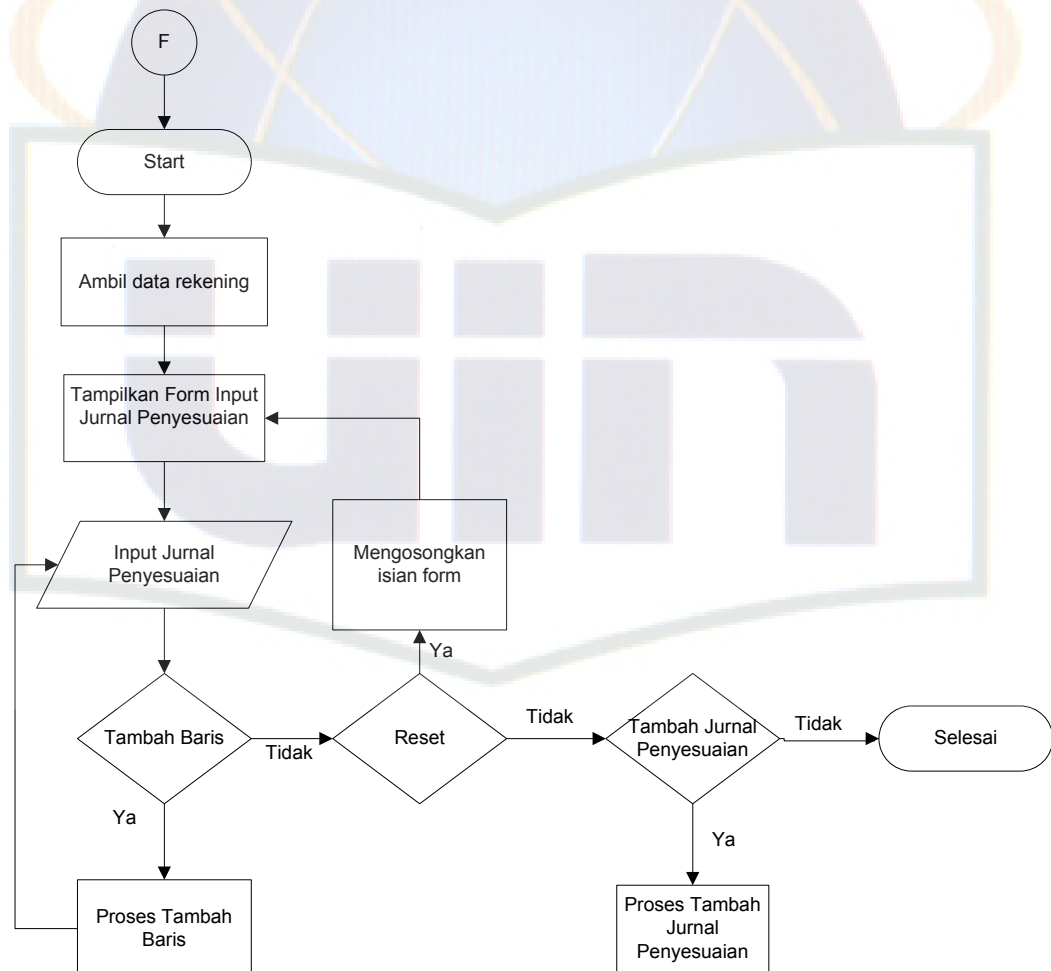
## b. Tambah jurnal

Gambar 4.47 *Flowchart* Tambah Jurnal Umum

Aplikasi akan mengecek, apakah *input* telah sesuai.

Bila sesuai, maka aplikasi akan menambahkan jurnal umum. Bila salah, maka akan tampil pesan *error*.

### 9. Flowchart Jurnal Penyesuaian

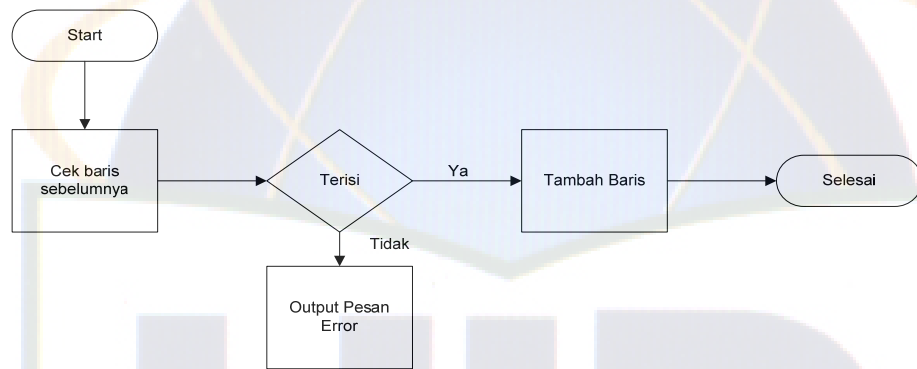


Gambar 4.48 Flowchart Halaman Jurnal Penyesuaian

Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan mengambil data rekening – rekening yang ada. Kemudian aplikasi menampilkan *form* untuk menambahkan

jurnal penyesuaian. Dalam proses ini terdapat dua subproses, yaitu :

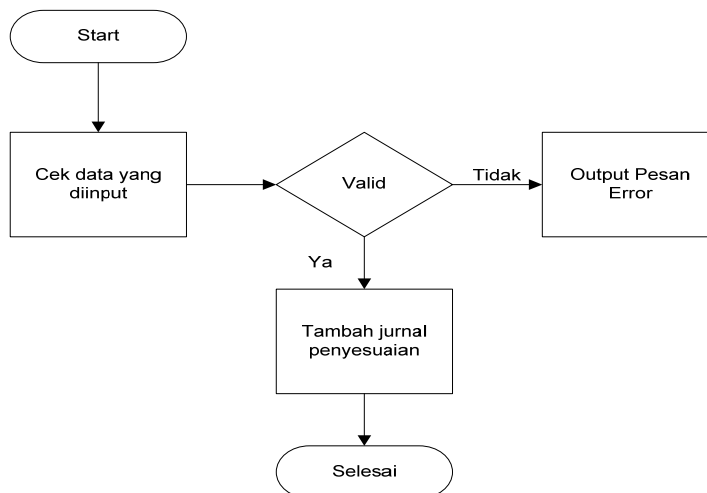
a. Tambah baris



Gambar 4.49 *Flowchart* Tambah Baris

Aplikasi akan mengecek, apakah baris sebelumnya telah diisi semua oleh data. Bila belum, maka tidak bisa menambahkan baris. Bila benar, maka baris akan bertambah

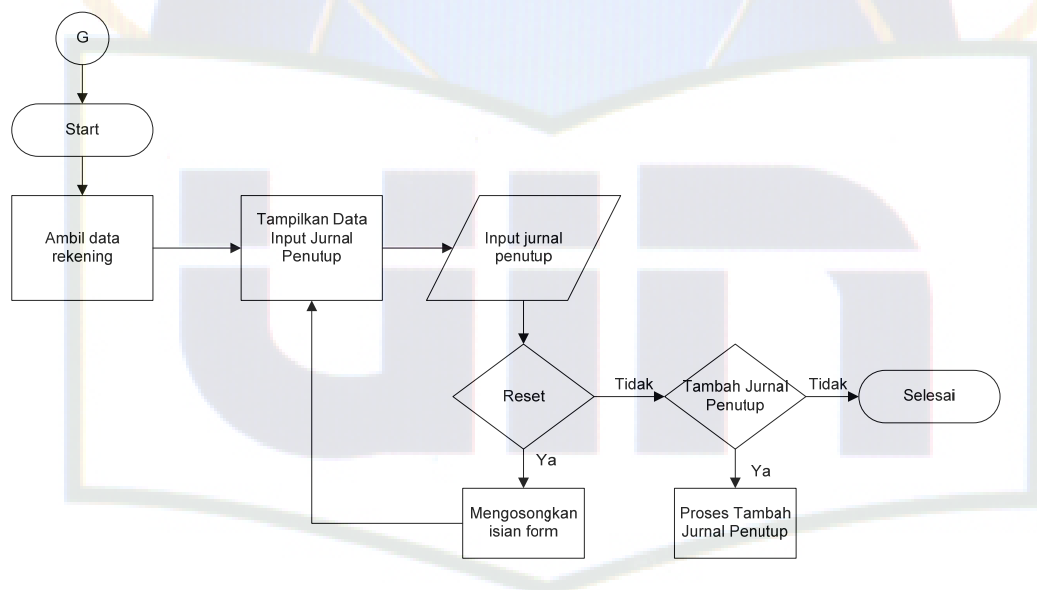
b. Tambah jurnal penyesuaian



Gambar 4.50 *Flowchart* Tambah Jurnal Penyesuaian

Aplikasi akan mengecek, apakah *input* telah sesuai. Bila sesuai, maka aplikasi akan menambahkan jurnal penyesuaian. Bila salah, maka akan tampil pesan *error*.

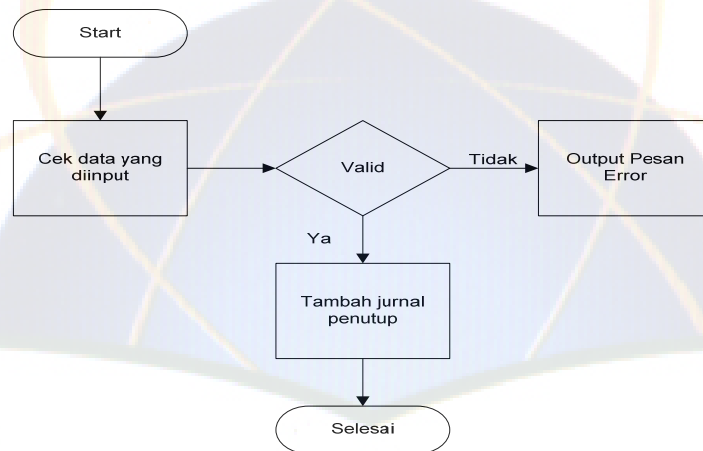
#### 10. Flowchart Jurnal Penutup



Gambar 4.51 Flowchart Halaman Jurnal Penutup

Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi akan mengambil data rekening – rekening bertipe pendapatan dan beban, kemudian dilakukan proses untuk menolkan rekening tersebut. Kemudian aplikasi menampilkan *form* untuk menambahkan jurnal penutup. Disini terdapat sebuah subproses:

### a. Tambah jurnal penutup

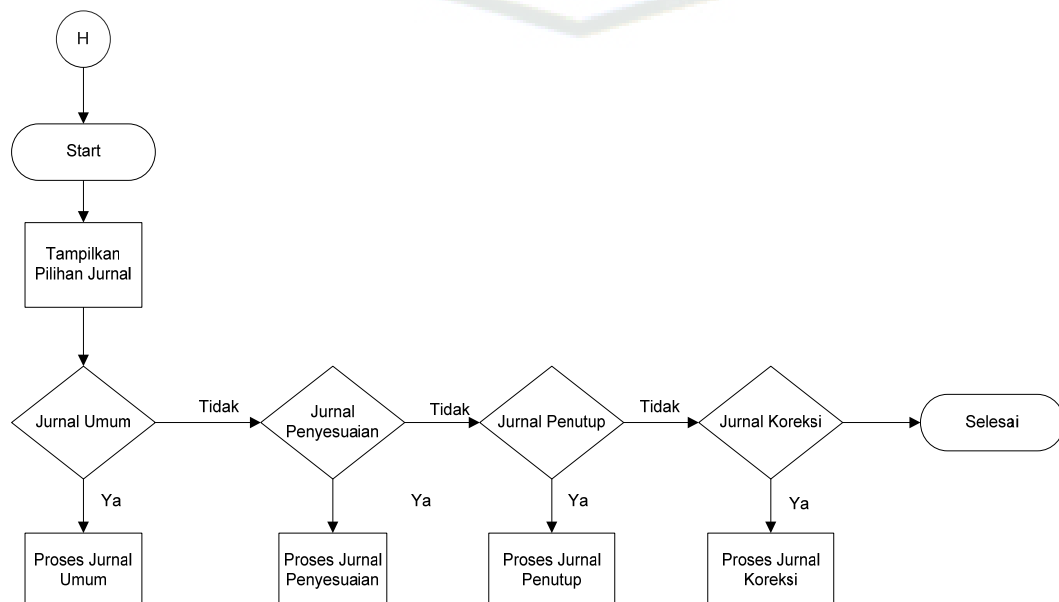


Gambar 4.52 *Flowchart* Proses Jurnal Penutup

Aplikasi akan mengecek, apakah *input* telah sesuai.

Bila sesuai, maka aplikasi akan menambahkan jurnal penutup. Bila salah, maka akan tampil pesan *error*.

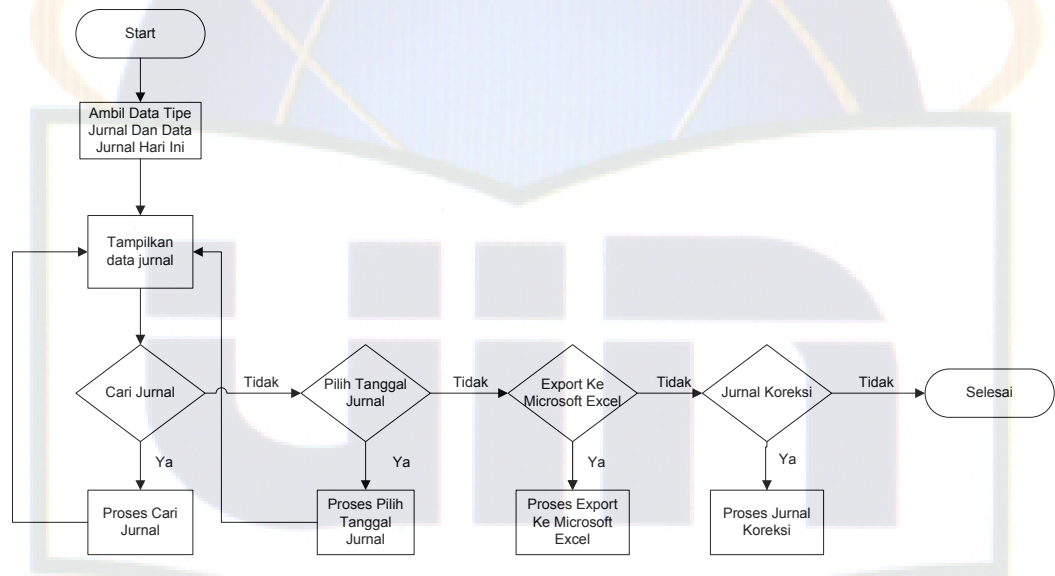
## 11. *Flowchart* Data Jurnal



Gambar 4.53 *Flowchart* Halaman Data Jurnal

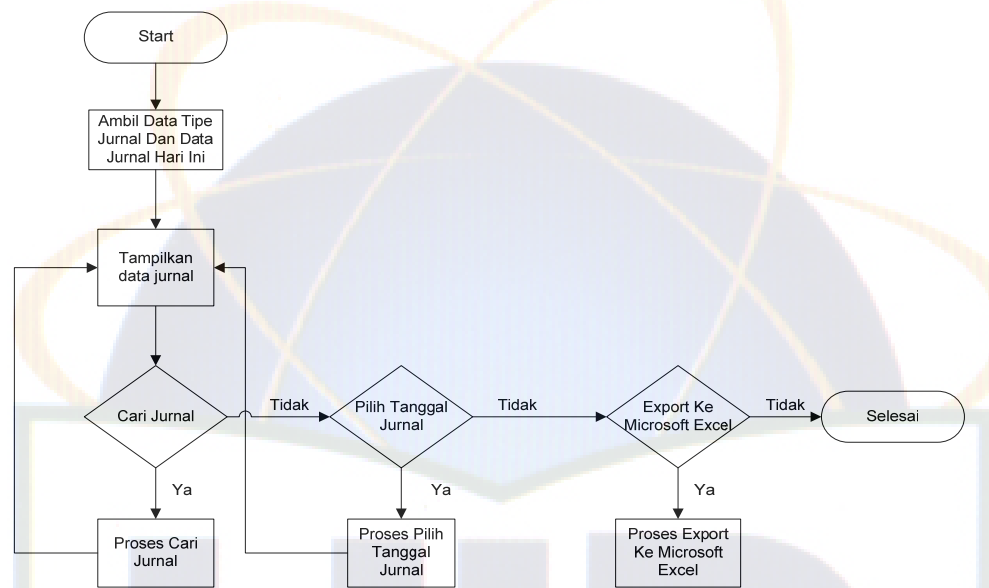


Halaman ini bisa diakses oleh *admin* dan *user*. Pertama, aplikasi akan mengambil data jurnal yang terdapat dalam tabel *tipe\_jurnal*. Kemudian *user* dapat memilih jurnal yang ingin dilihat.



Gambar 4.54 Flowchart Halaman Data Jurnal Untuk *admin*

Gambar diatas adalah *flowchart* melihat data jurnal untuk *user* yang berstatus *admin*. Pertama, aplikasi akan mengambil data tipe jurnal dari proses sebelumnya. Setelah itu, aplikasi akan menampilkan data jurnal pada tanggal yang sama dengan tanggal di *server*. Di halaman ini terdapat menu untuk mencari jurnal atau menampilkan jurnal pada hari tertentu. Jurnal ini bisa diekspor ke dalam Microsoft Excel. Perbedaan antara *user* dan *admin* adalah *admin* dapat memasukkan jurnal koreksi, sedangkan *user* tidak bisa.

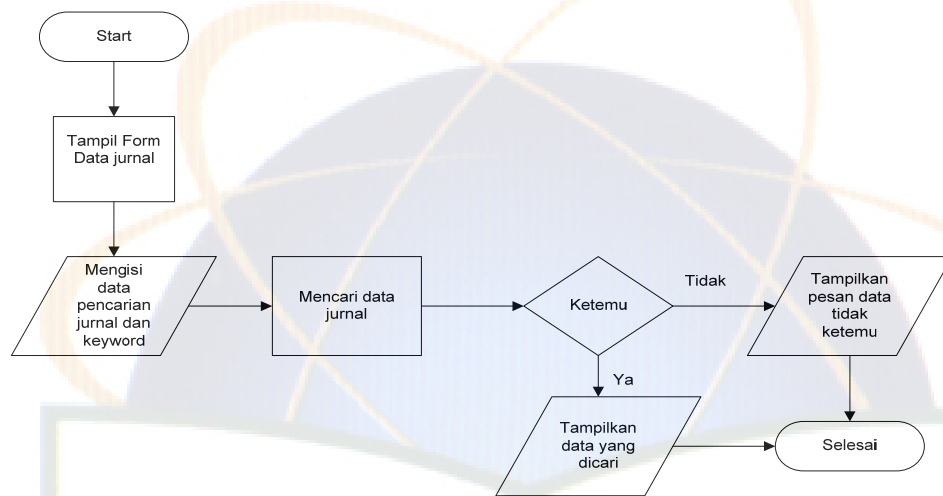


Gambar 4.55 *Flowchart* Halaman Data Jurnal Untuk *user*

Gambar diatas adalah *flowchart* melihat data jurnal untuk *user* yang berstatus *user*. Pertama, aplikasi akan mengambil data tipe jurnal dari proses sebelumnya. Setelah itu, aplikasi akan menampilkan data jurnal pada tanggal yang sama dengan tanggal di *server*. Di halaman ini terdapat menu untuk mencari jurnal atau menampilkan jurnal pada hari tertentu. Jurnal ini bisa diekspor ke dalam Microsoft Excel. Untuk *user* yang berstatus *user*, maka *user* tidak bisa memasukkan jurnal koreksi.

Dari *flowchart*, proses – proses tersebut dapat didetailkan seperti berikut :

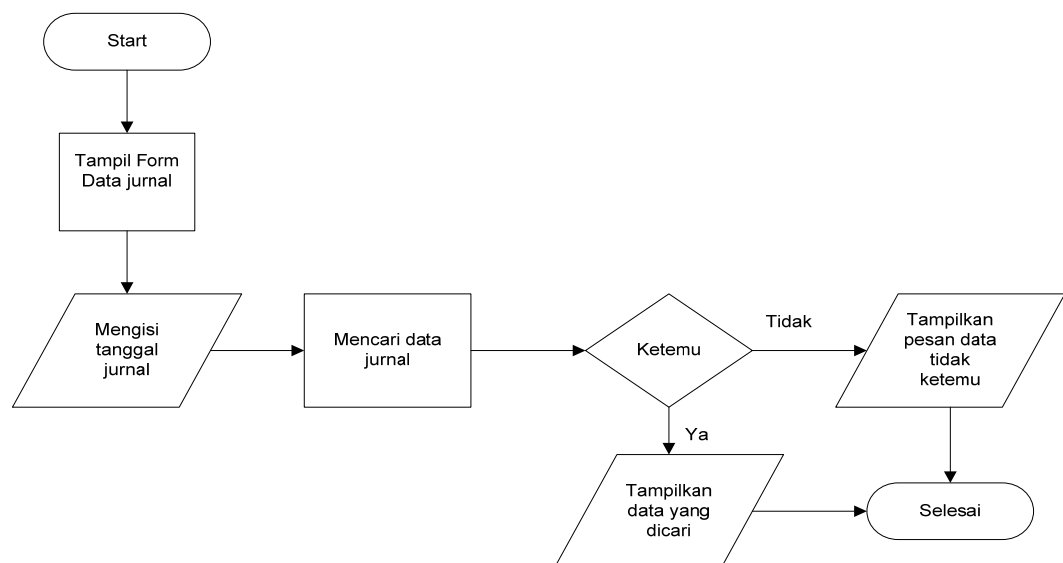
- a. Cari jurnal



Gambar 4.56 *Flowchart* Halaman Cari Data Jurnal

Pertama, akan ada tampilan data jurnal pada hari ini. Kemudian *user* mengisi *keyword* yang ingin dicari beserta kolomnya. Bila data ditemukan, maka data akan ditampilkan. Bila data tidak ditemukan, maka akan ada pesan data tidak ada.

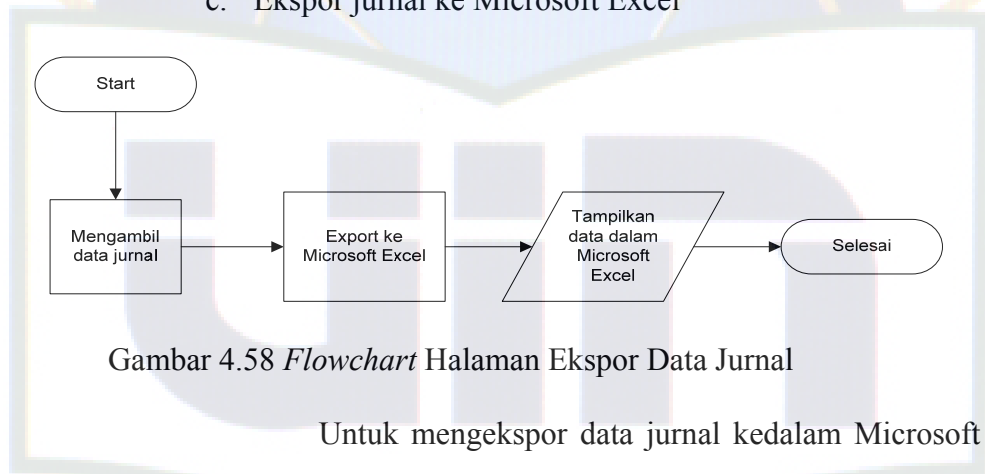
b. Pilih tanggal jurnal



Gambar 4.57 *Flowchart* Halaman Data Jurnal Berdasarkan Tanggal

Pertama, akan ada tampilan data jurnal pada hari ini. Kemudian *user* mengisi tanggal yang ingin dicari.. Bila data ditemukan, maka data akan ditampilkan. Bila data tidak ditemukan, maka akan ada pesan data tidak ada.

