BAB 3
PEMBAHASAN

3.1. Kegiatan Kerja Praktek

Jadwal kerja praktek dilaksanakan di Yayasan Raudhatul Alhfal Al-Muhajirin Bandung. Waktu dan penempatan kerja praktek adalah sebagai berikut:

Tempat : Sub Bidang Sistem Informasi Keuangan.

Waktu : 15 Juli - 16 agustus 2013

Jam : 09.00 - 16.00

Adapun kegiatan selama kerja praktek sebagai berikut:

Tabel 3.1 Kegiatan Kerja Praktek

No	Tanggal	Jumlah Jam	Uraian Kegiatan	Hasil
1	15 Juli	6	Pemberian Konsep Oleh Pembimbing	Bahan bacaan konsep
2	16 Juli	6	Analisis Konsep yang di berikan pembimbing	Paham konsep
3	17 Juli	6	Analisis Program yang akan di kerjakan(program)	Paham program yang akan dikerjakan
4	18 Juli	6	Analisis Sistem Offline	Mengetahui cara kerja sistem offline
5	19 Juli	6	Analisis Sistem Offline - Pemberian Struktur Table oleh pembimbing	Mengetahui cara kerja sistem offline + mendapat bahan analisis table
6	20 Juli	6	Perencanaan rancangan sistem	Mendapat gambaran system
7	21 Juli	6	Perencanaan rancangan database	Mendapat gambaran database
8	22 Juli	6	Perencanaan rancangan table database + pembuatan table relasi	Mendapat gambaran relasi
9	23 Juli	6	Pembuatan rancangan keseluruhan	Rancangan terbuat sebagian

10	24 Juli	6	Pembuatan rancangan keseluruhan	Rancangan Sudah bisa dilaporkan ke pembimbing
11	25 Juli	6	Pemeriksaan hasil rancangan + Perbaikan hasil rancangan yg telah diperiksa	Memperbaiki kesalahan di rancangan
12	26 Juli	6	Mempelajari SQL Server + Cara Koneksi ke MYSQL	Mendapat cara koneksi SQL Server ke Mysql
13	27 Juli	6	Mendesain Program	Program masih dasar dan coba-coba
14	28 Juli	6	Mendesain Program	Mulai pembuatan Program + mendapat tampilan
15	29 Juli	6	Mendesain Program	Desain Program terbuat 15%
16	1 Agustus	6	Mendesain Program	Desain Program terbuat 25%
17	2 Agustus	6	Mendesain Program + Scripting	Desain Program terbuat 35 % + Mencoba coding
18	3 Agustus	6	Mendesain Program + Scripting	Desain Program terbuat 50% + Coding
19	4 Agustus	6	Mendesain Program + Scripting	Desain Program terbuat 75 % + Coding
20	5 Agustus	6	Mendesain Program + Scripting	Desain Program terbuat 100 % + Coding
21	8 Agustus	6	Scripting	Coding 20%
22	9 Agustus	6	Scripting	Coding 50%
23	10 Agustus	6	Scripting	Coding Selesai
24	11 Agustus	6	Penyelesaian Web	Menemukan Bug
25	12 Agustus	6	Penyelesaian Web	Sudah bisa di presentasikan
26	16 Agustus	2	Presentasi	Pembimbing meminta tambahan feature

3.1.1. Hasil Kerja Praktek

Kegiatan kerja praktek yang dilakukan di Yayasan Raudhatul Alhfal Al-Muhajirin adalah merancang dan membangun sistem informasi keuangan. Adapun hasil analisis dan tahap perancangan dalam pembangunan sistem tersebut adalah sebagai berikut, langkah-langkah yang dilakukan :

1. Analisis sistem.
2. Analisis masalah.
3. Analisis sistem yang akan dikembangkan.
4. Analisis kebutuhan data
5. Analisis kebutuhan sistem.
6. Spesifikasi sistem.
7. Analisis fungsionalitas
8. Analisis non fungsional.
9. Implementasi.

