

SURAT TUGAS
0130/B.01/LPPM-UBSI/I/2024

Tentang

**Pelindungan Ciptaan di Bidang Ilmu Pengetahuan, Seni dan Sastra Berdasarkan Undang-Undang
Nomor 28 Tahun 2014 tentang Hak Cipta**
Nomor dan Tanggal Permohonan : EC00202407168, 21 Januari 2024
Nomor Pencatatan : 000582539

**PADA SURAT PENCATATAN CIPTAAN KEMENTERIAN HUKUM DAN HAK ASASI
MANUSIA REPUBLIK INDONESIA**

Program Komputer

Judul Ciptaan :

**RANCANG BANGUN PROGRAM APLIKASI KEUANGAN DENGAN JAVA SCRIPT (Studi
Kasus Pada PEMKOT BEKASI)**

MEMUTUSKAN

- Pertama : Kepada saudara
Melyani S.Pd, MM
Sebagai Pencipta yang mempublikasikan karyanya.
- Kedua : Mempunyai tugas sbb:
Melaksanakan Tugas yang diberikan dengan penuh rasa tanggung jawab.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan, dengan ketentuan apabila
dikemudian hari terdapat kekeliruan akan diubah dan diperbaiki sebagaimana
mestinya.

Jakarta, 15 Januari 2024

LPPM Universitas Bina Sarana Informatika

Ketua



Dr. Taufik Baidawi, M.kom

Tembusan

- Rektor Universitas Bina Sarana Informatika
- Arsip
- Ybs

SURAT PENCATATAN CIPTAAN

Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:

Nomor dan tanggal permohonan : EC00202407168, 21 Januari 2024

Pencipta

Nama : Melyani, Amin Nur Rais dkk

Alamat : Jl. Bunga Rampai VIII Gg.7 Rt.010 Rw.006 No. 20. Perumnas Klender-Jakarta Timur, Duren Sawit, Jakarta Timur, DKI Jakarta, 12860

Kewarganegaraan : Indonesia

Pemegang Hak Cipta

Nama : Melyani, Amin Nur Rais dkk

Alamat : Jl. Bunga Rampai VIII Gg.7 Rt.010 Rw.006 No. 20. Perumnas Klender-Jakarta Timur, Duren Sawit, Jakarta Timur, DKI Jakarta, 12860

Kewarganegaraan : Indonesia

Jenis Ciptaan : Program Komputer

Judul Ciptaan : RANCANG BANGUN PROGRAM APLIKASI KEUANGAN DENGAN JAVA SCRIPT (Studi Kasus Pada PEMKOT BEKASI)

Tanggal dan tempat diumumkan untuk pertama kali : 20 Januari 2024, di Jakarta Timur
di wilayah Indonesia atau di luar wilayah Indonesia

Jangka waktu perlindungan : Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.

Nomor pencatatan : 000582539

adalah benar berdasarkan keterangan yang diberikan oleh Pemohon.

Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.

a.n. MENTERI HUKUM DAN HAK ASASI MANUSIA
Direktur Hak Cipta dan Desain Industri



Anggoro Dasananto
NIP. 196412081991031002

Disclaimer:

Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Melyani	Jl. Bunga Rampai VIII Gg.7 Rt.010 Rw.006 No. 20. Perumnas Klender- Jakarta Timur, Duren Sawit, Jakarta Timur
2	Amin Nur Rais	Perumahan Griya Family 4 Blok F4 No 09, Cibitung, Kab. Bekasi, Jawa Barat, 17520, Cibitung, Bekasi
3	Ratih Dwi Asworowati	Perum Griya Sukadami Blok A3 No 32, Sukadami, Cikarang Selatan, Kab. Bekasi, Jawa Barat 17530, Cikarang Selatan, Bekasi
4	Yuli Komalasari	Perumahan Cikarang Baru. Jl. Kancil 3 Blok C No. 120. Cikarang, Bekasi Jawa Barat, Cikarang Pusat, Bekasi
5	Warjiyono	Pangkah RT. 003 RW. 002, Kel. Pangkah, Kab. Tegal, Jawa Tengah, Pangkah, Tegal
6	Popon Rabia Adawia	Jl. Kedasih 1 B1 No. 66, Mekarmukti, Cikarang, Jawa Barat 17550. , Cikarang Utara, Bekasi
7	Eka Rahmawati	Kedungombo RT05 RW01 Desa Candirejo Kec Borobudur Kab Magelang Jawa Tengah 56553, Borobudur, Magelang
8	Muhammad Rifqi Firdaus	Jl. Tundagan, Tundagan, RT.03 RW.007, Kel. Linggajaya, Kec. Mangkubumi, Kota Tasikmalaya Kode Pos : 46181 , Mangkubumi, Tasikmalaya
9	Unpris Yastanti	Jl. K.H Syatiri No 100 RT 02 RW 03 Ulujami Pesanggrahan, Jakarta Selatan. Kode Pos : 12250 , Pesanggrahan, Jakarta Selatan
10	Mike Amelia	Perumahan Sari Indah Permai Blok J21 Palumbonsari, Karawang Timur , Jawa Barat 41314, Karawang Timur, Karawang

LAMPIRAN PEMEGANG

No	Nama	Alamat
1	Melyani	Jl. Bunga Rampai VIII Gg.7 Rt.010 Rw.006 No. 20. Perumnas Klender- Jakarta Timur, Duren Sawit, Jakarta Timur
2	Amin Nur Rais	Perumahan Griya Family 4 Blok F4 No 09, Cibitung, Kab. Bekasi, Jawa Barat, 17520, Cibitung, Bekasi
3	Ratih Dwi Asworowati	Perum Griya Sukadami Blok A3 No 32, Sukadami, Cikarang Selatan, Kab. Bekasi, Jawa Barat 17530, Cikarang Selatan, Bekasi
4	Yuli Komalasari	Perumahan Cikarang Baru. Jl. Kancil 3 Blok C No. 120. Cikarang, Bekasi Jawa Barat, Cikarang Pusat, Bekasi
5	Warjiyono,	Pangkah RT. 003 RW. 002, Kel. Pangkah, Kab. Tegal, Jawa Tengah, Pangkah, Tegal
6	Popon Rabia Adawia	Jl. Kedasih 1 B1 No. 66, Mekarmukti, Cikarang, Jawa Barat 17550. , Cikarang Utara, Bekasi

7	Eka Rahmawati	Kedungombo RT05 RW01 Desa Candirejo Kec Borobudur Kab Magelang Jawa Tengah 56553, Borobudur, Magelang
8	Muhammad Rifqi Firdaus	Jl. Tundagan, Tundagan, RT.03 RW.007, Kel. Linggajaya, Kec. Mangkubumi, Kota Tasikmalaya Kode Pos : 46181 , Mangkubumi, Tasikmalaya
9	Unpris Yastanti	Jl. K.H Syatiri No 100 RT 02 RW 03 Ulujami Pesanggrahan, Jakarta Selatan. Kode Pos : 12250 , Pesanggrahan, Jakarta Selatan
10	Mike Amelia	Perumahan Sari Indah Permai Blok J21 Palumbonsari, Karawang Timur , Jawa Barat 41314, Karawang Timur, Karawang



Menu dan Paduan Penggunaan

RANCANG BANGUN PROGRAM APLIKASI KEUANGAN DENGAN JAVA SCRIPT

(Studi Kasus Pada PEMKOT BEKASI)

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dengan dikeluarkannya Undang- undang Nomor 6 Tahun 2014 Tentang Desa pada tanggal 15 Januari 2014, merupakan upaya untuk semakin mewujudkan semangat otonomi daerah hingga pada level pemerintahan desa (atau dengan nama lain telah diatur khusus atau tersendiri). Pemerintah mulai mengarahkan fokus pembangunannya ke desa. Pengaturan tentang desa mengalami perubahan secara signifikan. Desa - desa di Indonesia mengalami reposisi dan pendekatan baru dalam pelaksanaan pembangunan dan tata kelola pemerintahannya.

Implementasi otonomi bagi Pemkot Bekasi akan menjadi kekuatan bagi pemerintah Pemkot Bekasi untuk mengurus, mengatur dan menyelenggarakan rumah tangganya sendiri. Peran besar yang diterima oleh Pemkot Bekasi, tentunya disertai dengan tanggung jawab yang besar pula sekaligus bertambah pula beban tanggung jawab dan kewajiban Pemkot Bekasi, namun demikian penyelenggaraan pemerintahan tersebut tetap harus dipertanggungjawabkan. Pertanggungjawaban yang dimaksud diantaranya adalah pertanggungjawaban dalam pengelolaan keuangan mulai dari perencanaan, pelaksanaan, penatausahaan, pelaporan, dan pertanggungjawaban keuangan Pemkot Bekasi. PEMKOT Bekasi adalah salah satu kota di Provinsi Jawa Barat yang berbatasan langsung dengan DKI Jakarta. Nama Bekasi berasal dari kata Bagasasi yang artinya sama dengan Candrabaga yang tertulis di dalam Prasasti Tugu era Kerajaan

Tarumanegara, yaitu nama sungai yang melewati kota ini. Pada pertengahan tahun 2023, jumlah penduduk kota Bekasi berjumlah 2.496.198 jiwa. Kota Bekasi merupakan bagian dari metropolitan Jakarta Raya dan menjadi Kota satelit dengan jumlah penduduk terbanyak se-Indonesia. Saat ini Kota Bekasi berkembang menjadi tempat tinggal kaum urban dan sentra industri Kota Administratif Bekasi sendiri dibentuk pada tahun 1981 dari Kecamatan Bekasi (16 kelurahan dan 8 desa) dan sebagian dari Kecamatan Tambun (2 kelurahan, serta sebagian dari 2 desa) dengan 4 kecamatan yang terbagi ke 18 kelurahan dan 8 desa melalui Peraturan Pemerintah (PP) nomor 48 tahun 1981.

Peresmian Kota Administratif Bekasi dilakukan oleh Menteri Dalam Negeri pada tanggal 20 April 1982, dengan wali kota pertama dijabat oleh Bapak H. Soedjono dari tahun 1982 hingga 1988. Tahun 1988 Wali kota Bekasi dijabat oleh Bapak Drs. Andi Sukardi dari tahun 1988 hingga 1991, kemudian diganti oleh Bapak Drs. H. Khailani AR dari tahun 1991 hingga 1997.

Pada perkembangannya, Kota Administratif Bekasi terus bergerak dengan cepat. Hal ini ditandai dengan pertumbuhan penduduk yang cukup tinggi dan roda perekonomian yang makin tinggi. Dengan begitu status Kota Administratif Bekasi kembali ditingkatkan menjadi Kotamadya (sekarang "Kota") melalui Undang-Undang Nomor 9 Tahun 1996 yang menambahkan tiga kecamatan, yakni kecamatan Pondokgede, Jatiasih, dan Bantargebang.

Selanjutnya dengan terjadinya reformasi, penambahan kecamatan dan kelurahan dilakukan melalui peraturan daerah. Yang pertama adalah Peraturan Daerah Kota Bekasi Nomor 03 Tahun 2000 terkait pembentukan Kecamatan Jatisampurna dari Pondokgede, yang dilanjutkan dengan Peraturan Daerah (Perda) Kota Bekasi Nomor 14 Tahun 2000 Tentang Pembentukan Wilayah Administrasi Kecamatan Kota Bekasi

mengatur pembentukan dua kecamatan baru, yakni: Rawalumbu dari Bekasi Timur; dan Medan Satria dari Bekasi Barat. Dengan demikian, pembagian administratif Kota Bekasi pada tahun 2000 adalah 10 Kecamatan, yang dibagi lagi menjadi 35 Kelurahan dan 17 Desa.

Perda Kota Bekasi Nomor 02 Tahun 2002 tentang Penetapan Kelurahan mengatur bahwa seluruh desa yang ada di Kota Bekasi berubah status menjadi kelurahan. Dengan demikian, pembagian administratif Kota Bekasi pada tahun 2002 menjadi 10 Kecamatan dan 52 Kelurahan.

Kemudian pada tahun 2004, muncul lagi Perda Kota Bekasi Nomor 04 Tahun 2004 yang mengubah Perda Nomor 14 Tahun 2000. Perda ini membentuk 2 Kecamatan; Mustika Jaya dari Bantargebang; dan Pondok Melati dari Jatisampurna dan Pondokgede; serta menambah kelurahan baru. Hal ini menyebabkan pembagian administratif Kota Bekasi menjadi seperti sekarang ini, dengan 12 Kecamatan dan 56 Kelurahan. Jumlah penduduk Kota Bekasi saat ini lebih dari 2,4 juta jiwa yang tersebar di 12 kecamatan, yaitu Kecamatan Pondok Gede, Kecamatan Jatisampurna, Kecamatan Jati Asih, Kecamatan Bantar Gebang, Kecamatan Bekasi Timur, Kecamatan Rawa Lumbu, Kecamatan Bekasi Selatan, Kecamatan Bekasi Barat, Kecamatan Medan Satria, Kecamatan Bekasi Utara, Kecamatan Mustika Jaya, dan Kecamatan Pondok Melati. Dari total luas wilayahnya, lebih dari 50% sudah menjadi kawasan efektif perkotaan dengan 90% kawasan perumahan, 4% kawasan industri, 3% kawasan perdagangan, dan sisanya untuk bangunan lainnya. Pertumbuhan ekonomi merupakan indikator yang dapat menggambarkan kinerja perekonomian di suatu wilayah. Kecuali pada tahun 2004, pertumbuhan ekonomi Kota Bekasi selalu di atas Jawa Barat dan Indonesia. Pada tahun 2004 ekonomi Kota Bekasi tumbuh 5,38% dan pertumbuhan ini lebih tinggi dari Jawa Barat (4,77%) tetapi di bawah LPE Indonesia yang mencapai

5,50%. Pada tahun 2005 dengan 5,65%, LPE Kota Bekasi sedikit lebih tinggi dari Jawa Barat dan Indonesia dengan 5,62% dan 5,55%. Demikian pula pada tahun 2006, LPE Kota Bekasi yang mencapai 6,07% masih lebih baik dibandingkan Jawa Barat dan Indonesia yang hanya mencapai 6,01% dan 5,48%. Dengan besarnya dana yang dikelola maka Pemkot Bekasi dituntut untuk bisa melaksanakan pengelolaan keuangan dengan baik, tepat dan akurat sesuai dengan peraturan yang ada, saat ini pengolahan laporan keuangan Pemkot Bekasi sudah menggunakan komputerisasi dengan menggunakan sistem *microsoft office* dan *microsoft excel* tetapi belum efektif membutuhkan waktu yang lama dan sering terjadi kesalahan. Oleh karena itu dengan memanfaatkan perkembangan teknologi dan informasi maka perlu dicari solusi terbaik untuk mengatasi masalah pengolahan laporan keuangan tersebut, dan komputer dan program aplikasinya merupakan solusi terbaik dalam sistem pengolahan laporan keuangan.

Berdasarkan uraian diatas, maka penulis tertarik untuk mengangkat sebuah judul **“RANCANG BANGUN PROGRAM APLIKASI KEUANGAN DENGAN JAVA SCRIPT (Studi Kasus Pada PEMKOT BEKASI)** sebagai masalah utama dalam penulisan penelitian ini. Dengan tujuan ingin menciptakan sebuah program aplikasi komputerisasi khususnya dalam pengolahan laporan keuangan, sehingga sistem yang penulis rancang dapat diterapkan dan dimanfaatkan.

1.2 Identifikasi Masalah

Dari uraian diatas ada beberapa permasalahan dapat diidentifikasi yaitu :

1. Belum adanya sistem informasi pengolahan laporan keuangan Pemkot Bekasi sehingga nagari kesulitan dalam pembuatan laporan keuangan yang cepat, tepat dan akurat.
2. Kurang optimalnya pemanfaatan komputer dalam pengolahan laporan

keuangan Pemkot Bekasi yang menyebabkan nagari lambat dalam pembuatan laporan keuangan.

1.3 Batasan Masalah

Berdasarkan uraian permasalahan diatas agar pembahasan yang dilakukan secara terarah dan mempunyai tujuan yang jelas dan dengan mudah dapat mengatasi masalah yang ada, maka penulis membatasi penelitian pada masalah yang akan dibahas yaitu :

1. Pada sistem hanya menyediakan pengolahan laporan keuangan Pemerintah Kota Bekasi.
2. Penelitian terhadap pengolahan laporan keuangan Pemkot Bekasi menggunakan bahasa pemrograman Java.

1.4 Rumusan Masalah

Berdasarkan identifikasi masalah, maka dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana merancang sistem informasi pengolahan laporan keuangan Pemkot Bekasi agar bisa memperoleh informasi yang cepat, tepat dan akurat?
2. Bagaimana mengoptimalkan pemanfaatan komputer untuk pengolahan laporan keuangan Pemkot Bekasi?

1.5 Tujuan Penelitian

Setiap kegiatan pasti ada tujuan dan sasaran yang jelas akan diharapkan, maka dari tujuan penelitian yang dilakukan penulis adalah :

1. Untuk mendapatkan sistem informasi pengolahan laporan keuangan Pemkot Bekasi dengan cepat, tepat dan akurat.
2. Untuk memudahkan Pemkot Bekasi dalam pembuatan laporan keuangan.

BAB II

TINJAUAN PUSTAKA

2.1 Konsep Dasar Sistem Informasi

Pentingnya informasi digunakan dalam manajemen sebagai alat dalam pengambilan keputusan. Tidak hanya itu tetapi juga digunakan untuk mengetahui perkembangan organisasi dan lingkungan serta untuk perencanaan strategis untuk masa yang akan datang. Sistem yang tidak memperhatikan perkembangan informasi akan menjadi kerdil dan akhirnya akan berakhir. Informasi diperoleh dari sistem informasi (*information system*).

Suatu sistem dapat saja berbentuk abstrak maupun fisik. Sistem abstrak adalah suatu susunan teratur, gagasan atau konsep yang saling ketergantungan. Sistem fisik adalah sebuah sistem terdiri dari bagian-bagian yang saling berkaitan dan berarti, dan terdiri dari unsur-unsur yang dapat dikenal sebagai saling melengkapi karena satunya kesatuan maksud dan tujuan. Dari pernyataan diatas sebuah sistem informasi merupakan hal yang sangat penting bagi manajemen didalam pengambilan keputusan.

2.1.1 Pengertian Sistem

Sekumpulan unsur atau elemen yang saling berkaitan dan saling mempengaruhi dalam melakukan kegiatan bersama untuk mencapai suatu tujuan. Suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Sistem yang mempunyai fungsi mengumpulkan, memproses,

menyimpan, menganalisis, dan menyebarkan informasi untuk tujuan yang spesifik.

Istilah sistem yang digunakan untuk menunjukkan pengertian metode atau cara yang merupakan suatu himpunan unsur atau komponen yang saling berkaitan satu sama yang lainnya yang akan membentuk satu kesatuan yang utuh. Dan sistem dapat juga didefinisikan sebagai suatu elemen-elemen yang berhubungan satu sama yang lainnya yang saling berinteraksi untuk mencapai tujuan tertentu. Menurut Hanif Al Fatta (2007), sistem sebagai seperangkat elemen yang digabungkan satu dengan yang lainnya untuk suatu tujuan bersama. Sistem merupakan kumpulan dari sub-sub sistem, elemen-elemen, prosedur- prosedur yang saling berinteraksi, berintegrasi untuk mencapai tujuan tertentu seperti informasi, target dan tujuan lainnya.

2.1.2 Arti Perancangan Sistem

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk rancang bangun implementasi menggambarkan bagaimana suatu sistem dibentuk dengan penggambaran dan perencanaanya.

Setelah tahap analisis sistem selesai dilakukan, maka analisis sistem telah mendapatkan gambaran dengan jelas apa yang harus dikerjakan. Tiba waktunya sekarang bagi analisis sistem untuk memikirkan bagaimana membentuk sistem tersebut. Tahap ini disebut dengan perancangan sistem.

Perancangan sistem dapat dibagi dalam 2 bagian, yaitu :

- a. Perancangan sistem secara umum/perancangan konseptual,
perancangan logikal/perancangan secara makro.
- b. Perancangan sistem terinci/perancangan sistem secara fisik.

2.1.3 Pengertian Informasi

“(Jogiyanto, 2009:6), informasi adalah data yang diolah menjadi bentuk yang berguna bagi pada pemakainya.

Sumber dari informasi adalah data. Data merupakan bentuk jamak dari bentuk tunggal atau data item. Jadi data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian-kejadian (*event*) adalah sesuatu yang terjadi pada saat yang tertentu.

Informasi dibutuhkan oleh manajemen untuk menghindari proses *entropi*. Proses *entropi* adalah proses berakhirnya keberadaan suatu sistem manajemen yang didahului kondisi tanpa pola dan tidak menentu. Informasi adalah hasil pengolahan data yang dapat digunakan sebagai dasar pengambilan keputusan.

Informasi yang baik harusnya memenuhi kriteria sebagai berikut :

1. Akurat

Informasi yang akurat adalah informasi yang bebas dari kesalahan dan memiliki maksud yang jelas.

2. Tepat Waktu

Informasi yang datang pada penerima tidak boleh terlambat, karena informasi tersebut merupakan landasan untuk mengambil keputusan. Dengan demikian pentingnya informasi itu maka diperlukan teknologi untuk mendapatkan, mengolah dan mengirimnya secepat mungkin kepada penerima.

3. Relevan

Informasi harus mempunyai manfaat bagi pemakainya. Informasi yang diterima selalu berbeda-beda, setiap perusahaan akan menentukan informasi yang berbeda dengan perusahaan lain.

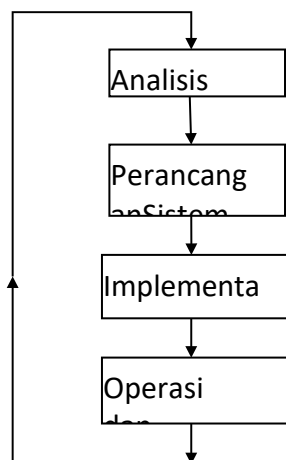
2.1.4 Pengertian Sistem Informasi

Menurut Hanif Al Fatta (2007) sistem informasi adalah suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya. Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan. Sistem informasi adalah data yang dikumpulkan, dikelompokkan dan diolah sedemikian rupa sehingga menjadi sebuah satu kesatuan informasi yang saling terkait dan saling mendukung sehingga menjadi suatu informasi yang berharga bagi yang menerimanya. Semua jenis informasi memiliki 3 kegiatan utama, yaitu :

1. Menerima data sebagai masukan (*input*).
2. mengerjakan perhitungan, penggolongan unsur data, pemutakhiran (*Updating*) dan lain sebagainya.
3. Memperoleh informasi sebagai keluaran (*output*).

2.1.5 Siklus Hidup Pengembangan Sistem (SDLC)

Metode siklus hidup pengembangan sistem atau *system devolopment life cycle* (SDLC) mempunyai beberapa tahapan. Sesuai dengan namanya, SDLC dimulai dari suatu tahapan sampai tahapan terakhir dan kembali lagi ketahapan awal membentuk suatu siklus atau daur hidup. Siklus atau daur hidup pengembangan sistem tampak jika sistem



Gambar 2.1 Siklus Hidup Pengembangan Sistem

Sumber : (Jogiyanto, 2009)

yang sudah dikembangkan dan dioperasikan tidak dapat dirawat lagi, sehingga dibutuhkan pengembangan sistem kembali yang dapat dilihat pada gambar 2.1 :

Tahapan-tahapan dalam metode SDLC adalah sebagai berikut ini :

1. Analisis sistem (*system analysis*)
 - a. Studi pendahuluan.
 - b. Studi kelayakan.
 - c. Mengidentifikasi permasalahan dan kebutuhan pemakai.
 - d. Memahami sistem yang ada.
 - e. Menganalisis hasil penelitian.
2. Perancangan sistem (*system design*)
 - a. Perancangan awal.
 - b. Perancangan rinci.
3. Implementasi sistem (*system implementation*)
4. Operasi dan perawatan sistem (*system operation and maintenance*)

2.2 Konsep Dasar Pengembangan Sistem Informasi

2.2.1 Pengertian Pengembangan Sistem

Pengembangan sistem atau lebih dikenal dengan rekayasa adalah suatu perancangan (arsitek) dari pada suatu sistem yang lama kepada bentuk yang baru dan lebih optimal penggunaannya. Pengembangan dari pada suatu sistem ini didalam organisasi memerlukan adanya suatu bentuk atau karakteristik dari pada model yang sering juga disebut dengan suatu gambaran bentuk abstrak dari pada suatu objek atau sistem realitas di dunia nyata. Pada tahap pengembangan sistem informasi disebut juga siklus hidup pengembangan sistem informasi. Siklus hidup sistem informasi dimulai dari perencanaan, pengembangan (survei, analisa, desain, pembuatan, implementasi,

pemeliharaan) dan dievaluasi secara terus-menerus untuk menetapkan apakah system informasi tersebut masih layak diaplikasikan, jika tidak sistem informasi tersebut akan diganti dengan yang baru dan dimulai dari perencanaan kembali.

Dari gambar diatas dapat dilihat bahwa tahap perencanaan dan tahap evaluasi melupakan kegiatan/tugas dari user dan manajemen, sedangkan tahap pengembangan merupakan tugas bagian EDP/Konsultan.

Tahapan Utama pengembangan sistem informasi:

- a. Survei, bertujuan untuk mengetahui ruang lingkup pekerjaan.
- b. Analisis, bertujuan untuk memahami system yang ada, mengidentifikasi masalah dan mencari solusinya.
- c. Desain, bertujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah-masalah yang dihadapi.
- d. Pembuatan membuat sistem baru (*hardware, software*).
- e. Implementasi, bertujuan untuk mengimplementasikan sistem yang baru.
- f. Pemeliharaan, bertujuan agar sistem dapat berjalan secara optimal.

Untuk lebih jelasnya, tahap-tahap yang harus dilaksanakan secara sistematis dalam pengembangan sistem informasi adalah sebagai berikut:

1. Kebijakan dan Perencanaan Sistem (*System Planning*).

- a. Permintaan untuk studi suatu system (*request for a system study*).
- b. Pengamatan/investigasi awal (*initial investigation*).
- c. Studi Kelayakan (*feasibility study*).

2. Analisa Sistem (System Analysis)

- a. Mendefinisikan kembali masalah (*redefine the problem*).
- b. Memahami sistem yang ada (*understand the existing system*).
- c. Menentukan kebutuhan-kebutuhan pemakai dan hambatan-hambatan pada suatu sistem baru (*determine user requirement and constraint on a new system*).
- d. Model logika dari pemecahan yang direkomendasikan (*logical model of the recommended solution*).

3. Desain Sistem (System Design)

- a. Desain output (*output design*).
- b. Desain input (*input design*).
- c. Desain file (*file design*).

4. Implementasikan Sistem (System Implementation)

- a. Pengembangan Sistem (*system building*).
- b. Pengetesan (*testing*).
- c. Instalasi/Konversi (*installation/conversion*).
- d. Operasi (*operation*).