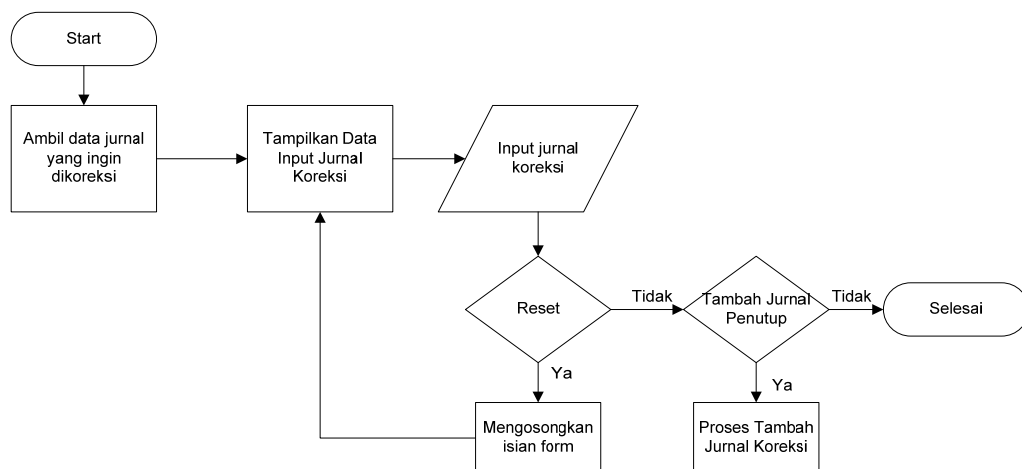
#### c. Ekspor jurnal ke Microsoft Excel



Gambar 4.58 *Flowchart* Halaman Ekspor Data Jurnal

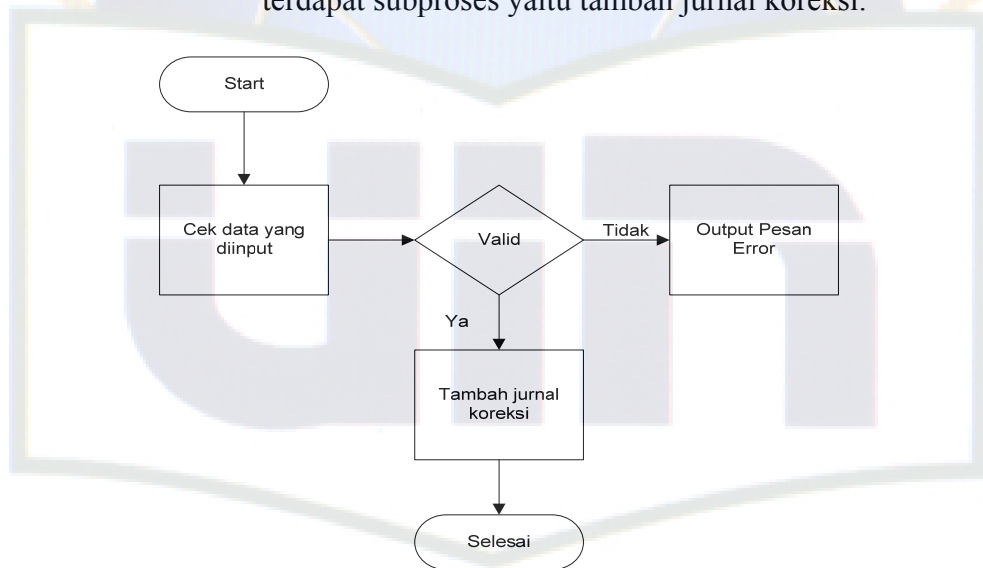
Untuk mengekspor data jurnal kedalam Microsoft Excel, maka aplikasi akan mengambil data yang telah ditampilkan. Data itu kemudian akan diekspor Microsoft Excel.

#### d. Jurnal Koreksi



Gambar 4.59 *Flowchart* Halaman Jurnal Koreksi

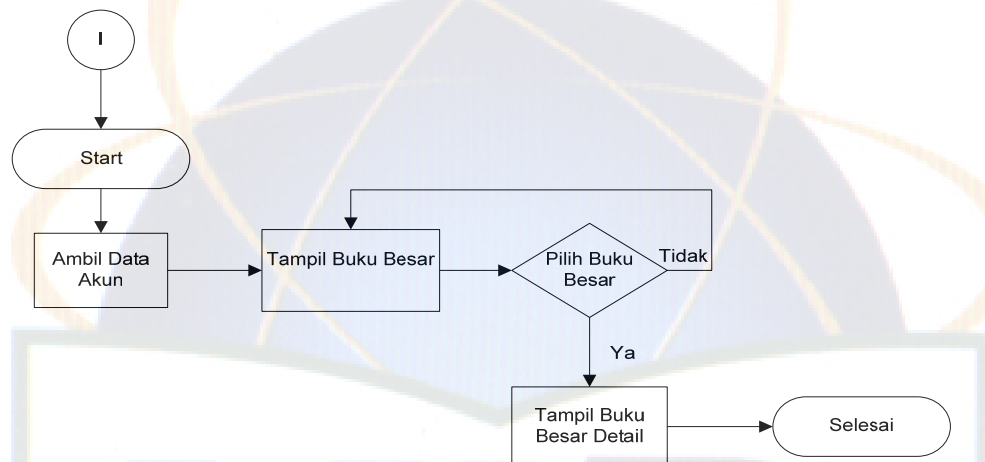
Halaman ini hanya bisa diakses oleh *admin*. Pertama, aplikasi menampilkan *form* untuk menambahkan jurnal koreksi, beserta data transaksi jurnal koreksi tersebut. *User* harus memasukkan tanggal jurnal, ID jurnal, dan deskripsi jurnal. Disini terdapat subproses yaitu tambah jurnal koreksi.



Gambar 4.60 *Flowchart* Halaman Jurnal Koreksi

Aplikasi akan mengecek, apakah *input* telah sesuai. Bila sesuai, maka aplikasi akan menambahkan jurnal koreksi. Bila salah, maka akan tampil pesan *error*.

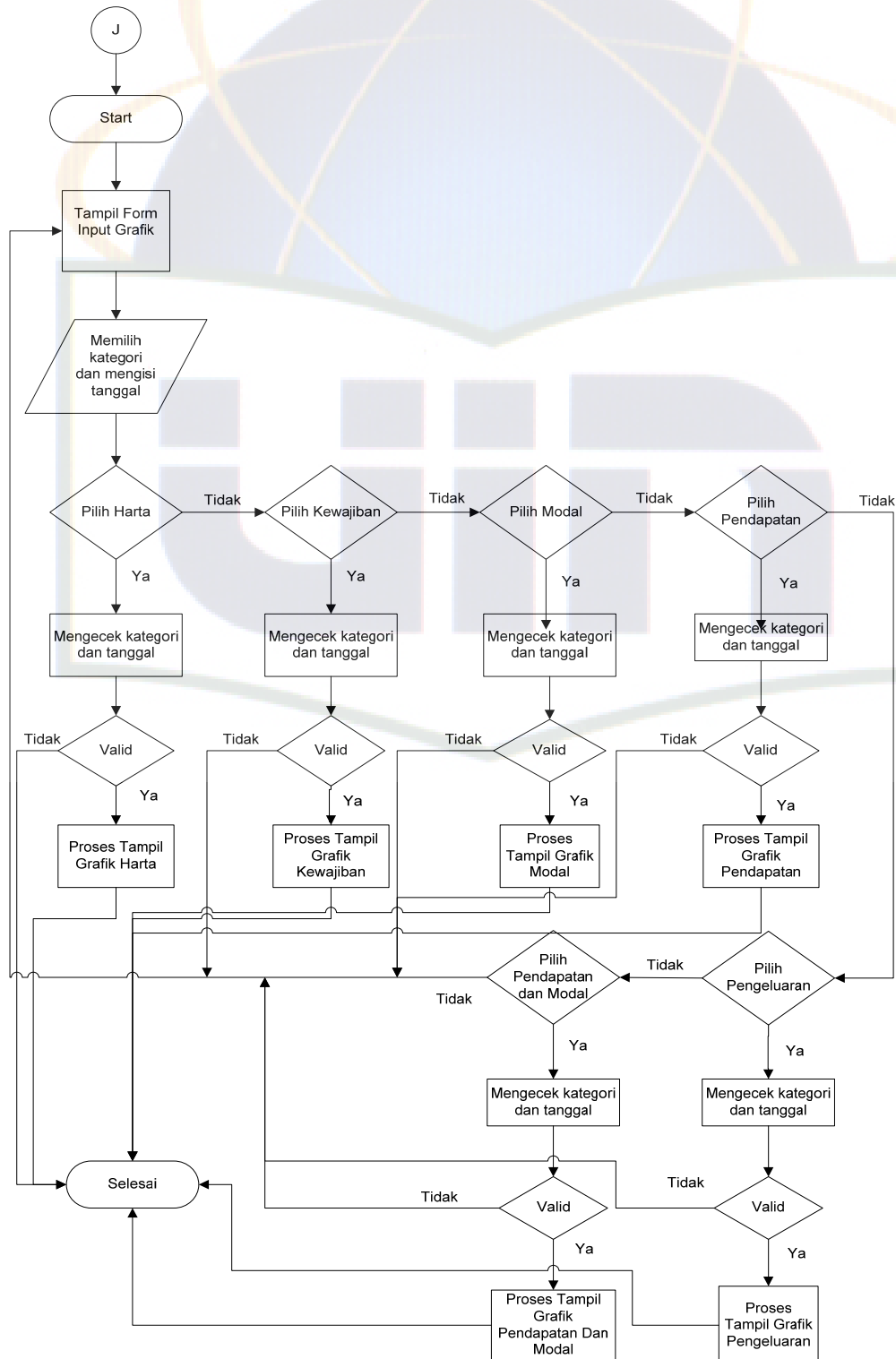
## 12. Flowchart Buku Besar



Gambar 4.61 Flowchart Halaman Data Buku Besar

Halaman ini bisa diakses oleh *admin* dan *user*. Pertama, aplikasi akan mengambil data rekening. Ketika *user* memilih buku besar detail, maka aplikasi akan mencari data rekening di dalam jurnal, kemudian menampilkannya.

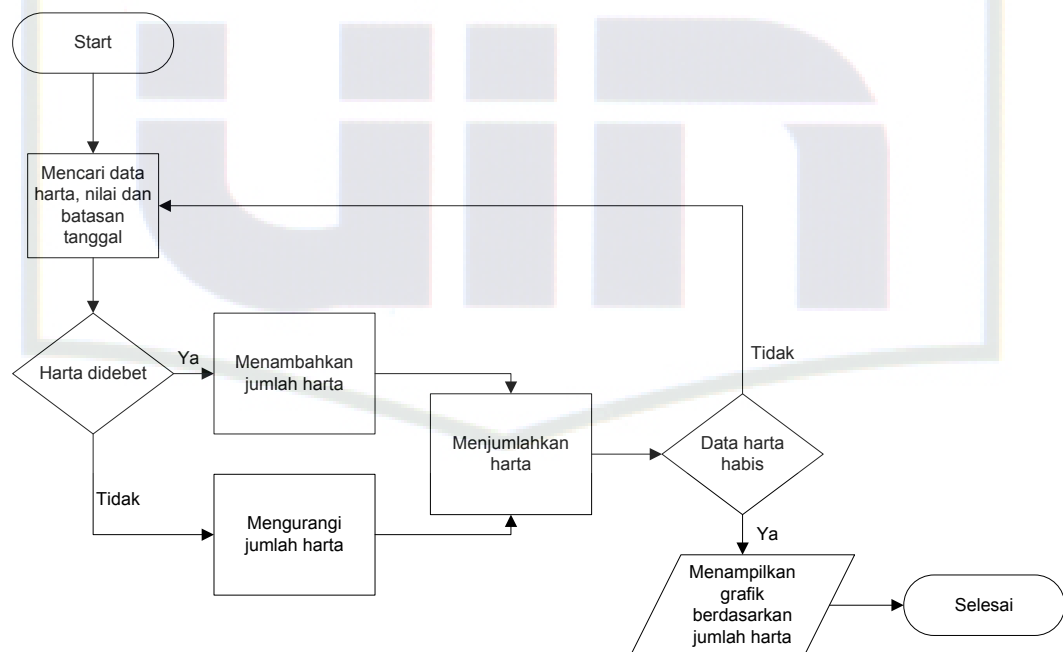
### 13. Flowchart Grafik



Gambar 4.62 Flowchart Halaman Grafik

Halaman ini bisa diakses oleh *admin* dan *user*. Pertama, aplikasi akan menampilkan *form input*. Ketika *user* telah meng-*input*, maka aplikasi akan mencari data tentang rekening tersebut, kemudian diproses untuk ditampilkan dalam bentuk diagram batang. Proses ini dibagi menjadi beberapa subproses, yaitu :

a. Grafik harta

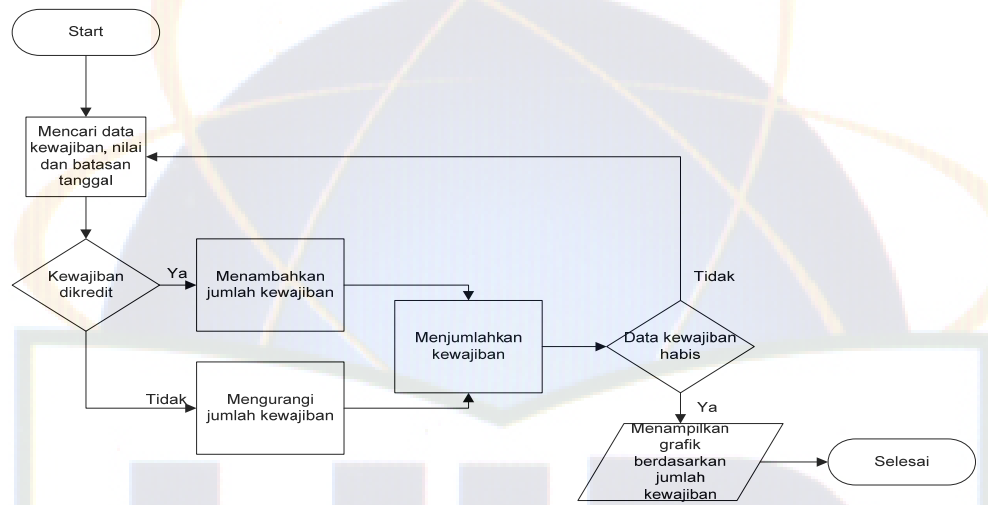


Gambar 4.63 *Flowchart* Halaman Grafik Harta

Aplikasi akan mencari jurnal yang memiliki rekening harta. Bila nilai harta adalah debit, maka akan ditambah. Bila kredit, maka akan dikurangi. Setelah itu, harta dijumlahkan untuk dibuat diagram batangnya.

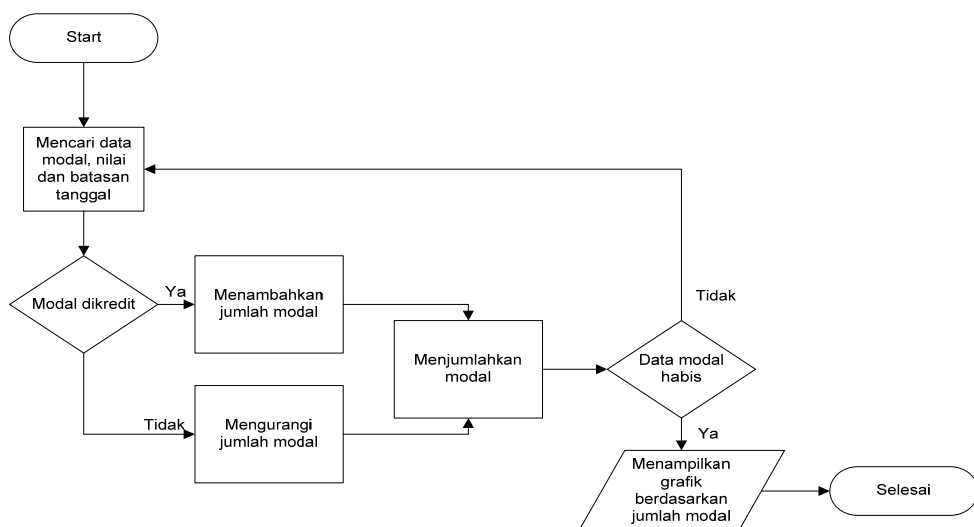


## b. Grafik kewajiban

Gambar 4.64 *Flowchart* Halaman Grafik Kewajiban

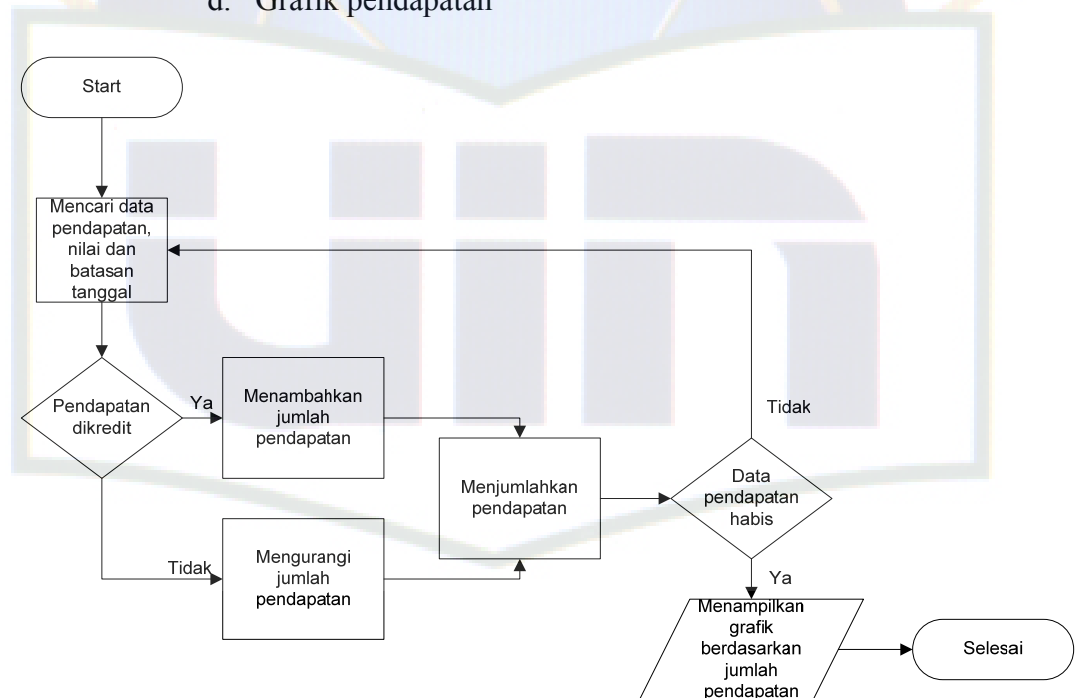
Aplikasi akan mencari jurnal yang memiliki rekening kewajiban. Bila nilai kewajiban adalah kredit, maka akan ditambah. Bila debit, maka akan dikurangi. Setelah itu, kewajiban dijumlahkan untuk dibuat diagram batangnya.

## c. Grafik modal

Gambar 4.65 *Flowchart* Halaman Grafik Modal

Aplikasi akan mencari jurnal yang memiliki rekening modal. Bila nilai modal adalah kredit, maka akan ditambah. Bila debit, maka akan dikurangi. Setelah itu, modal dijumlahkan untuk dibuat diagram batangnya.

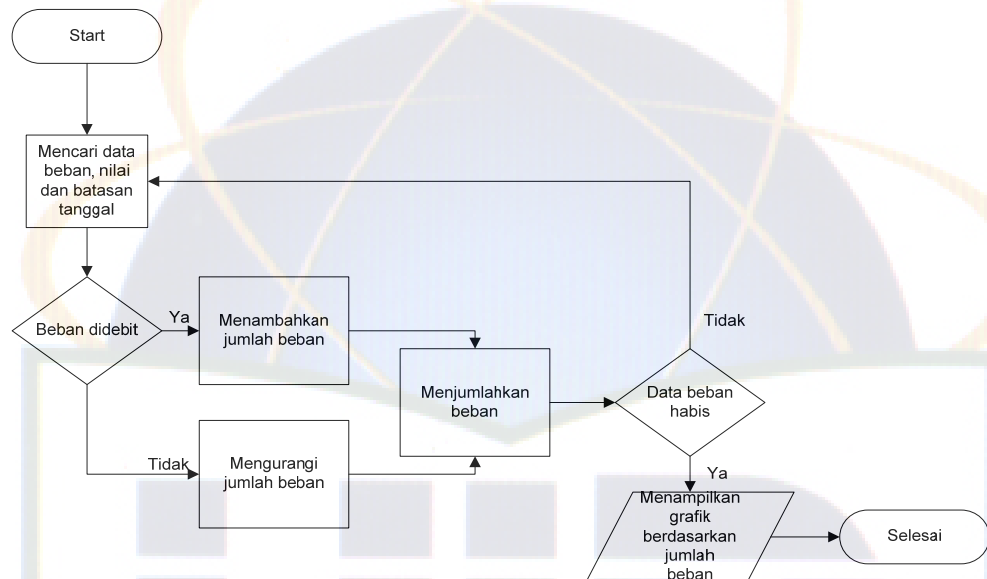
d. Grafik pendapatan



Gambar 4.66 *Flowchart* Halaman Grafik Pendapatan

Aplikasi akan mencari jurnal yang memiliki rekening pendapatan. Bila nilai pendapatan adalah kredit, maka akan ditambah. Bila debit, maka akan dikurangi. Setelah itu, pendapatan dijumlahkan untuk dibuat diagram batangnya.

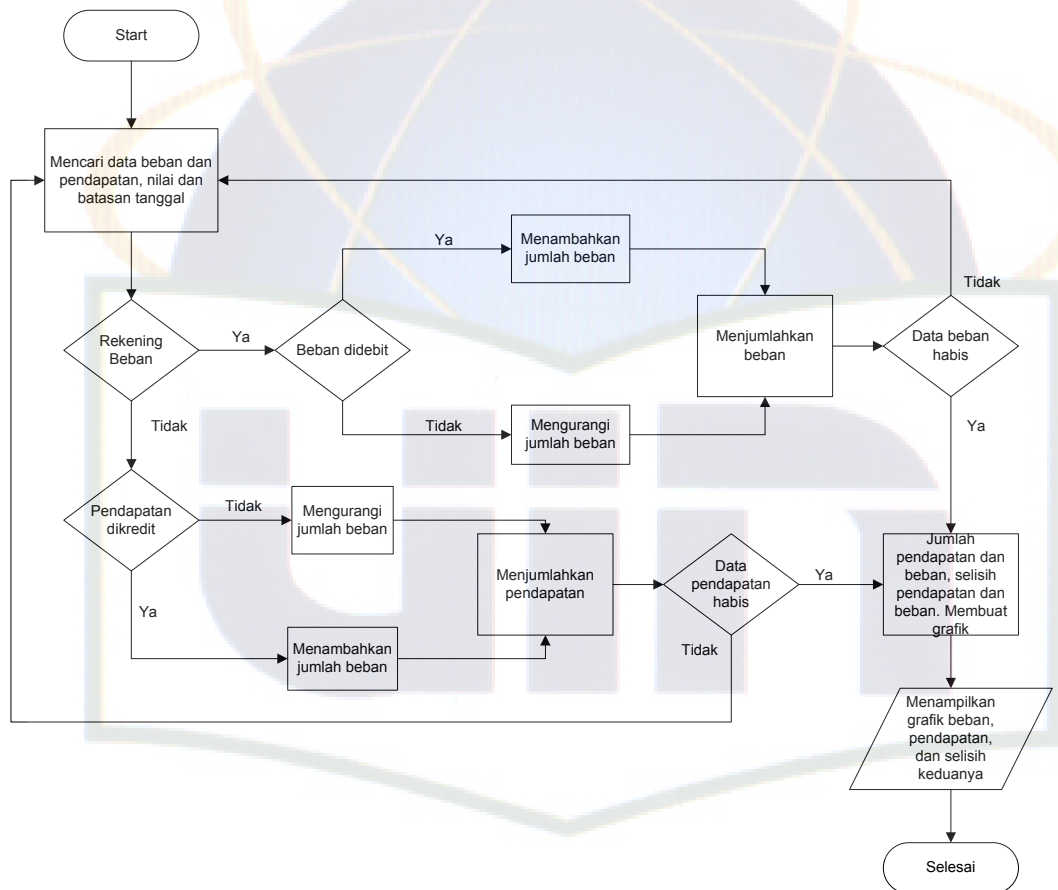
## e. Grafik beban



Gambar 4.67 Flowchart Halaman Grafik Beban

Aplikasi akan mencari jurnal yang memiliki rekening beban. Bila nilai beban adalah debit, maka akan ditambah. Bila kredit, maka akan dikurangi. Setelah itu, beban dijumlahkan untuk dibuat diagram batangnya.