3.2. Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.3. Analisis Masalah

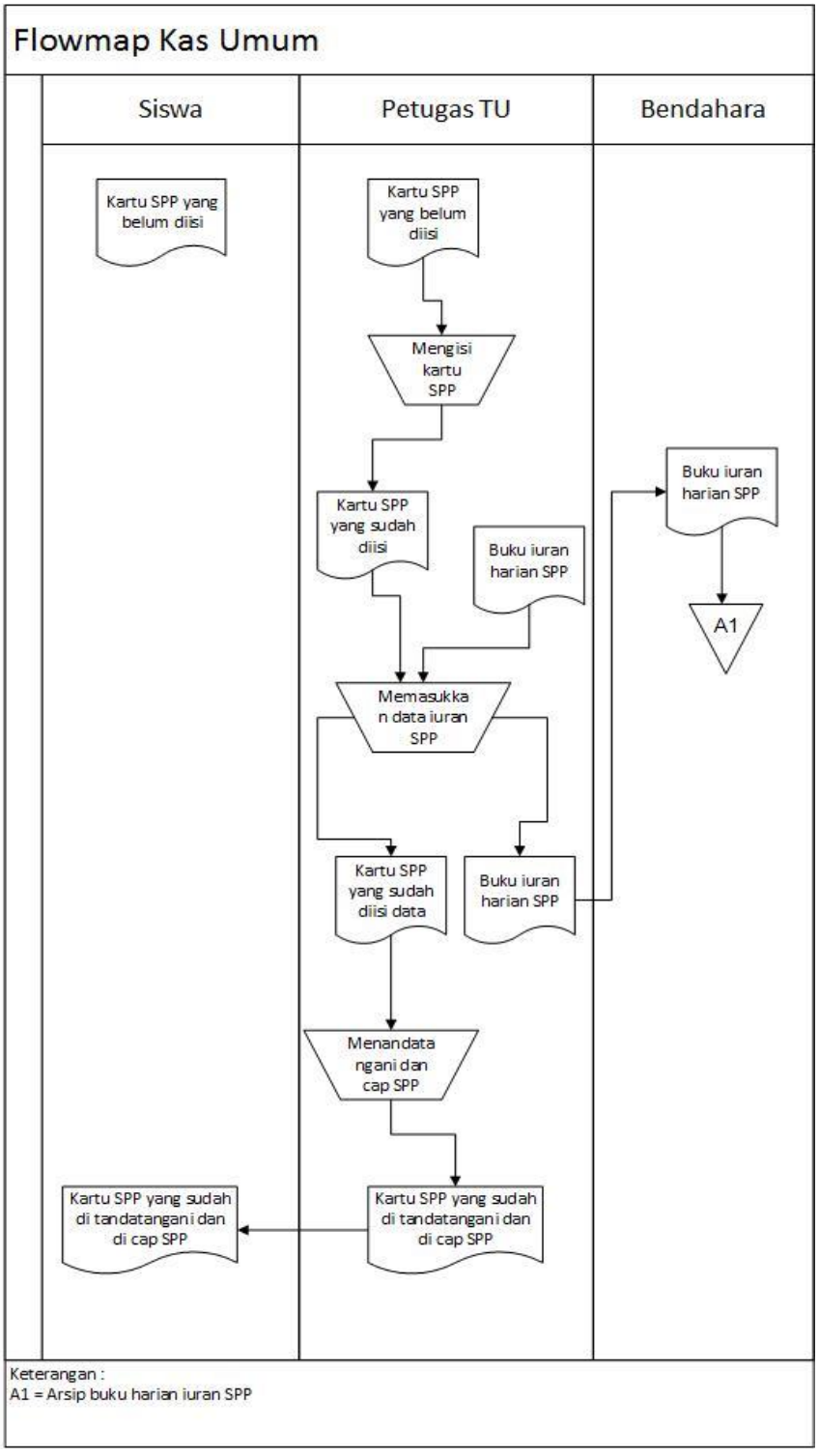
1. Tingkat keakuratan informasi yang kurang memadai baik dikarenakan keterbatasan dalam meng*input* data maupun banyaknya *inputa*n data mengakibatkan banyaknya waktu yang dibutuhkan untuk mengolah data tentang keuangan
2. Sulitnya mencari atau mengetahui informasi yang di butuhkan tentang data keuangan meliputi : gaji guru,anggaran pengeluaran dan pemasukan sekolah.
3. Tidak efektifnya pengolahan data keuangan maka mengakibatkan berkurangnya efisiensi dalam penyampaian data.

3.3.1 Analisis Prosedur Yang sedang Berjalan

Setelah diadakan pengamatan dan wawancara dari sistem yang sedang berjalan, maka akan dibangun sistem informasi yang baru sesuai dengan prosedur yang sedang berjalan di lembaga ini.