5. Perawatan (Maintenance)

- a. Menggunakan sistem.
- b. Audit sistem.
- c. Memelihara sistem.
- d. Perawatan dan peningkatan-peningkatan.

2.2.2 Perancangan Sistem

Sasaran-saran yang akan dicapai dalam perancangan suatu sistem adalah:

1. Perancangan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Perancangan sistem harus dapat mendukung tujuan utama perusahaan
 - a. Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan data transaksi manajemen dan mendukung keputusan yang diambil oleh pihak manajemen.
 - b. Perancangan sistem harus dapat mempersiapkan rancangan bangunan yang terinci untuk masing-masing komponen dari system informasi.

2.2.3 Sasaran Perancangan Sistem

Sasaran-sasaran yang akan dicapai dalam perancangan suatu sistem adalah:

1. Perancangan sistem harus berguna, mudah dipahami dan nantinya mudah digunakan.
2. Perancangan sistem harus dapat mendukung tujuan utama perusahaan.
 - a. Perancangan sistem harus efisien dan efektif untuk dapat mendukung pengolahan data transaksi manajemen dan mendukung keputusan yang diambil oleh pihak manajemen.
 - b. Perancangan sistem harus dapat mempersiapkan rancangan bangunan yang terinci untuk masing-masing komponen dari sistem informasi.

2.3 Alat Bantu dalam Perancangan Sistem

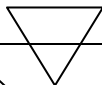
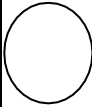
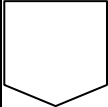
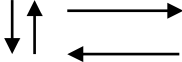
Agar perancangan sistem dapat berjalan lancar sesuai dengan rencana, selain metodologi juga dibutuhkan alat bantu (*tools*). Beberapa alat bantu perancangan yang sering digunakan adalah ASI, CD, DFD, ERD, dan *Flow-Chart*.

2.3.1 Aliran Sistem Informasi (ASI)

Aliran sistem informasi merupakan alat bantu yang menunjukkan arus pekerjaan secara keseluruhan dari sistem. ASI dapat digunakan dengan menggunakan simbol-simbol seperti pada tabel 2.1 :

Tabel 2.1 Simbol Aliran Sistem Informasi

Simbol	Nama Simbol	Keterangan
	Dokumen	Simbol ini digunakan untuk menggambarkan semua jenis dokumen yang digunakan untuk merekam data terjadinya suatu transaksi yang menunjukkan input dan output baik untuk proses manual, mekanik maupun komputerisasi.
	Proses manual	Simbol ini digunakan untuk menunjukkan pekerjaan atau kegiatan yang dilakukan secara manual.
	Proses	Simbol ini digunakan untuk menunjukkan kegiatan proses dari pengolahan data operasi dengan program komputer.
	Arsip	Simbol ini digunakan untuk menggambarkan file komputer/non komputer yang disimpan sebagai arsip. Didalam simbol ini biasanya ditulis huruf F atau huruf A.

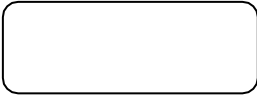
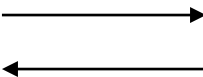
	Arsip	Simbol ini digunakan untuk pengarsipan file menurut numerik atau angka.
	Penghubung pada halaman yang sama	Simbol ini digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama. Didalam simbol ini dicantumkan nomor sebagai penghubung
	Penghubung pada halaman yang berbeda	Simbol ini digunakan untuk menunjukkan hubungan arus proses yang terputus dengan sambungan yang ada di halaman yang lain, sesuai dengan nomor yang tercantum dalam simbol.
	Garis Alir	Simbol ini menunjukkan aliran/arah dari proses pengolahan data.

Sumber : Buku Pengenalan komputer Karangan Jogianto HM, 2005

2.3.2 Context Diagram (CD)

Context diagram adalah gambaran umum dari sistem organisasi yang menunjukkan batasan sistem, entity eksternal yang berintegrasi dengan sistem dan adanya informasi yang mengalir diantara *entity* dan sistem. Simbol-simbol yang digunakan dalam Context Diagram dijelaskan pada tabel 2.2 berikut :

Tabel 2.2 Simbol Context Diagram

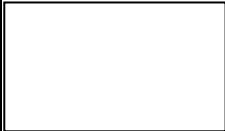
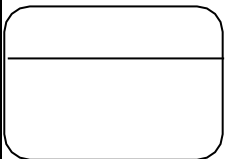
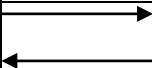
No.	Simbol	Keterangan
1.		External entity. Dapat berupa orang, kelompok orang (organisasi) serta sistem lain.
2.		Keterangan umum dari sistem yang dirancang.
3.		Diagram flow (Alur diagram). Menunjukkan arah aliran data.

Sumber : Buku Pengenalan komputer Karangan Jogianto HM, 2005

2.3.3 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah gambaran aliran data yang melewati sebuah sistem informasi. Dalam perancangan alat bantu ini digunakan untuk menentukan secara detail tentang batasan sistem, proses, dan entity-entity yang saling berhubungan. *Data Flow Diagram (DFD)* merupakan gambaran sistem secara logika yaang tidak tergantung pada perangkat keras, perangkat lunak, struktur data atau orgaanisasi file. Keuntungan dari DFD adalah memudahkan pemakai yang kurang menguasai bidang komputer untuk mengerti sitem yang dikerjakan atau dikembangkan. Dalam DFD digunakan empat simbol utama seperti yang dilihat pada tabel 2.3 berikut:

Tabel 2.3 Simbol Data Flow Diagram (DFD)

Simbol	Nama Simbol	Keterangan
	<i>External Entity/ Terminator</i>	Merupakan kesatuan luar (<i>entity</i>) di lingkungan luar sistem yang dapat berupa orang, organisasi, atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan <i>input</i> atau menerima <i>output</i> dari luar sistem.
	Proses	Merupakan kegiatan atau kerja yang dilakukan oleh orang, mesin atau komputer dari suatu arus data yang masuk ke dalam proses untuk dihasilkan arus data yang akan keluar dari proses yang berfungsi untuk mengolah data.
	<i>Data Store/ Penyimpanan Data</i>	Merupakan simpanan dari data yang dapat berupa suatu <i>file</i> atau <i>database</i> di sistem komputer.
	<i>Arus Data</i>	Arus data ini mengalir di antara proses (<i>process</i>), simpanan data(<i>data store</i>) dan kesatuan luar (<i>eksternal entity</i>). Arus data sebaiknya diberi nama yang jelas dan mempunyai arti.

Sumber : Buku Rekayasa Perangkat Lunak karangan Rosa A.S dan M.Shalahuddin, 2011

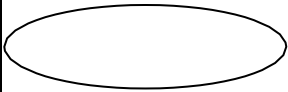
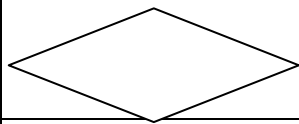
2.3.4 Entity Relationship Diagram (ERD)

ERD merupakan jaringan yang menggunakan susunan data yang tersimpan dari sistem secara acak. ERD berfungsi untuk menjelaskan hubungan anatara file pada suatu DFD.

Entity relationship diagram didasarkan pada presepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek. Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dari objek lain. (Basis Data : Janner Simarmata dan Iman Prayudi, 2010:59).

Model *Entity-Relationship* berisi komponen-komponen dari suatu himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan atribut-atribut yang mempresentasikan seluruh fakta yang ditinjau sehingga dapat diketahui hubungan antara *entity-entity* yang ada dengan atribut-atributnya. Relasi adalah hubungan antara beberapa entitas. Simbol-simbol ERD dapat dilihat pada tabel 2.4

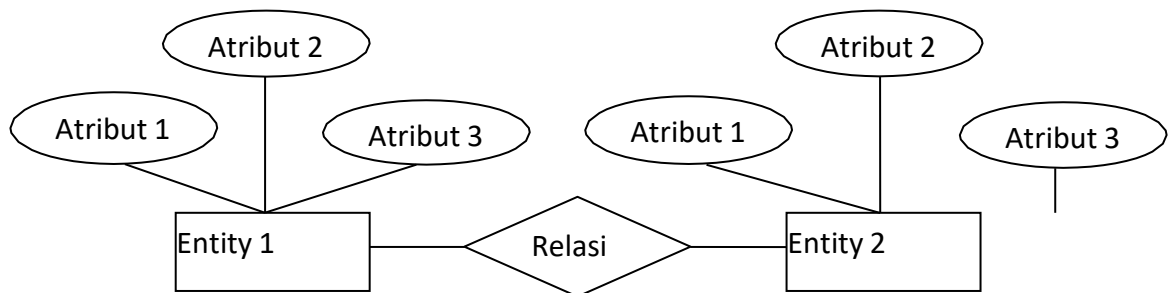
Tabel 2.4 Simbol-Simbol Entity Relationship Diagram (ERD)

Simbol	Nama Simbol	Keterangan
	Entitas / <i>entity</i>	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data.
	Atribut	Field atau kolom data yang butuh disimpan dalam suatu entitas.
	Atribut kunci primer	Field atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan; biasanya berupa id.
	Relasi	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja.
	Asosiasi / <i>association</i>	Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakai.

Sumber : Buku Rekayasa Perangkat Lunak karangan Rosa A.S dan M. Shalahuddin, 201

Berikut adalah contoh hubungan antara *entity* dan *atribut* pada *EntityRelationship*

Diagram (ERD) :



Gambar 2.2 Entity Relationship Diagram (ERD)

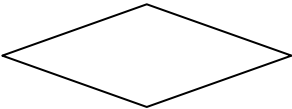
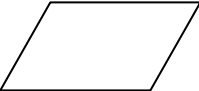
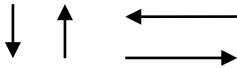
Sumber : Buku Basis Data Karangan Janner Simarmata dan Iman Prayudi, 2010

2.3.5 Program Flow-chart

Program *flow-chart* digunakan untuk menggambarkan algoritma atau proses. Aliran data tidak digambarkan dalam program *flow-chart* melainkan fokus pada urutan proses dalam sebuah aplikasi. Program *flow-chart* digambar dengan menggunakan simbol-simbol seperti pada tabel 2.5.

Tabel 2.5 Simbol Bagan Alir Program (Flow-Chart)

Simbol	Nama Simbol	Keterangan
	<i>Terminal Point</i>	Digunakan untuk menunjukan awal dan akhir program
	Preperation	Simbol persiapan digunakan untuk memberikan nilai awal pada suatu variabel atau <i>counter</i>
	<i>Process</i>	Simbol proses atau pengolahan digunakan untuk mewakili suatu kegiatan atau proses operasi program komputer

Simbol	Nama Simbol	Keterangan
	Keputusan	Digunakan untuk mewakili operasi perbandingan logika atau suatu penyeleksian kondisi di dalam program
	<i>Predifined Process</i>	Simbol proses terdefenisi digunakan untuk menunjukan suatu operasi yang rincinya ditunjukan ditempat empat atau untuk proses yang detailnya dijelaskan terpisah, misalnya berbetuk <i>subroutine</i>
	<i>Input/ Output</i>	Simbol input/output digunakan untuk menyatakan dan mewakili data masukan atau keluaran
	<i>Connector</i>	Simbol Penghubung digunakan untuk menunjukan hubungan artus proses yang terputus dalam halaman yang sama
	<i>Off Page Connector</i>	Simbol penghubung halaman lain digunakan untuk menunjukan hubungan artus proses yang terputus dengan sambungannya yang ada di halaman lain
	<i>Flow Lines</i>	Simbol garis alir digunakan untuk menunjukan arus dari proses

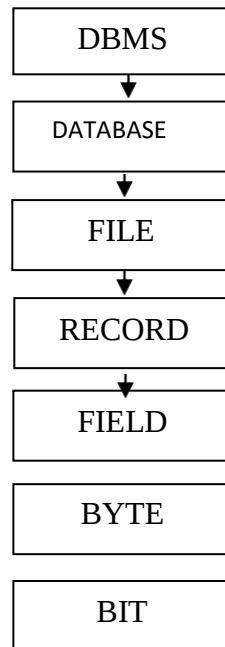
Sumber : Buku Pengenalan komputer Karangan Jogianto HM, 2005

2.4 Data Base

Database digunakan untuk memasukan, menyimpan, serta mengambil data untuk pembuatan laporan berdasarkan data yang telah disimpan. Database atau basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia daat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan

data agar dapat diakses dengan mudah dan cepat. (Rekayasa Perangkat Lunak :Rosa A.S dan M. Shalahuddin, 2011:44).

Hirarki Database



Gambar 2.3 Hirarki Database

Keterangan :

1. Database Management Sistem (DBMS) berfungsi sebagai manager database yaitu mengolah database sehingga dapat dilakukan manipulasi (perubahan, penghapusan, perbaikan perancangan, dan pelaporan terhadap data yang tersimpan.
2. *Database* yaitu kumpulan dari *file-file* atau berkas-berkas berdasarkan jenis dan fungsinya masing-masing yang saling berinteraksi satu sama lain yang membentuk bangunan data, dimana integrasi tersebut di jembatani oleh kunci pada setiap filenya.

3. *File* yaitu kumpulan *record-record* yang sejenis yang mempunyai panjang elemen yang sama, namun berbeda data value nya.
4. *Record/tuple* yaitu kumpulan elemen-elemen yang saling berkaitan, menginformasikan suatu entity secara lengkap. Suatu *record* mewakili informasi tentang seseorang atau data.
5. *Field* yaitu kumpulan nilai data atau data *value* yang sebenarnya disimpan.
6. *Byte* yaitu pembentukan data yang disimpan, baik berupa huruf, angka, simbol-simbol dan lain-lain.
7. *Bit* yaitu bagian terkecil dari *byte* atau untaian pembentuk *byte*.

2.5 Sekilas Mengenai Java

Bahasa pemrograman Java pada awalnya dibuat oleh James Gosling pada tahun 1995 sebagai bagian dari *Sun Microsystem Java Platform*. Sintaks Java banyak diturunkan dari C dan C++ tetapi lebih sederhana, ketat dan mempunyai akses ke OS yang lebih terbatas. Hal ini karena Java ditujukan sebagai bahasa pemrograman yang cukup sederhana untuk dipelajari dan mudah dibaca. (IfnuBima, 2011:2).

2.5.1 Pengenalan IDE NetBeans

NetBeans dibeli oleh Sun Microsystem dari sebuah perusahaan kecil bernama NetBeans di Chekoslowakia kemudian dilepas ke komunitas *Open Source*. Pada tahun 1996, Sun mengeluarkan JSDK (*Java Standard Development Kit*) yang kemudian pada tahun 1997 muncul JFC (*Java Foundation Class*) untuk kepentingan pembuatan aplikasi *window*. Keberadaan JFC didalam Java kemudian menjadi landasan dari Java JSDK versi 1.2 dan selanjutnya (yang kemudian dikenal sebagai Java2). Kemudian muncul Java2 SDK versi 1.4 dengan menambahkan fasilitas asersi untuk dukungan

design-by-contract.

2.5.2 Tipe Data Pada Java

IDE NetBeans yang merupakan kelas Java API sebagai *interface* untuk mengakses sumber daya atau layanan dari sistem operasi., mempunyai tipe-tipe data sebagai berikut:

1. Data teks

Platform Java mengandung tiga class yang berguna dalam bekerjadengan data teks diantaranya:

a. *Character*

Digunakan untuk menyimpan nilai dari suatu karakter (*unicode*) tunggal. Karena karakter Unicode disusun dari nilai 16-bit, maka tipe data char adalah 16-bit *unsigned integer* dengan range 0 sampai 65.535. Class ini juga berisi method-method untuk memanipulasi ataupun memeriksa nilai sebuah karakter tunggal.

b. *String*

Digunakan untuk data *immutable* (konstan, tidak dapat diubah) yang terdiri atas banyak karakter. Objek string bisa diisi dengan nilai null dan tidak akan bermasalah jika kita mencoba menampilkan hasilnya.

c. *String Buffer*

Merupakan class untuk menyimpan dan memanipulasi data yang dapat berubah yang terdiri atas banyak karakter.

2. Bilangan dan

Matematika Terdiri dari :

a. Angka-angka

Class *java.lang.Number* digunakan untuk menyimpan nilai bilangan. Ia mempunyai hirarki yang menggambarkan bahwa selain dari tipe-tipe primitif, java juga

menyediakan objek *wrapper* dari tipe-tipe primitifnya.

b. Fungsi-fungsi Sistematis

Class java.lang.math mendefinisikan sejumlah method static untuk operasi-operasi trigonometri, logaritmik, eksponensial, operasi pembulatan dan lain-lain.

c. Bilangan Besar

Package java.math mengandung *class BigInteger* dan *BigDecimal*. Class-class ini mengizinkan kita bekerja dengan angka-angka yang sangat besar dan ketelitian yang sangat tinggi.

Kelas *BigInteger* mengimplementasikan sejumlah operasi aritmatika untuk bilangan bulat sangat besar.

Kelas *BigDecimal* mengimplementasikan angka titik tetap sangat besar dan mempunyai parameter *scale* yang mengindikasikan seberapa banyak digit di sebelah kanan titik desimal.

Tabel 2.6 Tipe Data

No	Nama	Lebar (bit)	Rentang Nilai
1	Long	64	-9223372036864775808 s/d 9223372036864775807.
2	Int	32	-2147483648 s/d 2147483647
3	Short	16	-32768 s/d 32767
4	Byte	8	-128 s/d 127
5	Double	64	1,7E-308 s/d 1.7E+308
6	Float	32	3,4E-038 s/d 3,4E+038

Sumber : Konsep Dasar Pemrograman Java karangan Patrick Naughton Diterjemahkan Panji Gotama, 2002

2.5.3 Komponen Java

Komponen-komponen dapat diguna ulang sehingga mendukung RAD

(*Rapid Application development*).

1. Jendela utama

Disebut jendela utama karena merupakan pusat perintah pada IDE. Disini kita temukan menu bar, beberapa toolbar, *componen palette* dan *tab workspace*.

2. Workspaces

Sebuah *workspaces* adalah kumpulan-kumpulan dari jendela-jendela yang saling berhubungan erat untuk melakukan tugas-tugas tertentu.

Jenis-jenis *workspaces default*, yaitu:

1. Editing

Editing terdiri dari jendela *Explorer* dan *Properties*. Saat membuka sebuah file, Source Editor secara otomatis terbuka. Source Editor digunakan untuk mengedit file-file java, HTML dan *Plain Text*.

2. GUI Editing

Workspaces GUI Editing digunakan untuk merancang dan membangun GUI. Didalamnya terdapat *Explorer*, *Form Editor*, *Component Inspector* dan *Source Editor*.

- *Form Editor*

Jendela *Form Editor* merupakan daerah utama untuk membuat dan memodifikasi sebuah GUI

- *Component Inspector*

Component Inspector memungkinkan kita untuk melihat komponen- komponen apa saja, baik yang tampak maupun yang tidak tampak. Disini kita mengubah properti dari komponen- komponen tersebut.

3. Browsing

Didalamnya terdapat *Object Browser* yang terdiri atas tiga panel dan jendela *Properties*.

4. *Running*

Didalamnya terdapat jendela *Execution View* dan *Output*. Saat mengeksekusi program, maka IDE akan secara otomatis ke *Running Workspaces*.

5. *Debugging*

Debugging terdiri dari jendela-jendela *Debugger* dan *Output*. Jendela- jendela ini mempunyai tab panel untuk menset *breakpoint*, memonitor thread dan mengawasi nilai variabel-variabel. Jendela output menampilkan pesan dari *Debugger*.

2.5.4 **Komponen Utama JDBC**

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

Berikut komponen utama JDBC (Wahana Komputer, 2006:8) :

- a. *Driver* adalah komponen untuk menangani masalah komunikasi dengan database server.
- b. *Driver Manager* adalah komponen untuk menangani objek driver dimana objek *Driver Manager* juga mengabstraksikan detil dari proses kerja objek driver.
- c. *Connection* adalah komponen untuk mempresentasikan koneksi secara fisik ke database.
- d. *Statement* adalah komponen untuk mengirim perintah SQL ke database.
- e. *ResultSet* adalah komponen untuk menyimpan data yang didapat dari database setelah perintah SQL dijalankan dengan menggunakan komponen statement.
- f. *SQLException* adalah komponen untuk menangani kesalahan-kesalahan(error) yang mungkin terjadi dalam pengolahan database.

2.6 Sekilas Tentang Konsep MySQL

MySQL dikembangkan sekitar tahun 1994 oleh perusahaan pengembang *software* dan konsultan *database* bernama MySQL AB yang bertempat di Swedia. MySQL adalah satu dari sekian banyak sistem *database*, merupakan terobosan solusi yang tepat dalam aplikasi *database*. Didukung oleh ribuan bahkan jutaan komunitas pengguna internet yang siap membantu.

2.6.1 Pengertian Umum Database MySQL

MySQL adalah cepat, mudah untuk digunakan (*easy-to-use*) dan sebagai sistem manajemen database relasional (RDBMS) yang digunakan untuk database. Kecepatan adalah fokus utama pada pengembangan awal MySQL. Demi kepentingan kecepatan ini, mereka membuat keputusan untuk menawarkan fitur lebih sedikit dibanding pesaing utama mereka (sebagai contoh, Oracle dan Sybase). MySQL lebih mudah dalam instalasi dan penggunaannya dibanding pesaing komersialnya. Didalam harga, MySQL benar-benar murah. (Aplikasi Mobile Commerce Menggunakan PHP dan MySQL : Janner Simarmata, 2006 : 28).

2.6.2 Keunggulan MySQL

MySQL adalah suatu database yang sangat populer dengan pengembangan Web (*Web developers*). Kecepatan dan ukuran yang kecil membuatnya ideal untuk database perancangan sistem informasi. Ditambah lagi dengan fakta bahwa MySQL adalah *open source*, yang berarti gratis. (Aplikasi Mobile Commerce Menggunakan PHP dan MySQL : Janner Simarmata, 2006:29). Dibawah ini adalah beberapa keuntungannya :

1. Cepat

Tujuan utama dari pengembangan Mysql adalah kecepatan. Sebagai konsekuensi *software* yang dirancang dari awal untuk kecepatan.

2. Tidak mahal

MySQL adalah Cuma-Cuma dibawah lisensi GPL (*General Public License*) *open source*, sementara pembiayaan untuk lisensi komersialnya sangatlah pantas.

3. Mudah digunakan

Membangun dan berinteraksi dengan database MySQL hanya dengan menggunakan sedikit pernyataan (*statement*) sederhana bahasa SQL, yang menjadi bahasa standar untuk komunikasi dengan RDBMS.

4. Dapat berjalan pada beberapa sistem operasi

MySQL berjalan pada sistem operasi yang beragam, seperti Windows, Linux, Mac OS, kebanyakan variasi Unix (termasuk Solaris, AIX, dan DEC Unix), FreeBSD, OS/2, Irix, dan lainnya.

5. Aman

MySQL adalah sistem otorisasi fleksibel yang mengizinkan beberapa atau semua privilege database (sebagai contoh, privilege untuk menciptakan suatu database atau menghapus data) untuk pengguna khusus atau sekelompok pengguna.

6. Mendukung database yang besar

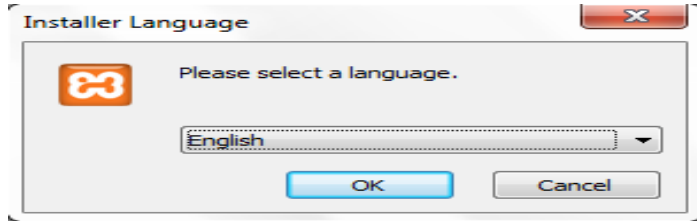
MySQL menangani database sampai 50 juta baris atau lebih. Batasan ukuran file secara default untuk tabel adalah 4 GB, tetapi dapat menaikannya (jika sistem operasi dapat menanganinya) hingga 8 juta terabytes (TB).

7. Customizable

Lisensi GPL *open source* mengizinkan pemograman untuk memodifikasi software MySQL untuk mencocokkannya dengan lingkungan tertentu.

2.7 Install xampp 1.6.8

- a) Untuk menginstall xampp 1.6.8, maka terlebih dahulu kita pilih bahasa yang akan digunakan, seperti pada Gambar 2.4.



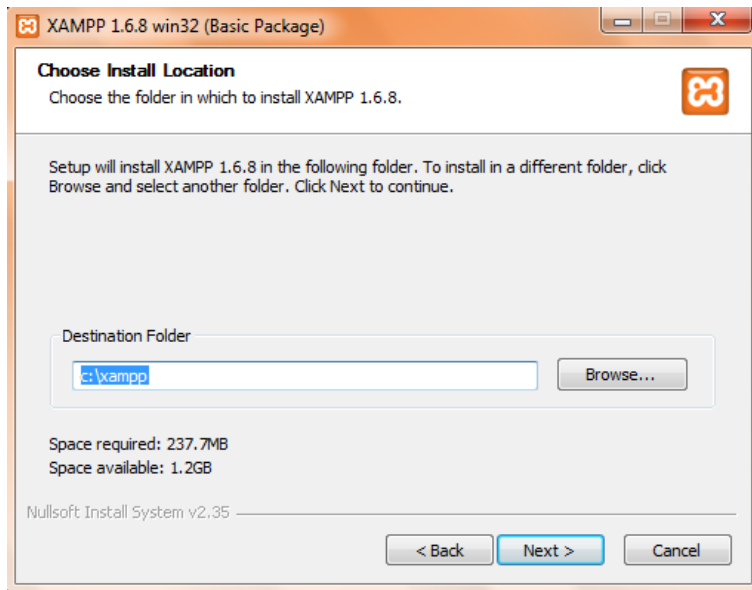
Gambar 2.4 Tahap Pertama Penginstalasian Xampp

- b) Setelah memilih bahasa yang akan digunakan, maka klik next, seperti pada Gambar 2.5.



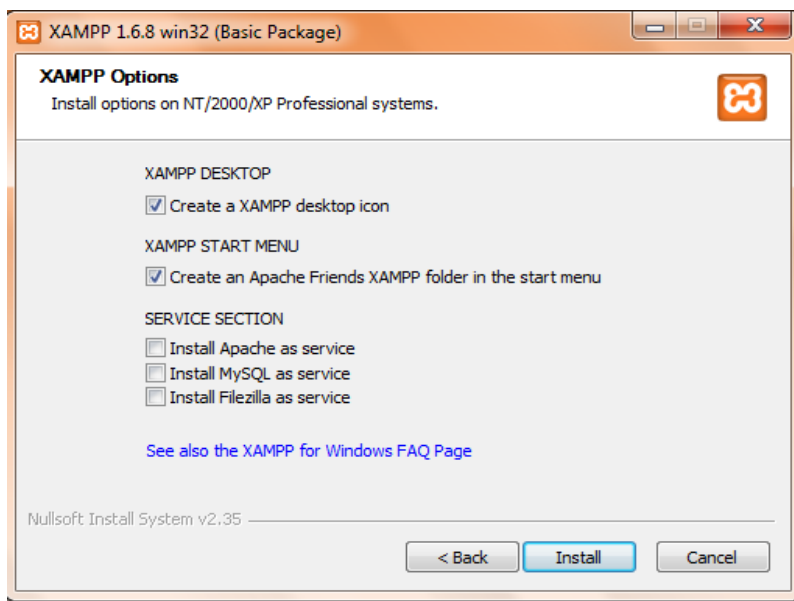
Gambar 2.5 Tahap Kedua Penginstalasian Xampp

- c) Pilih tempat penyimpanan, kemudian klik next, seperti pada Gambar 2.6.