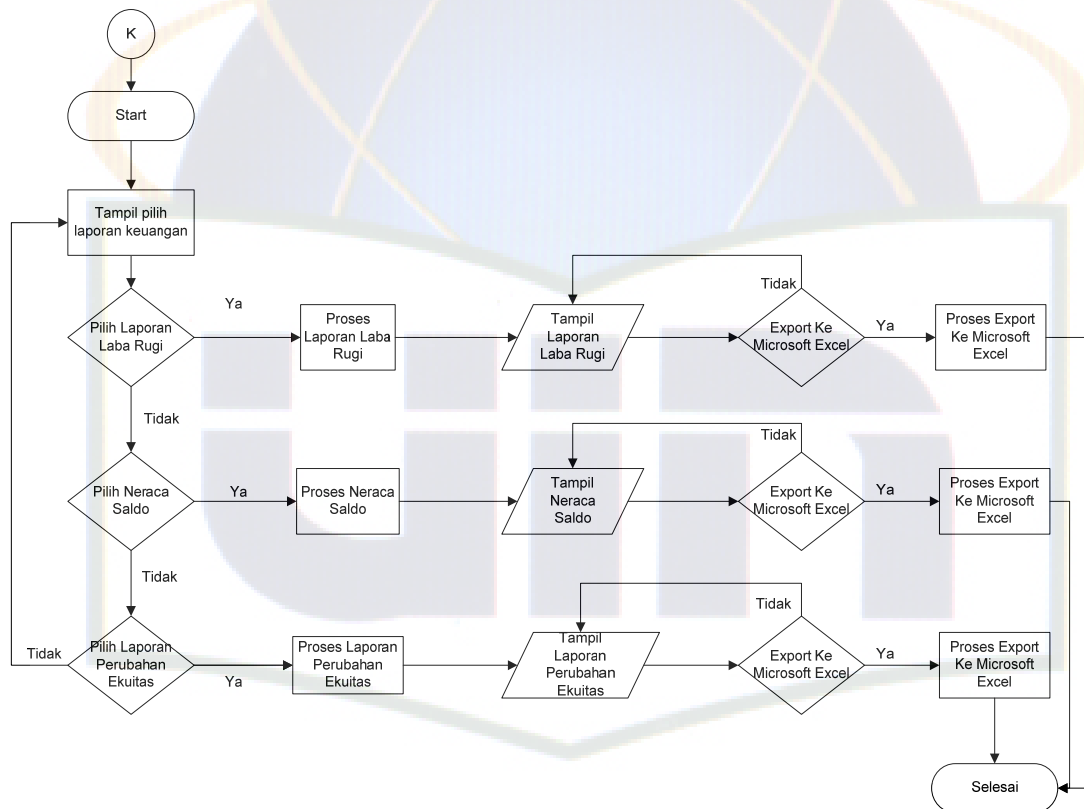
## f. Grafik laba rugi

Gambar 4.68 *Flowchart* Halaman Grafik Laba Rugi

Aplikasi akan mencari jurnal yang memiliki rekening pendapatan dan beban. Bila nilai beban adalah debit, maka akan ditambah. Bila kredit, maka akan dikurangi. Bila nilai pendapatan adalah kredit, maka akan ditambah. Bila debit, maka akan dikurangi. Setelah semua pendapatan dan beban dijumlahkan, maka akan dicari selisih antara pendapatan dan beban.

Jumlah pendapatan, jumlah beban, dan selisih keduanya kemudian dibuat diagram batangnya.

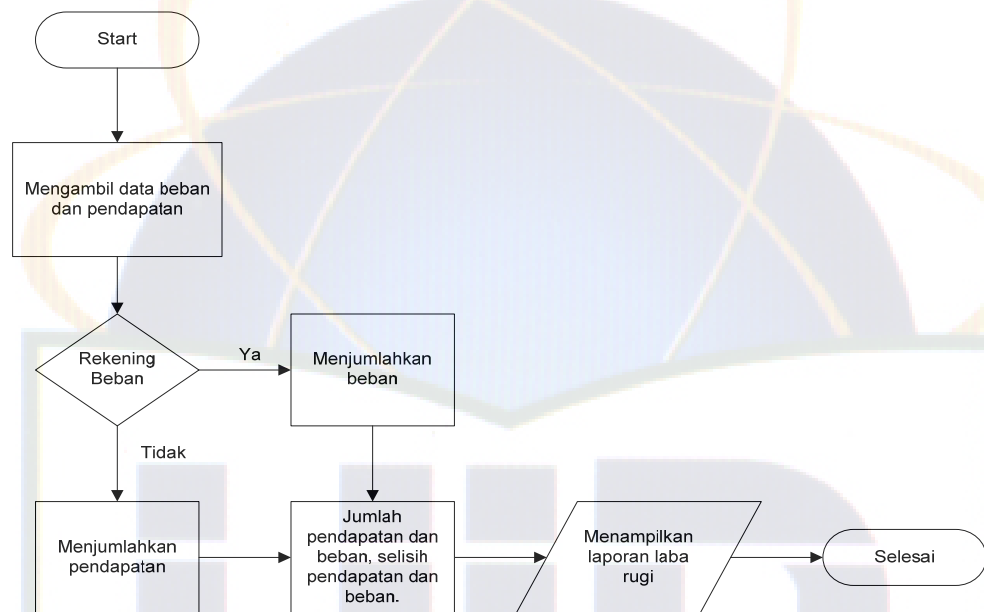
#### 14. Flowchart Laporan Keuangan



Gambar 4.69 Flowchart Halaman Laporan Keuangan

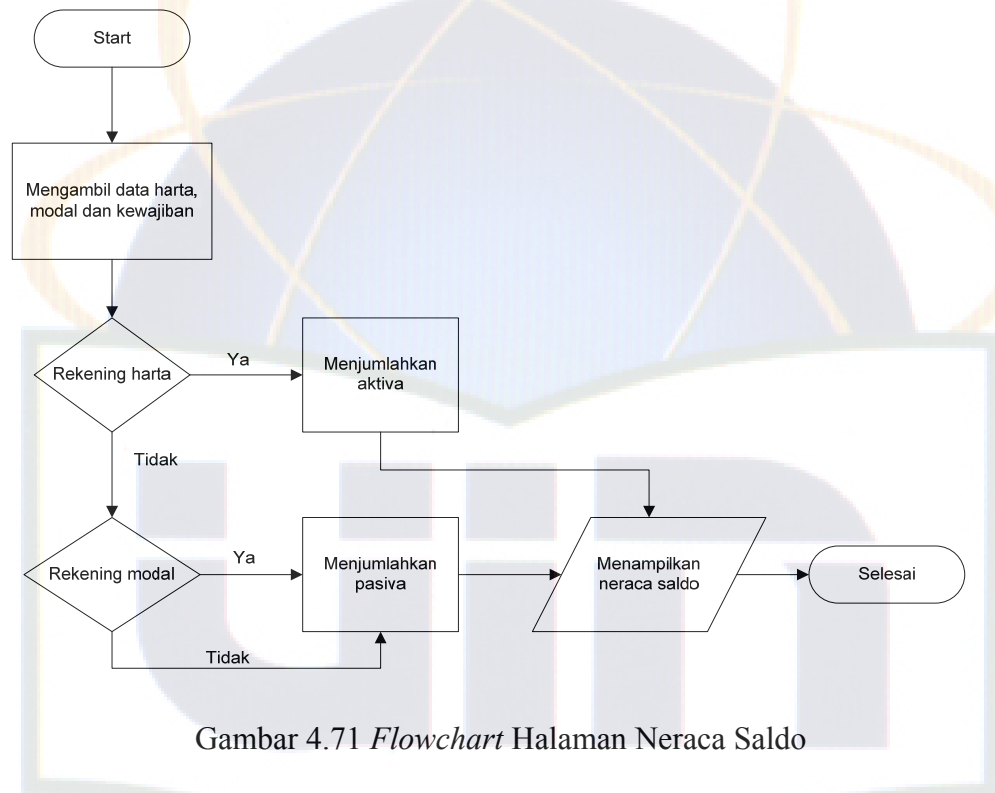
Halaman ini bisa diakses oleh *admin* dan *user*. Pertama, aplikasi akan menampilkan *form* untuk memilih laporan keuangan. Setelah *user* memilih, maka akan tampil laporan keuangan yang diinginkan. Laporan ini dapat diekspor kedalam Microsoft Excel. Disini terdapat beberapa subproses, yaitu :

## a. Proses laporan laba rugi

Gambar 4.70 *Flowchart* Halaman Laporan Laba Rugi

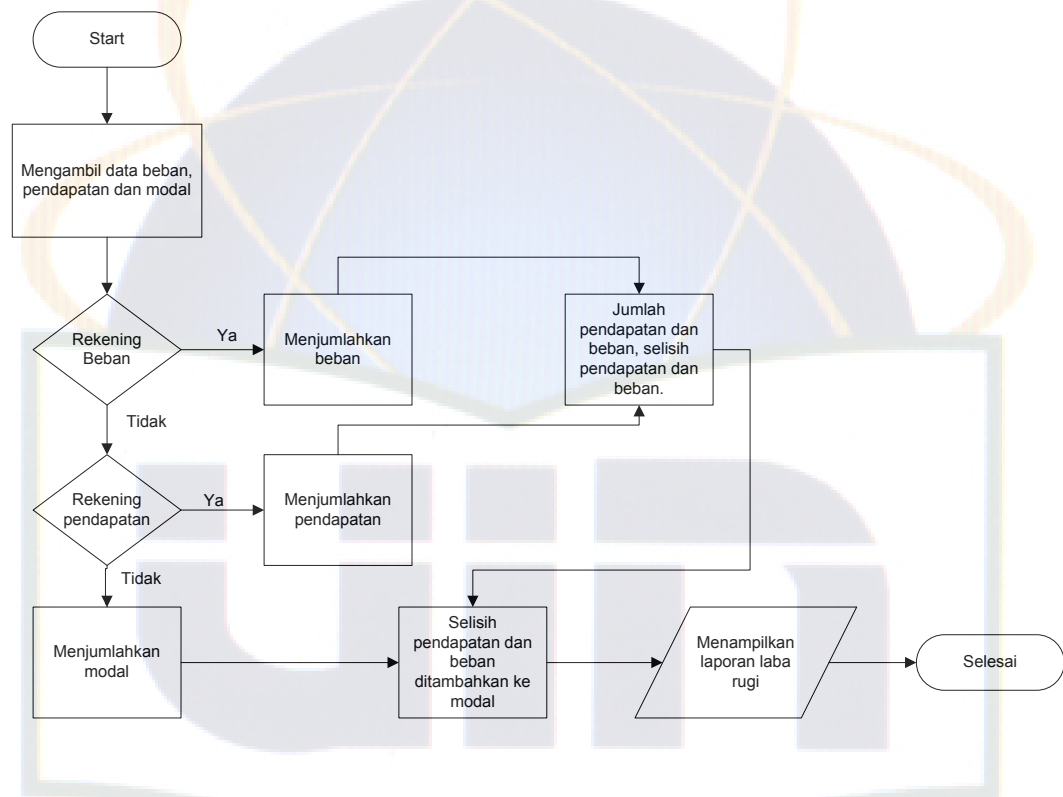
Pada proses ini, aplikasi akan mengambil saldo dari rekening pendapatan dan beban. Keduanya dijumlahkan pada masing – masing rekening. Setelah itu, akan dicari selisih antara pendapatan dan beban. Hasilnya yaitu rekening pendapatan dan beban beserta saldonya, kemudian laba atau rugi yang diterima IIUC.

## b. Proses neraca saldo

Gambar 4.71 *Flowchart* Halaman Neraca Saldo

Aplikasi akan mengambil data harta, modal, dan kewajiban beserta saldonya. Bila rekening harta, maka akan dijumlahkan ke aktiva. Bila modal dan kewajiban, maka akan dijumlahkan ke pasiva. Hasil akhirnya adalah rekening harta beserta jumlahnya di kolom aktiva dan rekening modal dan kewajiban beserta jumlahnya di kolom pasiva.

## c. Proses laporan perubahan modal

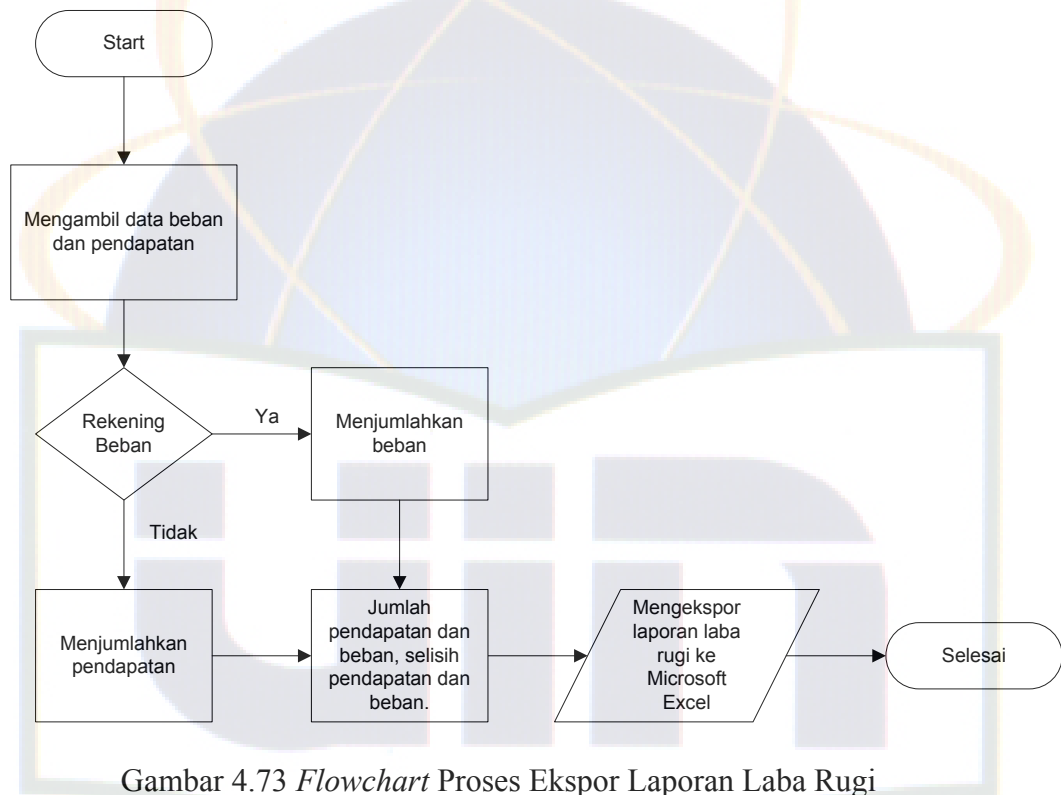


Gambar 4.72 Flowchart Halaman Laporan Perubahan Modal

Pada proses ini, aplikasi akan mengambil saldo dari rekening pendapatan dan beban. Keduanya dijumlahkan pada masing – masing rekening. Setelah itu, akan dicari selisih antara pendapatan dan beban. Hasil dari selisih ini kemudian dijumlahkan ke modal.



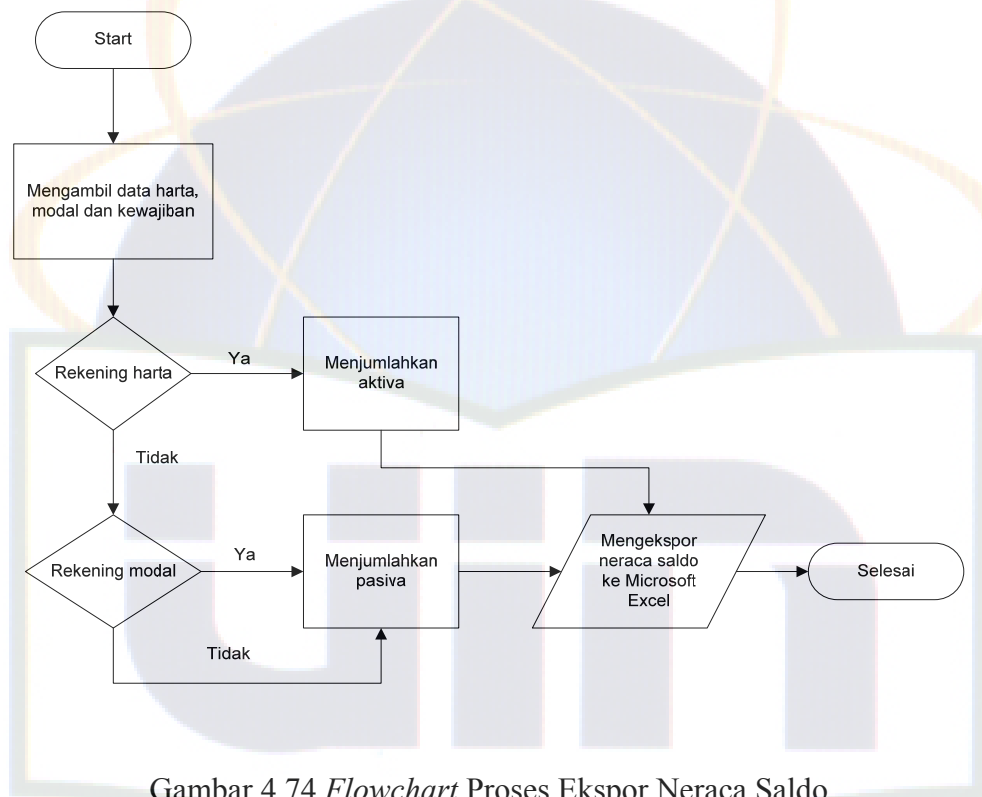
## d. Proses ekspor laporan laba rugi



Gambar 4.73 Flowchart Proses Ekspor Laporan Laba Rugi

Pada proses ini, aplikasi akan mengambil saldo dari rekening pendapatan dan beban. Keduanya dijumlahkan pada masing – masing rekening. Setelah itu, akan dicari selisih antara pendapatan dan beban. Hasilnya yaitu rekening pendapatan dan beban beserta saldonya, kemudian laba atau rugi. Ini semua kemudian akan diekspor ke Microsoft Excel.

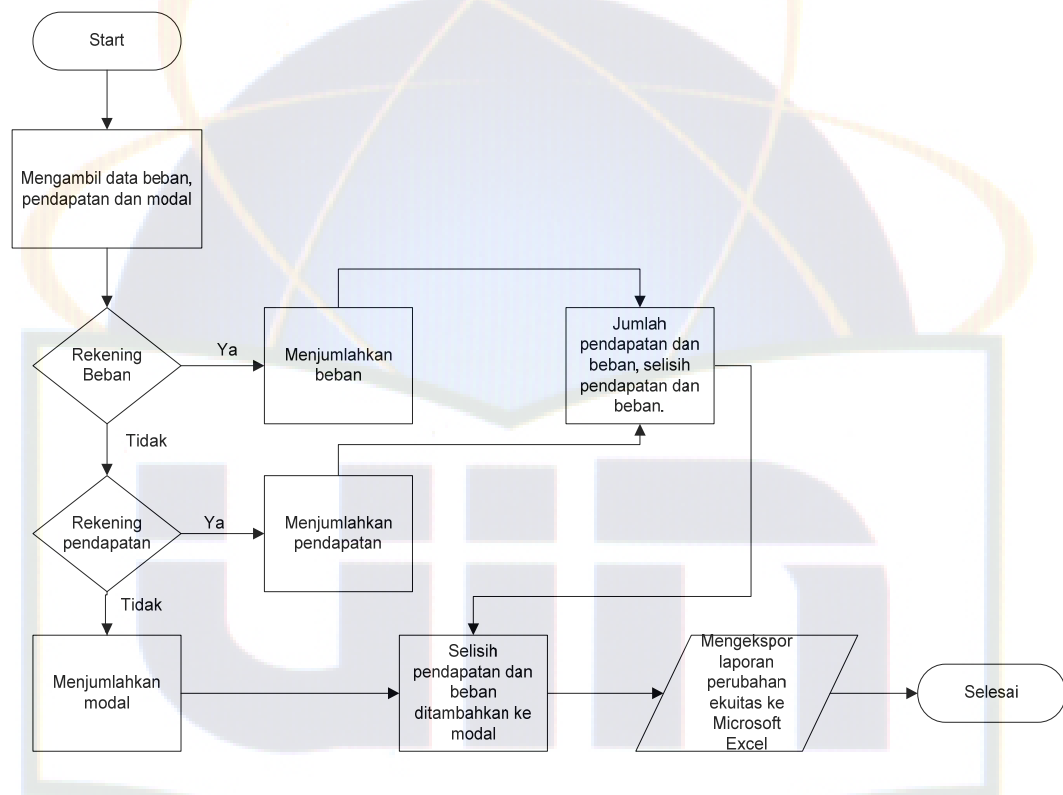
## e. Proses ekspor neraca saldo



Gambar 4.74 Flowchart Proses Ekspor Neraca Saldo

Aplikasi akan mengambil data harta, modal, dan kewajiban beserta saldonya. Bila rekening harta, maka akan dijumlahkan ke aktiva. Bila modal dan kewajiban, maka akan dijumlahkan ke pasiva. Hasil akhirnya adalah rekening harta beserta jumlahnya di kolom aktiva dan rekening modal dan kewajiban beserta jumlahnya di kolom pasiva. Hasil dari ini semua kemudian akan diekspor ke Microsoft Excel.

## f. Proses ekspor laporan perubahan modal

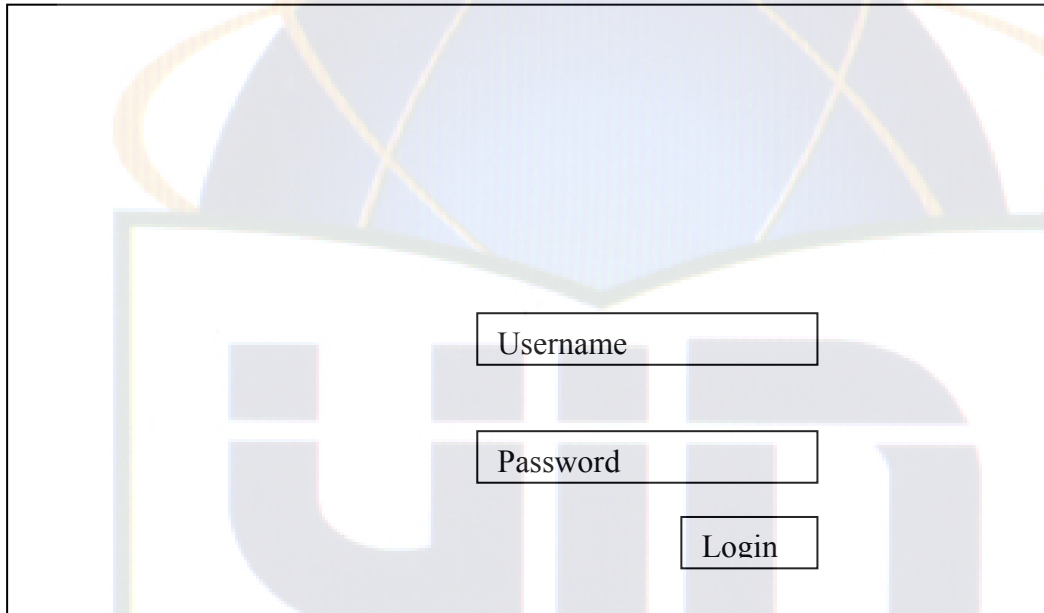


Gambar 4.75 Flowchart Proses Ekspor Laporan Perubahan Modal

Pada proses ini, aplikasi akan mengambil saldo dari rekening pendapatan dan beban. Keduanya dijumlahkan pada masing – masing rekening. Setelah itu, akan dicari selisih antara pendapatan dan beban. Hasil dari selisih ini kemudian dijumlahkan ke modal. Ini semua kemudian akan diekspor ke Microsoft Excel.

#### 4.3.7 Desain *Interface*

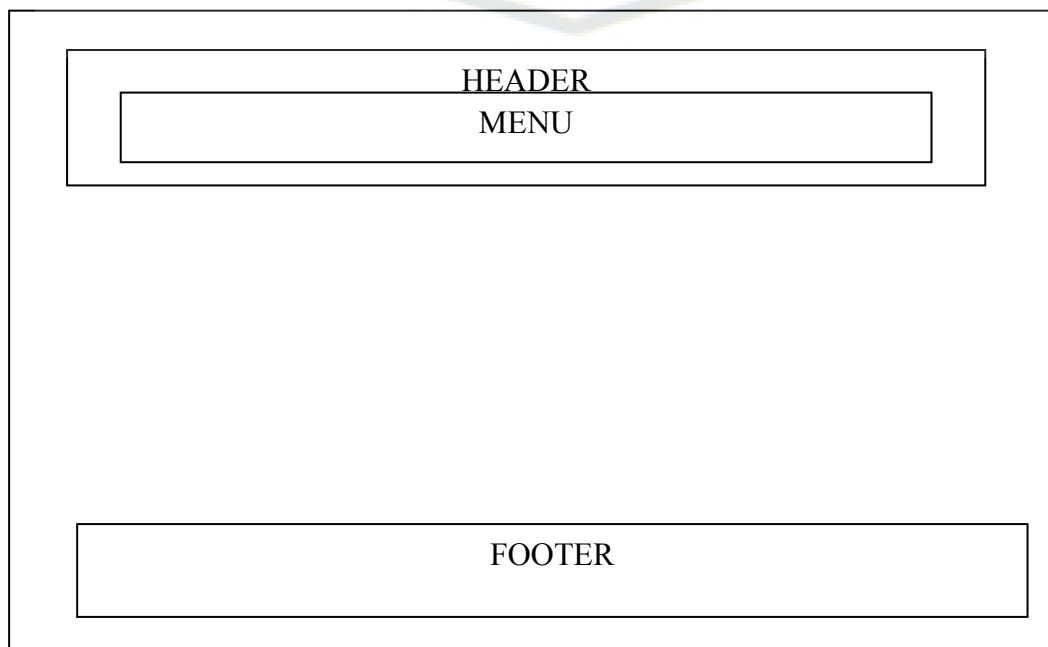
##### 1. Perancangan Halaman Login



The diagram illustrates the layout of a login page. It features a central rectangular area containing three input fields: a 'Username' field at the top, a 'Password' field below it, and a 'Login' button positioned to the right of the password field. The entire interface is overlaid on a background featuring a large, stylized blue dome and yellow orbital lines, with a faint watermark of the letters 'UIN'.

Gambar 4.76 *Interface* Halaman Login

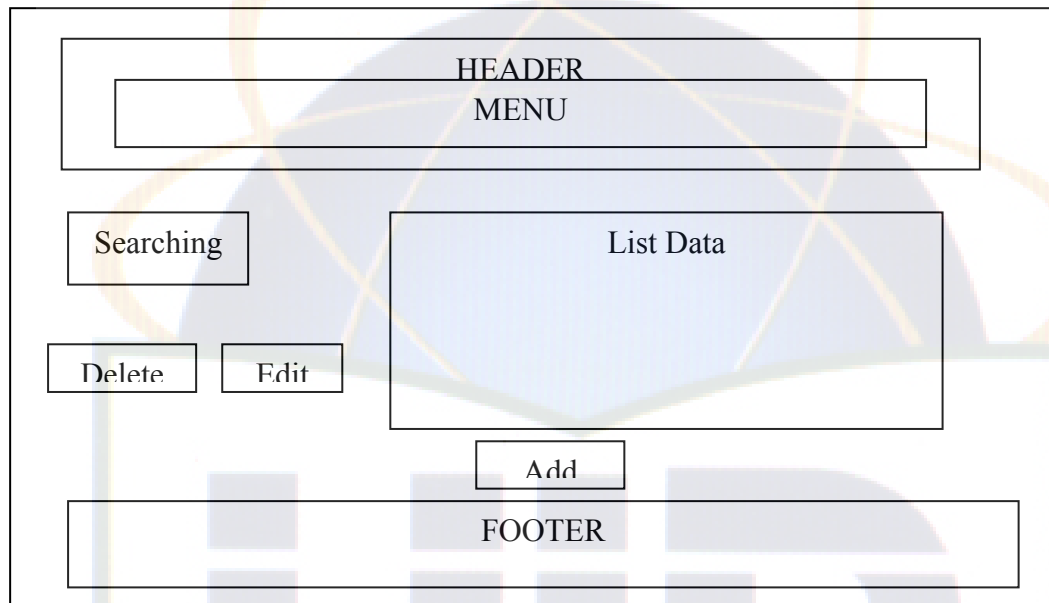
##### 2. Perancangan Halaman Menu Utama



The diagram shows the layout of a main menu page. It consists of a large outer rectangle. Inside, at the top, is a 'HEADER' section which contains a 'MENU' box. At the bottom of the main rectangle is a 'FOOTER' box. The central area between the header and footer is empty.

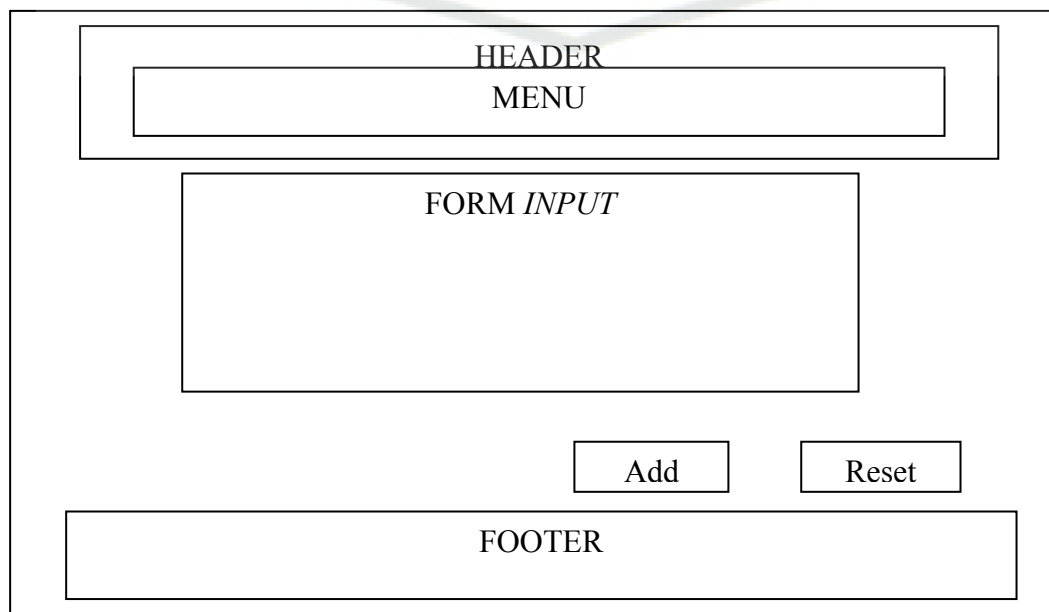
Gambar 4.77 *Interface* Halaman Menu Utama

### 3. Perancangan Halaman Data Rekening



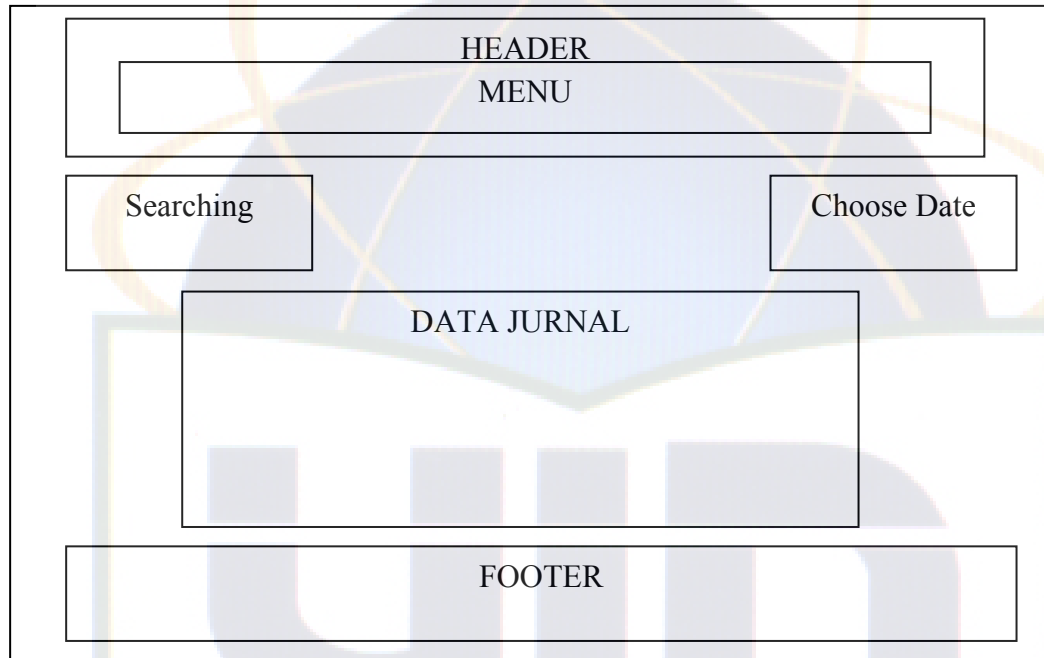
Gambar 4.78 *Interface* Halaman Data Rekening

### 4. Perancangan Halaman Jurnal Umum dan Jurnal Penyesuaian



Gambar 4.79 *Interface* Jurnal Umum, Jurnal Penyesuaian, dan Jurnal Penutup

### 5. Perancangan Halaman Jurnal



Gambar 4.80 *Interface* Halaman Jurnal

#### 4.4 Construction

Dalam fase konstruksi, bahasa yang digunakan adalah PHP dengan menggunakan *database* MySQL.

Dalam pengkodean ini, penulis menggunakan aturan – aturan akuntansi untuk memasukkan transaksi – transaksi ke dalam jurnal. Kemudian dari jurnal – jurnal tersebut akan dibuat laporan keuangan. Dari laporan keuangan tersebut dapat diketahui kondisi keuangan IIUC saat ini. Pada tahap ini penulis juga melakukan *testing* mandiri untuk mengetahui jalannya program yang telah dibuat.

## 4.5 *Deployment*

### 4.5.1 *Testing*

Pada tahapan ini, penulis mengadakan *blackbox testing* untuk menguji aplikasi yang telah dibuat. Cara pengujian *blackbox testing* dilakukan dengan menjalankan aplikasi keuangan dan melakukan *input* data serta melihat *input*-nya apakah sesuai dengan aturan – aturan akuntansi yang telah ditetapkan. Berikut adalah tabel pengujian :

Tabel 4.19. Hasil Pengujian

No	Nama Pengujian	Proses Pengujian	Hasil Yang Diharapkan	Hasil Aktual	Keterangan
1	<i>Login</i>	<i>User</i> akan diberikan halaman <i>login</i> . <i>User</i> harus mengisi <i>username</i> dan <i>password</i> untuk bisa masuk kedalam aplikasi. <i>Username</i> yang terdaftar adalah <i>iiuc</i> , sedangkan <i>password</i> yang terdaftar adalah <i>iiuc</i> .	Setelah <i>user</i> memasukkan <i>username</i> yaitu <i>iiuc</i> dan <i>password</i> yaitu <i>iiuc</i> , diharapkan aplikasi akan menampilkan halaman utama dari aplikasi.	Setelah <i>user</i> memasukkan <i>username</i> yaitu <i>iiuc</i> dan <i>password</i> yaitu <i>iiuc</i> , maka aplikasi menampilkan halaman utama dari aplikasi	OK
2	Menambah data akun	<i>User</i> akan menambahkan data akun baru. Untuk menambahkan, maka <i>user</i> harus memasukkan kode akun, nama akun, dan subklasifikasi akun. Bila kode akun sama, maka data tidak	Setelah <i>user</i> memasukkan kode akun, nama akun, dan subklasifikasi akun, diharapkan data akun akan bertambah. Data bertambah dengan catatan	Setelah <i>user</i> memasukkan kode akun, nama akun, dan subklasifikasi akun, maka data akun baru bertambah. Aplikasi menampilkan pesan sukses	OK



		masuk.	kode akun tidak sama.	menambah akun. Akun baru dapat dilihat pada menu akun.	
3	Menambah jurnal umum	<i>User</i> akan menambahkan jurnal umu sebagai transaksi baru yang akan dicatat pada aplikasi. Untuk memasukkannya, maka <i>user</i> harus memasukkan tanggal jurnal, kode jurnal, dan deskripsi jurnal. Selain itu, <i>user</i> juga harus memasukkan akun yang digunakan dalam transaksi beserta jumlahnya. Akun minimal harus terdiri dari dua akun berbeda, kemudian jumlah	Setelah <i>user</i> memasukkan tanggal jurnal, kode jurnal, deskripsi jurnal, akun yang digunakan beserta jumlah uang, diharapkan jurnal umum yang baru dapat ditambahkan. Jurnal umum yang baru bisa masuk dengan catatan tidak ada kode jurnal yang sama. Jurnal yang baru	Setelah <i>user</i> memasukkan tanggal jurnal, kode jurnal, deskripsi jurnal, akun yang digunakan beserta jumlah uang, maka jurnal umum yang baru dapat ditambahkan. Jurnal umum yang baru dapat dilihat pada <i>menu</i> jurnal. Selain itu, aplikasi telah mengganti saldo dari kas	OK

		<p>debet dan kredit harus seimbang. Bila ada dua akun yang sama atau jumlah debet dan kredit tidak sama, maka penambahan akun akan gagal. Dalam pengujian ini, penulis memasukkan akun kas pada posisi debit dan modal pada kredit. Uang yang digunakan berjumlah satu juta.</p>	<p>ditambahkan dapat dilihat pada <i>menu</i> jurnal. Selain itu, diharapkan aplikasi juga akan mengganti saldo akun, dalam hal ini kas dan modal. Saldo kas akan berjumlah satu juta pada posisi debit, sedangkan saldo modal berjumlah satu juta pada kredit.</p>	<p>dan modal. Saldo kas berjumlah satu juta pada debit, sedangkan saldo modal berjumlah satu juta pada kredit.</p>	
4	Menambah jurnal koreksi	<p><i>User</i> dapat mengkoreksi jurnal yang telah dimasukkan. Jurnal koreksi berfungsi untuk membalik posisi debit dan kredit pada akun</p>	<p>Setelah <i>user</i> memasukkan tanggal jurnal, kode jurnal, dan deskripsi jurnal, diharapkan aplikasi akan</p>	<p>Setelah <i>user</i> memasukkan tanggal jurnal, kode jurnal, dan deskripsi jurnal, maka aplikasi menambahkan</p>	OK

		yang digunakan pada jurnal tersebut. Selain itu, aplikasi juga akan mengganti saldo pada akun yang bersangkutan. <i>User</i> harus memasukkan tanggal jurnal, kode jurnal, dan deskripsi jurnal. Dalam pengujian ini, jurnal akan mengkoreksi transaksi sebelumnya, yaitu kas di debit sebesar satu juta, sedangkan modal di kredit sebesar satu juta.	menambahkan jurnal koreksi. Jurnal bisa ditambahkan dengan catatan tidak ada kode jurnal yang sama. Selain itu, diharapkan aplikasi akan mengganti saldo akun yang terkait. Diharapkan saldo kas akan berjumlah nol, sedangkan saldo modal akan berjumlah nol.	jurnal koreksi. Jurnal bisa ditambahkan dengan catatan tidak ada kode jurnal yang sama. Selain itu, aplikasi mengganti saldo akun yang terkait. Saldo kas akan berjumlah nol, sedangkan saldo modal akan berjumlah nol.	
5	Menambah jurnal penutup	<i>User</i> bisa menutup saldo akun – akun yang bertipe pendapatan dan	Setelah <i>user</i> memasukkan kode jurnal, tanggal jurnal,	Setelah <i>user</i> memasukkan kode jurnal, tanggal jurnal,	OK

		<p>beban. Akun – akun bertipe pendapatan dan beban akan bersaldo nol. Selisih dari jumlah akun – akun tersebut akan ditambahkan ke modal. <i>User</i> harus memasukkan tanggal jurnal, kode jurnal, dan deskripsi jurnal.</p>	<p>dan deskripsi jurnal, diharapkan aplikasi dapat menambahkan jurnal penutup. Selain itu, diharapkan aplikasi dapat menolak akun – akun bertipe pendapatan dan beban. Hasil dari selisih jumlah akun pendapatan dan jumlah akun beban diharapkan akan ditambahkan ke akun modal.</p>	<p>dan deskripsi jurnal, aplikasi menambahkan jurnal penutup. Selain itu, aplikasi menolak akun – akun bertipe pendapatan dan beban. Hasil dari selisih jumlah akun pendapatan dan jumlah akun beban ditambahkan ke akun modal.</p>	
6	Membuat laporan	<i>User</i> dapat membuat laporan laba	Diharapkan aplikasi akan	Aplikasi menampilkan	OK

	laba rugi	<p>rugi. Aplikasi akan mengambil saldo dari akun pendapatan dan beban. Selisih dari jumlah akun pendapatan dan akun beban akan menentukan laba atau rugi. Bila hasilnya positif, maka laba. Bila hasilnya negatif, maka rugi. Pada pengujian ini, akun pendapatan berjumlah satu juta, sedangkan akun beban berjumlah seratus ribu. <i>User</i> dapat mengekspor laporan ini ke format Microsoft Excel.</p>	<p>menampilkan nama – nama akun pendapatan dan akun beban serta saldonya masing – masing. Pada akhir tabel, diharapkan akan terlihat selisih dari kedua akun tersebut. Pada pengujian ini, diharapkan hasil akhir adalah laba sebesar sembilan ratus ribu.</p>	<p>nama – nama akun pendapatan dan akun beban serta saldonya masing – masing. Pada akhir tabel, terlihat selisih dari kedua akun tersebut. Hasil akhir adalah laba sebesar sembilan ratus ribu.</p>	
7	Membuat neraca	<i>User</i> dapat membuat neraca.	Diharapkan aplikasi akan	Aplikasi menampilkan	OK

		Neraca terdiri dari dua kolom, yaitu aktiva dan pasiva. Aktiva yaitu harta, sedangkan pasiva yaitu modal dan kewajiban. Jumlah dari aktiva dan pasiva harus sama. <i>User</i> dapat mengekspor ke dalam Microsoft Excel. Pada pengujian ini, jumlah aktiva dan pasiva adalah satu juta.	menampilkan nama akun harta pada kolom pasiva beserta saldonya. Aplikasi juga akan menampilkan akun modal dan kewajiban pada pasiva, beserta jumlah saldonya. Diharapkan jumlah saldo aktiva dan pasiva sama. Diharapkan aktiva dan pasiva sama – sama berjumlah satu juta.	nama akun harta pada kolom pasiva beserta saldonya. Aplikasi juga menampilkan akun modal dan kewajiban pada pasiva, beserta jumlah saldonya. Jumlah saldo aktiva dan pasiva sama, yaitu masing – masing sebesar satu juta.	
8	Membuat laporan perubahan ekuitas	<i>User</i> dapat membuat laporan perubahan ekuitas. Laporan	Diharapkan aplikasi akan menampilkan ekuitas awal	Aplikasi menampilkan ekuitas awal sebesar satu	OK

		<p>ekuitas terdiri dari saldo awal, laba atau rugi, pengambilan pribadi, dan total ekuitas. Aplikasi akan mengambil saldo dari modal, prive, akun pendapatan dan akun beban. Pada pengujian ini, ekuitas awal sebesar satu juta, kemudian sisa laba adalah sembilan ratus ribu, sehingga total ekuitas sebesar satu juta sembilan ratus ribu.</p>	<p>sebesar satu juta. Selain itu, akan ditampilkan sisa laba sebesar sembilan ratus ribu. Pengambilan pribadi berjumlah nol, sehingga total ekuitas diharapkan berjumlah satu juta sembilan ratus ribu.</p>	<p>juta. Selain itu, ditampilkan sisa laba sebesar sembilan ratus ribu. Pengambilan pribadi berjumlah nol, sehingga total ekuitas berjumlah satu juta sembilan ratus ribu.</p>	
--	--	---	---	--	--

*Screenshot* dari pengujian dapat dilihat pada lampiran 4.

#### **4.5.2 Implementation**

Implementasi yang dilakukan adalah meng-*upload* aplikasi yang telah jadi ke *web hosting* yang telah disediakan oleh IIUC. Alamatnya adalah <http://www.financial.iiuc.ac.id>.





## **BAB V**

### **KESIMPULAN DAN SARAN**

Bab terakhir ini berisi tentang kesimpulan dan saran. Berdasarkan uraian dan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, maka dapat dibuat kesimpulan tentang penelitian ini dan saran untuk penelitian selanjutnya.

#### **5.1 Kesimpulan**

1. Aplikasi akuntansi ini dapat diakses secara *online* sehingga dapat memudahkan bagian keuangan dalam memasukkan transaksi – transaksi keuangan yang berlangsung.
2. Aplikasi akuntansi ini memberikan fasilitas kemudahan bagi para *manager* untuk dapat melihat keadaan keuangan di IIUC sebagai informasi untuk mengambil keputusan.

#### **5.2 Saran**

1. Diharapkan pada pengembangan selanjutnya aplikasi ini terdapat jurnal yang lebih detail, seperti jurnal pendapatan atau jurnal pengeluaran.
2. Diharapkan pada pengembangan selanjutnya aplikasi dapat melakukan penyimpanan data untuk menyimpan data pada rentang waktu tertentu, misalkan untuk jangka waktu satu tahun. Transaksi – transaksi tersebut berguna untuk mengetahui kinerja dari IIUC selama rentang waktu itu.

3. Dilakukan pengembangan lebih lanjut sehingga dapat dibuat laporan arus kas secara otomatis untuk memudahkan pekerjaan bagian keuangan.



## DAFTAR PUSTAKA

- Febrian, Jack. 2007. *Kamus Komputer dan Teknologi Informasi*. Bandung : Informatika Bandung.
- Hakim, Lukmanul. 2010. *Bikin Website Super Keren Dengan PHP dan jQuery*. Yogyakarta : Lokomedia.
- Hariyanto, Bambang. 2004. *Sistem Manajemen Basis Data*. Bandung : Informatika Bandung.
- Husni.2007. *Pemrograman Database Berbasis Web*.Yogyakarta : Graha Ilmu
- Ibrahim, Ali. 2008.*Cara Praktis Membuat Website Dinamis Menggunakan XAMPP*. Yogyakarta : Neotekno.
- Jusup, Al Haryono. 2005. *Dasar – Dasar Akuntansi*. Yogyakarta : Sekolah Tinggi Ilmu Ekonomi YKPN.
- Kardiman, Agus Suranto.2007. *Prinsip – Prinsip Akuntansi 1 : SMA Kelas XI*. Jakarta : Yudhistira
- Komputer, Wahana.2006.*Seri Panduan Lengkap Menguasai Pemrograman Web Dengan PHP5*.Yogyakarta : Andi.
- Ladjamuddin, Al-bahra. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta : Graha Ilmu.
- Nader, J. C. 1992. *Prentice Hall's Illustrated Dictionary of Computing*. New South Wales : Prentice Hall Inc.
- Nugroho, Bunafit.2005.*Database Relasional Dengan MySQL*.Yogyakarta : Andi
- Peranginangin, Kasiman. 2006. *Aplikasi Web dengan PHP dan MySQL*. Yogyakarta : Andi.
- Post,Gerald V, David L. Anderson. 2003.*Management Information Systems : Solving Bussiness Problems With Information Technology*. New York : McGraw-Hill.
- Pressman,Roger. 2010. *Software Engineering, A Practitioner's Approach*. USA : McGraw-Hill
- Rizki Amalia, Tri. 2010. *Sistem Informasi Akuntansi Pengeluaran Kas (Studi Kasus : BNI Syariah Fatmawati Jakarta Selatan)*. Jakarta : Universitas Islam Negeri Syarif Hidayatullah.
- Setiawati, Wina. *Aplikasi Laporan Keuangan Berbasis Web Untuk Kelurahan Dukuh*. Jakarta : Universitas Gunadarma.

- Simamora, Henry. 2000. *Akuntansi, Basis Pengambilan Keputusan*. Jakarta : Salemba Empat.
- Simarmata, Janner. 2007. *Perancangan Basis Data*. Yogyakarta : Andi
- Turban, Efraim, *et all*. 2004. *Information Technology For Management : Transforming Organizations In The Digital Economy*. USA : John Wiley And Sons.
- Whitten,Jeffrey L,*et all*. 2004. *System Analysis And Design Methods*. USA : McGraw-Hill
- Wismakarma, Komang. 2010. *Panduan Lengkap Menguasai Pemrograman CSS*. Yogyakarta : Lokomedia.