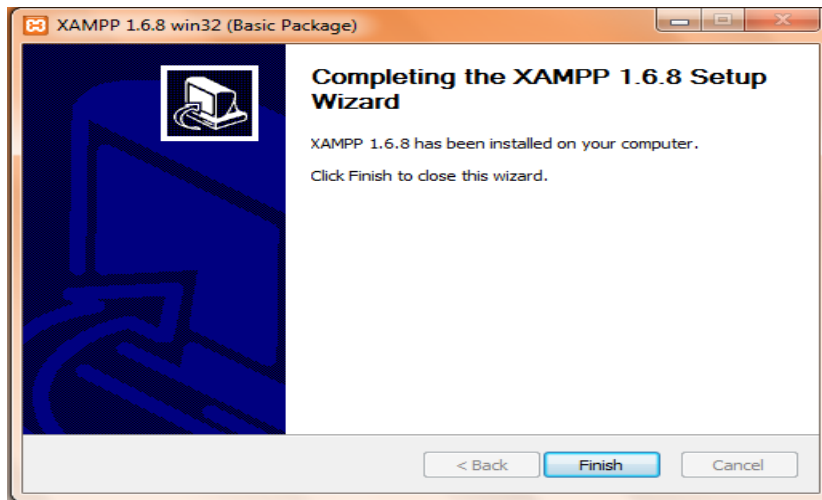
Gambar 2.6 Tahap Ketiga Penginstalasian Xampp

- d) Centang xampp desktop dan xampp start menu, lalu klik install, seperti pada Gambar 2.7



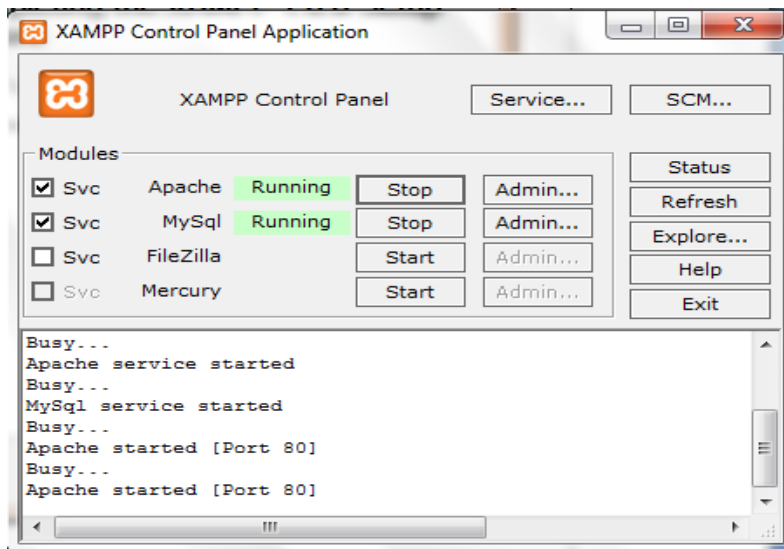
Gambar 2.7 Tahap Keempat Penginstalasian Xampp

- e) Setelah tahapan keempat, maka proses install akan berlangsung dan kemudian muncul tulisan “Completing the XAMPP 1.6.8 Setup Wizard”. Klik Finish, seperti pada Gambar 2.8.



Gambar 2.8 Tahap Kelima Penginstalasian Xampp

- f) Untuk menjalankan xampp, centang apache dan mysql lalu klik start, seperti pada Gambar 2.9.



Gambar 2.9 Tahapan Keenam Penginstalasian Xampp

2.8 Struktur Organisasi

Struktur organisasi adalah gambaran pembagian dan pengelompokan tugas di dalam suatu organisasi, sehingga terlihat dengan jelas bagian-bagian atau unit kerja serta hubungan komunikasi diantara unit kerja dan juga akan mewujudkan gambaran wewenang dan tanggung jawab pada setiap panjang hierarki.

2.8.1 Jenis / Bentuk Struktur Organisasi

Pada dasarnya struktur organisasi dapat dibedakan dalam tiga jenis yaitu :

1. Struktur Organisasi Garis

Struktur organisasi garis adalah tipe organisasi tertua dan yang paling sederhana. Dalam struktur organisasi garis tugas-tugas perencanaan, pengendalian dan pengawasan berada pada suatu garis wewenang langsung dari pimpinan pada bawahan.

2. Struktur Organisasi Fungsional.

Dalam sistem organisasi fungsional setiap pimpinan spesialisasi dalam melakukan tugas tugas tertentu dari banyaknya tugas yang ada dalam perusahaan.

3. Struktur Organisasi Garis dan Staf.

Dalam struktur organisasi ini setiap pimpinan atau atasan memberikan perintah langsung kepada bawahannya. Sistem ini merupakan penyempurnaan dari sistem organisasi dan sistem organisasi fungsional.

Struktur organisasi menunjukkan kerangka dan susunan perwujudan pola tetap hubungan diantara fungsi, bagian atau posisi yang menunjukkan kedudukan tugas dan wewenang serta tanggung jawab yang berbeda-beda dalam suatu organisasi.

Dengan gambaran yang jelas ini akan memudahkan kerja sama untuk mencapai tujuan organisasi yang telah ditentukan. Bila organisasi bertambah besar maka persoalan organisasi juga semakin sulit, hal ini disebabkan maju mundurnya suatu organisasi akan ditentukan oleh kemampuan organisasi tersebut mengadaptasi diri dengan perubahan lingkungan.

Sebelum membahas lebih lanjut mengenai struktur organisasi Pemerintah Kota Bekasi, ada baiknya dikemukakan apa yang dimaksud dengan organisasi.

Menurut M.Manulang (1981 : 68) bahwa :

- a. Organisasi dalam arti badan adalah sekelompok orang yang bekerjasama untuk mencapai tujuan atau beberapa tujuan tertentu.
- b. Organisasi dalam arti badan atau struktur adalah gambaran secara sistematis tentang hubungan-hubungan kerja sama dari orang-orang yang terdapat dalam usaha untuk mencapai suatu tujuan.

Dari penjelasan diatas maka dapat ditarik suatu kesimpulan bahwa organisasi adalah suatu badan yang mengelompokkan orang-orang untuk bekerja sama dimana setiap kelompok tersebut ada hubungan satu sama lain sehingga dapat mencapai tujuan yang diinginkan.

Kepala urusan berkedudukan sebagai unsur staf sekretariat. Kepala urusan bertugas membantu Sekretaris dalam urusan pelayanan administrasi pendukung pelaksanaan tugas-tugas pemerintahan. Untuk melaksanakan tugas kepala urusan mempunyai fungsi:

1. Kepala urusan tata usaha dan umum memiliki fungsi seperti melaksanakan urusan ketatausahaan seperti tata naskah, administrasi surat menyurat, arsip, dan ekspedisi, dan penataan administrasi perangkat nagari, penyediaan prasarana perangkat nagari dan kantor, penyiapan rapat, pengadministrasian aset, inventarisasi, perjalanan dinas, dan pelayanan umum.
2. Kepala urusan keuangan memiliki fungsi seperti melaksanakan urusan keuangan seperti pengurusan administrasi keuangan, administrasi sumber-sumber pendapatan dan pengeluaran, verifikasi administrasi keuangan, dan administrasi penghasilan Pemkot Bekasi, BPD, dan lembaga pemerintahan nagari lainnya.

3. Kepala urusan perencanaan memiliki fungsi mengoordinasikan urusan perencanaan seperti menyusun rencana anggaran pendapatan dan belanja nagari, menginventarisir data-data dalam rangka pembangunan, melakukan monitoring dan evaluasi program, serta penyusunan laporan.

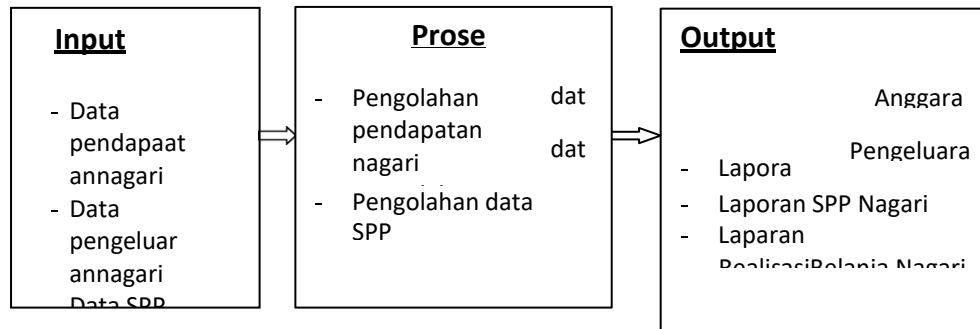
a. Kasi/ Staf

Kepala seksi berkedudukan sebagai unsur pelaksana teknis. Kepala seksi bertugas membantu Walikota sebagai pelaksana tugas operasional. Untuk melaksanakan tugas Kepala Seksi mempunyai fungsi:

1. Kepala seksi pemerintahan mempunyai fungsi melaksanakan manajemen tata praja Pemerintahan, menyusun rancangan regulasi nagari, pembinaan masalah pertanahan, pembinaan ketentraman dan ketertiban, pelaksanaan upaya perlindungan masyarakat, kependudukan, penataan dan pengelolaan wilayah, serta pendataan dan pengelolaan.
2. Kepala seksi kesejahteraan mempunyai fungsi melaksanakan pembangunan sarana prasarana pernagarian, pembangunan bidang pendidikan, kesehatan, dan tugas sosialisasi serta motivasi masyarakat di bidang budaya, ekonomi, politik, lingkungan hidup, pemberdayaan keluarga, pemuda, olahraga, dan karang taruna.
3. Kepala seksi pelayanan memiliki fungsi melaksanakan penyuluhan dan motivasi terhadap pelaksanaan hak dan kewajiban masyarakat, meningkatkan upaya partisipasi masyarakat, pelestarian nilai sosial budaya masyarakat, keagamaan, dan ketenagakerjaan.

2.9 Kerangka Konseptual

Adapun kerangka konseptual dari penelitian ini terdiri dari tiga bagian diantaranya :
input, Proses dan Output,



Gambar 2.9 Kerangka Konseptual

1. *Input*

Input atau masukan merupakan pengumpulan data-data tentang Pengelolaan Keuangan Pemkot Bekasi berupa :

- Data pendapatan
- Data belanja/ pengeluaran
- Data SPP
- Data SPJ

2. *Proses*

Proses yang dilakukan pada kegiatan pengolahan laporan keuangan Pemkot Bekasi adalah pengolahan data pendapatan, data belanja/pengeluaran, data SPP dan data SPJ.

3. *Output*

Output atau hasil dari kegiatan penelitian ini adalah untuk mengetahui laporan keuangan nagari berupa laporan anggaran, laporan belanja/pengeluaran Pemkot Bekasi, laporan SPP dan laporan realisasi penggunaan anggaran Pemkot Bekasi.

BAB III

METODOLOGI PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang peneliti lakukan adalah penelitian Terapan (*Applied Research*). Penelitian terapan adalah penelitian yang hati-hati, sistematis dan terus menerus terhadap suatu masalah dengan tujuan digunakan segera untuk keperluan tertentu.

3.2 Tempat dan Waktu Penelitian

3.2.1 Tempat Penelitian

Penelitian dilakukan di Pemkot Bekasi. Pemilihan tempat dilakukan secara purposif (sengaja) dengan pertimbangan akan kelengkapan data dan informasi yang diperlukan untuk kegiatan penelitian mengenai analisis pengolahan laporan keuangan Pemkot Bekasi.

3.2.2 Waktu Penelitian

Sedangkan waktu penelitian dilakukan pada bulan November 2023.

3.3 Langkah- langkah Metode penelitian

Adapun langkah – langkah metoda penelitian dapat diuraikan sebagai berikut :

1. Untuk mendapatkan data dan bahan laporan penelitian yang sesuai harapanteknik pengumpulan data yang digunakan ada dua jenis :

a. Studi Literatur

Studi Literatur merupakan pengumpulan data dengan cara literatur, paper dan bacaan yang ada kaitannya dengan jenis penelitian.

b. Metode Observasi

Metode observasi adalah pengumpulan data dengan mengadakan pengamatan atau kegiatan yang sistematis terhadap objek yang dituju secara langsung.

2. Metode Wawancara

Wawancara adalah suatu metode penelitian dengan mengadakan jawaban dengan pihak-pihak yang ada hubungannya dengan masalah akan dibahas.

Wawancara langsung dengan pegawai Pemkot Bekasi

3. Penelitian Perpustakaan

Pada penelitian Pustaka ini penulis menggunakan beberapa contoh dari berbagai buku yang berhubungan dengan proposal penelitian ini dan juga menggunakan buku pelajaran yang berhubungan dengan proposal penelitian.

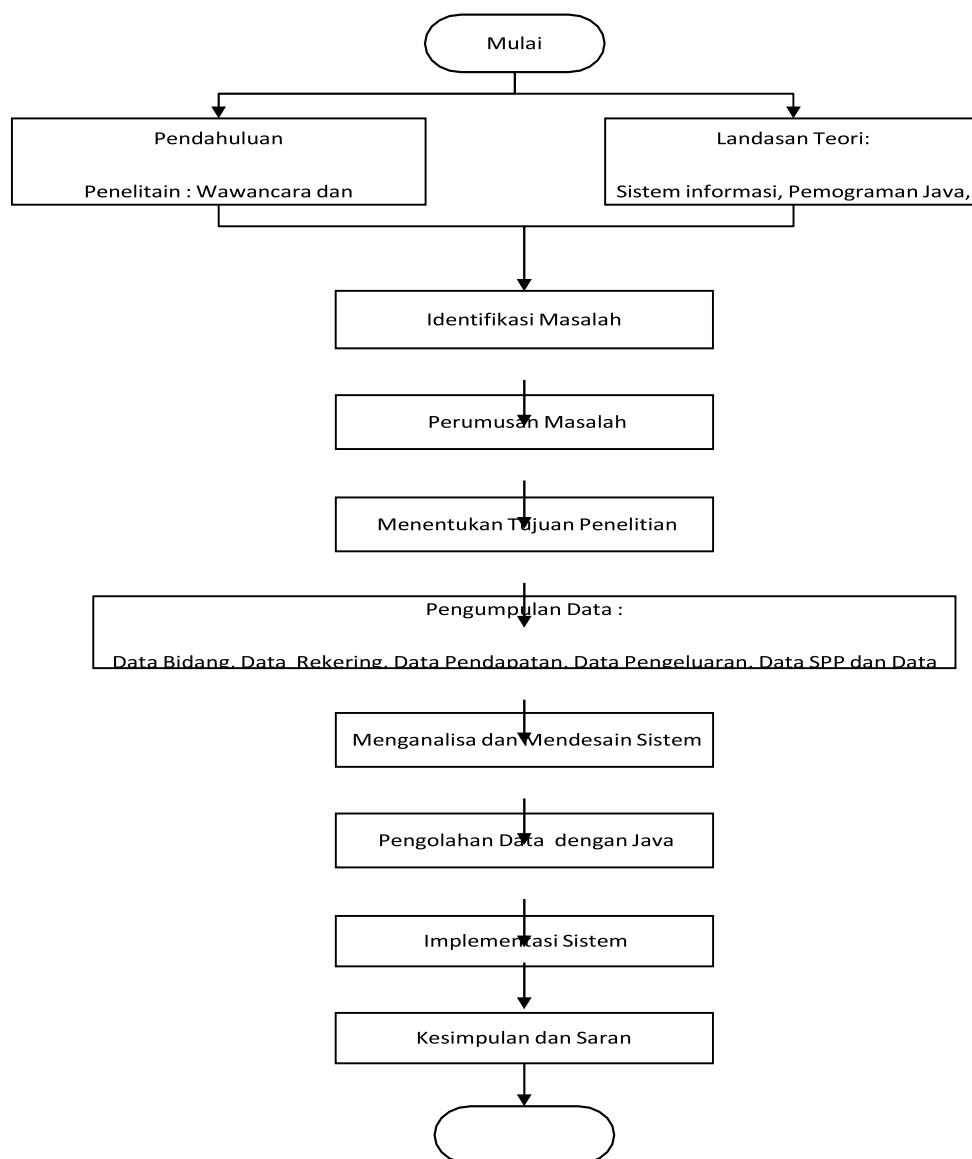
3.4 Data dan Sumber Data

Berdasarkan permasalahan dan tujuan penelitian, teknik analisa data yang digunakan dalam penelitian ini adalah teknik analisa data dengan menggunakan data kualitatif dengan metode pendekatan dekskriptif. Data yang digunakan untuk analisis adalah kualitatif, yaitu data yang digambarkan dengan kata-kata atau kalimat, dipisahkan menurut kategori untuk memperoleh satu kesimpulan

Data-data dan informasi yang dikumpulkan dalam penelitian ini kemudian dipisah-pisahkan sesuai dengan pengelompokan kemudian digambarkan dengan kata atau kalimat untuk mendapatkan kesimpulan dan saran.

3.5 Kerangka Metodologi

Untuk menjelaskan arah penelitian ini, dapat dilihat pada gambar kerangka metodologi berikut ini :



Gambar 3.1 Kerangka Metodologi Penelitian

Adapun penjelasan tentang kerangka kerja penelitian atau struktur kerja yang dilakukan dalam penelitian ini sebagai berikut :

1. Pendahuluan Penelitian

Melakukan wawancara dan pengamatan masalah secara langsung untuk mendapatkan data-data yang dibutuhkan.

2. Landasan Teori

Disini penulis juga memberikan penjelasan hal-hal yang berhubungan dalam perancangan sistem berupa pengertian sistem informasi, Pemrograman Java, Data Base MySql

3. Identifikasi Masalah

Kerangka kerja ini dimulai dari identifikasi masalah dalam pengelolaan laporan keuangan nagari Parit Malintang, yang bertujuan untuk menjaga konsistensi dari penelitian ini sehingga penelitian ini lebih terarah, sehingga tujuan dari penelitian yang diharapkan dapat tercapai.

4. Perumusan Masalah

Merumuskan suatu permasalahan yang timbul pada proses dengan berbagai tujuan.

5. Menentukan Tujuan Penelitian

Berdasarkan pada ruang lingkup masalah dan perumusan masalah yang telah dibuat pada tahap sebelumnya, tahap berikutnya adalah menentukan tujuan penelitian yang bertujuan untuk memperjelas kerangka tentang apa saja yang menjadi sasaran dari penelitian ini.

6. Pengumpulan Data

Pengumpulan data dan informasi pada tahap ini dilakukan untuk mengetahui mendapatkan data dan informasi yang nantinya akan mendukung penelitian ini, dalam pengumpulan data, terdapat beberapa metode yang digunakan yaitu penelitian lapangan (field research), penelitian perpustakaan

(library research), serta penelitian laboratorium (laboratory research).

7. Menganalisa dan Mendesain Sistem

menganalisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh. Setelah pengumpulan data maka langkah selanjutnya menganalisa dan mendesain sistem berdasarkan identifikasi masalah dan data yang didapat sehingga menghasilkan suatu informasi yang cepat, tepat dan akurat. Perancangan dari model sistem dengan menentukan rancangan input dan informasi apa saja yang dihasilkan nantinya.

8. Pengolahan Data dengan Java

Data yang didapat akan diolah dalam pemrograman java untuk mendapatkan informasi dalam pengelolaan laporan keuangan nagari yang cepat, tepat dan akurat.

9. Implementasi Sistem

Selanjutnya peneliti juga akan melakukan implementasi sistem untuk dapat memastikan apakah sistem yang telah dirancang layak atau belum untuk digunakan.

10. Kesimpulan dan Saran

Jika sistem sudah layak digunakan maka peneliti dapat menyimpulkan bahwa sistem yang telah dirancang sangat berguna bagi Pemkot Bekasi.

BAB IV

ANALISA DAN DESAIN SISTEM

4.1 Analisa dan Desain Sistem

Analisa sistem adalah penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mendefinisikan masalah dan mengevaluasi permasalahan-permasalahan, kesempatan dan hambatan-hambatan yang terjadi serta kebutuhan-kebutuhan yang diharapkan dapat diusulkan perbaikan-perbaikan. Sistem yang sudah ada penting dianalisa, karena merupakan dasar untuk merencanakan kesistem baru. Sistem lama akan dijadikan perbandingan terhadap sistem baru yang akan dirancang. Dalam analisa sistem tersebut dapat dilihat kelebihan dan kekurangannya pada sistem lama. Dengan adanya sistem baru diharapkan dapat meningkatkan fungsi yang lebih baik dari sistem yang sudah ada.

4.1.1 Sistem yang sedang berjalan

Dalam aliran sistem yang sedang berjalan diuraikan pertahapannya sebagai berikut : Pelaksana Kegiatan menyerahkan rencana kerja ke Sekretaris untuk diverifikasi dan untuk membuat APB. Rencana kerja yang telah diverifikasi dan APB tersebut diserahkan kepada Walikota untuk di setujui setelah itu rencana kerja dan APB yang telah disetujui tersebut diserahkan kembali pada Sekretaris dan Pelaksana Kegiatan untuk diarsipkan. Berdasarkan rencana kerja yang telah disetujui tersebut Pelaksana Kegiatan membuat SPP dan SPJ dengan 3 rangkap. SPP dan SPJ diberikan Sekretaris untuk diverifikasi kemudian serahkan kepada Walikota

untuk disetujui . SPP dan SPJ yang telah disetujui tersebut rangkap satu di berikan pada Sekretaris Pemkot Bekasi untuk diarsipkan, rangkap dua diberikan kepada Pelaksana Kegiatan untuk diarsipkan dan rangkar ketiga diberikan kepada Bendahara.

Dari SPP dan SPJ yeng telah disetujui tersebut Bendahara Pemkot Bekasi membuat laporan keuangan. Lalu laporan tersebut diserahkan kepada Walikota untuk di setujui.

Laporan keuangan yang telah di setujui tersebut kemudiandiserahkan kepada Sekretaris Pemkot Bekasi dan bendahara Pemkot Bekasi untuk di arsipkan. Setelah dilakukan analisis terhadap sistem yang sedang berjalan, maka dapat dibentuk Aliran Sistem Informasi (ASI) pengelolaan laporan keuangan Pemkot Bekasi sebagai berikut :



Gambar 4.1. Aliran Sistem Informasi lama

4.1.2 Aliran Sistem Informasi yang diusulkan

Aliran sistem informasi dalam pengelolaan laporan keuangan Pemkot Bekasi dari sistem yang diusulkan (sistem baru) pada Pemkot Bekasi tidak jauh beda dengan aliran sistem informasi yang sedang berjalan di Pemkot Bekasi.

Perbedaannya pada proses pembuatan SPP dan SPJ dimana pada aliran Sistem Informasi yang sedang berjalan telah menggunakan sarana komputer dengan aplikasi standar yaitu *microsoft excel*. Sedangkan pada aliran sistem informasi yang diusulkan proses dilakukan secara komputerisasi dengan database sebagai media penyimpanan dan memudahkan untuk pengelolaan laporan keuangan Pemkot Bekasi

Dalam aliran sistem yang diusulkan dapat diuraikan sebagai berikut : bendahara menginputkan data SPP dan SPJ yang telah di setuju oleh Walikota kedalam database. Dan membuat laporan keuangan nagari untuk diserahkan kepada Walikota untuk di setuju kemudian diserahkan kepada Sekretaris dan Bendahara untuk diarsipkan.

4.2 Disain Sistem Secara Global

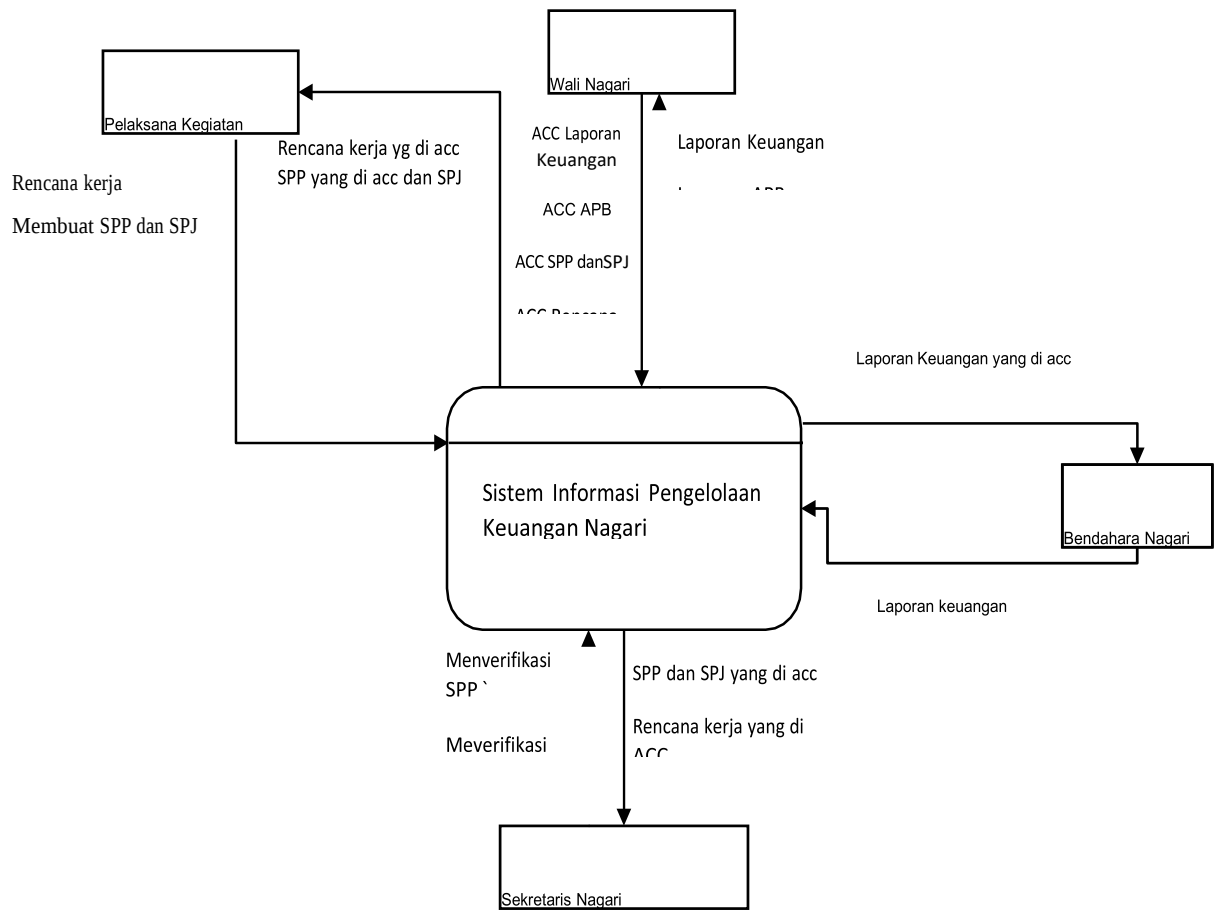
Disain *global* merupakan perancangan secara umum dari sistem baru yang akan dikembangkan oleh Penulis. Di dalam sistem pengelolaan keuangan nagari yang baru ini akan diberikan gambaran melalui bentuk Aliran Sistem Informasi Baru, *Context Diagram*, *Entity Relationship Diagram* dan serta *Struktur Program*.