## LAMPIRAN 4

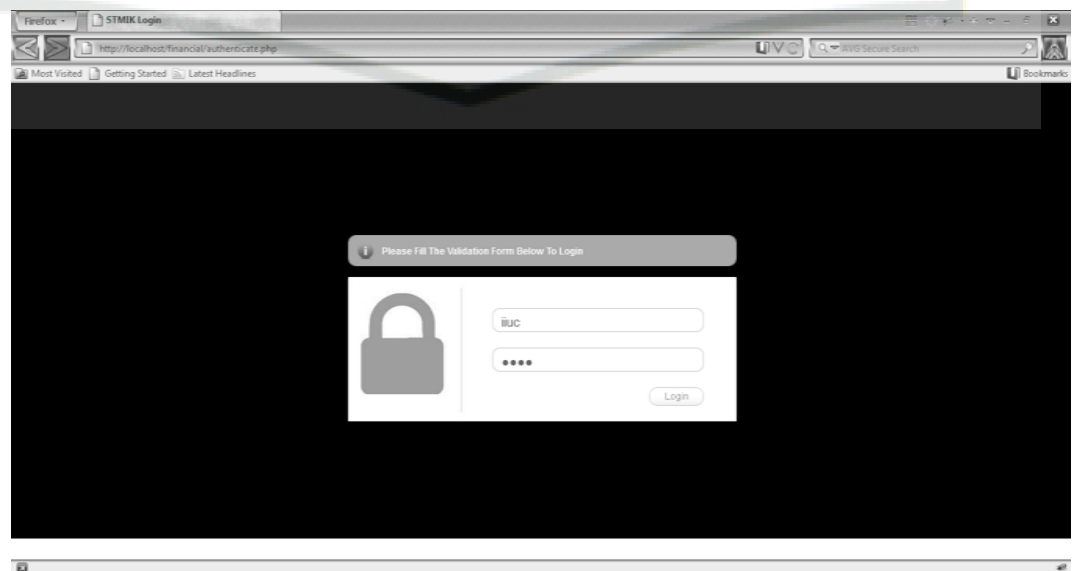
### BLACK BOX TESTING

No	Pengujian	Interface yang diharapkan	Hasil pengujian
1.	<i>Interface</i> halaman login	Halaman menu login	OK
2.	<i>Interface</i> halaman utama <i>admin</i>	Halaman utama <i>admin</i>	OK
3.	<i>Interface</i> halaman utama <i>user</i>	Halaman utama <i>user</i>	OK
4.	<i>Interface</i> data <i>user</i>	Halaman data <i>user</i>	OK
5.	<i>Interface</i> tambah <i>user</i>	Halaman tambah <i>user</i>	OK
6.	<i>Interface</i> ganti <i>user</i>	Halaman ganti <i>user</i>	OK
7.	<i>Interface</i> hapus <i>user</i>	Halaman hapus <i>user</i>	OK
8.	<i>Interface</i> data subklasifikasi akun	Halaman data subklasifikasi akun	OK
9.	<i>Interface</i> tambah subklasifikasi akun	Halaman tambah subklasifikasi akun	OK
10.	<i>Interface</i> ganti subklasifikasi akun	Halaman ganti subklasifikasi akun	OK
11.	<i>Interface</i> hapus subklasifikasi akun	Halaman hapus subklasifikasi akun	OK
12.	<i>Interface</i> data rekening	Halaman data rekening	OK
13.	<i>Interface</i> tambah rekening	Halaman tambah rekening	OK
14.	<i>Interface</i> ganti rekening	Halaman ganti rekening	OK
15.	<i>Interface</i> hapus rekening	Halaman hapus rekening	OK
16.	<i>Interface</i> cari rekening	Halaman cari rekening	OK
17.	<i>Interface</i> data tipe jurnal	Halaman data tipe jurnal	OK
18.	<i>Interface</i> tambah tipe jurnal	Halaman tambah tipe jurnal	OK
19.	<i>Interface</i> ganti tipe jurnal	Halaman ganti tipe jurnal	OK

20.	<i>Interface</i> hapus tipe jurnal	Halaman hapus tipe jurnal	OK
21.	<i>Interface</i> jurnal umum	Halaman jurnal umum	OK
22.	<i>Interface</i> tambah jurnal umum	Halaman tambah jurnal umum	OK
23.	<i>Interface</i> jurnal penyesuaian	Halaman jurnal penyesuaian	OK
24.	<i>Interface</i> tambah jurnal penyesuaian	Halaman tambah jurnal penyesuaian	OK
25.	<i>Interface</i> jurnal penutup	Halaman jurnal penutup	OK
26.	<i>Interface</i> tambah jurnal penutup	Halaman tambah jurnal penutup	OK
27.	<i>Interface</i> tampil data jurnal	Halaman tampil data jurnal	OK
28.	<i>Interface</i> cari jurnal	Halaman cari jurnal	OK
29.	<i>Export</i> data jurnal	<i>Export</i> data jurnal ke dalam Microsoft Excel	OK
30.	<i>Interface</i> jurnal koreksi	Halaman jurnal koreksi	OK
31.	<i>Interface</i> tambah jurnal koreksi	Halaman tambah jurnal koreksi	OK
32.	<i>Interface</i> buku besar	Halaman buku besar	OK
33.	<i>Interface</i> buku besar detail	Halaman buku besar detail	OK
34.	<i>Interface</i> grafik harta	Halaman grafik harta	OK
35.	<i>Interface</i> grafik kewajiban	Halaman grafik kewajiban	OK
36.	<i>Interface</i> grafik modal	Halaman grafik modal	OK
37.	<i>Interface</i> grafik pendapatan	Halaman grafik pendapatan	OK
38.	<i>Interface</i> grafik beban	Halaman grafik beban	OK
39.	<i>Interface</i> grafik pendapatan dan beban	Halaman grafik pendapatan dan beban	OK

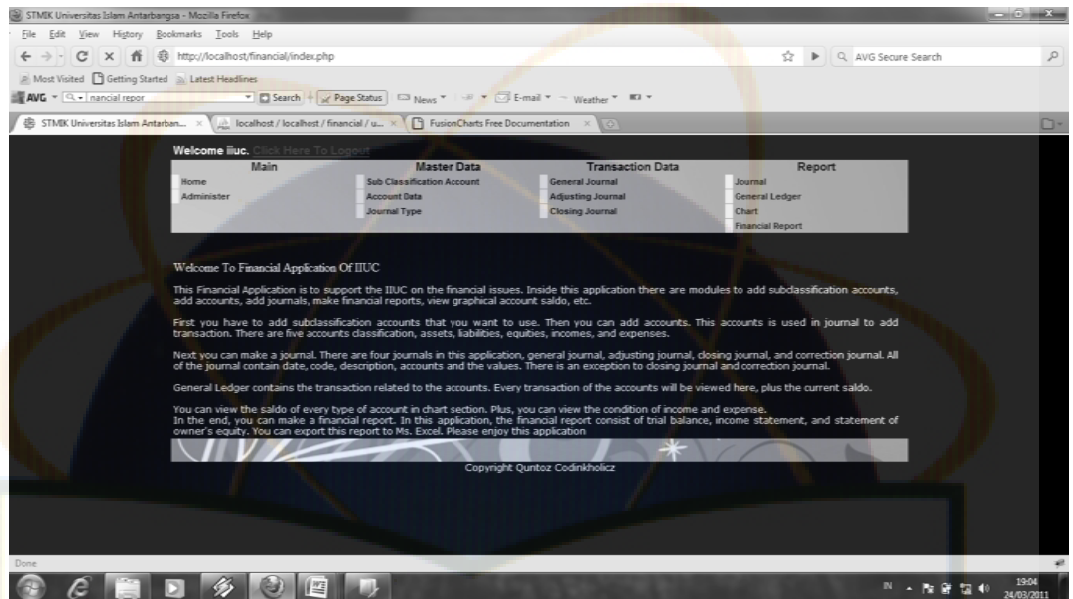
40.	<i>Interface</i> neraca saldo	Halaman neraca saldo	OK
41.	<i>Exported</i> neraca saldo	Neraca saldo dalam Microsoft Excel	OK
42.	<i>Interface</i> Laporan Laba Rugi	Halaman Laporan Laba Rugi	OK
43.	<i>Exported</i> Laporan Laba Rugi	Laporan Laba Rugi dalam Microsoft Excel	OK
44.	<i>Interface</i> Laporan Perubahan Modal	Halaman Laporan Perubahan Modal	OK
45.	<i>Exported</i> Laporan Perubahan Modal	Laporan Perubahan Modal dalam Microsoft Excel	OK

Berikut adalah skenario yang digunakan untuk mengetes halaman *login*.  
*Username* dan *password* yang terdaftar adalah iiuc, dengan *status* sebagai admin.  
Ini adalah tampilan dari halaman login tersebut.



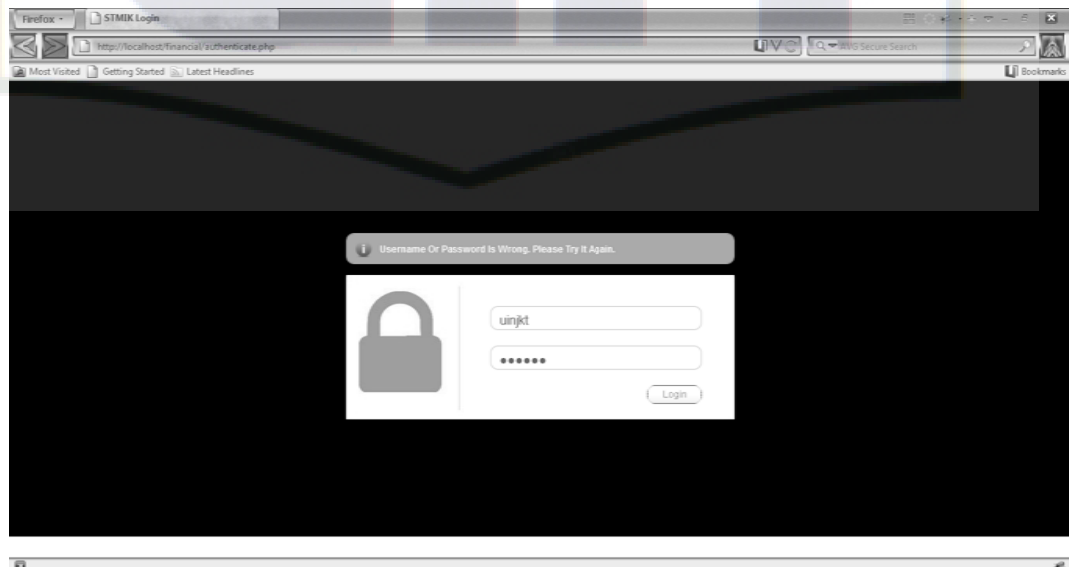
Gambar 1 Halaman *Login*

Maka ketika dimasukkan, aplikasi akan menampilkan halaman utama, seperti gambar 2.



Gambar 2. Menu Utama

Namun, ketika *user* memasukkan *username* dan *password* yang salah, maka aplikasi akan menampilkan pesan *error*, seperti gambar 4.



Gambar 3. *Username* atau *password* yang salah.

Kemudian ini adalah skenario untuk menguji penambahan data rekening.

Untuk menambahkan rekening, maka *user* harus memasukkan kode, nama akun,



dan subklasifikasi akun. Berikut adalah gambar untuk menambahkan akun dengan *input* data yang benar.

Firefox - STMIK Universitas Islam Antarbangsa  
http://localhost/financial/add\_akun.php

Welcome iluc. [Click Here To Logout](#)

**Main** | **Master Data** | **Transaction Data** | **Report**

Home | Sub Classification Account | General Journal | Journal  
Administer | Account Data | Adjusting Journal | General Ledger  
Journal Type | Closing Journal | Financial Report  
Chart

**Add New Account**

Code: 550-90  
Account Name: Beban Kendaraan  
Sub Classification: Biaya - Beban Usaha

Add Reset

Copyright Quntoz Codinkholicz

Gambar 4. *Input* data akun baru

Bila, data akun yang dimasukkan benar, maka aplikasi akan menampilkan pesan sukses, seperti gambar berikut.

Firefox - STMIK Universitas Islam Antarbangsa  
http://localhost/financial/add\_akun.php

Welcome iluc. [Click Here To Logout](#)

**Main** | **Master Data** | **Transaction Data** | **Report**

Home | Sub Classification Account | General Journal | Journal  
Administer | Account Data | Adjusting Journal | General Ledger  
Journal Type | Closing Journal | Financial Report  
Chart

**Add New Account**

Code:   
Account Name:   
Sub Classification: Aktiva - Harta

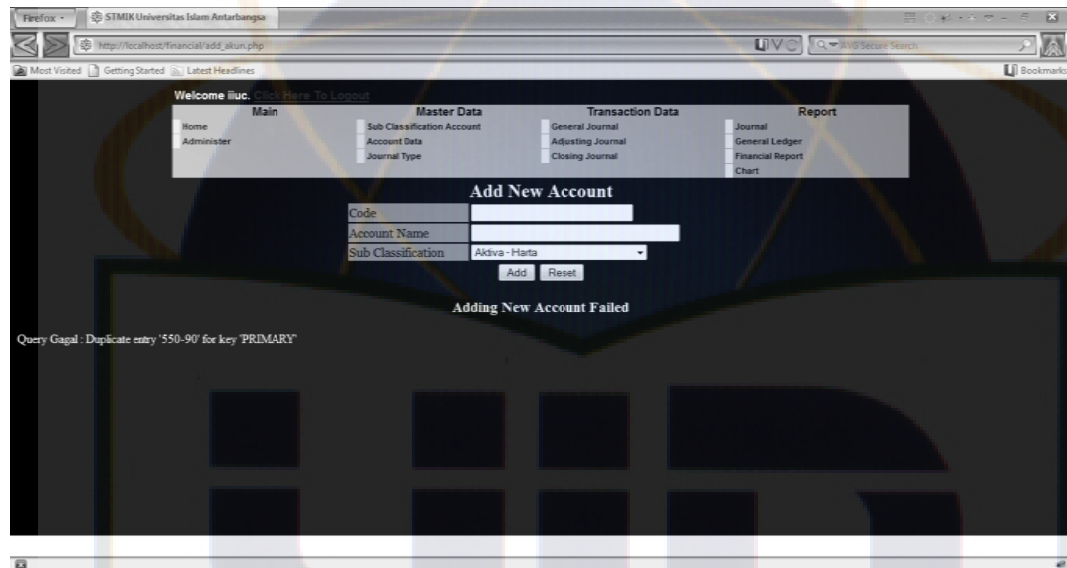
Add Reset

Database : Sub Classification  
Code : 550-90 Successfully Added

Copyright Quntoz Codinkholicz

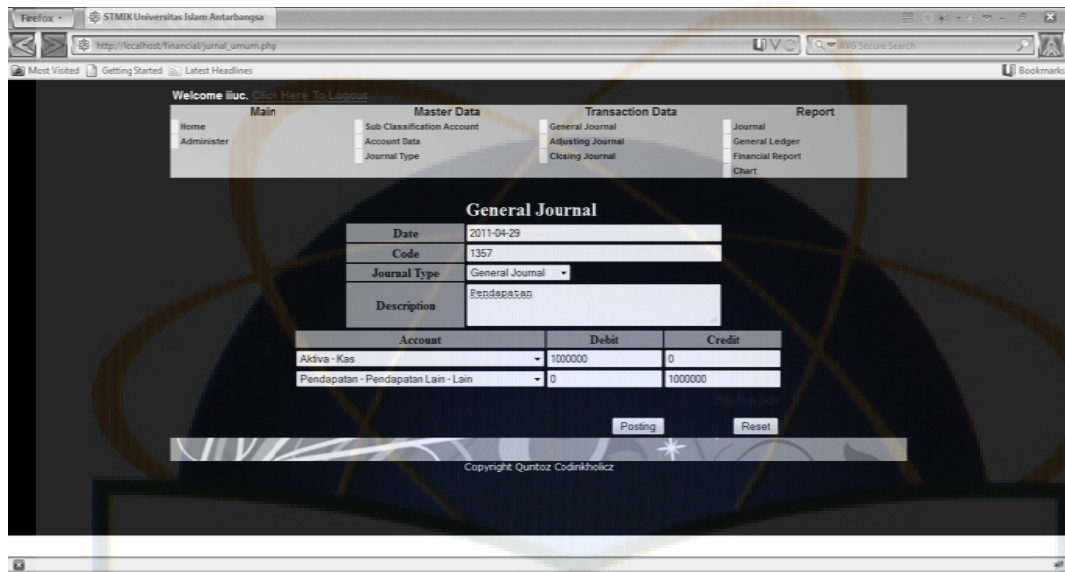
Gambar 5. Pesan sukses ketika menambahkan akun baru

Namun, bila data yang ingin dimasukkan telah ada pada *database*, maka aplikasi akan menampilkan pesan berikut.



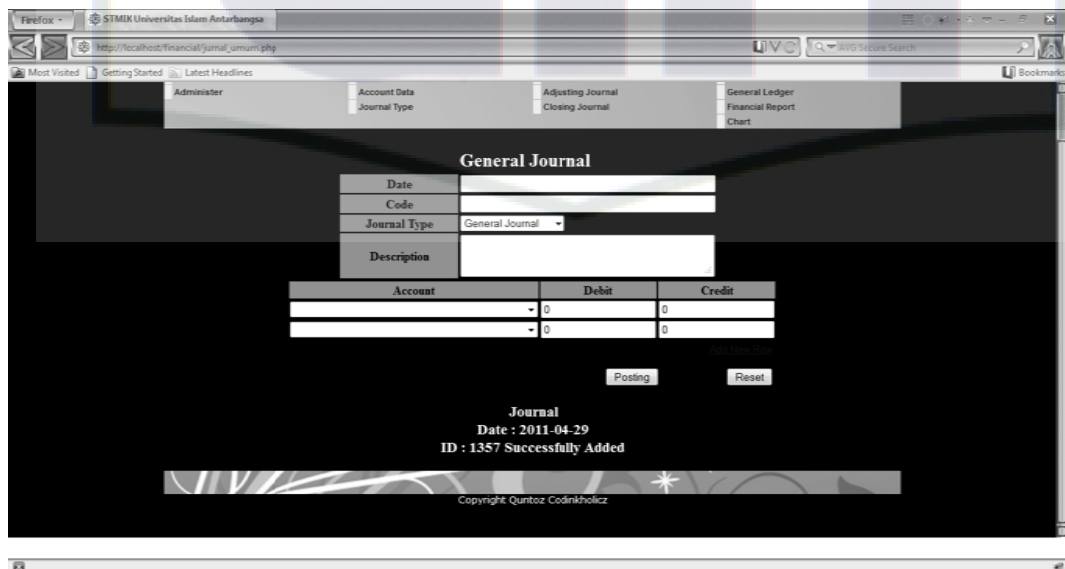
Gambar 6. Pesan gagal ketika menambahkan akun baru.

Berikut adalah skenario untuk menguji proses menambahkan jurnal umum atau jurnal penyesuaian. Untuk menambahkan jurnal baru, maka *user* harus memasukkan tanggal, kode, deskripsi, serta akun dan jumlah akun. Berikut adalah tampilan ketika *input* data jurnal yang benar.



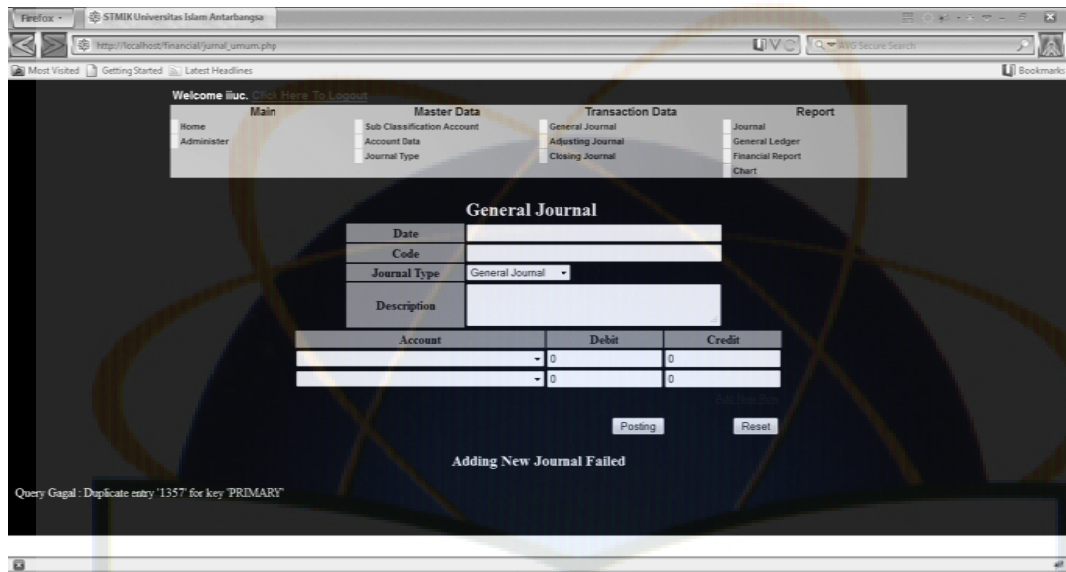
Gambar 7. Tampilan pada saat memasukkan data jurnal

Bila data yang dimasukkan benar, maka aplikasi akan menambahkan jurnal dan menampilkan pesan sukses seperti berikut.



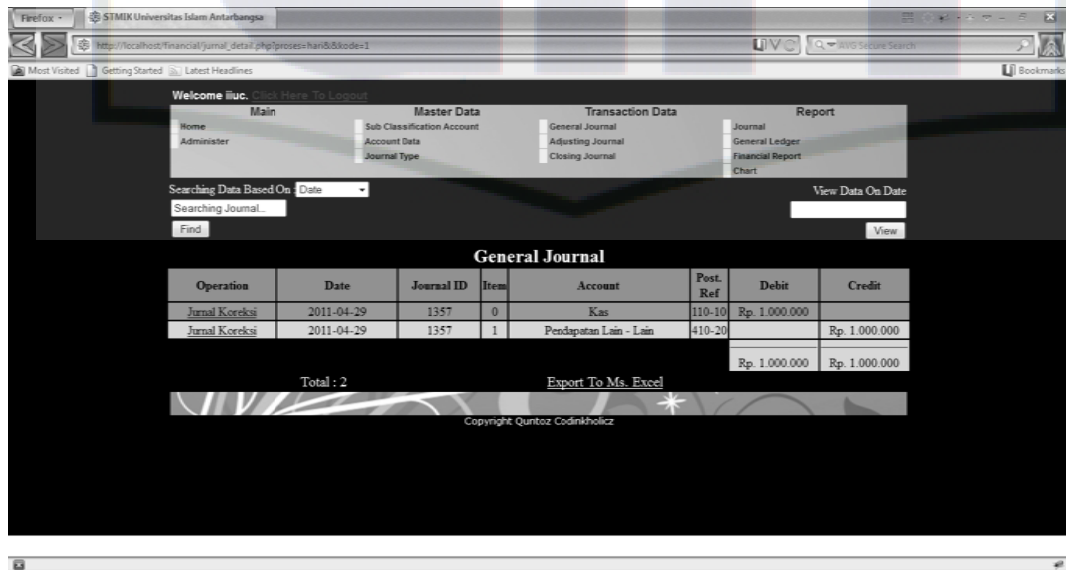
Gambar 8. Pesan ketika sukses menambahkan jurnal baru.

Namun, bila data yang dimasukkan tidak *valid*, maka aplikasi akan menampilkan pesan *error* sebagai berikut.



Gambar 9. Pesan *error* ketika menambahkan jurnal baru

Data jurnal baru yang telah ditambahkan dapat dilihat pada menu jurnal, dengan tampilan sebagai berikut.



Gambar 10. Data jurnal yang telah ditambahkan

## LAMPIRAN 5

### CONTOH KODE PROGRAM

akun.php

```
<?php include("login.php");
?>
<html>
<head>
<title>STMIK Universitas Islam Antarbangsa</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<!--Fireworks MX 2004 Dreamweaver MX 2004 target. Created Wed
Jun 22 19:32:45 GMT+0800 (China Standard Time) 2005-->
<link type="text/css" href="paging.css" rel="stylesheet"
media="screen" />
<link type="text/css" href="scripts/themes/base/ui.all.css"
rel="stylesheet" />
<link href="styles.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="akuntansi.js"></script>
<script type="text/javascript" src="scripts/jquery-
1.4.js"></script>
<script type="text/javascript"
src="scripts/ui/ui.core.js"></script>
<script type="text/javascript"
src="scripts/ui/ui.datepicker.js"></script>
</head>
<?php include("header.php");
include_once("paging.class.php"); // include the paging class
?>
<table width="960" border="0" align="center" height = '195'>
<tr><td colspan='3' align='left'>
<form name = 'form2' action='cari.php' method='post'>
<font color = '#FFFFFF'>Searching Data Based On :</font>
<select name='kolom'>
<option value='kode'>Code</option>
<option value='Nama Akun'>Account Name</option>
<option value='saldo'>Saldo</option>
</select></td></tr>
<tr><td colspan = '3' align = 'left'>
```

```
<input type="text" id ='kata' name="kata" maxlength="50"
value="Searching Account..." onblur="if(this.value=='')
this.value='Searching Account...';"
onfocus="if(this.value=='Searching Account...') this.value='';"
/>
<input name = 'type' value = 'Account' type =
'hidden'></td></tr>
<tr><td>
<input type="submit" onMouseOver='Filter()' onFocus =
'Filter()' value="Find" />
</tr></td>
</form></td></tr>
<tr><td colspan='14' align='center'>
<p><b><font size=5 color = '#FFFFFF'>Account
Data</font></b></p>
</td></tr>
<tr>
<td align="center" bgcolor='#FF8000'
width='100'><b>Operation</b></td>
<td align="center" bgcolor='#FF8000'
width='80'><b>Code</b></td>
<td align="center" bgcolor='#FF8000'
width='300'><b>Account Name</b></td>
<td align="center" bgcolor='#FF8000' width='200'><b>Sub
Classification</b></td>
<td align="center"
bgcolor='#FF8000'><b>Classification</b></td>
<td align="center" bgcolor='#FF8000'
width='150'><b>Saldo</b></td>
</tr>
<?php
$query = "SELECT akun.kode, akun.nama_akun, akun.saldo,
kelompok_akun.nama_akun, sub_klas_akun.nama FROM (akun INNER
JOIN sub_klas_akun ON akun.sub_klasifikasi = sub_klas_akun.kode
INNER JOIN kelompok_akun ON sub_klas_akun.kelompok =
kelompok_akun.id) ORDER BY akun.kode";
$pg1 = new spa_paging; //create an instance of paging class
$query = $pg1->pagingSql($query); //parse the sql thru paging
class
$result = mysql_query($query);
$bagi = 0;
$num = 0;
if($result)
```

```

{
    $num = mysql_num_rows($result);
}
if($num!=0)
{
while($row = mysql_fetch_array($result)){
    $Kode = htmlspecialchars($row[0]);
    $Nama = htmlspecialchars($row[1]);
    $Saldo = htmlspecialchars($row[2]);
    $Kelompok_Akun = htmlspecialchars($row[3]);
    $Sub_Klasifikasi = htmlspecialchars($row[4]);

    if($bagi%2 == 0)
    {
        $td = "<td align = 'center' bgcolor =
'#999999'>";
    }
    else
    {
        $td = "<td align = 'center' bgcolor =
'#B0DFFF'>";
    }

    echo"<tr>$td<a href='delete_akun.php?Kode=$Kode'
target='_self' ><img src=images/Delete.png hspace='10' border=0
title=Delete_{$Kode} alt=Delete></a>";
    echo"<a href='edit_akun.php?Kode=$Kode'
target='_self' ><img src=images/edit.png hspace='10' border=0
title=Edit_{$Kode} alt=Edit></td>";
    echo"$td $Kode</a></td>";
    echo"$td $Nama</td>";
    echo"$td $Sub_Klasifikasi</td>";
    echo"$td $Kelompok_Akun</td>";
    echo"$td $cfg_curr
".number_format($Saldo,0,',','.') . "</td></tr>";
    $bagi++;
}
}
$shal = $pg1->getInfo();
echo "<tr><td colspan = '4' align = 'center'>$shal</td></tr>";

```

```

echo"<tr><td colspan = '2' align = 'center'><font size='4'
color = '#FFFFFF'>Total : $num</font></td>";
echo"<td align = 'center' colspan = '2'><a href
='add_akun.php'><font size='4' color = '#FFFFFF'>Add New
Account</font></a></td></tr></table>";

```

```

?>

```

```

<?php include("footer.php"); ?>
</body>
</html>

```

```

sub_klas.php

```

```

<?php include("login.php"); ?>
<html>
<head>
<title>STMIK Universitas Islam Antarbangsa</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<!--Fireworks MX 2004 Dreamweaver MX 2004 target. Created Wed
Jun 22 19:32:45 GMT+0800 (China Standard Time) 2005-->
<link type="text/css" href="paging.css" rel="stylesheet"
media="screen" />
<link type="text/css" href="scripts/themes/base/ui.all.css"
rel="stylesheet" />
<link href="styles.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="akuntansi.js"></script>
<script type="text/javascript" src="scripts/jquery-
1.4.js"></script>
<script type="text/javascript"
src="scripts/ui/ui.core.js"></script>
<script type="text/javascript"
src="scripts/ui/ui.datepicker.js"></script>
</head>
<?php include("header.php");
include_once("paging.class.php"); // include the paging class
?>
<table width="960" border="0" align="center" height = '195'>
<tr><td colspan='14' align='center'>
<p><b><font size=5 color = '#FFFFFF'>Sub Classification
Account Data</font></b></p>
</td></tr>
<tr>

```

```

        <td align="center" bgcolor='#FF8000'
width='100'><b>Operation</b></td>
        <td align="center" bgcolor='#FF8000'
width='80'><b>Code</b></td>
        <td align="center" bgcolor='#FF8000' width='300'><b>Sub
Classification Name</b></td>
        <td align="center"
bgcolor='#FF8000'><b>Classification</b></td>
    </tr>
<?php
$query = "SELECT sub_klas_akun.kode, sub_klas_akun.nama,
kelompok_akun.nama_akun FROM (sub_klas_akun INNER JOIN
kelompok_akun ON sub_klas_akun.kelompok = kelompok_akun.id)";
$pgl = new spa_paging; //create an instance of paging class
$query = $pgl->pagingSql($query); //parse the sql thru paging
class
$result = mysql_query($query);
$bagi = 0;
$num = 0;
if($result)
{
    $num = mysql_num_rows($result);
}
if($num!=0)
{
while($row = mysql_fetch_array($result)){
    $Kode = htmlspecialchars($row[0]);
    $Nama = htmlspecialchars($row[1]);
    $Kelompok_Akun = htmlspecialchars($row[2]);

    if($bagi%2 == 0)
    {
        $td = "<td align = 'center' bgcolor =
'#999999'>";
    }
    else
    {
        $td = "<td align = 'center' bgcolor =
'#B0DFFF'>";
    }
}
}

```

```

        echo"<tr>$td<a
href='delete_sub_klas_akun.php?Kode=$Kode' target='_self' ><img
src=images/Delete.png hspace='10' border=0 title=Delete_$Kode
alt=Delete></a>";
        echo"<a href='edit_sub_klas_akun.php?Kode=$Kode'
target='_self' ><img src=images/edit.png hspace='10' border=0
title=Edit_$Kode alt=Edit></td>";
        echo"$td $Kode</a></td>";
        echo"$td $Nama</td>";
        echo"$td $Kelompok_Akun</td></tr>";
        $bagi++;
    }
}
$hal = $pgl->getInfo();
echo "<tr><td colspan = '4' align = 'center'>$hal</td></tr>";

echo"<tr><td colspan = '2' align = 'center'><font size='4'
color = '#FFFFFF'>Total : $num</font></td>";
echo"<td align = 'center' colspan = '2'><a href
='add_sub_klas_akun.php'><font size='4' color = '#FFFFFF'>Add
New Sub Classification</font></a></td></tr></table>";

?>

<?php include("footer.php"); ?>
</body>
</html>

```

```

jurnal_tipe.php
<?php include("login.php"); ?>
<html>
<head>
<title>STMIK Universitas Islam Antarbangsa</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<!--Fireworks MX 2004 Dreamweaver MX 2004 target. Created Wed
Jun 22 19:32:45 GMT+0800 (China Standard Time) 2005-->
<link type="text/css" href="paging.css" rel="stylesheet"
media="screen" />
<link type="text/css" href="scripts/themes/base/ui.all.css"
rel="stylesheet" />
<link href="styles.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="akuntansi.js"></script>

```

```

<script type="text/javascript" src="scripts/jquery-1.4.js"></script>
<script type="text/javascript" src="scripts/ui/ui.core.js"></script>
<script type="text/javascript" src="scripts/ui/ui.datepicker.js"></script>
</head>
<?php include("header.php");
include_once("paging.class.php"); // include the paging class
?>
<table width="960" border="0" align="center" height = '195'>
<tr><td colspan="3" align="left">
<tr><td colspan="14" align="center">
<p><b><font size=5 color = '#FFFFFF'>Journal Type
Data</font></b></p>
</td></tr>
<tr>
<td align="center" bgcolor='#FF8000'
width='100'><b>Operation</b></td>
<td align="center" bgcolor='#FF8000'
width='80'><b>Code</b></td>
<td align="center" bgcolor='#FF8000'
width='300'><b>Journal Name</b></td>
</tr>
<?php
$query = "SELECT * FROM tipe_jurnal ORDER BY id";
$result = mysql_query($query);
$bagi = 0;
$num = 0;
if($result)
{
    $num = mysql_num_rows($result);
}
if($num!=0)
{
    while($row = mysql_fetch_array($result)){
        $Kode = htmlspecialchars($row[0]);
        $Nama = htmlspecialchars($row[1]);

        if($bagi%2 == 0)
        {
            $td = "<td align = 'center' bgcolor =
            '#999999'>";

```

```

        }
        else
        {
            $td = "<td align = 'center' bgcolor =
            '#B0DFFF'>";
        }

        echo"<tr>$td<a
href='delete_jurnal_tipe.php?Kode=$Kode' target='_self' ><img
src=images/delete.png hspace='10' border=0 title=Delete_$Kode
alt=Delete></a>";
        echo"<a href='edit_jurnal_tipe.php?Kode=$Kode'
target='_self' ><img src=images/edit.png hspace='10' border=0
title=Edit_$Kode alt=Edit></td>";
        echo"$td $Kode</a></td>";
        echo"$td $Nama</td>";
        $bagi++;
    }
}
echo"<tr><td colspan = '2' align = 'center'><font size='4'
color = '#FFFFFF'>Total : $num</font></td>";
echo"<td align = 'center' colspan = '2'><a href
='add_jurnal_tipe.php'><font size='4' color = '#FFFFFF'>Add New
Journal Type</font></a></td></tr></table>";

?>

<?php include("footer.php"); ?>
</body>
</html>

```

```

buku_besar_detail.php
<?php include("login.php"); ?>
<html>
<head>
<title>STMIK Universitas Islam Antarbangsa</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1">
<!--Fireworks MX 2004 Dreamweaver MX 2004 target. Created Wed
Jun 22 19:32:45 GMT+0800 (China Standard Time) 2005-->
<link href="styles.css" rel="stylesheet" type="text/css" />
<link type="text/css" href="paging.css" rel="stylesheet"
media="screen" />

```



```

<script language="JavaScript" src="akuntansi.js"
type="text/JavaScript">
</script>
</head>
<?php include("header.php");
include_once("paging.class.php");?>
<table width="950" border="0" align="center">
<tr><td colspan="14" align="center"><div>

<?php
$Akun = $_GET['Akun'];

$cari = "SELECT Nama_Akun FROM akun WHERE Kode = '$Akun'";
$dapat = mysql_query($cari);
while($baris = mysql_fetch_array($dapat))
{
    $Nama_Akun = htmlspecialchars($baris[0]);
}

echo"<p><b><font size=5 color = '#FFFFFF'>General Ledger
Details :  $Nama_Akun</font></b></p>
</div></td></tr>
<tr>;
    echo"<td align='center'
bgcolor='#FF8000'><b>Date</b></td>";
    echo"<td align='center'
bgcolor='#FF8000'><b>Description</b></td>";
    echo"<td align='center' bgcolor='#FF8000'><b>Journal
Tipe</b></td>";
    echo"<td align='center'
bgcolor='#FF8000'><b>Debit</b></td>";
    echo"<td align='center'
bgcolor='#FF8000'><b>Credit</b></td>";
    echo"<td align='center'
bgcolor='#FF8000'><b>D/C</b></td>";
    echo"<td align='center'
bgcolor='#FF8000'><b>Saldo</b></td>";
    echo"</tr>";
$bagi = 0;
//echo"Akun yang dipilih adalah : $Akun";
$query = "SELECT * FROM jurnal_detail WHERE akun_id = '$Akun'";
//$pgl = new spa_paging; //create an instance of paging class

```

```

//$query = $pgl->pagingSql($query); //parse the sql thru paging
class
$result = mysql_query($query);
if($result)
{
    while($row=mysql_fetch_array($result))
    {
        $Jurnal_ID = htmlspecialchars($row[1]);
        $Akun_ID = htmlspecialchars($row[3]);
        $Deb_Kre = htmlspecialchars($row[4]);
        $nilai = htmlspecialchars($row[5]);

        //$Saldo = $nilai;
        $query2 = "SELECT Tipe,Tanggal,Deskripsi FROM
jurnal WHERE ID = '$Jurnal_ID'";
        $result2 = mysql_query($query2);

        while($row2 = mysql_fetch_array($result2))
        {
            $Tipe = htmlspecialchars($row2[0]);
            $Tanggal = htmlspecialchars($row2[1]);
            $Deskripsi = htmlspecialchars($row2[2]);

            $query3 = "SELECT Nama FROM tipe_jurnal
WHERE ID = '$Tipe'";
            $result3 = mysql_query($query3);

            while($row3 =
mysql_fetch_array($result3))
            {
                $Tipe_Jurnal =
htmlspecialchars($row3[0]);

                //ambil Tipe Akun
                $query4 = "SELECT
kelompok_akun.id FROM (kelompok_akun INNER JOIN sub_klas_akun
ON kelompok_akun.id = sub_klas_akun.kelompok INNER JOIN akun ON
sub_klas_akun.kode = akun.sub_klasifikasi) WHERE akun.kode =
'$Akun_ID'";

                //echo"Query 4 adalah :

$query4<br>";

                $result4 = mysql_query($query4);
                while($row4 =
mysql_fetch_array($result4))

```

```

{
    $tipe_akun = $nilai;
    //echo"Tipe Akun adalah :
    $Saldo = $Saldo +
    if($Saldo >= 0)
    {
        echo"$td
    }
    echo"$td
    $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
    }
    else if($Saldo <
    {
        echo"$td
    }
    echo"$td
    $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
    }
}
else if ($Deb_Kre ==
{
    $Saldo = $Saldo -
    {
        echo"$td
    }
    echo"$td
    $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
    }
    else if($Saldo <
    {
        echo"$td
    }
    echo"$td
    $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
    }
}

htmlspecialchars($row4[0]);
$tipe_akun<br>";

if($bagi%2 == 0)
{
    $td = "<td align =
    0)
    C</td>";
    $td = "<td align =
    $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
    }

echo"<tr>$td $Tanggal</td>";
echo"$td $Deskripsi</td>";
echo"$td $Tipe_Jurnal</td>";
//pilih nilai untuk debit atau
kredit
if($Deb_Kre == 0)
{
    echo"$td </td>$td
    $cfg_curr ".number_format($nilai,0,',','.')."</td>";
    }
    else if($Deb_Kre == 1)
    {
        echo"$td $cfg_curr
        ".number_format($nilai,0,',','.')."</td>$td</td>";
    }

    //Pilih akun debit atau kredit.
    Proses nilai yang ada.
    if($tipe_akun == 1 || $tipe_akun
    == 5)// akun yang debit tambah kredit kurang
    {
        if($Deb_Kre == 1)//Nilai
        Debit
        {

```

```

else if($tipe_akun == 2 || $tipe_akun == 3 || $tipe_akun == 4)//Akun yang Kredit bertambah
Debit berkurang
{
    if($Deb_Kre == 0)//Nilai
    {
        $Saldo = $Saldo +
        if($Saldo >= 0)
        {
            echo"$td
            echo"$td
            $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
        }
        else if($Saldo <
        {
            echo"$td
            echo"$td
            $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
        }
    }
    else if ($Deb_Kre ==
    {
        $Saldo = $Saldo -
        if($Saldo >= 0)
        {
            echo"$td
            echo"$td
            $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
        }
        else if($Saldo <
        {
            echo"$td
            echo"$td
            $cfg_curr ".number_format($Saldo,0,',','.')."</td></tr>";
        }
    }
}

//setcookie('saldo',$Saldo);
$bagi++;
}

//$shal = $pg1->getInfo();
if($shal == 1)
{
    echo"<br>Masih 1";
}
else if($shal == 2)
{
    echo"<br>Sudah 2";
}
echo "<tr><td colspan = '4' align = 'center'>$shal</td></tr>";
?>
<?php include("footer.php"); ?>
</body>
</html>

```

## DAFTAR LAMPIRAN

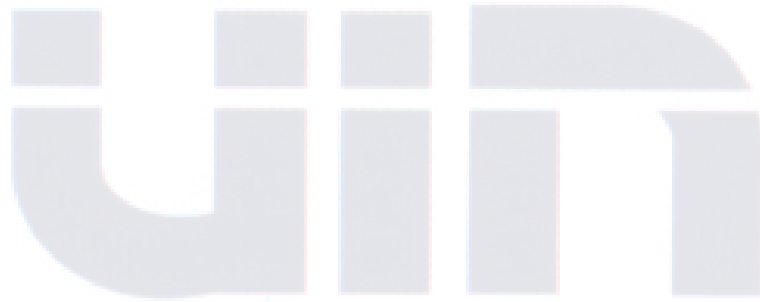
Lampiran 1 (Surat Keterangan Penelitian)

Lampiran 2 (Surat Keterangan Penunjukkan Dosen Pembimbing)

Lampiran 3 (Daftar Pertanyaan Wawancara)

Lampiran 4 (*Black Box Testing*)

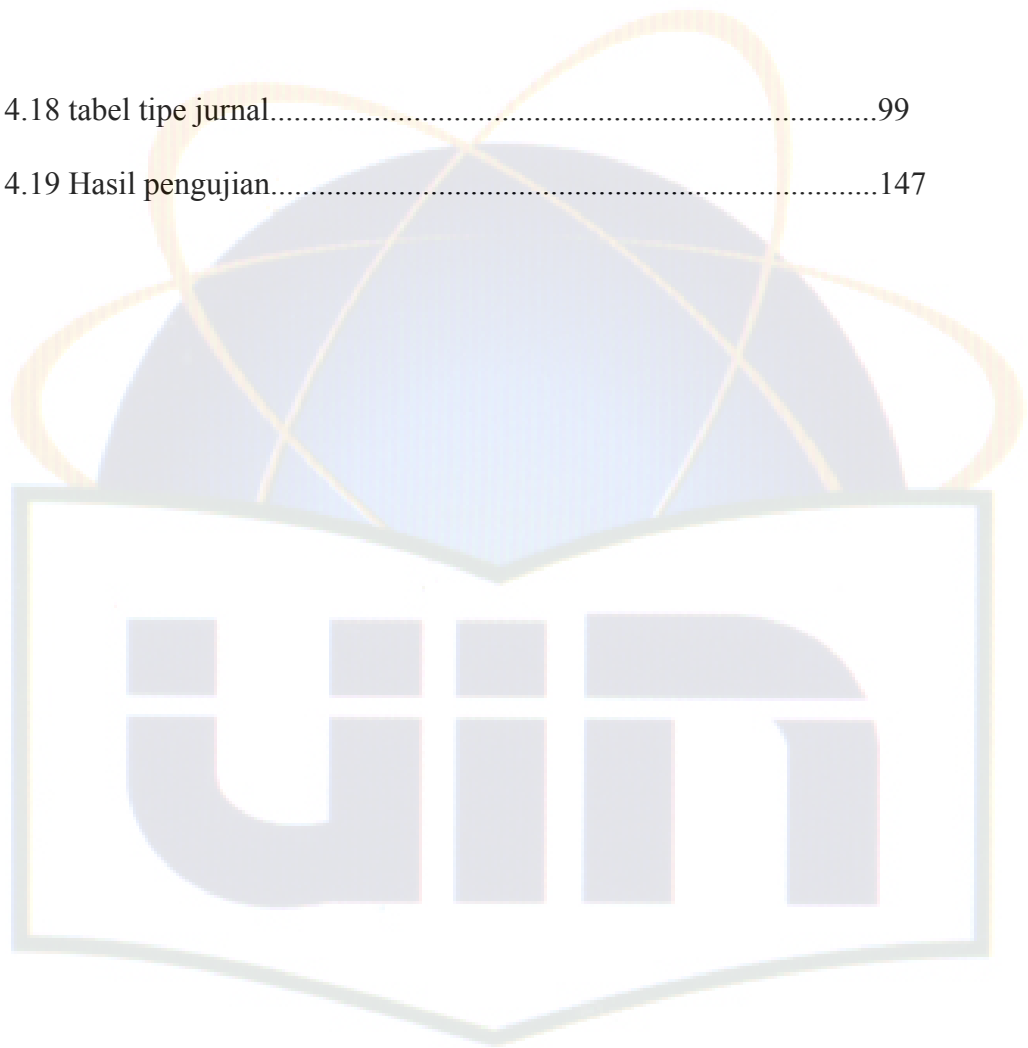
Lampiran 5 (Contoh Kode Program)



## DAFTAR TABEL

	Halaman
Tabel 2.1 Tabel karakteristik rekening.....	17
Tabel 2.2 Simbol Penghubung.....	27
Tabel 2.3 Simbol Proses <i>Flowchart</i> .....	29
Tabel 2.4 Simbol input output <i>flowchart</i> .....	30
Tabel 4.1 Table transaksi <i>unnormalized</i> .....	85
Tabel 4.2 Tabel transaksi 1NF.....	87
Tabel 4.3 Tabel tipe_jurnal 2NF.....	89
Tabel 4.4 Tabel jurnal 2NF.....	89
Tabel 4.5 Tabel akun 2NF.....	90
Tabel 4.6 Tabel tipe_jurnal 3NF.....	92
Tabel 4.7 Tabel jurnal 3NF.....	92
Tabel 4.8 Tabel jurnal_detail 3NF.....	92
Tabel 4.9 Tabel kelompok_akun 3NF.....	93
Tabel 4.10 Tabel sub_klas_akun 3NF.....	93
Tabel 4.11 Tabel akun 3NF .....	93
Tabel 4.12 Tabel kelompok akun.....	95
Tabel 4.13 Tabel sub_klas_akun.....	96
Tabel 4.14 tabel akun.....	96
Tabel 4.15 tabel jurnal.....	97
Tabel 4.16 tabel jurnal_detail.....	98
Tabel 4.17 tabel tipe jurnal.....	99

Tabel 4.18 tabel tipe jurnal.....	99
Tabel 4.19 Hasil pengujian.....	147



# ANALISIS DAN PEMBUATAN APLIKASI AKUNTANSI BERBASIS WEB PADA IIUC

Arif Kunto Wibisono (106091002876)

Teknik Informatika, Universitas Islam Negeri Syarif Hidayatullah Jakarta  
Di bawah bimbingan Yusuf Durachman, M.Sc, MIT dan Rusdianto  
Rustam, M.Sc, Dr.

## ABSTRAK

Saat ini, pemasukan data-data transaksi keuangan yang berlangsung pada IIUC dilakukan secara *manual* oleh staf keuangan. Transaksi keuangan hanya bisa dilakukan pada komputer lokal yang ada pada IIUC saja, sehingga bila staf keuangan atau orang yang bersangkutan sedang berada di luar IIUC, maka ia tidak bisa melakukan transaksi. Ini merupakan suatu hambatan bagi manajer, karena data – data ini sangat penting untuk mengetahui keadaan keuangan saat ini pada IIUC. Melihat permasalahan yang terjadi, maka perlu dibuat aplikasi akuntansi yang dapat membantu pekerjaan bagian keuangan dan serta manajer dalam melihat informasi keuangan. Metode pengembangan sistem yang dipakai dalam penelitian ini adalah *incremental*. Dalam aplikasi ini terdapat logika – logika akuntansi yang digunakan untuk menghitung transaksi – transaksi serta saldo dari rekening yang digunakan. Hasil dari aplikasi ini adalah laporan keuangan yang ditujukan kepada manajer. Pada akhirnya, aplikasi ini dapat membantu pekerjaan bagian keuangan dalam memasukkan transaksi dan manajer dalam melihat laporan keuangan.