4.2.1 Context Diagram

Context diagram merupakan sebuah pandangan yang menyeluruh darisebuah sistem organisasi yang menunjukkan batasan sistem, *entity external* yang berintegrasi dengan sistem informasi yang didapatkan antara *entity* dengan sistem.

Context diagram merupakan merupakan garis besar dari suatu rancangan sistem yang akan diusulkan. *Context diagram* hanya memiliki satu proses dan menunjukkan hubungan antara sistem dengan lingkungan luarnya. Berikut dijelaskan *context diagram* yang menggambarkan garis besar dari sistem komputerisasi pengelolaan keuangan nagari.

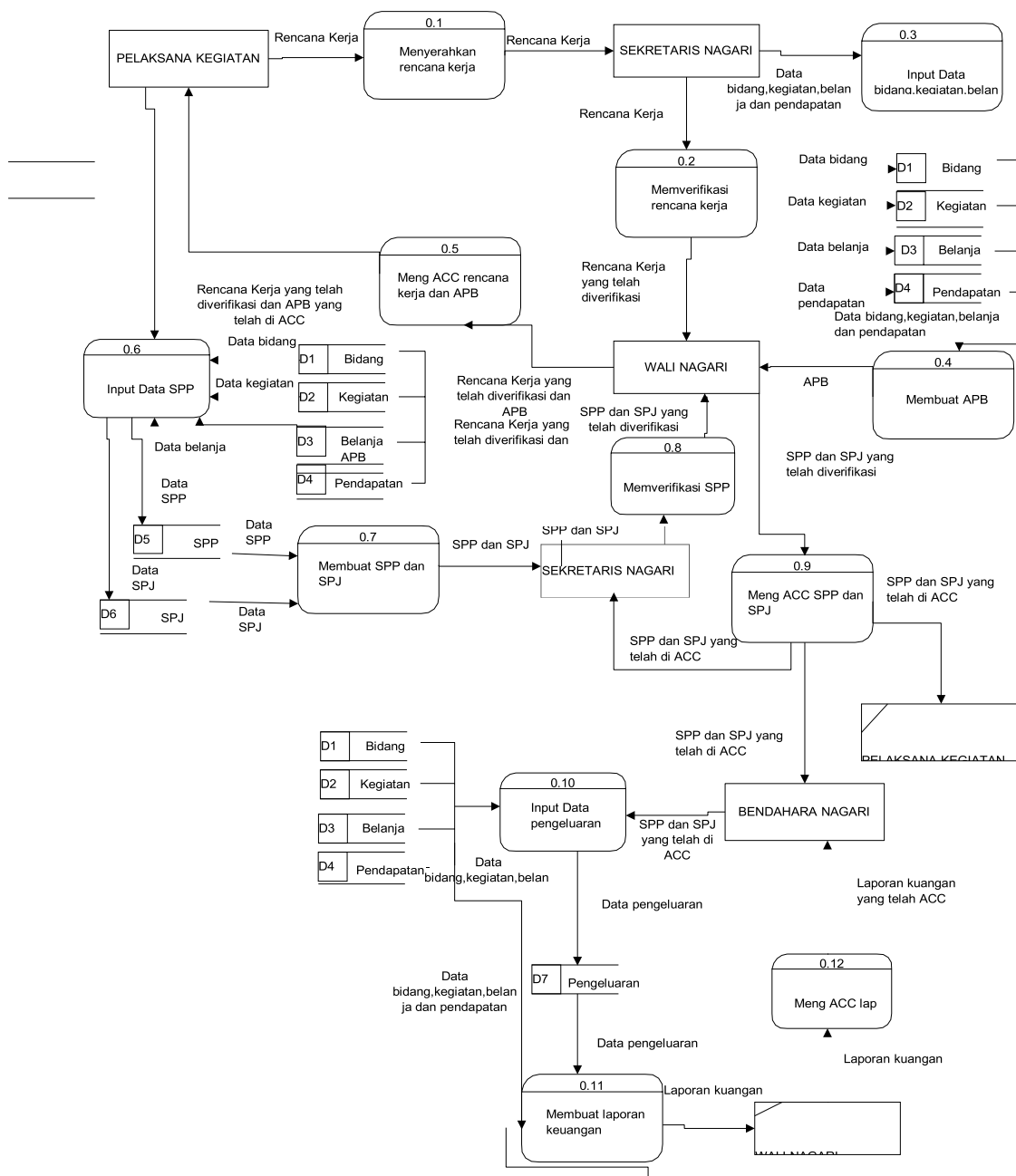
Untuk entitasnya melibatkan beberapa buah *entity* yaitu Pelaksana Kegiatan, Sekretaris Nagari, Bendahara dan Walikota. Kesemua entitas tersebut saling terlibat dalam sistem komputerisasi pengelolaan keuangan nagari. Untuk lebih jelasnya dapat kita lihat gambar 4.3 *context diagram* berikut :



Gambar 4.3 Context Diagram (CD)

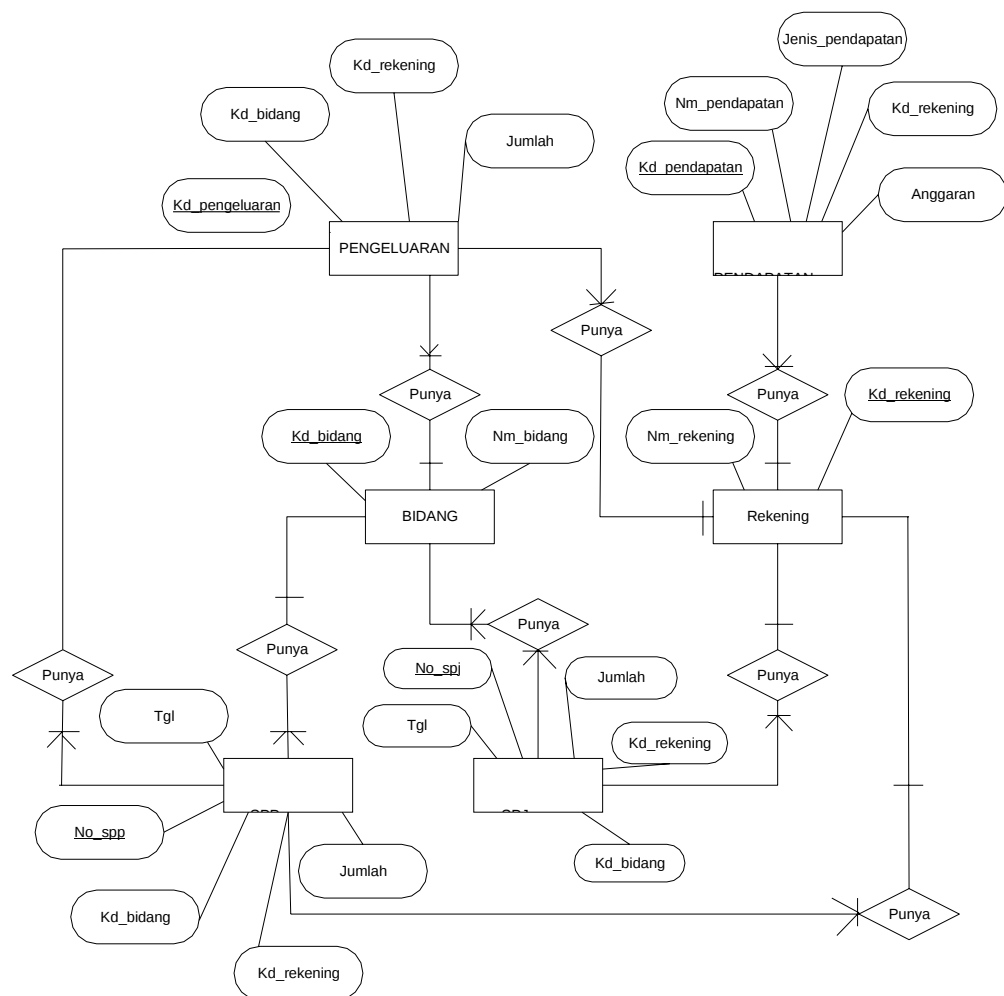
4.2.3 Data Flow Diagram (DFD)

Data flow diagram merupakan bentuk aliran data dalam semua sistem informasi. *Data flow diagram* menjelaskan proses terbentuknya informasi dari program aplikasi yang digunakan.



4.2.4 Entity relationship diagram (ERD)

Entity relationship diagram merupakan gambaran tentang relasi dari setiap *entity* atau atribut yang dari *file* yang digunakan untuk menggabungkan atau menghubungkan data sehingga dapat dirancang laporan atau input data yang diinginkan. Berikut ini penulis jelaskan aliran Entity relationship diagram dari *file-file* yang dirancang untuk aplikasi sistem komputerisasi pengelolaan keuangan, yaitu :



Gambar 4.5 Entity Relationship Diagram (ERD)

4.3 Desain Input

Bentuk rancangan input tersebut dapat dilihat pada gambar di bawah ini :

a. Desain Input Data User / Admin

Gambar 4.6 Tampilan Input Data User / Admin

(untuk proses tombol save)

```
try {  
    sql="insert into users values('"+user.getText()+"', '"+  
        +jComboBox1.getSelectedItem()+"', '"+pass.getText()+"')";  
  
    st=con.createStatement();st.execute(sql);  
    hapus();  
    JOptionPane.showMessageDialog(rootPane, "data tersimpan");  
} catch (Exception e) { JOptionPane.showMessageDialog(rootPane,  
    "gagal tersimpan");  
}
```

(untuk proses tombol cancel)

```
user.setText("");  
pass.setText(""); jComboBox1.setSelectedItem(null);  
pass.requestFocus();
```

(untuk proses tombol close)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data operator yang akan login ke sistem

b. Tampilan Form Login

Gambar 4.7 Tampilan Form Login(untuk proses tombol login)

```
try {  
    sql="select * from users where username='"+user.getText()+"';  
    st=con.createStatement();  
    rs=st.executeQuery(sql);while(rs.next()){  
        if (user.getText().equals(rs.getString("username")) &&  
pass.getText().equals(rs.getString("pass"))&&  
jComboBox1.getSelectedItem().equals(rs.getString("level")))  
        {  
            new menu().show();atur();  
        } else  
            JOptionPane.showMessageDialog(null, "Maaf masukkan usernamedan password anda dengan  
benar...");  
pass.setText(""); user.requestFocus();  
    }  
} catch (Exception e) {  
}
```

(untuk proses tombol cancel)

```
user.setText("");  
pass.setText(""); jComboBox1.setSelectedItem(null);  
pass.requestFocus();
```

(untuk proses tombol close)

```
this.dispose();
```

Form di atas digunakan untuk proses login ke sistem sesuai level.

c. Desain Form Input Data Bidang



No	KODE BIDANG	NAMA BIDANG
1	1	Pemerintahan
2	2	Pembangunan
3	3	Pembinaan
4	4	Pemberdayaan

Gambar 4.8 Tampilan Form Input Data Bidang(untuk proses tombol simpan)

```
try {  
    sql = "insert into bidang values('" + a.getText() + "','"  
    + b.getText() + "')"; st = con.createStatement();  
    st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin  
    menyimpan data ini??", "simpan data di sini!", JOptionPane.YES_NO_OPTION); if (ok == 0) {  
        hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap hubungi  
    programmer kami!" + e.getMessage());  
}  
(untuk proses tombol batal)
```

```
a.setText("");  
b.setText(""); a.requestFocus();
```

(untuk proses tombol hapus)

```
try {
    sql = "delete from bidang where kd_bidang='" + a.getText() + "'"; st = con.createStatement();
    st.execute(sql);
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???", "hapus
    di sini ya!!", JOptionPane.YES_NO_OPTION);
        if (ok == 0) {hapus();
                                tampil();
        }
    } catch (Exception e) {
    }
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? "+
"\nDengan kd_bidang"+a.getText()+"",
"Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){
try {
    koneksi();
    sql = "UPDATE bidang set nm_bidang='"+b.getText()+

                                "'where kd_bidang='"+a.getText()+"';st =
                                con.createStatement();
                                st.execute(sql);
                                JOptionPane.showMessageDialog(null, "Data berhasil diedit
                                !!!"+sql);

                                hapus();
                                tampil();
                                } catch (Exception e) {
                                JOptionPane.showMessageDialog(null, "Data gagal di update,
                                Kesalahan pada"+e);
                                }
    }
}
```

(untuk proses tombol keluar)

```
this.dispose();
Form di atas digunakan untuk menginputkan data bidang.
```

d. Desain Form Input Data Rekening

No	KODE REKENING	NAMA REKENING
1	1.1	Pendapatan Asli Nagari
2	1.2	Alokasi Dana Desa
3	1.3	Alokasi Dana Nagari
4	1.4	Bantuan Propinsi
5	2.1	Bantuan Daerah

Gambar 4.10 Tampilan Form Input Data Belanja(untuk proses tombol simpan)

```
try {  
    sql = "insert into belanja values('" + a.getText() + "','"  
    + b.getText() + "','"  
    + b1.getText() + "')"; st =  
    con.createStatement();st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??", "simpan  
data di sini!", JOptionPane.YES_NO_OPTION);  
    if (ok == 0) {hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap hubungi  
programmer kami!" + e.getMessage());  
}
```

(untuk proses tombol tambah)

```
a.setText("");  
b.setText("");  
b1.setText(""); a.requestFocus();
```

(untuk proses tombol hapus)

```
try {
    sql = "delete from belanja where kd_belanja=" + a.getText() + """;st = con.createStatement();
    st.execute(sql);
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???", "hapus
    di sini ya!!", JOptionPane.YES_NO_OPTION);
        if (ok == 0) {hapus();
                                tampil();
        }
    } catch (Exception e) {
    }
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? "+
"\nDengan kd_belanja"+a.getText()+"",
"Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){
try {
    koneksi();
    sql = "UPDATE belanja set jenis_belanja="+b.getText()+
    ",nm_belanja="+b1.getText()+
    "where kd_belanja="+a.getText()+""";st = con.createStatement();
    st.execute(sql);
    JOptionPane.showMessageDialog(null, "Data berhasil diedit
    !!!"+sql);
        hapus();
        tampil();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data gagal di update,
        Kesalahan pada"+e);
    }
}
```

(untuk proses tombol keluar)

```
this.dispose();
Form di atas digunakan untuk menginputkan data belanja.
```

e. Desain Form Input Data Pendapatan

No	NO pendapatan	NAMA PENDAP...	KODE BELANJA	PENDAPATAN
1	1	Pendapatan A...	1.1	1000000
2	2	Dana Transfer	1.2	800000000
3	3	Dana Transfet	1.3	500000000
4	4	Dana Transfer	1.4	12000000

Gambar 4.11 Tampilan Form Input Data Pendapatan(untuk proses tombol simpan)

```
try {
    sql = "insert into pendapatan values('" + a.getText() + "','"
    + b1.getText() + "','"
    + d.getSelectedItem() + "','"
    + c1.getText() + "','"
    + e.getText() + "')"; st =
    con.createStatement();st.execute(sql);
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??", "simpan
    data di sini!", JOptionPane.YES_NO_OPTION);
    if (ok == 0) {hapus();
        tampil();
    }
} catch (Exception e) {
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap hubungi
    programmer kami!" + e.getMessage());
}
```

(untuk proses tombol tambah)

```
a.setText("");
b1.setText("");
c1.setText(""); d.setSelectedItem(null);e.setText("");
a.requestFocus();
```

(untuk proses tombol hapus)

```
try {
    sql = "delete from pendapatan where kd_pendapatan='" + a.getText() + "'";
    st = con.createStatement(); st.execute(sql);
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???", "hapus
    di sini ya!!", JOptionPane.YES_NO_OPTION);
        if (ok == 0) {hapus();
                                tampil();
        }
    } catch (Exception e) {
    }
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? "+
"\nDengan kd_pendapatan='"+a.getText()+"'", "Konfirmasi edit
data",JOptionPane.YES_NO_OPTION);
if (ok==0){try {
    koneksi();
    sql = "UPDATE pendapatan set nm_pendapatan='"+
    b1.getText()+

                                "','kd_belanja='"+d.getSelectedItem()+
                                "','kd_belanja='"+c1.getText()+
                                "','anggaran='"+e.getText()+
                                "'where kd_pendapatan='"+a.getText()+"';st =
                                con.createStatement();
                                st.execute(sql);
                                JOptionPane.showMessageDialog(null, "Data berhasil diedit
                                !!!"+sql);

                                hapus();
                                tampil();
                                } catch (Exception e) {
                                JOptionPane.showMessageDialog(null, "Data gagal di update,
                                Kesalahan pada"+e);
                                }
    }
}
```

(untuk proses tombol keluar)

this.dispose();

Form di atas digunakan untuk menginputkan data pendapatan.

f. Desain Form Input Data Pengeluaran

No	NO pengeluaran	KODE BIDANG	KODE BELANJA	JUMLAH
1	1	1	2.1	500000000
2	2	1	2.2	2000000
3	3	1	2.3	5000000
4	4	2	2.4	2000000

Gambar 4.12 Tampilan Form Input Data Pengeluaran(untuk proses tombol simpan)

```
try {  
    sql = "insert into pengeluaran values('" + a.getText() + "','"  
    + b.getSelectedItem() + "','"  
    + c.getSelectedItem() + "','"  
    + d.getSelectedItem() + "','"  
    + e.getText() + "')"; st =  
    con.createStatement();st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??", "simpan  
data di sini!", JOptionPane.YES_NO_OPTION);  
    if (ok == 0) {hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap hubungi  
programmer kami!" + e.getMessage());  
}
```

(untuk proses tombol simpan)

```

a.setText(""); b.setSelectedItem(null);
c.setSelectedItem(null);
d.setSelectedItem(null);e.setText("");
a.requestFocus();

```

(untuk proses tombol hapus)

```

try {
sql = "delete from pengeluaran where kd_pengeluaran=" + a.getText() + "";
st = con.createStatement();st.execute(sql);
int ok = JOptionPane.showConfirmDialog(null, " anda yakin menghapus data ini???", "hapus
di sini ya!!", JOptionPane.YES_NO_OPTION);
    if (ok == 0) {hapus();
                                tampil();
    }
} catch (Exception e) {
}

```

(untuk proses tombol edit)

```

int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? "+
"\nDengan kd_pendapatan="+a.getText()+""", "Konfirmasi edit
data",JOptionPane.YES_NO_OPTION);
if (ok==0){try {
koneksi();
sql = "UPDATE pendapatan set nm_pendapatan="+b1.getText()+
"",kd_belanja="+d.getSelectedItem()+ "",kd_belanja="+c1.getText()+
"",anggaran="+e.getText()+
""where kd_pendapatan="+a.getText()+""";st = con.createStatement();
st.execute(sql);
JOptionPane.showMessageDialog(null, "Data berhasil diedit
!!!"+sql);
        hapus();
        tampil();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data gagal di update,
Kesalahan pada"+e);
    }
}

```

(untuk proses tombol keluar)

this.dispose();

Form di atas digunakan untuk menginputkan data pengeluaran.

g. Desain Form Input Data SPP

No	NO SPP	TANGGAL	KODE BIDA...	KODE BELA...	JUMLAH
1	1	2017-03-01	1	2.1	2000000
2	1	2017-03-02	1	2.1	2000000
3	1	2017-03-02	1	2.2	500000
4	2	2017-03-01	2	2.2	100000
5	2	2017-03-02	2	2.2	100000

Gambar 4.13 Tampilan Form Input Data SPP(untuk proses tombol simpan)

```
try {  
    sql = "insert into spp values('" + a.getText() + "','"  
    + tgl + "','"  
    + b.getSelectedItem() + "','"  
    + c.getSelectedItem() + "','"  
    + d.getSelectedItem() + "','"  
    + e.getText() + "')"; st =  
    con.createStatement(); st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??", "simpan  
data di sini!", JOptionPane.YES_NO_OPTION);  
    if (ok == 0) {hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap hubungi  
programmer kami!" + e.getMessage());  
}
```

(untuk proses tombol simpan)

```

a.setText(""); b.setSelectedItem(null);
c.setSelectedItem(null);
d.setSelectedItem(null); e.setText("");
a.requestFocus();

```

(untuk proses tombol hapus)

```

try {
    sql = "delete from spp where no_spp=" + a.getText() + """; st = con.createStatement();
    st.execute(sql);
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???", "hapus
    di sini ya!!", JOptionPane.YES_NO_OPTION);
        if (ok == 0) {hapus();
                                tampil();
        }
    } catch (Exception e) {
    }

```

(untuk proses tombol edit)

```

int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? " +
"\nDengan no_spp=" + a.getText() + "",
"Konfirmasi edit data", JOptionPane.YES_NO_OPTION); if (ok==0){
try {
koneksi();
sql = "UPDATE spp set tgl=" + tgl + ", kd_bidang=" + b.getSelectedItem() +
", kd_kegiatan=" + c.getSelectedItem() + ", kd_belanja=" + d.getSelectedItem() +
", jumlah=" + e.getText() +
"where no_spp=" + a.getText() + """; st = con.createStatement();
st.execute(sql);
JOptionPane.showMessageDialog(null, "Data berhasil diedit
!!!" + sql);
        hapus();
        tampil();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Data gagal di update,
        Kesalahan pada" + e);
    }

```

```
}
```

(untuk proses tombol keluar)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data spp.

h. Desain Form Input Data SPJ

No	NO SPP	TANGGAL	KODE BIDA...	KODE BELA...	JUMLAH
1	1	2017-03-01	1	2.1	2000000
2	1	2017-03-02	1	2.1	2000000
3	1	2017-03-02	1	2.2	500000
4	2	2017-03-01	2	2.2	100000
5	2	2017-03-02	2	2.2	1000000

Gambar 4.14. Tampilan Form Input Data SPJ(untuk proses tombol simpan)

```
try {  
    sql = "insert into spj values('" + a.getText() + "','"  
    + tgl + "','"  
    + b.getSelectedItem() + "','"  
    + c.getSelectedItem() + "','"  
    + d.getSelectedItem() + "','"  
    + e.getText() + "')"; st =  
    con.createStatement();st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??", "simpan  
data di sini!", JOptionPane.YES_NO_OPTION);  
    if (ok == 0) {hapus();
```

```

tampil();
}
} catch (Exception e) {
JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap
hubungi programmer kami!" + e.getMessage());
}

```

(untuk proses tombol simpan)

```

a.setText(""); b.setSelectedItem(null);
c.setSelectedItem(null);
d.setSelectedItem(null); e.setText("");
a.requestFocus();

```

(untuk proses tombol hapus)

```

try {
sql = "delete from spp where no_spj=" + a.getText() + """; st =
con.createStatement();
st.execute(sql);
int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",
"hapus di sini ya!!", JOptionPane.YES_NO_OPTION);
if (ok == 0) {hapus();
tampil();
}
} catch (Exception e) {
}
}

```

(untuk proses tombol edit)

```

int ok = JOptionPane.showConfirmDialog(null, "Anda yakin ingin mengedit data??
"+
"\n Dengan no_spj=" + a.getText() + "",
"Konfirmasi edit data", JOptionPane.YES_NO_OPTION); if (ok == 0) {
try {
koneksi();
sql = "UPDATE spj set tgl=" + tgl + ", kd_bidang=" + b.getSelectedItem() +
", kd_kegiatan=" + c.getSelectedItem() +
", kd_belanja=" + d.getSelectedItem() + ", jumlah=" + e.getText() +
" where no_spj=" + a.getText() + """; st = con.createStatement();
st.execute(sql);

```

```

JOptionPane.showMessageDialog(null, "Data berhasil diedit
!!!"+sql);
hapus();
tampil();
} catch (Exception e) {
JOptionPane.showMessageDialog(null, "Data gagal di update,
Kesalahan pada"+e);
}
}

```

(untuk proses tombol keluar)

this.dispose();

Form di atas digunakan untuk menginputkan data SPJ.

4.4. Desain Output

bentuk rancangan output tersebut dapat dilihat pada gambar di bawah ini :

a. Laporan Anggaran Pendapatan



Gambar 4.15 Tampilan Laporan Anggaran Pendapatan dan Belanja(untuk

proses tombol print)

String periode= a.getText().toString(); String bk =

b.getText().toString();try {

JDialog window = new JDialog(this, "Print Preview");

window.setSize(Toolkit.getDefaultToolkit().getScreenSize()); SwingReportViewer viewer

= new SwingReportViewer(); URLRenderData renderConnection = new

URLRenderData(

"http://localhost:9000/?report=file:" +

```

System.getProperty("user.dir")+"/src/Laporan/apb.rpt&prompt0=" + periode +
"&prompt1=" + bk + "");
viewer.addNewReportView( renderConnection ); window.getContentPane().add(
BorderLayout.CENTER, viewer);window.setVisible( true );
} catch (Exception e) {
e.printStackTrace();
}

```

Form di atas digunakan untuk menampilkan laporan anggaran pendapatan dan belanja.

b. Laporan Anggaran Belanja

Gambar 4.16 Tampilan Laporan Anggaran Belanja(untuk proses tombol print)

```

String periode= a.getText().toString(); String bk =
b.getText().toString();
String bi = b1.getSelectedItem().toString();try {
JDialog window = new JDialog(this, "Print Preview");
window.setSize(Toolkit.getDefaultToolkit().getScreenSize()); SwingReportViewer viewer =
new SwingReportViewer(); URLRenderData renderConnection = new URLRenderData(
"http://localhost:9000/?report=file:" +
System.getProperty("user.dir")+"/src/Laporan/belanja.rpt&prompt0="
+ periode + "&prompt1=" + bk + "&prompt2=" + bi + "");
viewer.addNewReportView( renderConnection ); window.getContentPane().add(
BorderLayout.CENTER, viewer);window.setVisible( true );
}

```

```

} catch (Exception e) {
e.printStackTrace();
}

```

Form di atas digunakan untuk menampilkan laporan anggaran belanja.

c. Desain Laporan Surat Perintah Pembayaran

Gambar 4.17 Tampilan Desain Laporan Surat Perintah Pembayaran

(untuk proses tombol print)

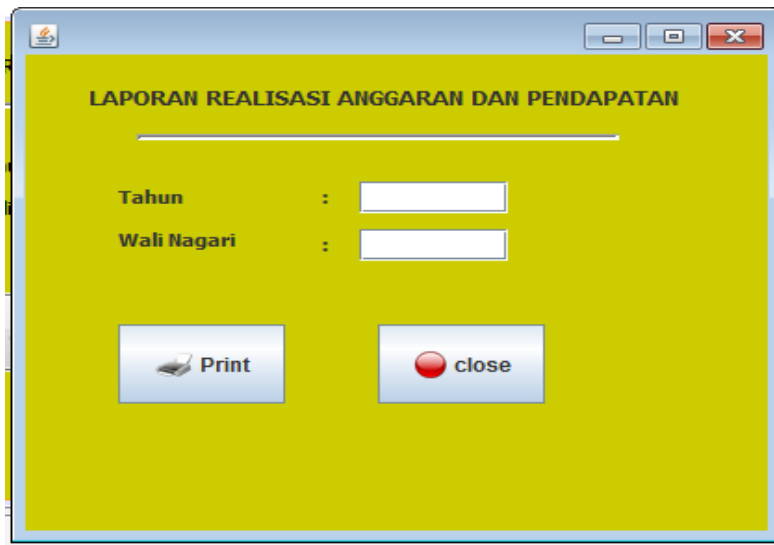
```

String periode= a.getText().toString(); String bk
= b.getText().toString(); String seknag=
c.getText().toString();String pk =
d.getText().toString(); try {
JDialog window = new JDialog(this, "Print Preview");
window.setSize(Toolkit.getDefaultToolkit().getScreenSize());
SwingReportViewer viewer = new SwingReportViewer(); URLRenderData
renderConnection = new URLRenderData(
"http://localhost:9000/?report=file:" +
System.getProperty("user.dir")+"/src/Laporan/spp.rpt&prompt0=" + periode +
"&prompt1=" + bk + "&prompt2=" + seknag + "&prompt3=" + pk + "");
viewer.addNewReportView( renderConnection );
window.getContentPane().add( BorderLayout.CENTER, viewer);
window.setVisible( true );
} catch (Exception e) {
e.printStackTrace();
}

```

Form di atas digunakan untuk mencetak laporan surat perintah pembayaran.

d. Desain Laporan Realisasi Anggaran dan Pendapatan



The image shows a Java Swing window with a yellow background. At the top, the title 'LAPORAN REALISASI ANGGARAN DAN PENDAPATAN' is displayed. Below the title, there are two text input fields. The first field is labeled 'Tahun' and the second is labeled 'Wali Nagari'. Below these fields, there are two buttons: 'Print' with a printer icon and 'close' with a red circle icon.