Kata Kunci : Analisis, Perancangan, Aplikasi, Akuntansi, Web.

## I. PENDAHULUAN

### 1.1 LATAR BELAKANG

Saat ini kesuksesan organisasi sangat bergantung kepada kemampuan mengumpulkan, memburu, dan mengorganisasi data mengenai operasi – operasi perusahaan secara akurat dan tepat, mampu mengelola data secara efektif, serta menggunakannya untuk analisis dan menuntun aktivitas – aktivitas perusahaan. Jumlah informasi yang telah tersedia telah meledak sedemikian besar serta pandangan data sebagai aset organisasi telah mulai tumbuh. Namun tanpa kemampuan yang memadai dalam pengelolaan data sehingga data mampu memberi informasi yang relevan secara cepat terhadap suatu persoalan, maka jumlah data yang besar justru akan menjadi beban yang harus ditanggung perusahaan. (Hariyanto, 2004 : 3)

Saat ini, pemasukan data-data transaksi keuangan yang berlangsung pada IIUC dilakukan secara *manual* oleh staf

keuangan. Transaksi keuangan hanya bisa dilakukan pada komputer lokal yang ada pada IIUC saja, sehingga bila staf keuangan atau orang yang bersangkutan sedang berada di luar IIUC, maka ia tidak bisa melakukan transaksi. Ini merupakan suatu hambatan bagi manajer, karena data – data ini sangat penting untuk mengetahui keadaan keuangan saat ini pada IIUC.

Dengan pembuatan aplikasi baru diharapkan akan mempermudah memproses transaksi keuangan. Para staf keuangan dapat memproses transaksi keuangan secara cepat, manajer dapat melihat keadaan keuangan meskipun tidak berada pada kantor pusat, dan pemeliharaan data dapat dilakukan dengan lebih mudah.

Berdasarkan hal inilah, penulis melakukan penelitian skripsi dengan judul

## **Analisis Dan Pembuatan Aplikasi Akuntansi Berbasis Web Pada IIUC**

### **1.2 Rumusan Masalah**

Sesuai dengan permasalahan yang diangkat pada latar belakang penulisan, maka penulis merumuskan permasalahan yang ada yaitu :

1. Bagaimana membuat aplikasi akuntansi berbasis *web*.
2. Bagaimana membuat aplikasi akuntansi yang bisa melakukan pemeliharaan data secara lebih teratur.
3. Bagaimana cara penyajian aplikasi akuntansi ini secara mudah dan interaktif.

### **1.3 Batasan Masalah**

Dalam penelitian ini terdapat batasan, yaitu :

1. Pencatatan transaksi keuangan yang ada ke dalam jurnal.
2. Pemrosesan transaksi keuangan yang berlangsung.
3. Pembuatan laporan keuangan dari transaksi – transaksi yang telah berlangsung ke dalam Microsoft Excel.
4. Laporan keuangan yang dibuat adalah neraca saldo, laporan laba rugi, dan laporan perubahan ekuitas.
5. Aplikasi keuangan dibuat untuk lingkungan IIUC.

### **1.4 Tujuan Penelitian**

Berdasarkan uraian latar belakang, maka tujuan penelitian ini adalah untuk membantu bagian keuangan pada IIUC untuk memproses transaksi yang ada serta membuat laporan keuangan secara otomatis.

## **II. LANDASAN TEORI**

### **2.1 Akuntansi**

Akuntansi adalah proses pengidentifikasian, pencatatan, dan pengkomunikasian kejadian – kejadian ekonomi suatu organisasi (perusahaan

ataupun bukan perusahaan) kepada para pemakai informasi yang berkepentingan. Akuntansi memberikan jasa yang sangat vital dengan memasok informasi yang dibutuhkan oleh para pengambil keputusan untuk membuat “pilihan – pilihan yang masuk akal di antara alternatif – alternatif penggunaan sumber – sumber daya yang langka dalam menjalankan bisnis dan kegiatan - kegiatan ekonomi”(Simamora, 2000 : 4). Pengertian langka disini adalah sumber daya tersebut istimewa. Sumber daya itu ada dalam perusahaan, namun penggunaannya untuk aktivitas ekonomi tidak dapat langsung dipakai oleh karyawan. Misalkan ketika ingin mengambil dana dari kas untuk pembelian komputer, maka karyawan harus melapor kepada bagian keuangan untuk mendapat izin. Ketika melakukan penjualan barang produksi, maka karyawan harus melapor ke bagian keuangan untuk dicatat pendapatannya.

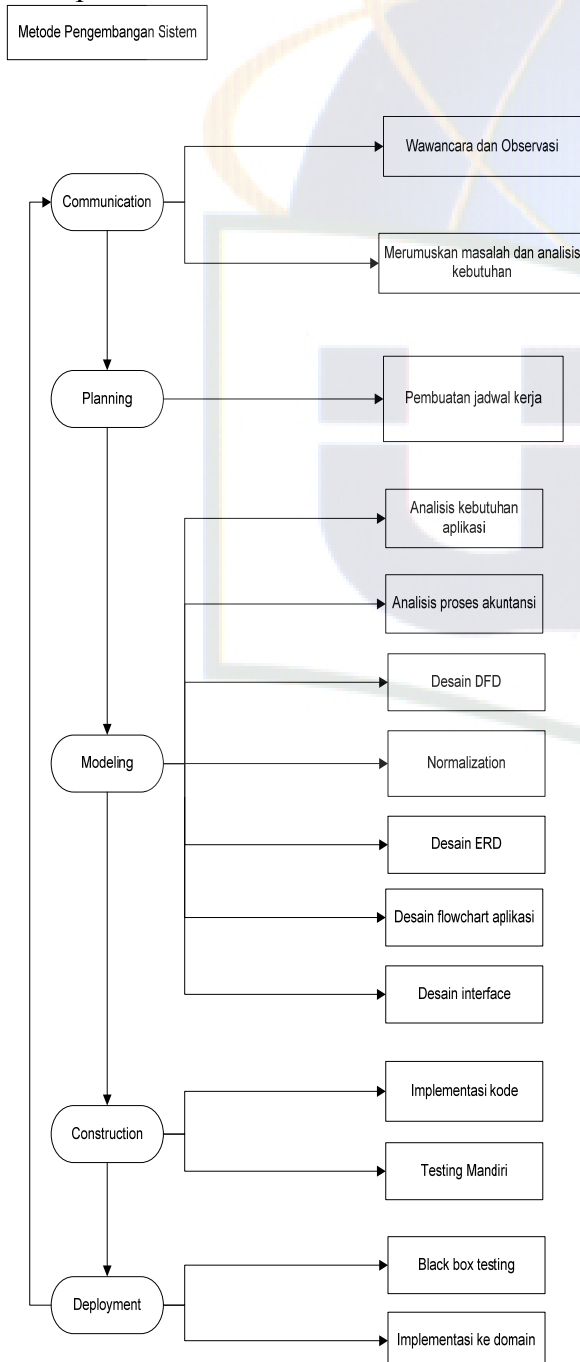
Akuntansi dapat didefinisikan sebagai sistem informasi yang menghasilkan laporan kepada pihak – pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi perusahaan.(Warren, *et al.*2006 : 10). Aktivitas ekonomi disini adalah aktivitas perusahaan dalam menggunakan sumber daya perusahaan untuk kelangsungan hidup perusahaan tersebut. Misalkan pembelian peralatan, penggunaan kas, penjualan saham, dan lain – lain. Kondisi perusahaan yang dimaksud disini yaitu keadaan keuangan pada perusahaan tersebut. Dari pencatatan aktivitas – aktivitas ekonomi tersebut, maka dapat diketahui pendapatan, pengeluaran, laba, rugi, aktiva dan pasiva dalam perusahaan tersebut.



### III. METODOLOGI PENELITIAN

#### 3.1 Kerangka Berpikir

Dalam melakukan penelitian ini, penulis melakukan tahapan – tahapan yang tertulis pada kerangka berpikir yang meliputi metode pengumpulan data dan metode pengembangan aplikasi.



Gambar 3.1 Kerangka Berpikir

#### 3.2 Metode Pengumpulan Data

##### 1. Studi Lapangan atau Observasi

Observasi merupakan teknik penelusuran fakta dimana analis sistem berpartisipasi atau melihat seseorang melakukan aktivitas untuk mempelajari sistem (Whitten, 2004 : 245). Observasi yang dilakukan, adalah melakukan tinjauan langsung kelapangan guna mendapatkan informasi dan fakta pendukung dalam penelitian. Sasaran dari observasi adalah :

1. Mengetahui sistem keuangan yang telah berjalan pada IIUC.
2. Mengetahui proses akuntansi yang ada di IIUC.

Penelitian dilakukan di IIUC, Jl. Melawai Raya no. 8C Kebayoran Baru, Jakarta Selatan dari tanggal 3 Juli 2010 sampai dengan 30 Oktober 2010.

##### 2. Studi Pustaka

Studi Pustaka, yakni mengumpulkan data primer dan referensi melalui literatur, buku, artikel maupun secara online menggunakan media *internet* untuk mendapatkan referensi yang berhubungan dengan penulisan skripsi ini. Sasaran dari tahapan ini adalah :

1. Mencari literatur – literatur yang berhubungan dengan akuntansi, PHP, MySQL, dan jQuery.
  2. Mencari studi sejenis yang berhubungan dengan aplikasi akuntansi.
- ##### 3. Wawancara

Wawancara merupakan teknik penelusuran fakta dimana analis sistem mengumpulkan informasi dari individu - individu melalui interaksi *face to face* (Whitten, 2004 : 250).

Pada tahap ini dilakukan wawancara dengan tujuan untuk mengumpulkan data, dan data yang diperoleh adalah melalui tanya jawab dengan Ibu Shinta Mardhatillah, selaku Manajer

Akademik. Sasaran dari tahapan ini adalah :

1. Mendapatkan informasi tentang sistem keuangan di IIUC.
2. Pemanfaatan teknologi informasi pada bagian keuangan IIUC.
3. Rencana penggunaan aplikasi berbasis *web* untuk menunjang keuangan di IIUC.

### 3.3 Metode Pengembangan Aplikasi

Dalam penelitian ini, penulis menggunakan metode penelitian *incremental*. Ada banyak situasi dimana syarat awal sebuah *software* telah didefinisikan secara jelas, namun lingkungan pengembangannya secara umum tidak selalu bisa linier. Sebagai tambahan, bisa saja ada kebutuhan mendesak untuk memberikan beberapa fungsi dari *software* kepada *user*, kemudian diperbaiki, dan dikembangkan dalam fungsi tersebut dalam pengembangan *software* selanjutnya. Dalam kasus ini, pengembang bisa memilih model proses yang didesain untuk memproduksi *software* dalam penambahan – penambahan. (Pressman, 2010 : 41)

Penulis memilih metode ini karena masalah yang dihadapi sudah dimengerti dan ditetapkan dari awal, namun dalam prosesnya terdapat penambahan – penambahan dalam penyempurnaan aplikasi.

Tahapan – tahapan dari proses *incremental* adalah :

#### 1. *Communication*

Tahap pertama adalah komunikasi, dimana ini sangat penting untuk mendapatkan syarat –syarat kebutuhan dari *user*. Target dari tahapan ini adalah mendapatkan kebutuhan dari *user* yaitu aplikasi akuntansi berbasis *web*, dimana didalam aplikasi tersebut dapat membuat laporan

keuangan dan penginformasian transaksi – transaksi yang ada.

#### 2. *Planning*

Disini dibuat jadwal – jadwal perencanaan untuk membuat aplikasi tersebut. Target dari tahapan ini adalah membuat penjadwalan dari tahap *planning* , konstruksi, dan sampai tahap *deployment*.

#### 3. *Modeling*

Tahap ini bertujuan untuk mendesain aplikasi yang akan dibuat. Setelah informasi keuangan diformulasikan secara lengkap, kemudian diimplementasikan dengan membuat perancangan aplikasi yang akan dibangun. Target dari tahapan ini, yakni:

1. Menganalisis kebutuhan aplikasi yang akan dibuat.
2. Menganalisis tahapan – tahapan proses akuntansi di IIUC.
3. Merancang *normalization*, *entity relationship diagram*, *data flow diagram*, *kamus data*, dan *flowchart*.
4. Merancang *user interface*

#### 4. *Construction*

Target dari tahap konstruksi adalah :

1. Mengimplementasikan kode – kode program.
2. Menguji aplikasi ini secara mandiri.

#### 5. *Deployment*

Pada tahap ini dilakukan pengujian kepada *user*. Target dari tahapan ini adalah :

1. Bersama – sama dengan *user* untuk melakukan *blackbox testing*. *Black box testing* dilakukan dengan Ibu Shinta Mardhatillah.
2. Meng-*upload* aplikasi yang telah dites kedalam *hosting*

## IV. HASIL DAN PEMBAHASAN

#### 4.1 Communication

Melalui wawancara dan observasi yang dilakukan, diketahui bahwa sistem keuangan di IIUC bersifat *manual*, yaitu dengan menggunakan Microsoft Excel. Maka disimpulkan permasalahan yang didapat adalah pembuatan aplikasi akuntansi berbasis *web*. Berangkat dari hasil observasi yang dilakukan kemudian dilakukan pengkajian dan pembatasan masalah yang akan diimplementasikan ke dalam aplikasi akuntansi ini.

#### 4.2 Planning

Dalam tahap ini dibuat penjadwalan untuk membuat aplikasi. Beberapa tahapan kerja yang akan dilakukan antara lain :

1. Analisis kebutuhan aplikasi yang akan dibuat
2. Analisis tahapan proses akuntansi di IIUC
3. Perancangan DFD.
4. *Normalization*
5. Perancangan ERD.
6. Perancangan *flowchart*.
7. Perancangan *interface* serta pengkodean aplikasi.
8. *Testing* kepada *user* untuk mendapatkan *feedback*.

#### 4.3 Modelling

##### 4.3.1 Analisis Kebutuhan Aplikasi

Berdasarkan wawancara dan observasi, maka aplikasi yang akan dibuat adalah aplikasi akuntansi berbasis *web*. *Web server* yang digunakan adalah Apache, menggunakan bahasa PHP dan *database* MySQL. Semua ini dapat di-*install* menggunakan XAMPP. Kemudian untuk membuat *chart* menggunakan FusionChart Free dan menggunakan jQuery untuk membuat aplikasi lebih interaktif.

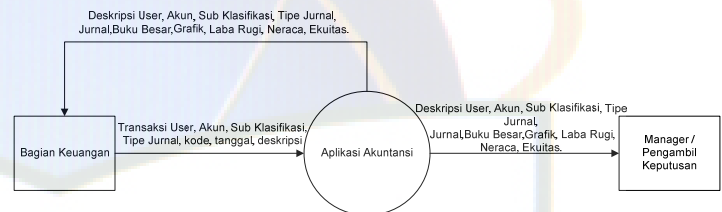
##### 4.3.2 Analisis Proses Akuntansi

Laporan Keuangan dibuat berdasarkan tahapan – tahapan dalam mengolah informasi keuangan. Tahapan – tahapan itu antara lain :

1. Gambaran proses akuntansi
2. Mekanisme pencatatan transaksi
3. Proses akuntansi
4. Target akhir

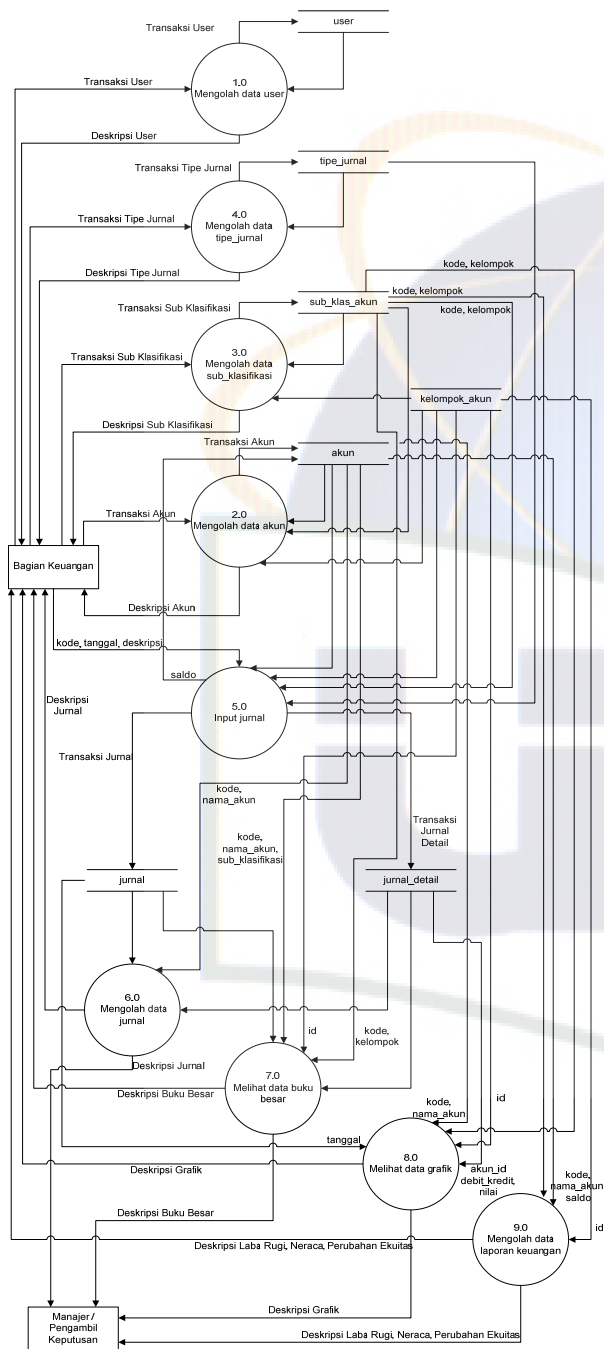
##### 4.3.3 Data Flow Diagram

Penulis menggunakan *data flow diagram* untuk mengetahui alur data yang mengalir dari aplikasi yang dibuat. *Data flow diagram* yang dibuat antara lain :

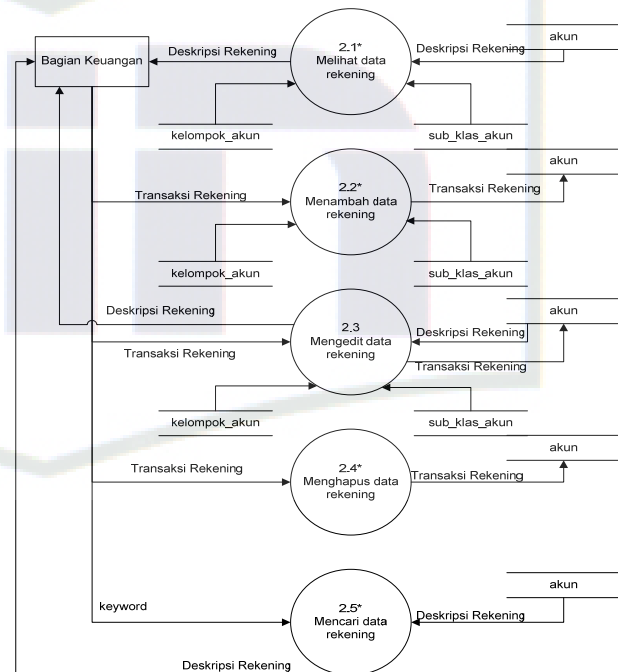
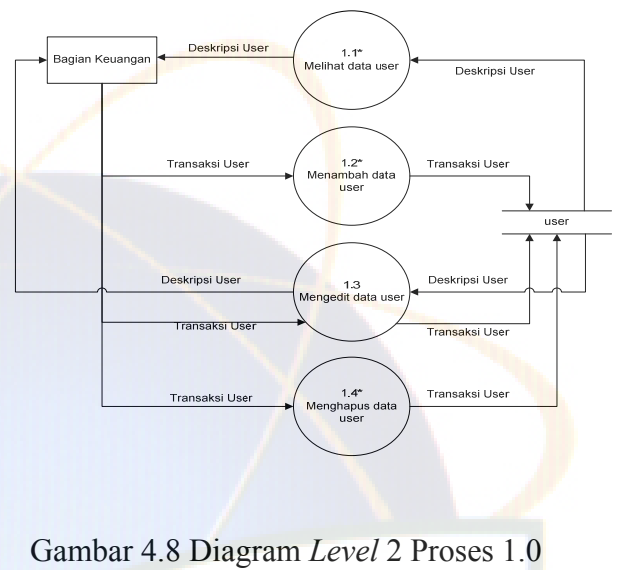


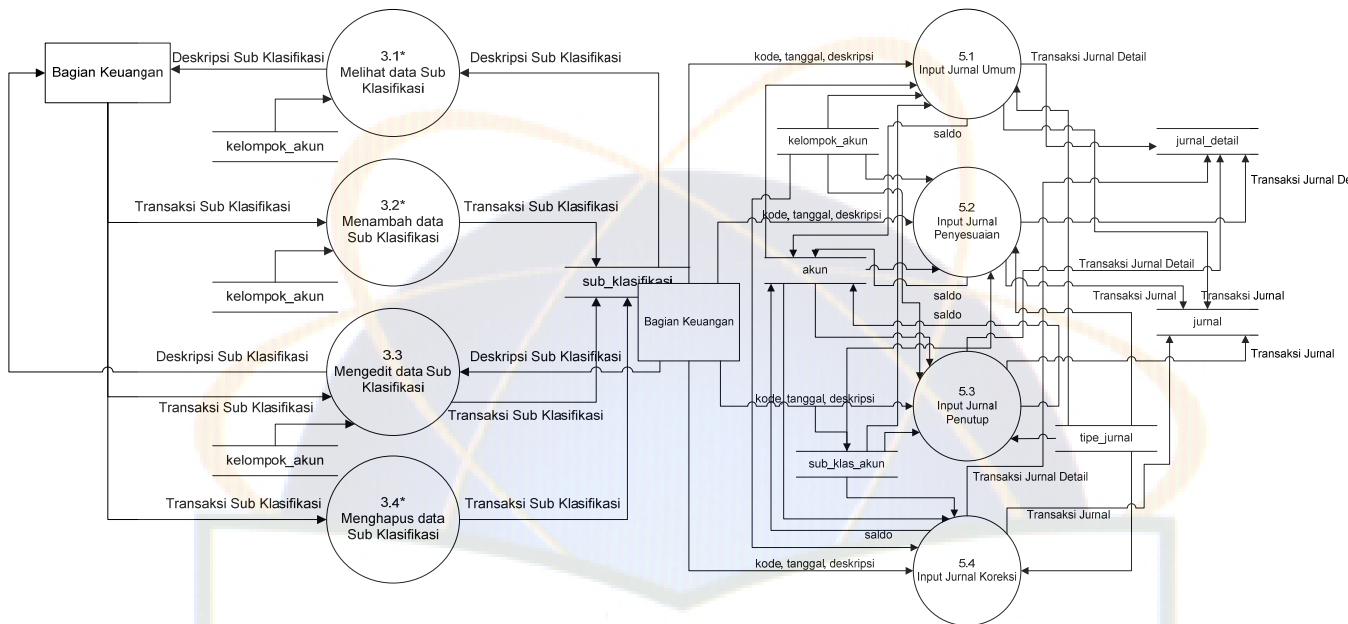
Gambar 4.6 Diagram Konteks Aplikasi Akuntansi

Dalam diagram konteks diatas terdapat dua buah entitas, yaitu bagian keuangan dan *manager* atau pengambil keputusan. Dalam hal ini, yang bisa memasukkan transaksi hanyalah bagian keuangan saja, sedangkan *manager* tidak bisa memasukkan transaksi. *Manager* hanya bisa menampilkan data – data tentang keuangan serta melihat laporan keuangan.