Gambar 4.18 Tampilan Desain Laporan Realisasi Anggaran dan Pendapatan (untuk proses tombol print)

```
String periode= a.getText().toString(); String  
bk = b.getText().toString();
```

```
try {  
    JDialog window = new JDialog(this, "Print Preview");  
    window.setSize(Toolkit.getDefaultToolkit().getScreenSize());  
    SwingReportViewer viewer = new SwingReportViewer(); URLRenderData  
    renderConnection = new URLRenderData(  
        "http://localhost:9000/?report=file:" +  
        System.getProperty("user.dir")+"/src/Laporan/rap.rpt&prompt0=" +periode +  
        "&prompt1=" + bk + "");  
    viewer.addNewReportView( renderConnection );  
    window.getContentPane().add( BorderLayout.CENTER, viewer);  
    window.setVisible( true );  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

Form di atas digunakan untuk mencetak laporan realisasi anggaran dan pendapatan.

4.5 Desain File

Dalam merancang suatu sistem yang baik dibutuhkan beberapa file yang bertujuan untuk memudahkan pengambilan informasi data. Bentuk desain file akan mendeskripsikan file-file dalam database yang digunakan dalam perancangan sebuah sistem. Desain file merupakan suatu media penyimpanan yang bersifat sementara, disamping itu fungsi desain file tersebut digunakan untuk proses pengolahan data, baik itu proses pengentrian data maupun proses pembuatan laporan data atau menu tampil. Adapun bentuk rancangan file tersebut dapat dilihat pada format berikut:

1. Desain File Bidang

Desain file bidang merupakan desain tabel bidang pada database yang terdiri dari dua field, seperti Tabel 4.1.

Nama Database : dbkeuangan.sql Nama Tabel :

bidang

Field Key : **Kd_bidang**

Tabel 4.1 Desain File Bidang

No.	Field Name	Type	Width	Description
1	Kd_bidang	Varchar	11	Primary Key
2	Nm_bidang	Varchar	40	

2. Desain File rekening

Desain file belanja merupakan desain tabel anggaran pada database yang terdiri dari tiga field, seperti Tabel 4.2.

Nama Database : dbkeuangan.sql Nama Tabel :

belanja

Field Key : **kd_rekening**

Tabel 4.2 Desain File belanja

No.	Field Name	Type	Width	Description
1	Kd_rekening	Varchar	11	Primary Key
2	Nm_rekening	Varchar	50	

3. Desain File Pendapatan

Desain file pendapatan merupakan desain tabel pendapatan pada database yang terdiri dari lima field, seperti Tabel 4.3.

Nama Database : dbkeuangan.sql Nama Tabel :

pendapatan

Field Key : **kd_pendapatan**

Tabel 4.3 Desain File Pendapatan

No.	Field Name	Type	Width	Description
1	Kd_pendapatan	Varchar	11	Primary Key
2	Nm_pendapatan	Varchar	40	
3	Kd_rekening	Varchar	11	
4	Jenis_pendapatan	Varchar	30	
5	Anggaran	Int	8	

4. Desain File Pengeluaran

Desain file pengeluaran merupakan desain tabel pengeluaran pada database yang terdiri dari lima field, seperti Tabel 4.5.

Nama Database : dbkeuangan.sql Nama Tabel :

pengeluaran

Field Key : **kd_pengeluaran**

Tabel 4.5 Desain File Pengeluaran

No.	Field Name	Type	Width	Description
1	Kd_pengeluaran	Varchar	11	Primary Key
2	Kd_bidang	Varchar	11	
3	Kd_rekening	Varchar	11	
4	Kd_kegiatan	Varchar	11	

5. Desain File SPP

Desain file SPP merupakan desain tabel SPP pada database yang terdiri dari enam field, seperti Tabel 4.6.

Nama Database : dbkeuangan.sql Nama Tabel :

SPP

Field Key : **no_spp**

Tabel 4.6 Desain File SPP

No.	Field Name	Type	Width	Description
1	No_SPP	Varchar	11	Primary Key
2	Tanggal	Date	10	
2	Kd_bidang	Varchar	11	
3	Kd_rekening	Varchar	11	
4	Jumlah	Int	8	

6. Desain File SPJ

Desain file SPJ merupakan desain tabel SPJ pada database yang terdiri dari enam field, seperti Tabel 4.7.

Nama Tabel : SPJ Field Key :

no_spj

Tabel 4.7 Desain File SPJ

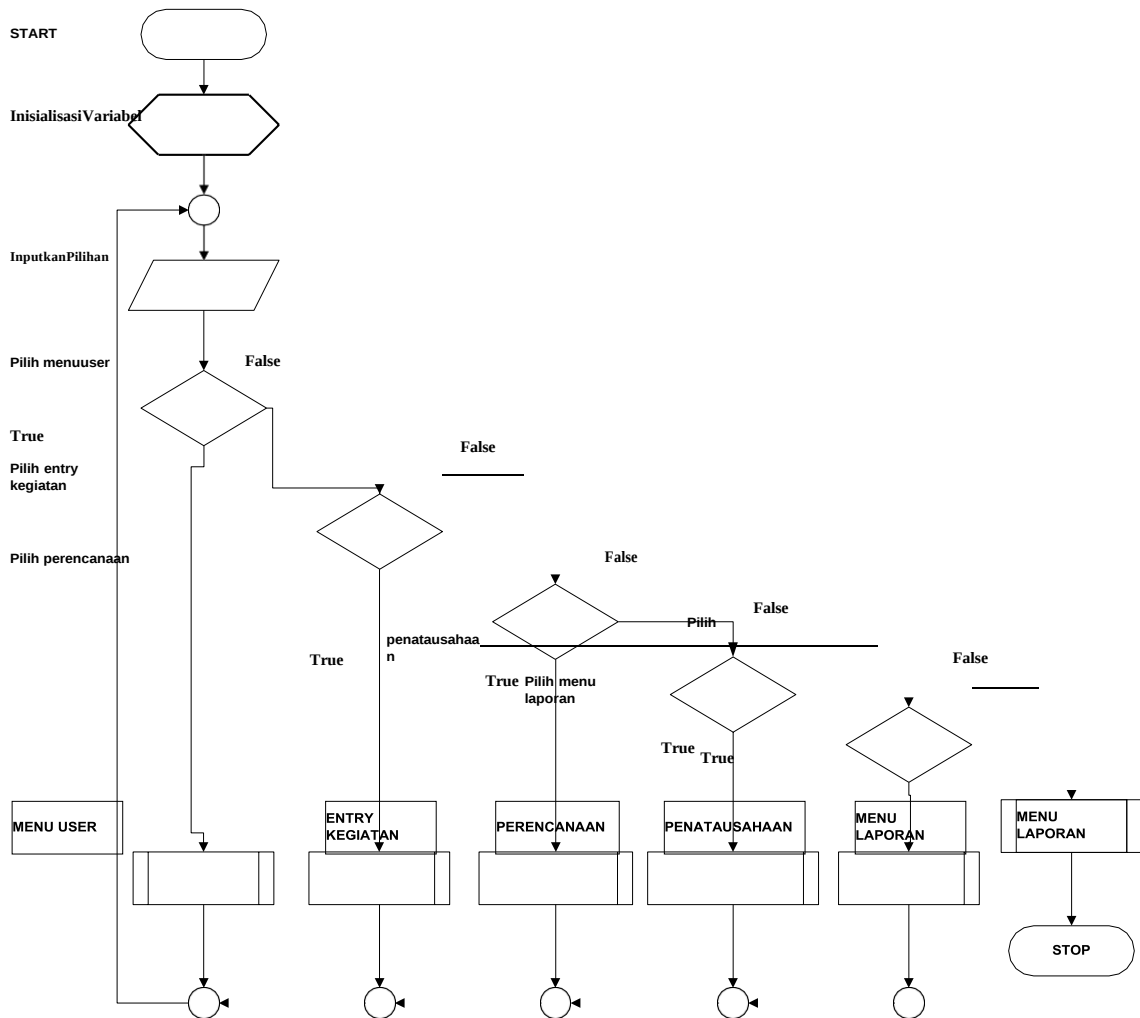
No.	Field Name	Type	Width	Description
1	No_spj	Varchar	11	Primary Key
2	Tanggal	Date	10	
2	Kd_bidang	Varchar	11	
3	Kd_rekening	Varchar	11	
4	Jumlah	Int	8	

4.5 Logika Program

Dalam merancang suatu sistem baru dibutuhkan beberapa alat bantu. Pada perancangan sistem informasi pengolahan penjualan rumah ini digunakan alat Bantu logika program berupa flowchart program. Ada pun flowchart yang digunakan antara lain:

1. Menu Utama

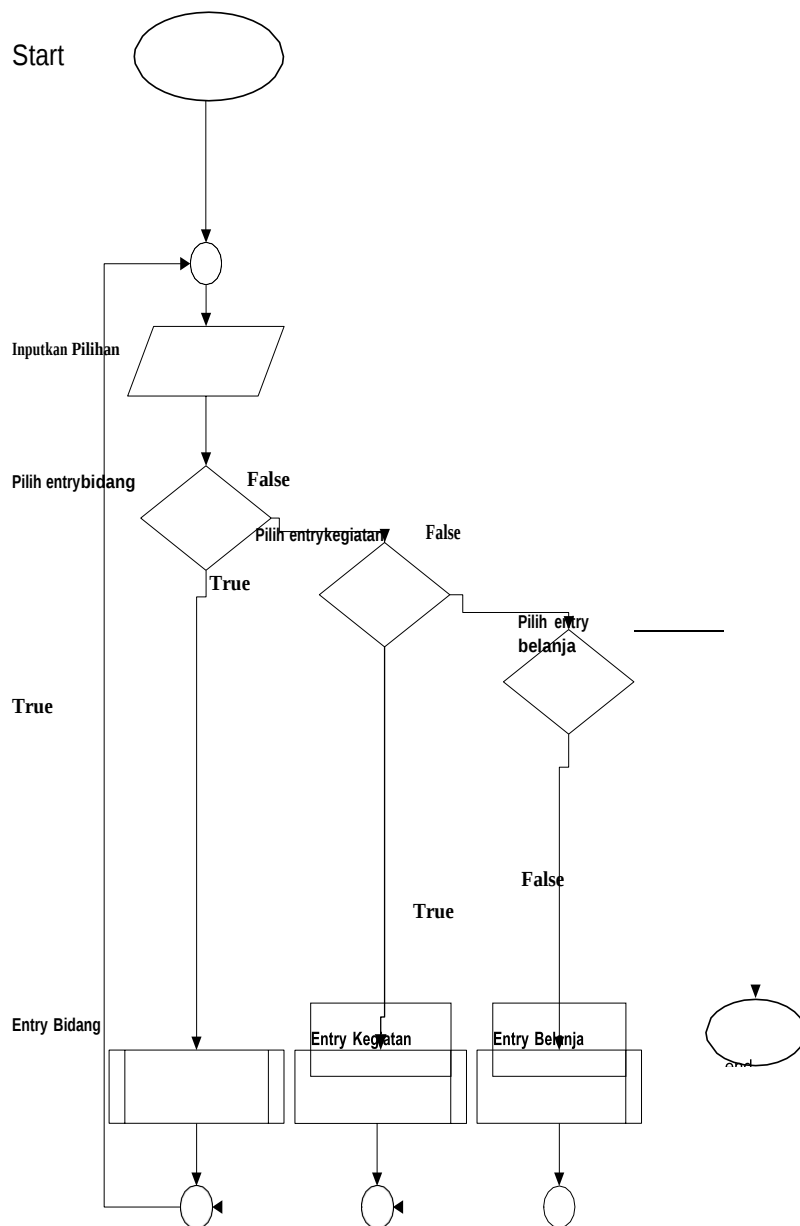
Aliran flowchart menu utama ini menggambarkan aliran program menu utama:



Gambar 4.19 Program Flowchart Menu Utama

2. Program Flowchart Entry Kegiatan

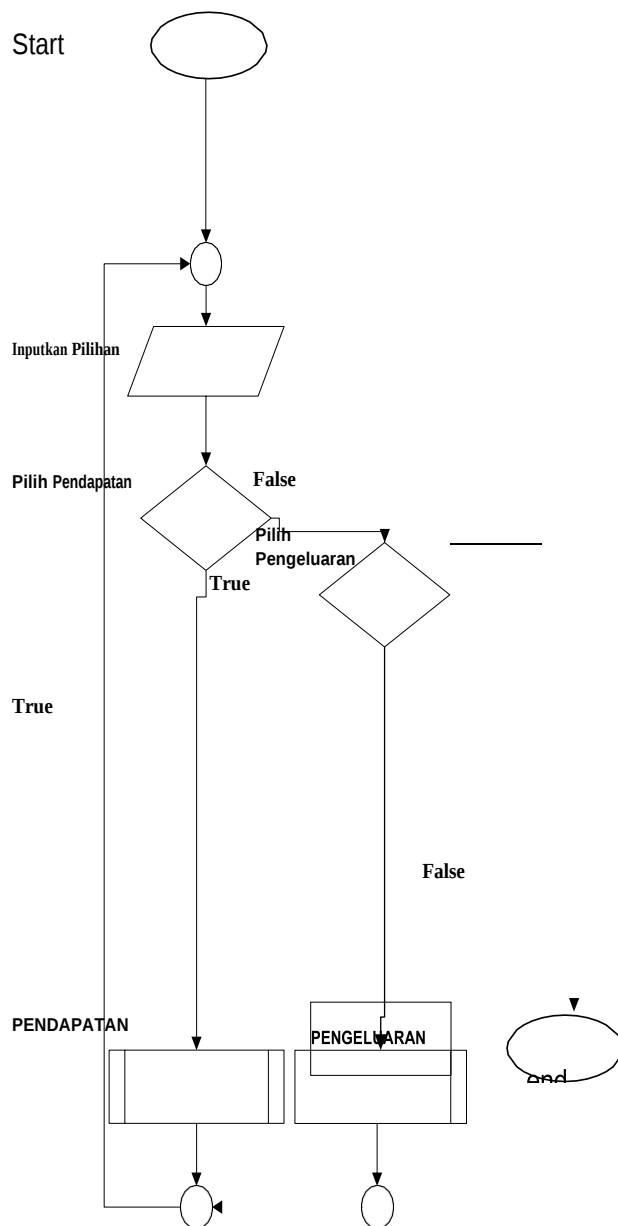
Dalam program flowchart berikut ini digambarkan bagaimana logika program dari pembuatan entry kegiatan, agar lebih jelas dapat dilihat pada gambar berikut ini:



Gambar 4.20 Program Flowchart Entry Kegiatan

3. Program Flowchart Perencanaan

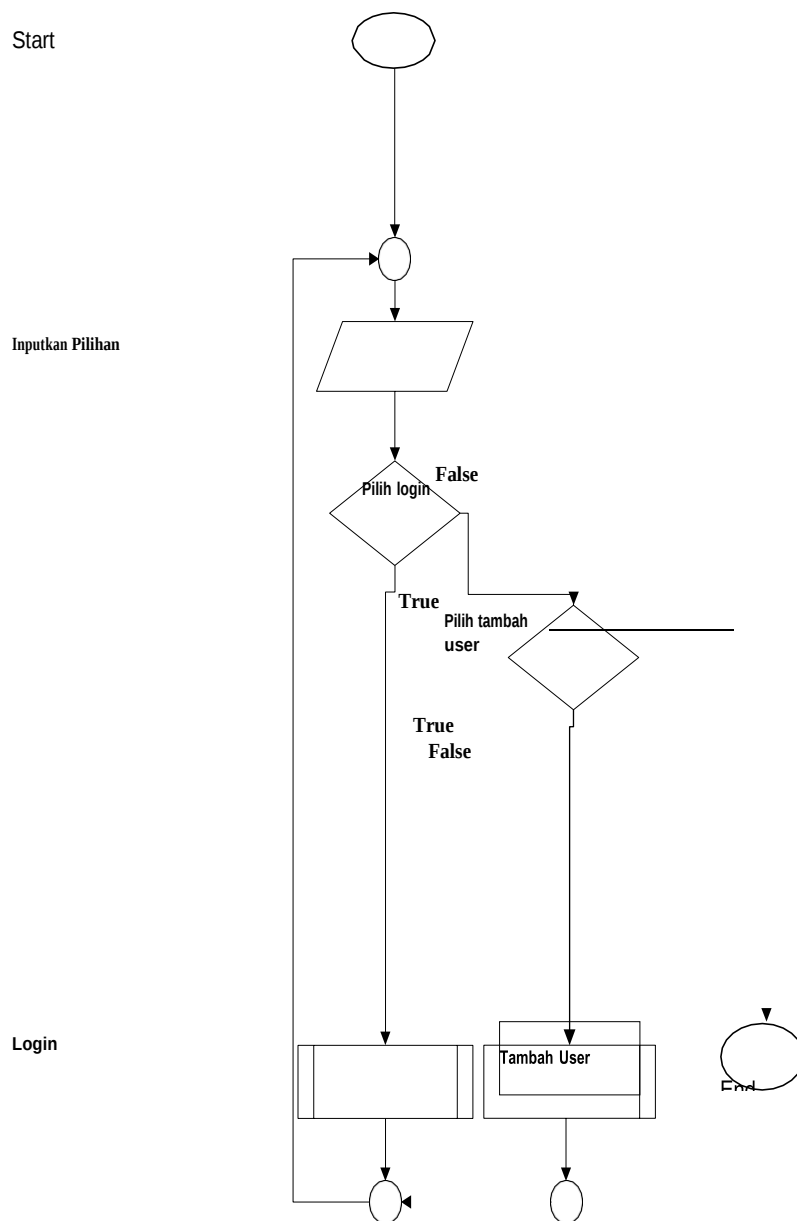
Dalam program flowchart berikut ini digambarkan bagaimana logika program dari pembuatan perencanaan, agar lebih jelas dapat dilihat pada gambar berikut ini:



Gambar 4.21 Program Flowchart Perencanaan

4. Program Flowchart Menu User

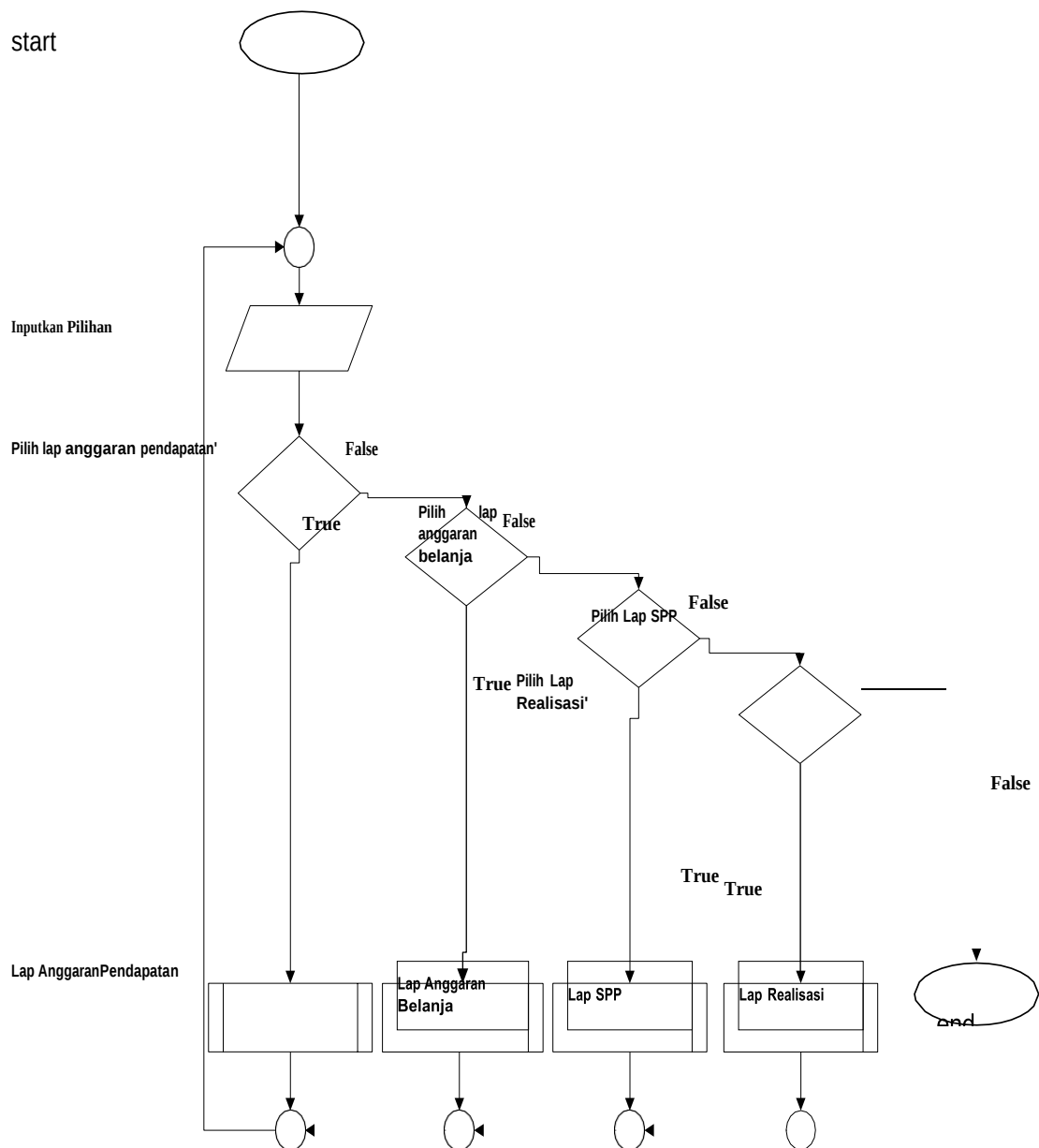
Dalam program flowchart berikut ini digambarkan bagaimana logika program dari pembuatan user dan login ke sistem, agar lebih jelas dapat dilihat pada gambar berikut ini:



Gambar 4.22 Program Flowchart Menu User

5. Program Flowchart Menu Laporan

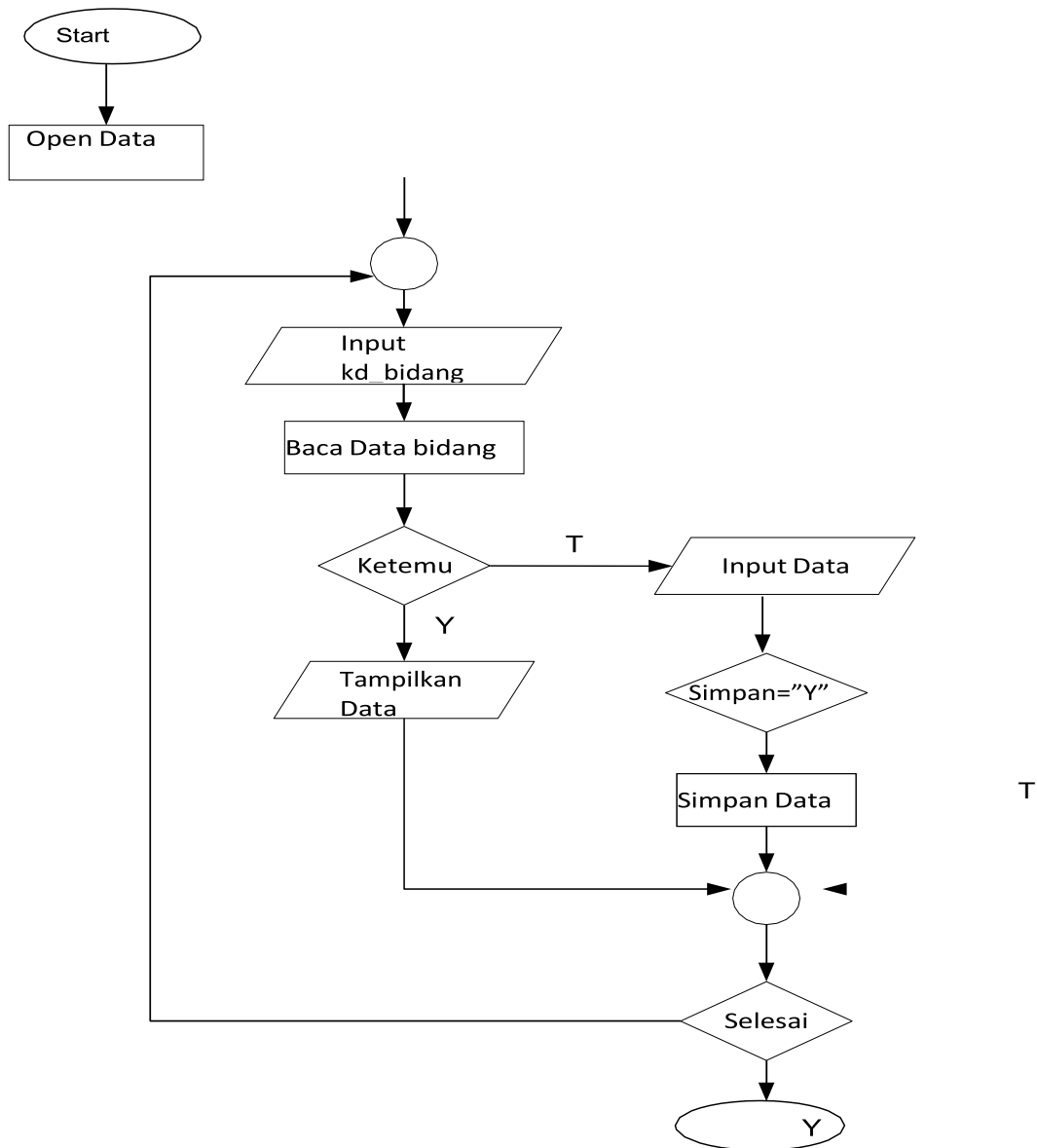
Dalam program flowchart berikut ini digambarkan bagaimana logika program dari pembuatan menu cetak, agar lebih jelas dapat dilihat pada gambar berikut ini:



Gambar 4.23 Program Flowchart Menu Laporan

6. Program Flowchart Entry Bidang

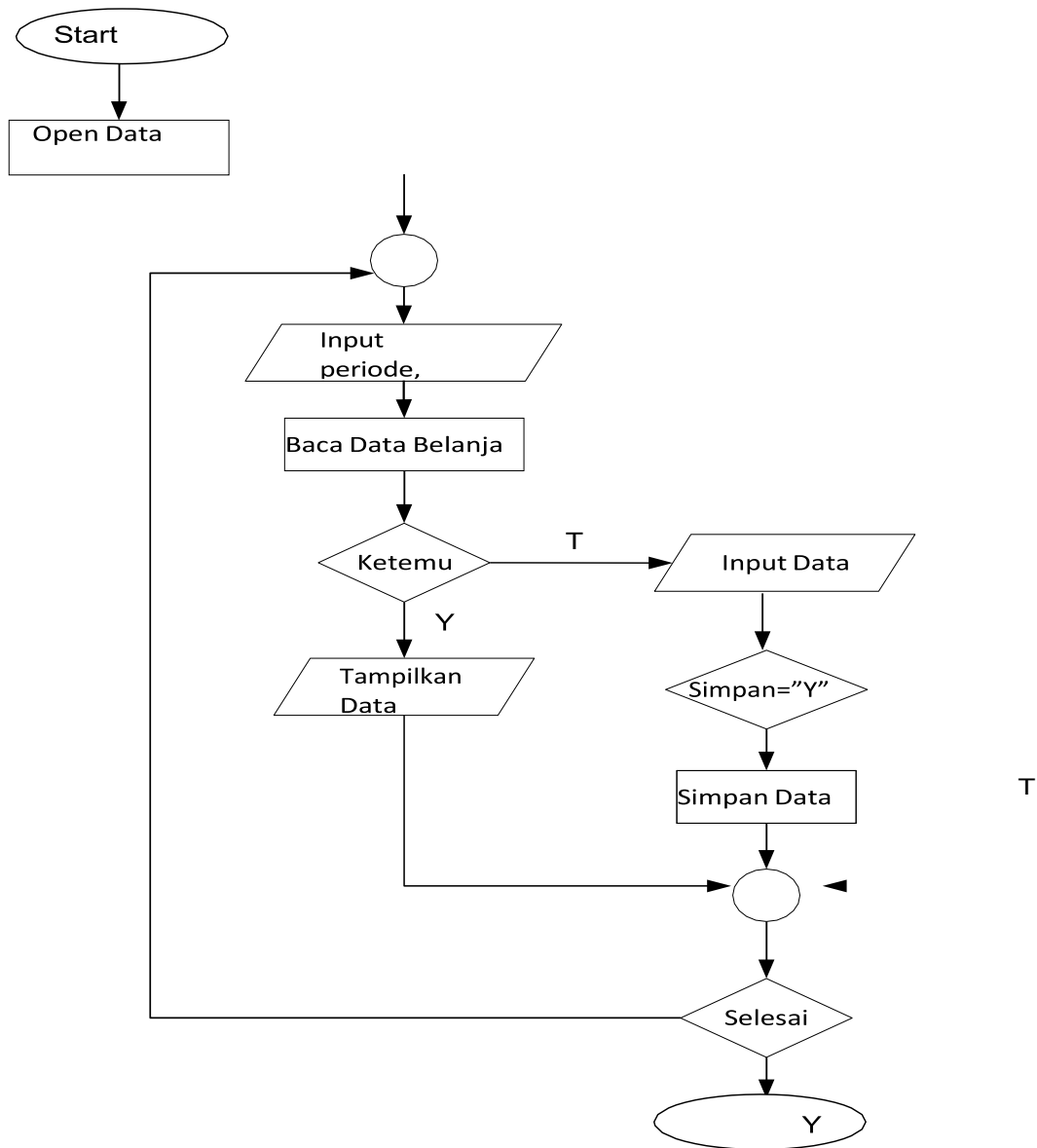
Dalam program flowchart berikut ini digambarkan bagaimana logika dari pembuatan entry bidang , agar lebih jelas dapat dilihat pada gambar berikut ini :



Gambar 4.24 Program Flowchart Entry Bidang

7. Program Flowchart Entry Belanja

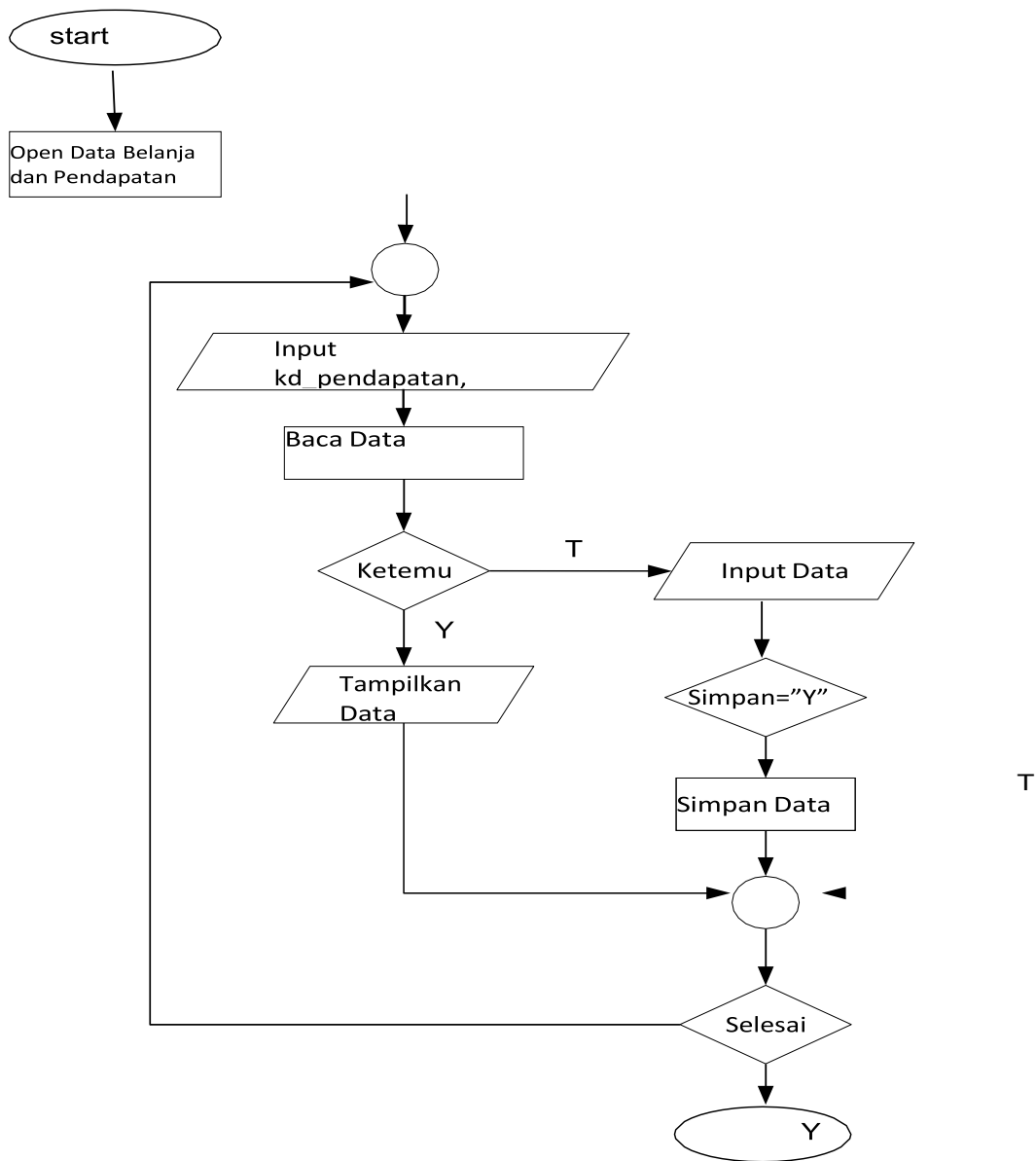
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan entry belanja, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.25 Program Flowchart Entry Rekening

8. Program Flowchart Pendapatan

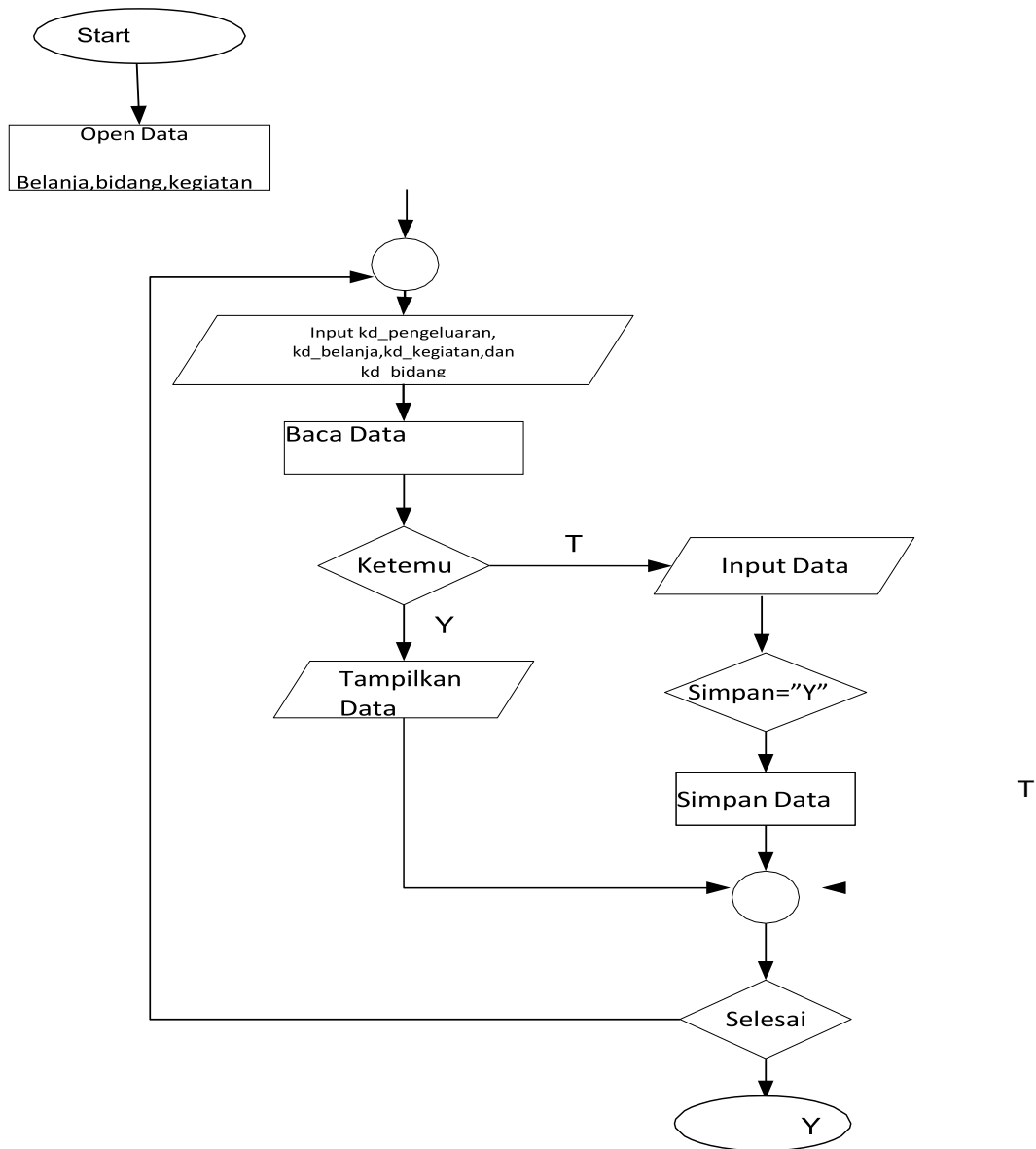
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan pendapatan, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.26 Program Flowchart Pendapatan

9. Program Flowchart Pengeluaran

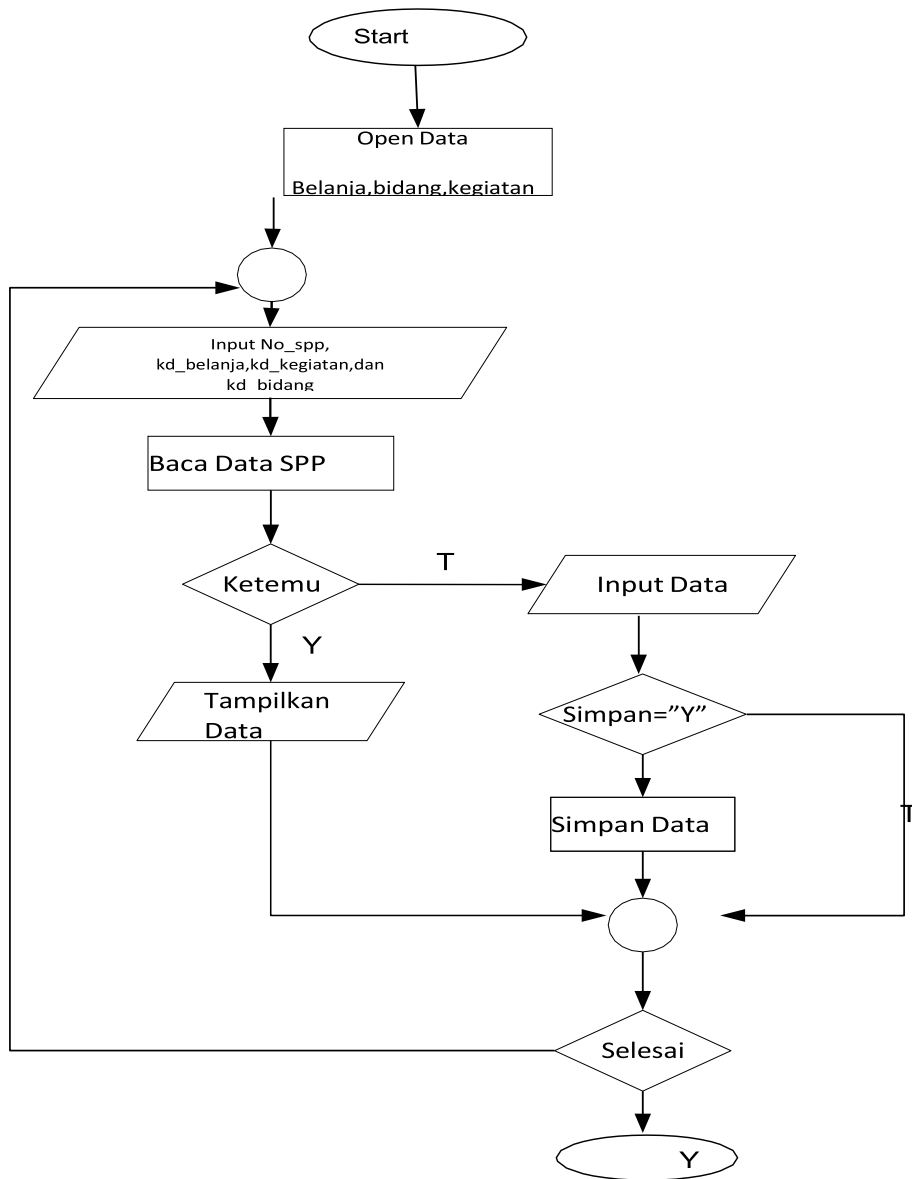
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan pengeluaran, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.28 Program Flowchart Pengeluaran

10. Program Flowchart SPP

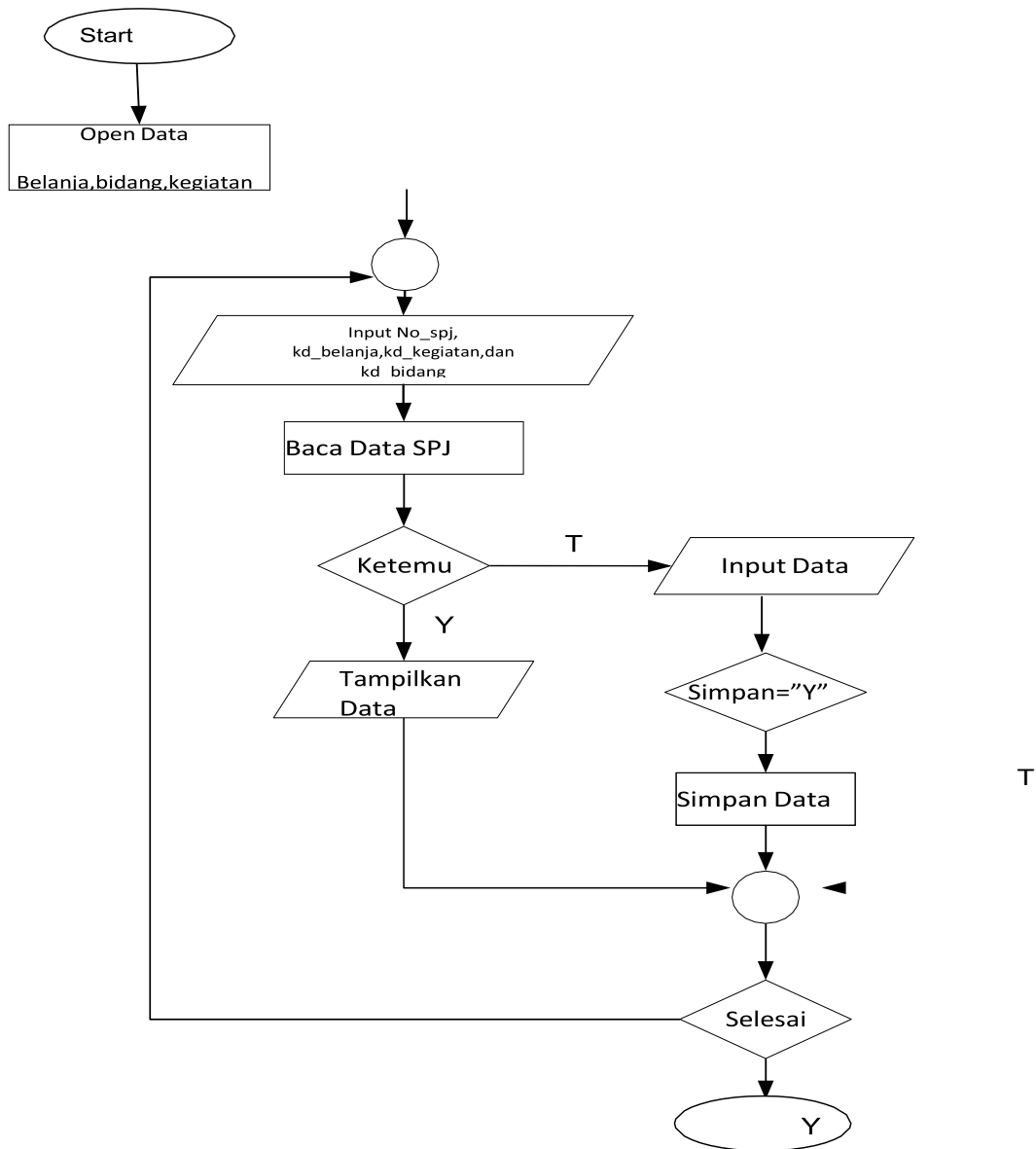
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan SPP, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.29 Program Flowchart SPP

11. Program Flowchart SPJ

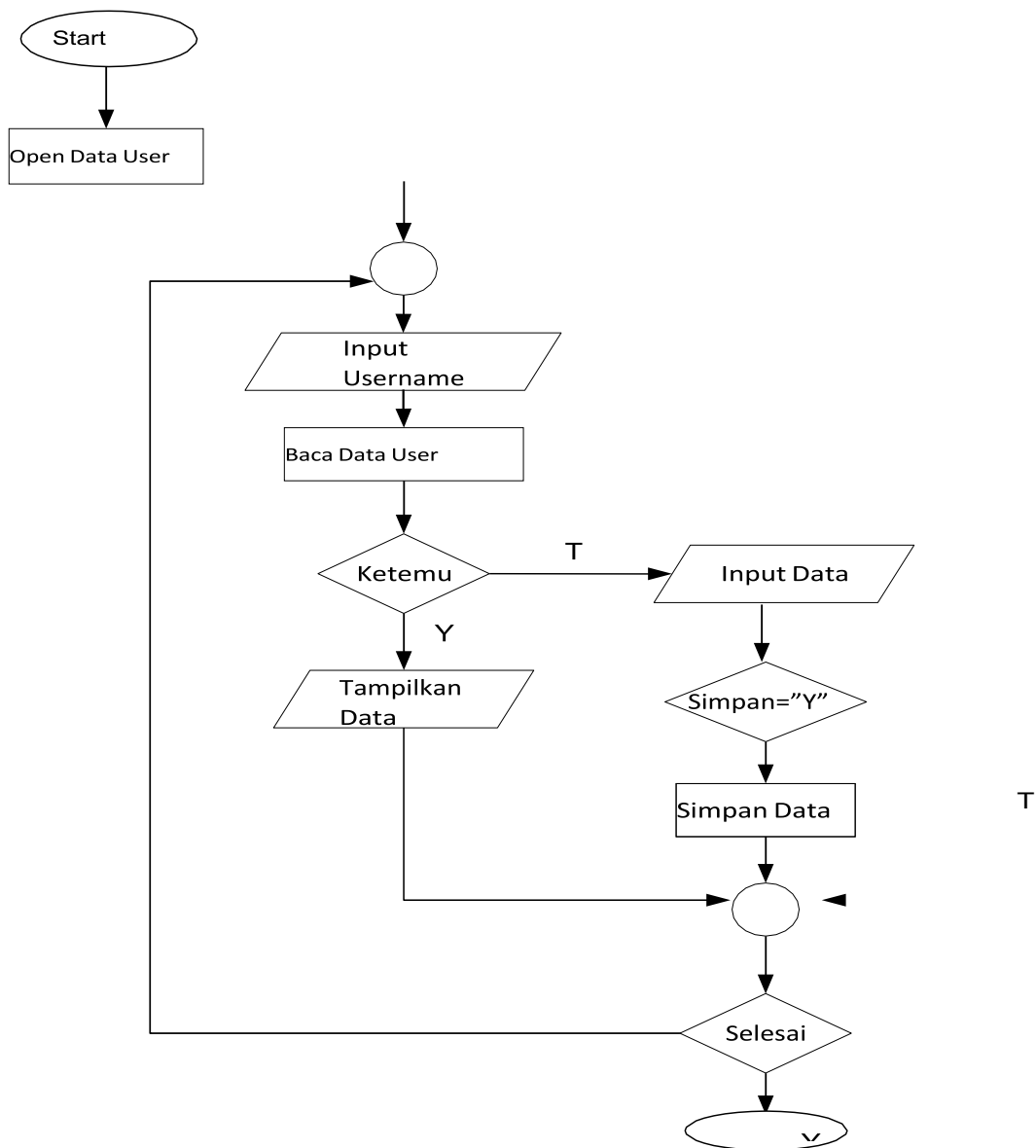
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan SPJ, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.30 Program Flowchart SPJ

12. Program Flowchart Tambah User

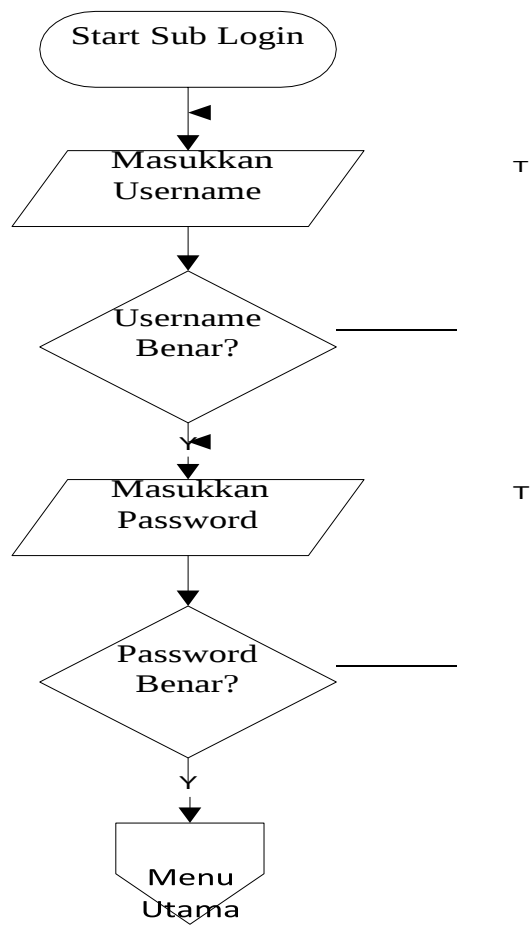
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan tambah user, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.31 Program Flowchart Tambah User

13. Program Flowchart Login

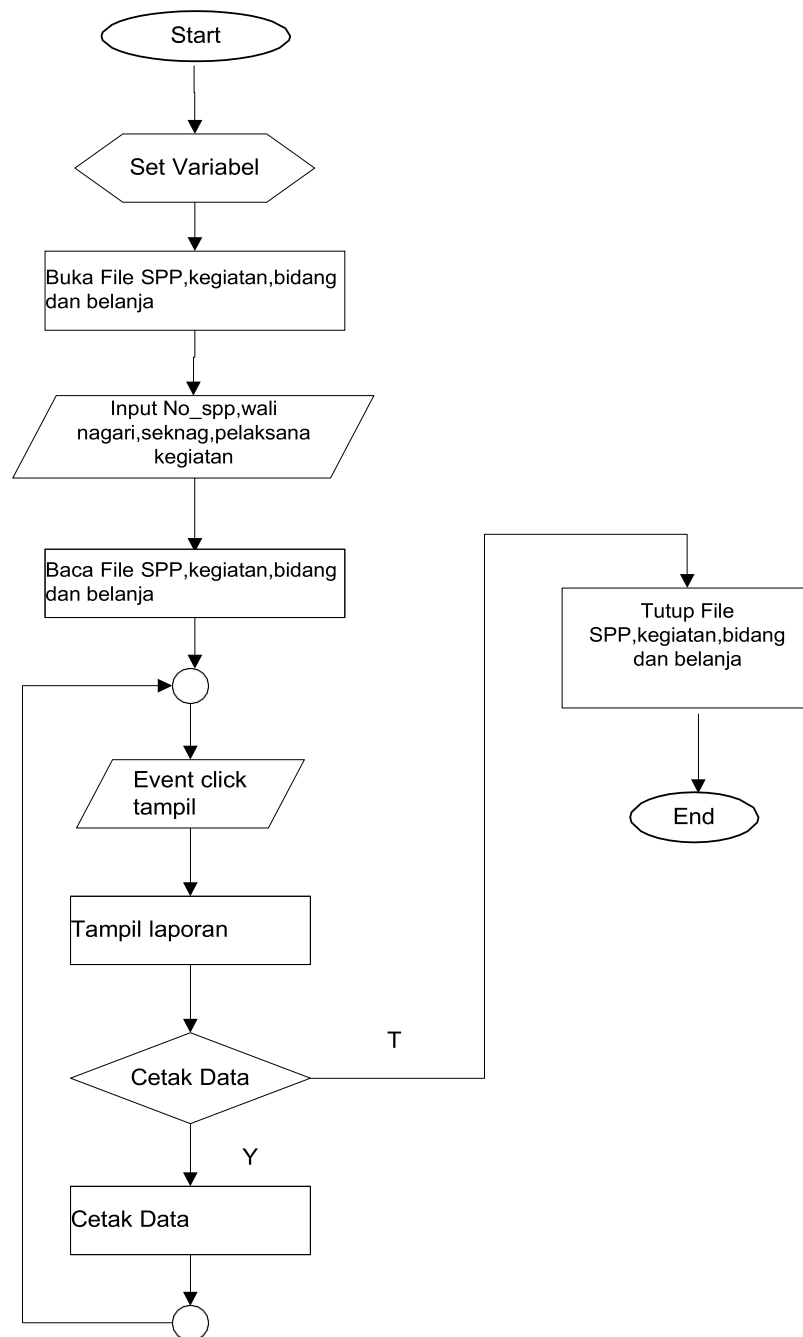
Pada program flowchart berikut ini digambarkan bagaimana logika dari login, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.32 Program Flowchart Login

14. Program Flowchart Laporan SPP

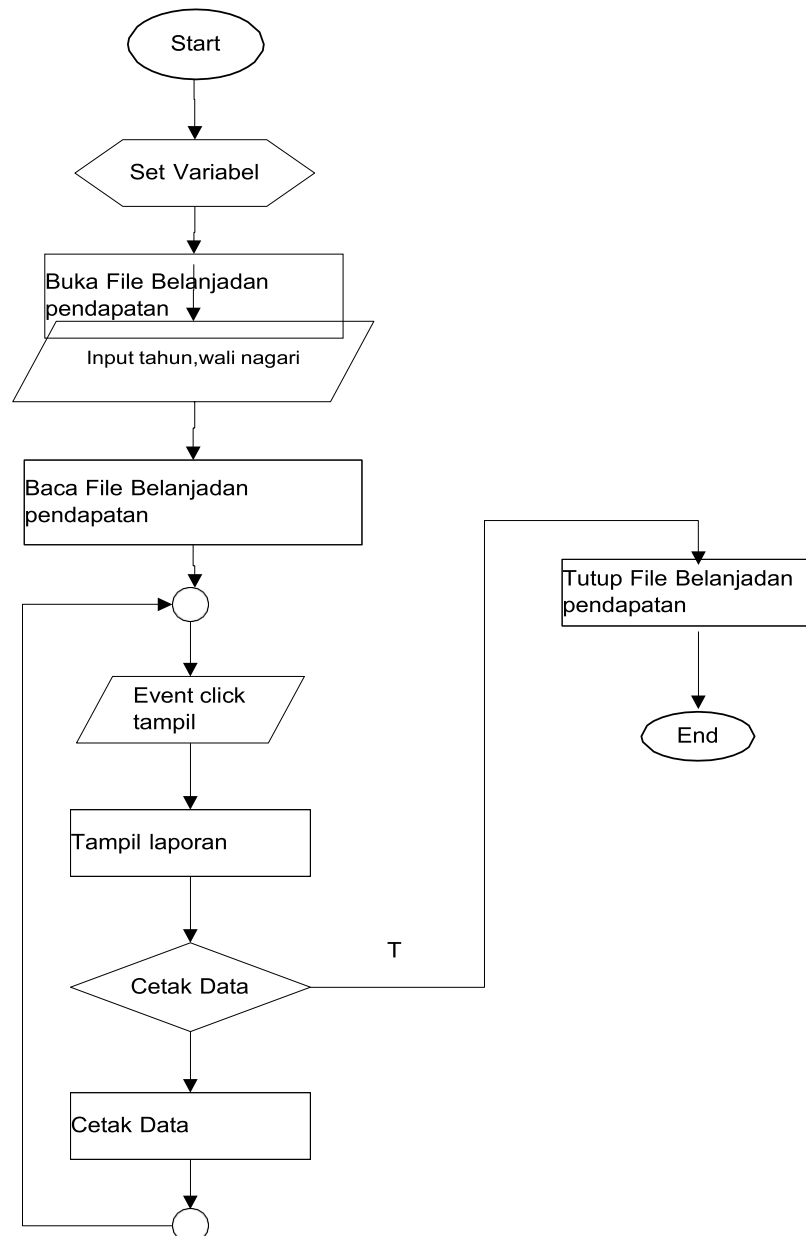
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan laporan SPP ini, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.33 Program Flowchart SPP

15. Program Flowchart Laporan Anggaran Pendapatan

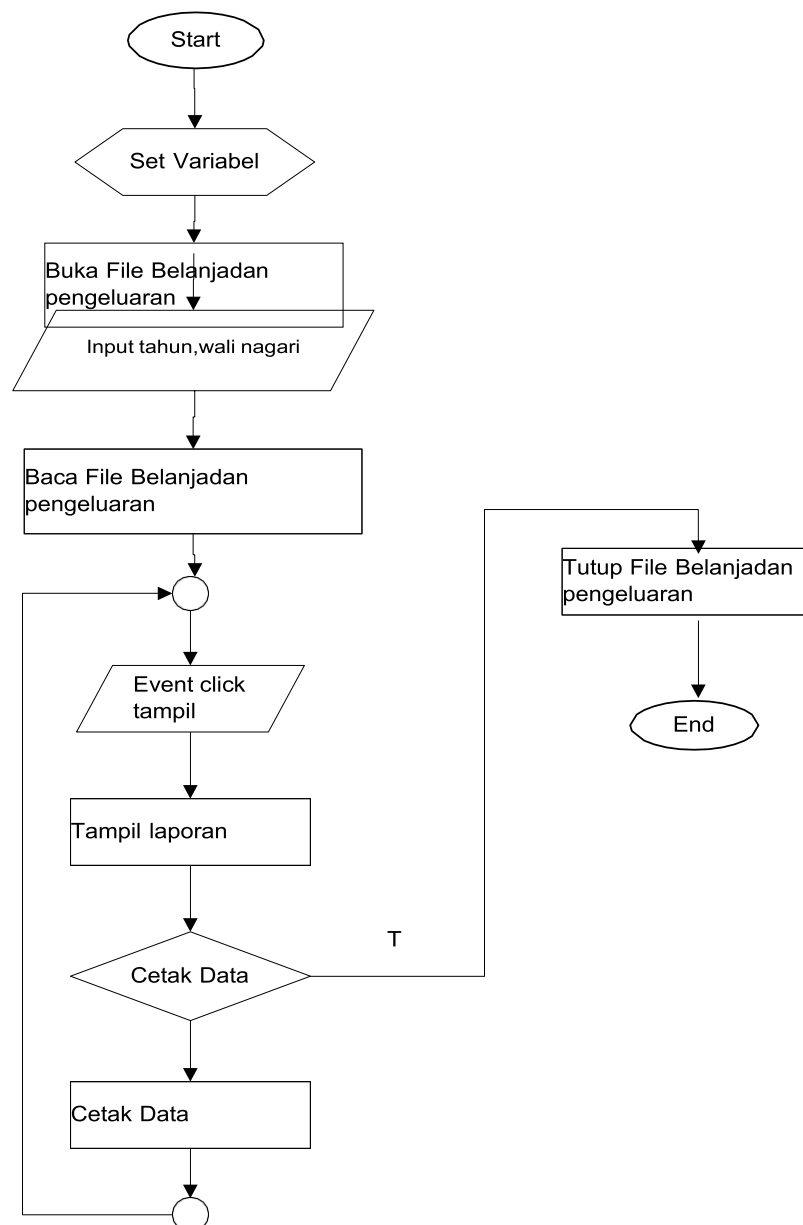
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan laporan Anggaran Pendapatan ini, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.34 Program Flowchart Laporan Anggaran Pendapatan

16. Program Flowchart Laporan Anggaran Belanja

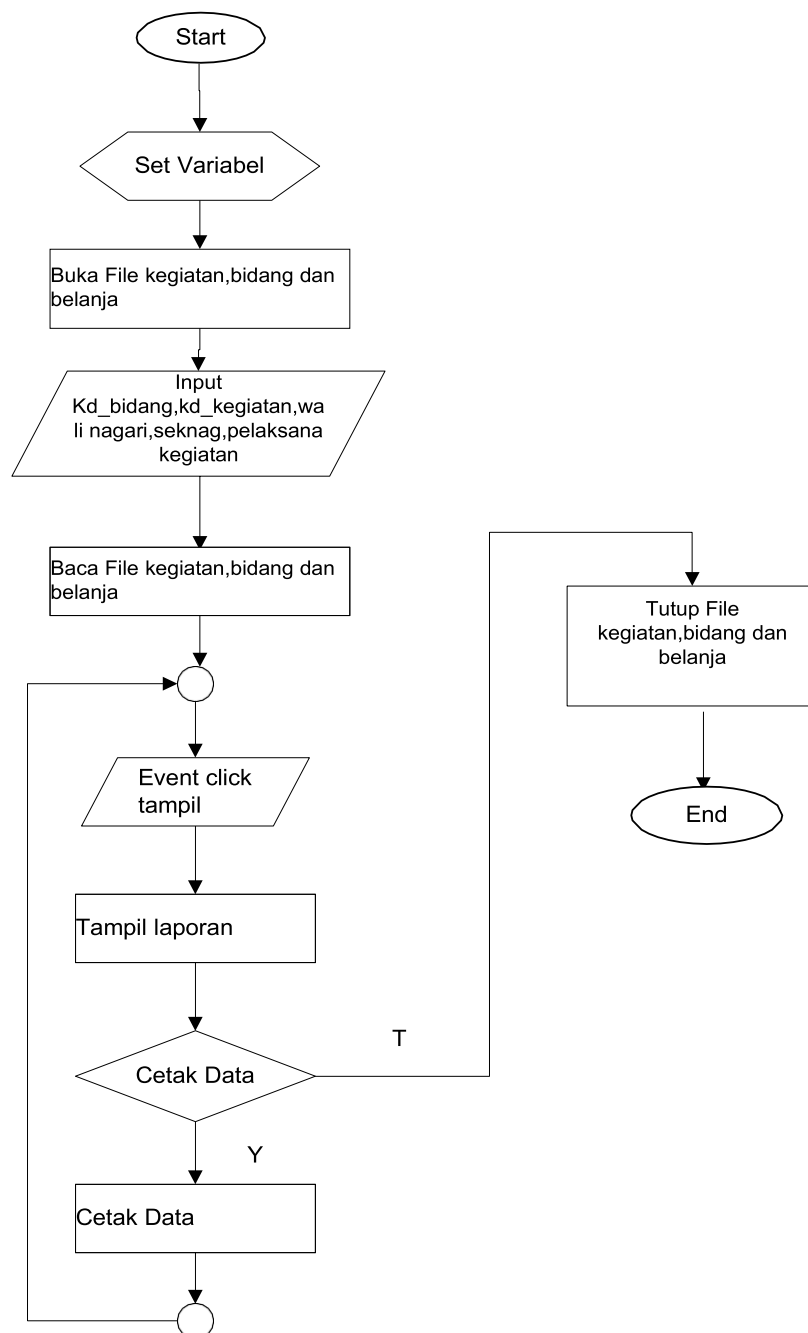
Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan laporan Anggaran Belanja ini, agar lebih jelas dapat dilihat padagambar :



Gambar 4.35 Program Flowchart Laporan Anggaran Belanja

17. Program Flowchart Laporan Realisasi

Pada program flowchart berikut ini digambarkan bagaimana logika dari pembuatan laporan realisasi ini, agar lebih jelas dapat dilihat pada gambar :



Gambar 4.36 Program Flowchart Laporan Realisasi

BAB V

IMPLEMENTASI SISTEM

5.1 Pengertian dan Tujuan Implementasi Sistem

Implementasi sistem adalah prosedur yang dilakukan pada tahap sistem dalam dokumen yang sedang disetujui dan menguji kemudian menginstall dan menggunakan program yang dibuat, penerapan sistem baru ini akan diterapkan secara menyeluruh.

Tujuan dari implementasi sistem adalah :

1. Menyelesaikan desain sistem yang ada dalam dokumen sistem yang disetujui.
2. Menulis, menguji dan mendokumentasikan program dan prosedur-prosedur yang diperlukan oleh dokumen desain sistem yang diuji.
3. Memastikan bahwa personal dapat mengoperasikan sistem yang telah dibuat.
4. Memastikan bahwa konversi ke sistem yang baru berjalan dengan benar.
5. Memperhitungkan bahwa sistem sesuai dengan kebutuhan pemakai.

5.2 Tahapan Implementasi Sistem

Tahapan dalam implementasi sistem ada tiga, yaitu :

1. Perencanaan dan proses implementasi

Perencanaan dan proses implementasi merupakan langkah penting dalam penerapan sebuah sistem baru. Tujuan dari pemberlakuan proses ini adalah

agar penerapan sistem baru ini betul-betul sesuai dengan tujuan dan sasaran pengadaannya.

2. Pelaksanaan proses implementasi

Pelaksanaan proses implementasi bertujuan untuk melakukan proses penerapan sistem baru. Bentuk kegiatan pelaksanaan proses implementasi sendiri, yaitu sebagai berikut :

- a. Pelatihan personil (*training*)
- b. *Install hardware dan software*
- c. Pemrograman
- d. Pengujian program
- e. Pengujian sistem
- f. Konversi sistem baru

3. Pelatihan Personil

Personil pengguna sistem merupakan orang yang memegang peran penting dalam pemanfaatan sistem baru. Pelatihan personil dilakukan dengan tujuan agar personil yang ditunjuk untuk menjalankan sistem baru tidak mengalami kesulitan dalam pengoperasiannya.

5.3 Pemrograman

Penulisan program dalam bahasa komputer, dalam hal ini menggunakan JAVA kemudian proses menjalankan program untuk melakukan pengujian program.

5.3.1 Form Menu Utama

Form menu utama adalah antar muka (*interface*) yang digunakan sebagai *form* induk / *form* utama. *Form* utama ini akan selalu ditampilkan saat program dijalankan. Untuk lebih jelasnya dapat dilihat pada Gambar 5.1:



Gambar 5.1 *Form* Menu Utama

Menu utama terdiri dari tiga menu yaitu:

1. Menu *User*

Pada menu *user* terdiri dari dua sub menu yaitu:

a. Sub Menu Login

Menampilkan *form* menu *login* dimana *user* harus login terlebih dahulu sebelum memasuki sistem.

b. Sub Menu Tambah User

Menampilkan *form* menu *entry* data *user*, dimana data-data tersebut akan dimasukkan pada *database* program.

Berikut gambar dari menu dan sub menu *entry* data yang dapat dilihat pada Gambar 5.2 :



Gambar 5.2 Sub Menu User

2. Menu *Entry* Kegiatan

Terdiri dari tiga sub menu *entry* kegiatan data yaitu:

a. Sub Menu *Entry* Data Bidang

Menampilkan *form* menu *entry* data bidang, dimana data-data tersebut akan dimasukkan pada *database* program.

b. Sub Menu *Entry* Data Rekening

Menampilkan *form* menu *entry* data belanja, dimana data-data tersebut akan dimasukkan pada *database* program.

Berikut gambar dari menu dan sub menu entry kegiatan pada Gambar 5.3 :



Gambar 5.3 Sub Menu Entry Kegiatan

3. Menu Perencanaan

Terdiri dari dua sub menu perencanaan yaitu:

a. Sub Menu Pendapatan

Menampilkan *form* menu pendapatan, dimana data-data tersebut akan dimasukkan pada *database* program.

b. Sub Menu Pengeluaran

Menampilkan *form* menu pengeluaran, dimana data-data tersebut akan dimasukkan pada *database* program.

Berikut gambar dari menu dan sub menu laporan pada Gambar 5.4 :



Gambar 5.4 Sub Menu Perencanaan

4. Menu Penatausahaan

Terdiri dari dua sub menu penatausahaan yaitu:

c. Sub Menu SPP

Menampilkan *form* menu SPP, dimana data-data tersebut akan dimasukkan pada *database* program.

d. Sub Menu SPJ

Menampilkan *form* menu SPJ, dimana data-data tersebut akan dimasukkan pada *database* program.

Berikut gambar dari menu dan sub menu laporan pada Gambar 5.5 :



Gambar 5.5 Sub Menu Penatausahaan

5. Menu Laporan

Terdiri dari empat sub menu laporan yaitu:

a. Sub Menu Laporan Anggaran Pendapatan

Menampilkan *form* laporan Anggaran Pendapatan yang ada pada keuangan nagari.

b. Sub Menu Laporan Anggaran Belanja

Menampilkan *form* laporan Anggaran Belanja yang ada pada keuangan nagari.

c. Sub Menu Laporan SPP

Menampilkan *form* laporan SPP yang ada pada keuangan nagari.

d. Sub Menu Laporan Realisasi

Menampilkan *form* laporan realisasi yang ada pada keuangan nagari. Berikut gambar dari menu dan sub menu laporan pada Gambar 5.6 :



Gambar 5.6 Sub Menu Laporan

3. Menu Exit

Menu ini terdiri dari sub menu keluar program. Sub menu ini menjalankan perintah pada *NetBeans 6.9* untuk keluar dari program keseluruhan.

5.3.2 Form Menu User

Form menu user digunakan untuk mengentry data *user* dan menyimpan data-data ke *database*. Data yang ada di *database* dapat ditampilkan kembali pada *form* yang ada. Selain itu juga terdapat form login untuk memasuki sistem aplikasinya.

5.3.2.1 Login

Ini merupakan awal untuk masuk ke sistem, dengan kata lain ini merupakan hak akses untuk masuk bagi *user*. Fungsi dari *form* login adalah untuk menjaga agar sistem tidak sembarangan orang bisa mengaksesnya demi keamanan data yang ada di dalamnya. *Form* login untuk sistem informasi pengelolaan laporan keuangan nagari Parit Malintang yang di dalamnya ada beberapa *user* yang menggunakannya dapat dilihat pada berikut ini :

Gambar 5.7 Form Login

5.3.2.2 Input Data User

Form ini merupakan *entri* data *user* atau pemberian hak akses untuk menjalankan aplikasi ini. Dalam hal ini yang diberikan hak akses adalah admin yang mengurus semua data keuangan. Dalam *form* ini dirancang beberapa tombol yaitu tombol *save* dan *cancel* serta *close*. Tombol *save* berfungsi untuk menyimpan data yang telah diinputkan sedangkan tombol *cancel* berfungsi untuk membatalkan data yang akan disimpan. *Form input data user* dapat dilihat pada gambar dibawahini :

Gambar 5.8 Form Input User

5.3.3 Form Entry Data

Form entry data digunakan untuk mengentry data dan menyimpan data- data ke *database*. Data yang ada di *database* dapat ditampilkan kembali pada *form* yang ada.

5.3.3.1 Form Input Data Bidang

Form ini merupakan *form* untuk mengentrikan data-data bidang pada Keuangan Nagari Kabupaten Padang Pariaman. Dalam *form* ini dirancang beberapa tombol yaitu simpan, tambah, hapus, edit dan keluar. Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambahkan data yang akan dientrikan.

Tombol edit berfungsi untuk mengedit data yang ada. Tombol hapus berfungsi untuk menghapus data yang ada. *Form input* data bidang dapat dilihat pada gambar dibawah ini:

No	KODE BIDANG	NAMA BIDANG
1	1	Pemerintahan
2	2	Pembangunan
3	3	Pembinaan
4	4	Pemberdayaan

Gambar 5.9 *Form Input* Data Bidang

5.3.3.2 *Form Input* Data Rekening

Form ini merupakan *form* untuk mengentrikan data-data rekening. Dalam *form* ini dirancang beberapa tombol yaitu simpan, tambah, hapus, edit dan keluar. Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambah data yang akan dientrikan. Tombol hapus berfungsi untuk menghapus data yang ada. Tombol edit berfungsi untuk mengedit data yang ada. *Form input* data belanja dapat dilihat pada gambar dibawah ini :

No	KODE REKENING	NAMA REKENING
1	1.1	Pendapatan Asli Nagari
2	1.2	Alokasi Dana Desa
3	1.3	Alokasi Dana Nagari
4	1.4	Bantuan Propinsi
5	2.1	Bantuan Daerah Lain

Gambar 5.10 *Form Input* Data Rekening

5.3.4 Form Perencanaan

Form perencanaan data digunakan untuk mengentry data dan menyimpan data-data ke database. Data yang ada di database dapat ditampilkan kembali pada form yang ada.

5.3.4.1 Form Pendapatan

Form ini merupakan form untuk mengentrikan data-data pendapatan. Dalam form ini dirancang beberapa tombol yaitu simpan, tambah, hapus, edit dan keluar. Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambahkan data yang akan dientrikan. Tombol edit berfungsi untuk mengedit data yang ada. Tombol hapus berfungsi untuk menghapus data yang ada. Form input data biaya dapat dilihat pada gambar dibawah ini :

No	NO pendapatan	NAMA PENDAP...	KODE BELANJA	PENDAPATAN
1	1	Pendapatan A...	1.1	1000000
2	2	Dana Transfer	1.2	800000000
3	3	Dana Transfet	1.3	500000000
4	4	Dana Transfer	1.4	12000000

Gambar 5.12 Form Pendapatan

5.3.4.2 Form Pengeluaran

Form ini merupakan form untuk mengentrikan data-data pengeluaran/ belanja Dalam form ini dirancang beberapa tombol yaitu simpan, tambah, hapus,

edit dan keluar. Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambahkan data yang akan dientrikan. Tombol hapus berfungsi untuk menghapus data yang ada. Tombol edit berfungsi untuk mengedit data yang ada. *Form input* data biaya dapat dilihat pada gambar dibawahini :

No	NO pengeluaran	KODE BIDANG	KODE BELANJA	JUMLAH
1	1	1	2.1	500000000
2	2	1	2.2	2000000
3	3	1	2.3	5000000
4	4	2	2.4	2000000

Gambar 5.13 Form Pengeluaran

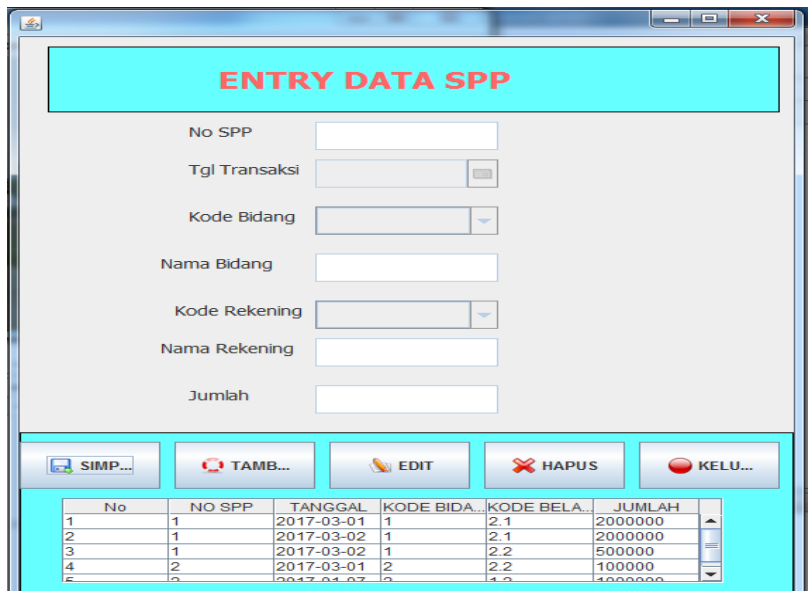
5.3.5 Form Penatausahaan

Form penatausahaan data digunakan untuk mengentry data dan menyimpan data-data ke *database*. Data yang ada di *database* dapat ditampilkan kembali pada *form* yang ada.

5.3.5.1 Form SPP

Form ini merupakan *form* untuk mengentrikan data-data SPP pada Keuangan Nagari Parit Malintang Kab. Padang Pariaman. Dalam *form* ini dirancang beberapa tombol yaitu simpan, tambah, hapus, edit dan keluar.

Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambahkan data yang akan dientrikan. Tombol edit berfungsi untuk mengedit data yang ada. Tombol hapus berfungsi untuk menghapus data yang ada. *Form input* data biaya dapat dilihat pada gambar dibawah ini :



No	NO SPP	TANGGAL	KODE BIDA.	KODE BELA.	JUMLAH
1	1	2017-03-01	1	2.1	2000000
2	1	2017-03-02	1	2.1	2000000
3	1	2017-03-02	1	2.2	500000
4	2	2017-03-01	2	2.2	100000
5	2	2017-03-02	2	2.2	1000000

Gambar 5.14 *Form SPP*

5.3.5.2 *Form SPJ*

Form ini merupakan *form* untuk mengentrikan data-data SPJ pada Keuangan Nagari Parit Malintang Kabupaten Padang Pariaman. Dalam *form* ini dirancang beberapa tombol yaitu simpan, batal, hapus, edit dan keluar. Tombol simpan berfungsi untuk menyimpan data yang dientrikan. Tombol tambah berfungsi untuk menambahkan data yang akan dientrikan. Tombol hapus berfungsi untuk menghapus data yang ada. Tombol edit berfungsi untuk mengedit data yang ada. *Form input* data biaya dapat dilihat pada gambar dibawah ini :

No	NO SPP	TANGGAL	KODE BIDA...	KODE BELA...	JUMLAH
1	1	2017-03-01	1	2.1	2000000
2	1	2017-03-02	1	2.1	2000000
3	1	2017-03-02	1	2.2	500000
4	2	2017-03-01	2	2.2	100000
5	2	2017-03-02	2	2.2	100000

Gambar 5.15 Form SPJ

5.3.6 Menu Laporan

Form laporan digunakan untuk mencetak laporan dengan mengentrykan data terlebih dahulu pada *form input* data, kemudian data ditampilkan dari *form* laporan dan dicetak ke kertas.

5.3.6.1 Laporan Anggaran Pendapatan

Laporan ini mencatat Anggaran Pendapatan Nagari Parit Malintang Kab. Padang Pariaman. Bentuk laporan pada sistem ini seperti pada gambar dibawahini :

ANGGARAN PENDAPATAN NAGARI
Pemerintah Kota BEKASI

Tahun: 2017

Kode Rekening	Uraian Pendapatan	Anggaran
1.1	Pendapatan Asli Nagari	1.006.000.00
1.2	Pajak Bumi dan Bangunan	800.000.000.00
1.3	Pajak Daerah	500.000.000.00
1.4	Bantuan Operasional	12.000.000.00
Jumlah :		1.313.000.000.00

Parit Malintang: 05 Maret 2017
Dikur Waki Nagari

xii

Gambar 5.16 Laporan Anggaran Pendapatan

5.3.6.2 Laporan Anggaran Belanja

Laporan ini mencatat Anggaran Belanja Nagari Parit Malintang Kab.Padang Pariaman.

Bentuk laporan pada sistem ini seperti pada gambar dibawah ini :

ANGGARAN BELANJA NAGARI
Pemerintah Kota Bekasi

Tahun: 2017

Kode Bidang : 1 Nama Bidang : Pemerintahan

Kode Belanja	Uraian Belanja	Anggaran
2.1	Belanja Penghasilan Tetap	500.000.000.00
2.2	Belanja Alat Tulis Kantor	2.000.000.00
2.3	Belanja Perjalanan Dinas Dalam Daerah	5.000.000.00
Jumlah :		507.000.000.00

Parit Malintang: 05 Maret 2017
Dikur Waki Nagari

xii

Gambar 5.17 Laporan Anggaran Belanja

5.3.6.3 Laporan SPP

Laporan ini mencatat SPP di lingkungan Kab. Padang Pariaman. Bentuk laporan pada sistem ini seperti pada gambar dibawah ini :



SURAT PERINTAH PEMBAYARAN
Pemerintah Kota BEKASI

Nomor : 1

Nama Bidang : Pemerintahan

Kode Belanja	Nama Belanja	Jumlah
2.1	Belanja Penghasilan Tetap	2.000.000,00
2.1	Belanja Penghasilan Tetap	2.000.000,00
2.2	Belanja Aset Tetap Kantor	800.000,00
Jumlah		4.800.000,00

Part Matndang: 03-Maret-2017

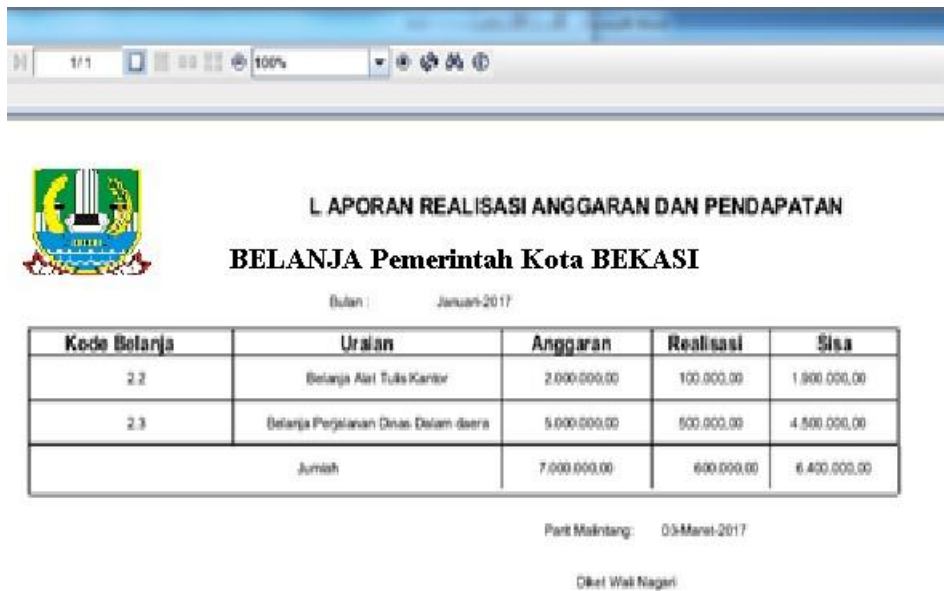
Diket Wakil Nagari
Team Verifikasi Sakrag
Pelaksana Kegiatan

all
Budi
wali

Gambar 5.18 Laporan SPP

5.3.6.4 Laporan Realisasi

Laporan ini mencatat realisasi keuangan, bentuk laporan pada sistem ini seperti pada gambar dibawah ini :



L APORAN REALISASI ANGGARAN DAN PENDAPATAN
BELANJA Pemerintah Kota BEKASI

Bulan : Januari 2017

Kode Belanja	Uraian	Anggaran	Realisasi	Sisa
2.2	Belanja Alat Tulis Kantor	2.000.000,00	100.000,00	1.900.000,00
2.3	Belanja Perjalanan Dinas Dalam daerah	5.000.000,00	500.000,00	4.500.000,00
Jumlah		7.000.000,00	600.000,00	6.400.000,00

Pada Tanggal: 03-Maret-2017

Diket: Wali Negeri

Gambar 5.19 Laporan Realisasi

BAB VI

PENUTUP

6.1 Kesimpulan

Adapun kesimpulan yang dapat penulis ambil dari tugas akhir antara lain :

1. Mendapatkan aplikasi Sistem Informasi dengan menggunakan bahasa pemrograman Bahasa Java untuk pengolahan laporan keuangan dan menggunakan *database MYSQL* sebagai tempat penyimpanan datanya, maka laporan pengolahan laporan keuangan dapat dilakukan secara cepat dan akurat.
2. Memberi kemudahan pada PEMKOT BEKASI dalam pengelolaan laporan keuangan dengan cepat, tepat dan akurat
3. Dengan penerapan sistem yang baru maka akan dapat membantu pegawai dalam bekerja di era globalisasi ini yang serba modern.

6.2 Saran

Penulis akan menyampaikan beberapa saran agar Sistem Informasi ini dapat dimanfaatkan sebaik mungkin.

1. Dalam penerapan sistem baru ini, perlu dilakukan pelatihan terhadap staf nagari untuk menghindari terjadinya masalah dan kesalahan-kesalahan dalam pengoperasian sistem.
2. Ketelitian dan kecermatan dibidang komputer harus diperhatikan dengan sungguh-sungguh dan perlu tenaga ahli yang terampil baik dalam pengoperasian maupun pengontrolan hardware.
3. Sistem baru ini merupakan desain sistem yang mampu menghasilkan laporan dan informasi yang akurat, namun demikian perlu ketelitian dan kecermatan

dalam penerapannya karena jika terjadi satu kesalahan akan mempengaruhi data yang lain.

4. Diharapkan sistem yang baru ini dapat diterapkan dan berguna dalam kelancaran proses kinerja pada pegawai.
5. Agar kelangsungan sistem lebih baik dimasa yang akan datang, perlunya dilakukan pendekatan, pemeliharaan (*maintenance*) dan pengembangan terhadap sistem.

DAFTAR PUSTAKA

Abdul Kadir, *Pengenalan Sistem Informasi*, Yogyakarta: Andi Offset, 2013.

Ernita, T, dkk., 2013, *Buku Panduan Penulisan Proposal dan Skripsi STTIND Padang*.

Hanif Al-Fatta, *Analisis & perancangan system informasi*, Yogyakarta: Andi Offset, 2007.

<http://octavianti92.blogspot.com/2013/01/pengertian-sistem-informasi-dan-sistem.html>,

Ifnu Bima, *Java Desktop Aplikasi POS Berarsitektur Three Tier Menggunakan Swing, Hibernate, Dan Spring*, Bogor: 2011.

Janner Simarmata, Iman Paryudi. *Basis Data*, Yogyakarta: Andi Offset, 2010.

Jogiyanto HM, *Analisis dan Design Sistem Informasi*, Yogyakarta: Andi Offset, 2009.

Jogiyanto HM, *Pengenalan Komputer*, Yogyakarta: Andi Offset, 2005.

Jurnal *Al-Azhar Indonesia Seri Sains dan Teknologi*, vol . 2, no. 2, september 2013.

Jurnal Angga Adhe Saputra, *Sistem Informasi Berbasis Client Server*, Semarang, 2014.

Nue Efendi Husda, Yvonne Wangdra, *Pengantar Teknologi Informasi*, Jakarta: *Baduose Media*, 2016

Patrick Naughton, *Konsep Dasar Pemrograman Java*, terj. Panji Gotama, Yogyakarta: Andi, 2002.

Rosa AS, M. Shalahuddin, *Modula Pembelajaran Rekayasa Perangkat Lunak*, Modula, Yogyakarta: 2011.

Wahana Komputer, *Seri Panduan Aplikasi Membuat Aplikasi Database Dengan Java 2*, Yogyakarta: Andi Offset, 2006.

Listing program

(untuk proses tombol save)