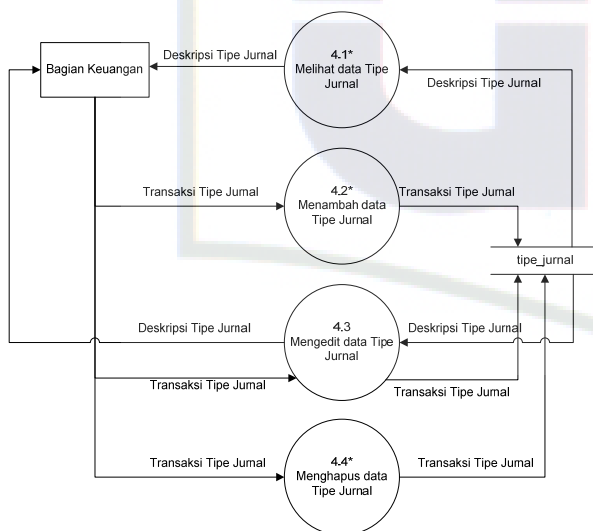
Gambar 4.7 Diagram *Level 1* Aplikasi Akuntansi





Gambar 4.12 Diagram *Level 2* Proses 3.0

Gambar 4.16 Diagram *Level 2* Proses 5.0



Gambar 4.14 Diagram *Level 2* Proses 4.0

#### 4.3.4 Normalization

Berikut akan dijelaskan proses normalisasi, mulai dari tabel yang belum dinormalisasi sampai 3NF.

##### 1. *Unnormalized Form*

Tahap pertama ini, semua data transaksi akan dicatat. Pada tabel berikut, kolom item, tipe akun, subklasifikasi, kode\_akun, nama\_akun, debit\_kredit, nilai, dan saldo belum memiliki nilai atomik. Ini akan diperbaharui pada bentuk 1NF.

##### 2. *First Normal Form (1NF)*

Syarat pada 1NF adalah semua kolom data harus memiliki nilai atomik. Maka dari itu, kolom item, kode\_tipe\_akun, tipe\_akun, kode\_sub\_klasifikasi, subklasifikasi, kode\_akun, nama\_akun, debit\_kredit, nilai, dan saldo harus diubah nilainya untuk mendapatkan nilai atomik. Untuk mendapatkan nilai atomik itu, maka data kedua yang terdapat pada



kolom tersebut akan dipindahkan untuk dibuat baris baru.

### 3. Second Normal Form (2NF)

Persyaratan pada 2NF adalah harus berada dalam 1NF dan menghilangkan dependensi parsial. Data yang bukan *primary key* harus memiliki dependensi fungsional terhadap *primary key*. Berikut adalah *primary key* tersebut beserta kolom yang dipengaruhi :

- a. Kode\_Tipe -> Tipe\_Jurnal
- b. Kode -> Tanggal, Deskripsi, Item, Debit\_Kredit, Kode\_Akun, Nilai
- c. Kode\_Tipe\_Akun -> Tipe\_Akun, Kode\_Sub\_Klasifikasi, Subklasifikasi, Kode\_Akun, Nama\_Akun, Saldo

Maka dari tabel transaksi, akan dipecah dan dibuat tiga tabel baru, yaitu tabel tipe\_jurnal, jurnal, kemudian kelompok\_akun.

### 4. Third Normal Form (3NF)

Persyaratan 3NF adalah harus berada dalam 1NF dan 2NF, kemudian penghilangan dependensi transitif. Data yang bukan *primary key* tidak boleh memiliki dependensi transitif dengan *primary key*. Pada tabel tipe\_jurnal, nama berdependensi parsial dengan id, kemudian tidak ada dependensi transitif. Maka tabel tipe\_jurnal sudah berada dalam 3NF.

Pada tabel jurnal, tanggal, deskripsi, item, kode\_akun, debit\_kredit, dan nilai berdependensi parsial dengan kolom id dan berdependensi transitif dengan kolom tipe. Tabel jurnal ini kemudian

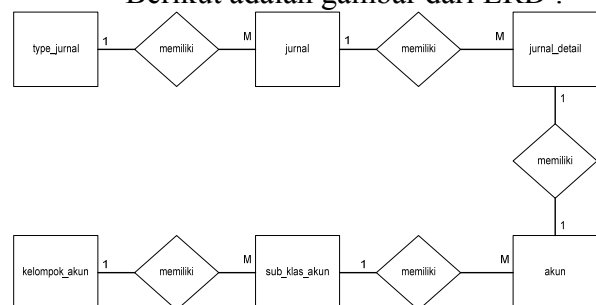
dipecah menjadi dua, yaitu tabel jurnal dan tabel jurnal\_detail untuk menghilangkan dependensi transitif tersebut.

Pada tabel akun, kolom sub\_klasifikasi berdependensi parsial dengan kolom kode\_subklasifikasi dan berdependensi transitif dengan kolom kode\_tipe\_akun. Kemudian kolom nama\_akun berdependensi parsial dengan kode\_akun dan berdependensi transitif dengan kode\_tipe\_akun. Maka dari itu, dependensi transitif ini harus dihilangkan dengan membuat tabel baru. Ini adalah komposisi dari tabel – tabel tersebut :

- a. Tabel tipe\_jurnal  
id -> nama.
- b. Tabel jurnal  
id -> tipe, tanggal, deskripsi.
- c. Tabel jurnal\_detail  
id -> jurnal\_id, item, akun\_id, debit\_kredit, nilai.
- d. Tabel kelompok\_akun  
id -> nama\_akun
- e. Tabel sub\_klas\_akun  
kode -> kelompok, nama.
- f. Table akun  
kode -> nama\_akun, sub\_klasifikasi, saldo

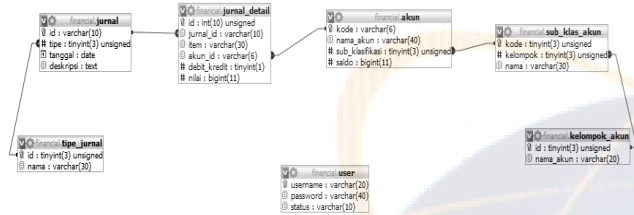
### 4.3.5 Entity Relationship Diagram

Berikut adalah gambar dari ERD :



Gambar 4.24 Model ERD

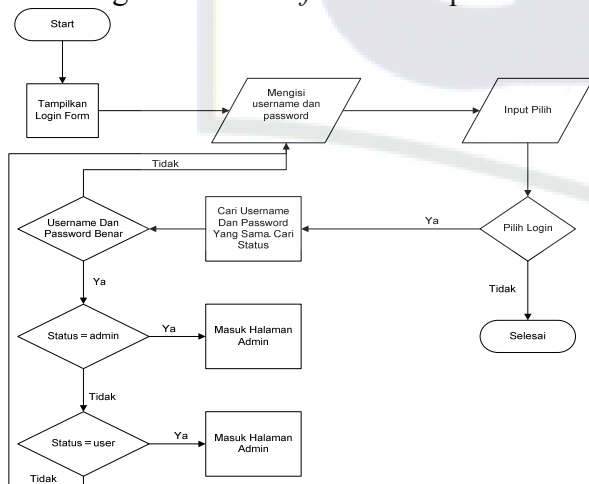
Berikut adalah penerjemahan ERD ke LRS



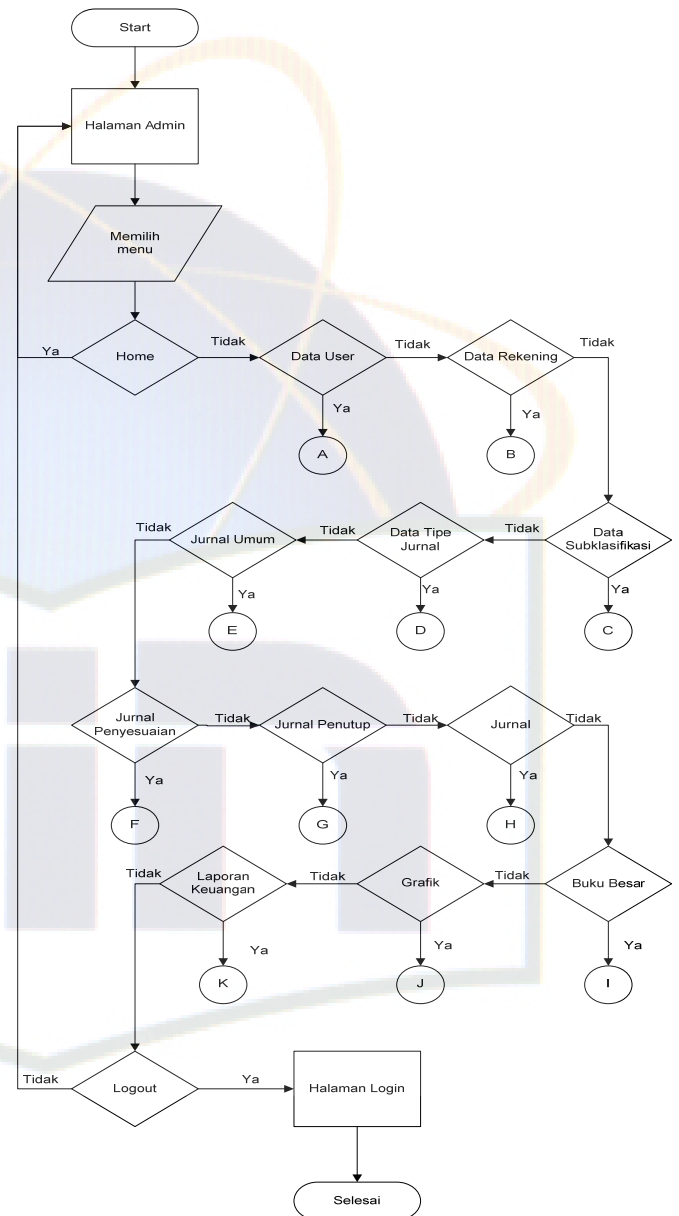
Seperti terlihat diatas, terdapat tujuh buah tabel. Semua tabel tersebut berhubungan dengan tabel yang lain, kecuali tabel *user*. Tabel jurnal memiliki *foreign key* dari tabel tipe\_jurnal. Tabel akun memiliki *foreign key* dari tabel sub\_klas\_akun. Tabel sub\_klas\_akun memiliki *foreign key* dari tabel kelompok\_akun. Sedangkan tabel jurnal\_detail memiliki *foreign key* dari tabel jurnal dan tabel akun.

#### 4.3.6 Flowchart

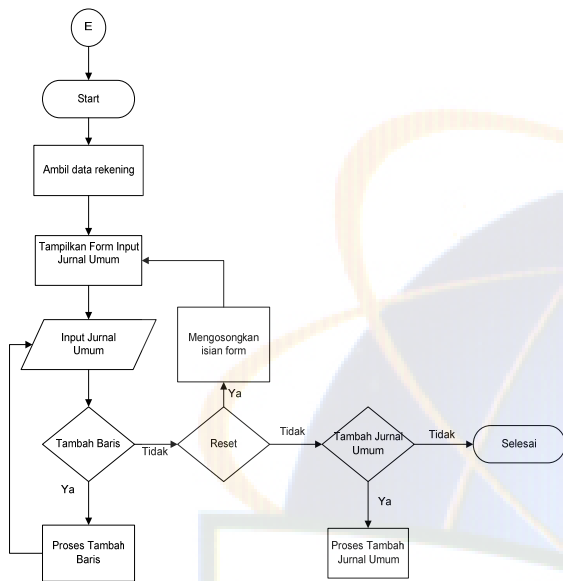
Berikut ini adalah beberapa gambaran dari *flowchart* aplikasi :



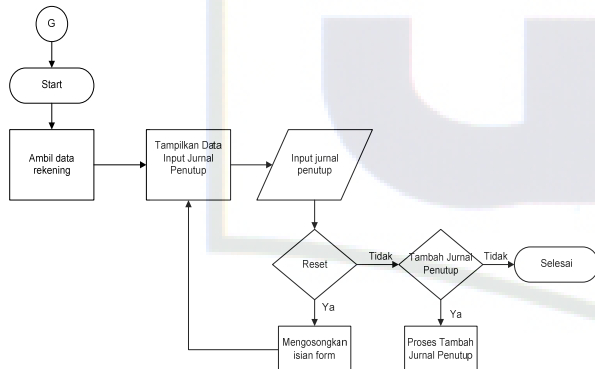
Gambar 4.25 *Flowchart* Halaman Login



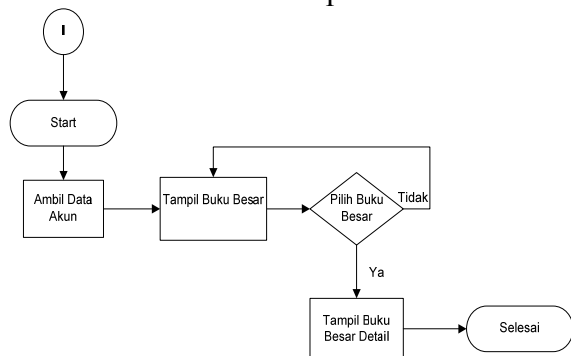
Gambar 4.26 *Flowchart* Halaman Admin



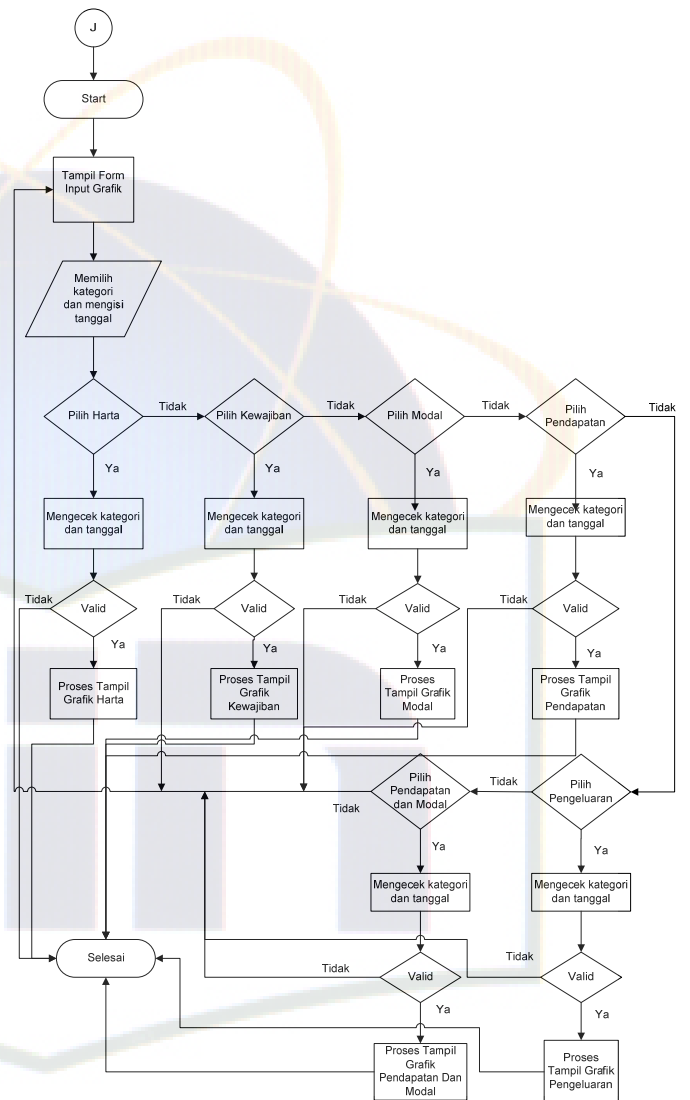
Gambar 4.45 Flowchart Halaman Jurnal Umum



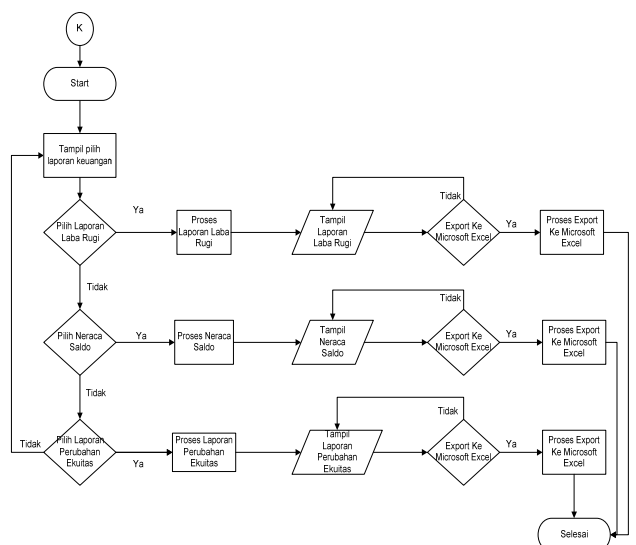
Gambar 4.51 Flowchart Halaman Jurnal Penutup



Gambar 4.61 Flowchart Halaman Data Buku Besar



Gambar 4.62 Flowchart Halaman Grafik



Gambar 4.69 Flowchart Halaman Laporan Keuangan



#### 4.4 Construction

Dalam fase konstruksi, bahasa yang digunakan adalah PHP dengan menggunakan *database* MySQL.

Dalam pengkodean ini, penulis menggunakan aturan – aturan akuntansi untuk memasukkan transaksi – transaksi ke dalam jurnal. Kemudian dari jurnal – jurnal tersebut akan dibuat laporan keuangan. Dari laporan keuangan tersebut dapat diketahui kondisi keuangan IIUC saat ini. Pada tahap ini penulis juga melakukan *testing* mandiri untuk mengetahui jalannya program yang telah dibuat.

#### 4.5 Deployment

##### 4.5.1 Testing

Pada tahapan ini, penulis mengadakan *blackbox testing* untuk menguji aplikasi yang telah dibuat. Cara pengujian *blackbox testing* dilakukan dengan menjalankan aplikasi keuangan dan melakukan *input* data serta melihat *input*-nya apakah sesuai dengan aturan – aturan akuntansi yang telah ditetapkan.

##### 4.5.2 Implementation

Implementasi yang dilakukan adalah meng-*upload* aplikasi yang telah jadi ke *web hosting* yang telah disediakan oleh IIUC. Alamatnya adalah <http://www.financial.iiuc.ac.id>.

### V. KESIMPULAN DAN SARAN

Bab terakhir ini berisi tentang kesimpulan dan saran. Berdasarkan uraian dan pembahasan yang telah dijelaskan pada bab-bab sebelumnya, maka dapat dibuat kesimpulan tentang penelitian ini dan saran untuk penelitian selanjutnya.

#### 5.1 Kesimpulan

1. Aplikasi akuntansi ini dapat diakses secara *online* sehingga dapat memudahkan bagian keuangan dalam memasukkan

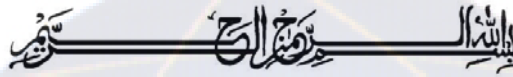
transaksi – transaksi keuangan yang berlangsung.

2. Aplikasi akuntansi ini memberikan fasilitas kemudahan bagi para *manager* untuk dapat melihat keadaan keuangan di IIUC sebagai informasi untuk mengambil keputusan.

#### 5.2 Saran

1. Diharapkan pada pengembangan selanjutnya aplikasi ini terdapat jurnal yang lebih detail, seperti jurnal pendapatan atau jurnal pengeluaran.
2. Diharapkan pada pengembangan selanjutnya aplikasi dapat melakukan penyimpanan data untuk menyimpan data pada rentang waktu tertentu, misalkan untuk jangka waktu satu tahun. Transaksi – transaksi tersebut berguna untuk mengetahui kinerja dari IIUC selama rentang waktu itu.
3. Dilakukan pengembangan lebih lanjut sehingga dapat dibuat laporan arus kas secara otomatis untuk memudahkan pekerjaan bagian keuangan.

## KATA PENGANTAR



*Assalamualaikum Wr. Wb*

Alhamdulillah hirabbil a'lamin, kata yang dapat penulis ucapkan kepada Allah SWT, atas segala rahmat dan hidayah yang telah dilimpahkan-Nya, sehingga penulis bisa menyelesaikan laporan skripsi ini. Shalawat serta salam senantiasa tercurah kepada Rasulullah SAW, insan yang dapat membawa dunia ini kepada kehidupan yang lebih baik.

Dengan rasa syukur yang mendalam penulis dapat menyelesaikan laporan skripsi dengan judul : "Analisis Dan Pembuatan Aplikasi Akuntansi Berbasis Web Pada IIUC".

Sehubungan dengan selesainya laporan ini, penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu penulis, juga atas dorongan dan bimbingannya sehingga laporan ini bisa selesai tanpa ada hambatan yang berarti. Semoga Allah SWT membalas semua amal baik mereka. Terima kasih penulis ucapkan kepada :

1. Bapak Prof. Dr. Komarudin Hidayat sebagai rektor Universitas Islam Negeri Syarif Hidayatullah Jakarta.
2. Bapak Dr. Syopiansyah Jaya Putra, M.Sis selaku Dekan Fakultas Sains dan Teknologi
3. Bapak Yusuf Durachman, M.Sc selaku Ketua Program Studi Teknik Informatika dan pembimbing pertama.
4. Bapak Rusdianto Rustam, MSc, Dr, selaku pembimbing kedua

5. Dosen-dosen jurusan Teknik Informatika yang telah memberikan ilmu, pengetahuan, kemampuan dan *skill* manajemen beserta pemupukan moral dan mental yang baik.
6. Kedua orang tuaku yang aku cintai (Bapak Satrio & Ibu Floura), yang selalu memberikan doanya dengan ikhlas, *allâhummaghfirlî wa liwâlidayya warhamhummâ kamâ rabbayânî shighîrâ*, yang selalu memberikan nasihat, motivasi, mengajarkan arti hidup dan senantiasa mencurahkan dukungannya demi kesuksesan penulis dalam proses kuliah dan kehidupan ini.
7. Ibu Yuniar Ernawati, yang selalu membantu penulis dalam segala hal.
8. Seluruh teman-teman KKN 15, Heri, Fikri, Iche, dan seluruh mahasiswa TI angkatan 2006. Sukses untuk kita semua.
9. Seluruh pihak yang telah membantu dan namanya tidak dapat disebutkan satu per satu. Terima kasih atas dukungan dan motivasinya,

Semoga Allah SWT memberikan balasan yang setimpal atas segala bantuan dan kebaikan yang telah mereka berikan kepada penulis, Amin. Demikianlah, Semoga apa yang tertulis dalam laporan skripsi ini dapat bermanfaat bagi semua pihak. Dalam hal ini, penulis mengharapkan saran dan kritik yang membangun demi menambah kesempurnaan laporan skripsi ini.

***Wassalamualaikum Wr.Wb.***

Jakarta, Maret 2011

(Arif Kunto Wibisono)

# ANALISIS DAN PEMBUATAN APLIKASI AKUNTANSI BERBASIS WEB PADA IIUC

Skripsi  
Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
Sarjana Komputer  
Pada Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta

Oleh:  
ARIF KUNTO WIBISONO  
106091002876

Menyetujui,

Pembimbing 1

Pembimbing 2

Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1 002

Rusdianto Rustam, M.Sc, Dr.  
NIP. 19570205 198303 1 015

Mengetahui,

Ketua Program Studi Teknik Informatika,

Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1 002

## PENGESAHAN UJIAN

Skripsi yang berjudul “Analisi Dan Pembuatan Aplikasi Akuntansi Berbasis Web Pada IIUC” telah diuji dan dinyatakan lulus pada sidang Munaqosyah Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada hari Jum’at, 1 April 2011. Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) Program Studi Teknik Informatika.

Jakarta, 1 April 2011

Menyetujui,

Penguji,

Penguji 1

Penguji 2

Husni Teja Sukmana, Ph. D  
NIP. 19771030 2001121 003

Imam M. Shofi, MT  
NIP. 19720205 2008011 010

Pembimbing,

Pembimbing 1

Pembimbing 2

Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1 002

Rusdianto Rustam, M.Sc, Dr.  
NIP. 19570205 198303 1 015

Mengetahui,

Dekan  
Fakultas Sains dan Teknologi

Ketua  
Program Studi Teknik Informatika

DR. Syopiansyah Jaya Putra, M.Sis  
NIP. 19680117 200112 1 001

Yusuf Durachman, M.Sc, MIT  
NIP. 19710522 200604 1 002

## PERNYATAAN

DENGAN INI SAYA MENYATAKAN BAHWA SKRIPSI INI ADALAH HASIL KARYA SENDIRI YANG BELUM PERNAH DIAJUKAN SEBAGAI SKRIPSI ATAU KARYA ILMIAH PADA PERGURUAN TINGGI ATAU LEMBAGA MANAPUN

Jakarta, Maret 2011

Arif Kunto Wibisono

106091002876