```
try {  
  
    sql="insert into users values('"+user.getText()+"',"  
    +jComboBox1.getSelectedItem()+  
    "','"+pass.getText()+"')";  
  
    st=con.createStatement();st.execute(sql);  
  
    hapus();  
  
    JOptionPane.showMessageDialog(rootPane, "data tersimpan");  
    } catch (Exception e) {  
  
        JOptionPane.showMessageDialog(rootPane, "gagal tersimpan");  
    }
```

```
}
```

(untuk proses tombol cancel)

```
user.setText("");
```

```
pass.setText(""); jComboBox1.setSelectedItem(null);
```

```
pass.requestFocus();
```

(untuk proses tombol close)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data operator yang akan login ke sistem

```
try {
```

```
    sql="select * from users where username='"+user.getText()+"'";
```

```
    st=con.createStatement();
```

```
    rs=st.executeQuery(sql);while(rs.next()){
```

```
        if (user.getText().equals(rs.getString("username")) &&
```

```
pass.getText().equals(rs.getString("pass"))&&
```

```
jComboBox1.getSelectedItem().equals(rs.getString("level")))
```

```
{
```

```
    new menu().show();atur();
```

```
} else
```

```
JOptionPane.showMessageDialog(null, "Maaf masukkan usernamedan password anda  
dengan benar...");
```

```
pass.setText(""); user.requestFocus();
```

```
}
```

```
} catch (Exception e) {
```

```
}
```

(untuk proses tombol cancel)

```
user.setText("");
```

```
pass.setText(""); jComboBox1.setSelectedItem(null);
```

```
pass.requestFocus();
```

(untuk proses tombol close)

```
this.dispose();
```

Form di atas digunakan untuk proses login ke sistem sesuai level.

(untuk proses tombol simpan)

```
try {  
    sql = "insert into bidang values('" + a.getText() + "','"  
    + b.getText() + "')"; st =  
    con.createStatement(); st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin  
    menyimpan data ini??", "simpan data di sini!", JOptionPane.YES_NO_OPTION); if (ok  
    == 0) {  
        hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap  
    hubungi programmer kami!" + e.getMessage());  
}
```

(untuk proses tombol batal)

```
a.setText("");  
b.setText("");  
a.requestFocus();
```

(untuk proses tombol hapus)

```
try {  
  
    sql = "delete from bidang where kd_bidang='" + a.getText() + "'";  
    st = con.createStatement();  
    st.execute(sql);  
  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",  
    "hapus di sini ya!!", JOptionPane.YES_NO_OPTION);  
  
        if (ok == 0) {hapus();  
            tampil();  
        }  
  
    } catch (Exception e) {  
  
    }  
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data?? "+  
    "\nDengan kd_bidang'" + a.getText() + "'",  
    "Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){  
  
    try {  
  
        koneksi();  
  
        sql = "UPDATE bidang set nm_bidang='" + b.getText() +  
  
            "!!'" + sql);
```

```

"        bidang="'+a.getText()+''";st =
,        con.createStatement();
w        st.execute(sql);
h        JOptionPane.showMessageDialog(null, "Data berhasil diedit
e
r        hapus();
e        tampil();
        } catch (Exception e) {
k        JOptionPane.showMessageDialog(null, "Data gagal di update,
d
—
Kesalahan pada"+e);
}
}

(untuk proses tombol keluar)

this.dispose();

```

Form di atas digunakan untuk menginputkan data bidang.

```

try {

    sql = "insert into belanja values('" + a.getText() + "','"
        + b.getText() + "','"
        + b1.getText() + "')"; st =
        con.createStatement();st.execute(sql);

    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??",
        "simpan data di sini!", JOptionPane.YES_NO_OPTION);

        if (ok == 0) {hapus();

            tampil();

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap
        hubungi programmer kami!" + e.getMessage());

    }

```

(untuk proses tombol tambah)

```

a.setText("");

b.setText("");

b1.setText(""); a.requestFocus();

```

(untuk proses tombol hapus)

```
try {  
  
    sql = "delete from belanja where kd_belanja=" + a.getText() + """;st =  
    con.createStatement();  
  
    st.execute(sql);  
  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",  
    "hapus di sini ya!!", JOptionPane.YES_NO_OPTION);  
  
        if (ok == 0) {hapus();  
            tampil();  
        }  
  
    } catch (Exception e) {  
  
    }  
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data??  
"+  
    "\nDengan kd_belanja"+a.getText()+"",  
    "Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){  
    try {  
        koneksi();  
  
        sql = "UPDATE belanja set jenis_belanja="+b.getText()+  
        ",nm_belanja="+b1.getText()+  
        "where kd_belanja="+a.getText()+"";st = con.createStatement();  
  
        st.execute(sql);  
  
        JOptionPane.showMessageDialog(null, "Data berhasil diedit  
!!!"+sql);  
  
        hapus();
```

```
tampil();      (Exception e) {  
} catch      JOptionPane.showMessageDialog(null, "Data gagal di update,  
Kesalahan pada"+e);  
}  
}
```

(untuk proses tombol keluar)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data belanja.

```

try {

    sql = "insert into pendapatan values('" + a.getText() + "','"
        + b1.getText() + "','"
        + d.getSelectedItem() + "','"
        + c1.getText() + "','"
        + e.getText() + "')"; st =
        con.createStatement(); st.execute(sql);

    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??",
        "simpan data di sini!", JOptionPane.YES_NO_OPTION);

        if (ok == 0) {hapus();

            tampil();

        }

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap
        hubungi programmer kami!" + e.getMessage());

    }

```

(untuk proses tombol tambah)

```

a.setText("");
b1.setText("");
c1.setText(""); d.setSelectedItem(null);
e.setText(""); a.requestFocus();

```

(untuk proses tombol hapus)

```
try {  
  
    sql = "delete from pendapatan where kd_pendapatan=" + a.getText() + "";  
  
    st = con.createStatement(); st.execute(sql);  
  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",  
    "hapus di sini ya!!", JOptionPane.YES_NO_OPTION);  
  
        if (ok == 0) {hapus();  
  
            tampil();  
  
        }  
  
    } catch (Exception e) {  
  
    }  
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data??  
  
"+  
  
"\nDengan kd_pendapatan="+a.getText()+"", "Konfirmasi edit  
data",JOptionPane.YES_NO_OPTION);  
  
if (ok==0){try {  
  
    koneksi();  
  
    sql = "UPDATE pendapatan set nm_pendapatan="+  
    b1.getText()+  
  
    "!!!" +sql);
```

```
"      =""+d.getSelectedItemAt()+
,      ""',kd_belanja=""+c1.getText()+
,      ""',anggaran=""+e.getText()+
k      ""where kd_pendapatan=""+a.getText()+""";st =
d      con.createStatement();
-      st.execute(sql);
b      JOptionPane.showMessageDialog(null, "Data berhasil diedit
e
l      hapus();
a      tampil();
n      } catch (Exception e) {
j      JOptionPane.showMessageDialog(null, "Data gagal di update,
a
Kesalahan pada"+e);
}
}
```

(untuk proses tombol keluar)

this.dispose();

Form di atas digunakan untuk menginputkan data pendapatan.

Tampilan Form Input Data Pengeluaran(untuk proses tombol simpan)

try {

sql = "insert into pengeluaran values('" + a.getText() + ", "

+ b.getSelectedItemAt() + ", "

+ c.getSelectedItemAt() + ", "

+ d.getSelectedItemAt() + ", "

+ e.getText() + "')"; st =

con.createStatement();st.execute(sql);

int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpendata ini??",

"simpan data di sini!", JOptionPane.YES_NO_OPTION);

if (ok == 0) {hapus();

tampil();

}

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap

hubungi programmer kami!" + e.getMessage());

}

(untuk proses tombol simpan)

```
a.setText(""); b.setSelectedItem(null);  
c.setSelectedItem(null);  
d.setSelectedItem(null);e.setText("");  
a.requestFocus();
```

(untuk proses tombol hapus)

```
try {  
    sql = "delete from pengeluaran where kd_pengeluaran=" + a.getText() + "";  
    st = con.createStatement();st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",  
    "hapus di sini ya!!", JOptionPane.YES_NO_OPTION);  
        if (ok == 0) {hapus();  
            tampil();  
        }  
    } catch (Exception e) {  
    }  
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data??  
"+  
    "\nDengan kd_pendapatan="+a.getText()+"", "Konfirmasi edit  
data",JOptionPane.YES_NO_OPTION);  
if (ok==0){try {  
    koneksi();  
    sql = "UPDATE pendapatan set nm_pendapatan="+b1.getText()+  
    ",kd_belanja="+d.getSelectedItem()+ ",kd_belanja="+c1.getText()+
```

```
"" ,anggaran="" + e.getText() +  
"" where kd_pendapatan="" + a.getText() + """; st =  
con.createStatement();  
st.execute(sql);  
JOptionPane.showMessageDialog(null, "Data berhasil diedit  
!!!"+sql);  
  
        hapus();  
  
        tampil();  
  
    } catch (Exception e) {  
  
        JOptionPane.showMessageDialog(null, "Data gagal di update,  
Kesalahan pada"+e);  
  
    }  
}
```

(untuk proses tombol keluar)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data pengeluaran.

Form Input Data SPP(untuk proses tombol simpan)

```
try {  
    sql = "insert into spp values('" + a.getText() + "','"  
    + tgl + "','"  
    + b.getSelectedItem() + "','"  
    + c.getSelectedItem() + "','"  
    + d.getSelectedItem() + "','"  
    + e.getText() + "')"; st =  
    con.createStatement();st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??",  
    "simpan data di sini!", JOptionPane.YES_NO_OPTION);  
    if (ok == 0) {hapus();  
        tampil();  
    }  
} catch (Exception e) {  
    JOptionPane.showMessageDialog(null, "data tidak berhasil disimpan,harap  
    hubungi programmer kami!" + e.getMessage());  
}
```

(untuk proses tombol simpan)

```
a.setText(""); b.setSelectedItem(null);  
c.setSelectedItem(null);  
d.setSelectedItem(null);e.setText("");  
a.requestFocus();
```

(untuk proses tombol hapus)

```
try {  
    sql = "delete from spp where no_spp=" + a.getText() + """;st =  
    con.createStatement();  
    st.execute(sql);  
    int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",  
    "hapus di sini ya!!", JOptionPane.YES_NO_OPTION);  
        if (ok == 0) {hapus();  
                                tampil();  
        }  
    } catch (Exception e) {  
    }  
}
```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit  
data?? "+  
    "\nDengan no_spp="+a.getText()+"",  
    "Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){  
    try {
```

```

koneksi();

sql = "UPDATE spp set tgl='"+tgl+ "',kd_bidang='"+b.getSelectedItemAt()+
"',kd_kegiatan='"+c.getSelectedItemAt()+
"',kd_belanja='"+d.getSelectedItemAt()+ "',jumlah='"+e.getText()+
"'where no_spp='"+a.getText()+"''";st =
con.createStatement(); st.execute(sql);

JOptionPane.showMessageDialog(null, "Data berhasil diedit

!!!"+sql);

        hapus();

        tampil();

    } catch (Exception e) {

        JOptionPane.showMessageDialog(null, "Data gagal di update,

Kesalahan pada"+e);

    }

```

```
}
```

(untuk proses tombol keluar)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data spp.

Form Input Data SPJ(untuk proses tombol simpan)

```
try {
```

```
sql = "insert into spj values('" + a.getText() + "','"
```

```
+ tgl+ "','"
```

```
+ b.getSelectedItem()+ "','"
```

```
+ c.getSelectedItem()+ "','"
```

```
+ d.getSelectedItem()+ "','"
```

```
+ e.getText() + "')"; st =
```

```
con.createStatement();st.execute(sql);
```

```
int ok = JOptionPane.showConfirmDialog(null, "anda yakin menyimpandata ini??",
```

```
"simpan data di sini!", JOptionPane.YES_NO_OPTION);
```

```
if (ok == 0) {hapus();
```

```

    tampil();

}

} catch (Exception e) {

OptionPane.showMessageDialog(null, "data tidak berhasil disimpan, harap
hubungi programmer kami!" + e.getMessage());

}

```

(untuk proses tombol simpan)

```

a.setText(""); b.setSelectedItem(null);

c.setSelectedItem(null);

d.setSelectedItem(null); e.setText("");

a.requestFocus();

```

(untuk proses tombol hapus)

```

try {

sql = "delete from spp where no_spj=" + a.getText() + """; st =
con.createStatement();

st.execute(sql);

int ok = JOptionPane.showConfirmDialog(null, "anda yakin menghapus data ini???",
"hapus di sini ya!!", JOptionPane.YES_NO_OPTION);

    if (ok == 0) {hapus();

                                tampil();

}

} catch (Exception e) {

}

```

(untuk proses tombol edit)

```
int ok = JOptionPane.showConfirmDialog(null, "Anda yakin inginmengedit data??  
"+  
"\nDengan no_spj='"+a.getText()+"'",  
"Konfirmasi edit data",JOptionPane.YES_NO_OPTION);if (ok==0){  
try {  
koneksi();  
sql = "UPDATE spj set tgl='"+tgl+ "',kd_bidang='"+b.getSelectedItem()+  
"',kd_kegiatan='"+c.getSelectedItem()+  
"',kd_belanja='"+d.getSelectedItem()+ "',jumlah='"+e.getText()+  
"'where no_spj='"+a.getText()+"'";st = con.createStatement();  
st.execute(sql);
```

```

JOptionPane.showMessageDialog(null, "Data berhasil diedit
!!!"+sql);

hapus();

tampil();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Data gagal di update,
Kesalahan pada"+e);

}

}

```

(untuk proses tombol keluar)

```
this.dispose();
```

Form di atas digunakan untuk menginputkan data SPJ.

Laporan Anggaran Pendapatan dan Belanja(untuk proses tombol print)

```

String periode= a.getText().toString(); String bk =
b.getText().toString();try {

JDialog window = new JDialog(this, "Print Preview");

window.setSize(Toolkit.getDefaultToolkit().getScreenSize()); SwingReportViewer viewer
= new SwingReportViewer(); URLRenderData renderConnection = new
URLRenderData(

"http://localhost:9000/?report=file:" +

```

```

System.getProperty("user.dir")+"/src/Laporan/apb.rpt&prompt0=" + periode +
"&prompt1=" + bk + "");

viewer.addNewReportView( renderConnection ); window.getContentPane().add(
BorderLayout.CENTER, viewer);window.setVisible( true );

} catch (Exception e) {

e.printStackTrace();

}

```

Form di atas digunakan untuk menampilkan laporan anggaran pendapatan dan belanja.

Laporan Anggaran Belanja(untuk proses tombol print)

```

String periode= a.getText().toString(); String bk =
b.getText().toString();

String bi = b1.getSelectedItem().toString();try {

JDialog window = new JDialog(this, "Print Preview");

window.setSize(Toolkit.getDefaultToolkit().getScreenSize()); SwingReportViewer viewer =
new SwingReportViewer(); URLRenderData renderConnection = new URLRenderData(
"http://localhost:9000/?report=file:" +

System.getProperty("user.dir")+"/src/Laporan/belanja.rpt&prompt0="

+ periode + "&prompt1=" + bk + "&prompt2=" + bi + "");

viewer.addNewReportView( renderConnection ); window.getContentPane().add(
BorderLayout.CENTER, viewer);window.setVisible( true );

```

```

} catch (Exception e) {

e.printStackTrace();

}

```

Form di atas digunakan untuk menampilkan laporan anggaran belanja.

Desain Laporan Surat Perintah Pembayaran

(untuk proses tombol print)

```

String periode= a.getText().toString(); String
bk = b.getText().toString(); String seknag=
c.getText().toString();String pk =
d.getText().toString(); try {
JDialog window = new JDialog(this, "Print Preview");
window.setSize(Toolkit.getDefaultToolkit().getScreenSize());
SwingReportViewer viewer = new SwingReportViewer(); URLRenderData
renderConnection = new URLRenderData(
"http://localhost:9000/?report=file:" +
System.getProperty("user.dir")+"/src/Laporan/spp.rpt&prompt0=" + periode +
"&prompt1=" + bk + "&prompt2=" + seknag + "&prompt3=" + pk + "");
viewer.addNewReportView( renderConnection );
window.getContentPane().add( BorderLayout.CENTER, viewer);
window.setVisible( true );

```

```
} catch (Exception e) {  
e.printStackTrace();  
}
```

Form di atas digunakan untuk mencetak laporan surat perintah pembayaran.

Desain Laporan Realisasi Anggaran dan Pendapatan

(untuk proses tombol print)

```
String periode= a.getText().toString();
```

```
String bk = b.getText().toString();
```

```
try {
```

```
    JDialog window = new JDialog(this, "Print Preview");
```

```
    window.setSize(Toolkit.getDefaultToolkit().getScreenSize());
```

```
    SwingReportViewer viewer = new SwingReportViewer(); URLRenderData
```

```
    renderConnection = new URLRenderData(
```

```
    "http://localhost:9000/?report=file:" +
```

```
    System.getProperty("user.dir")+"/src/Laporan/rap.rpt&prompt0=" +periode +
```

```
    "&prompt1=" + bk + """);
```

```
    viewer.addNewReportView( renderConnection );
```

```
window.getContentPane().add( BorderLayout.CENTER, viewer);
```

```
window.setVisible( true );
```

```
} catch (Exception e) {
```

```
e.printStackTrace();
```

```
}
```

Form di atas digunakan untuk mencetak laporan realisasi anggaran dan pendapatan.