3.3.2 Analisis Prosedur Kas Umum

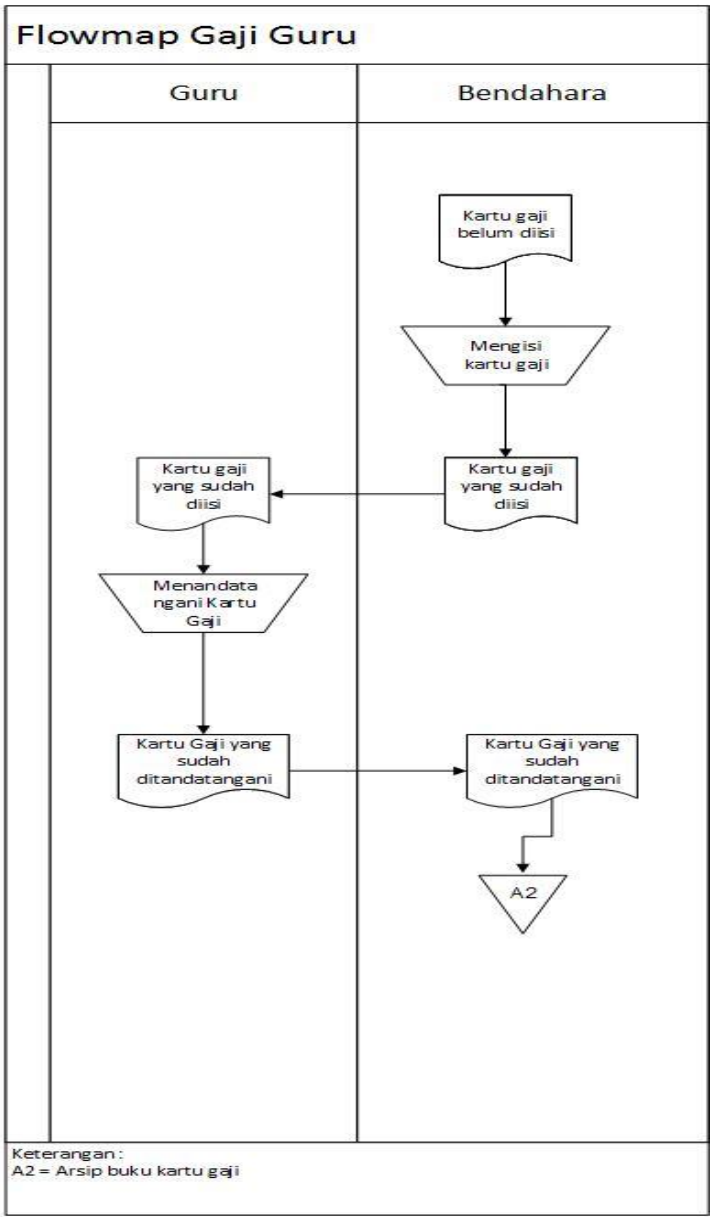
Prosedur kas umum adalah proses dimana petugas melakukan pegolahan data kas umum. Proses pengolahan data kas umum dapat dilihat pada gambar berikut :



Gambar 3.1 Analisi Prosedur Kas Umum

3.3.3 Analisis Prosedur Gaji Guru

Prosedur Gaji Guru adalah proses dimana petugas melakukan pegolahan data Gaji Guru. Proses pengolahan data Gaji Guru dapat dilihat pada gambar berikut



Gambar 3.2 Analisis Prosedur Data Gaji Guru

3.4 Analisis Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan non fungsional adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan.

3. 4.1. Analisis Perangkat Lunak (*Software Analysis*)

Adapun Perangkat lunak yang digunakan di Yayasan Raudhatul Atfhfal Al-muhajirin saat ini adalah sebagai berikut:

1. Microsoft Windows XP
2. Microsoft Office 2007

Spesifikasi perangkat lunak yang dibutuhkan untuk mendukung aplikasi yang akan dibangun adalah sebagai berikut:

1. Microsoft Windows XP / Microsoft Windows 7
2. Borland Delphi
3. Adobe Photoshop

3.4.2. Analisis Perangkat Keras (*Hardware Analysis*)

Adapun spesifikasi komputer pada di Yayasan Raudhatul Atfhfal Al-muhajirin, khususnya pada bagian Keuangan adalah sebagai berikut:

1. Processor Intel Pentium IV 2.0 Ghz
2. Memory RAM DDR 512 MB
3. Harddisk 120 GB
4. VGA 128 MB
5. Monitor 17” LCD
6. Keyboard dan Mouse Standar

3. 4.3 Analisis pengguna (*User Analysis*)

Dalam sistem yang akan berjalan, melibatkan dua (2) pengguna yaitu Administrator dan petugas dengan karakteristiknya yaitu:

Tabel 3.2 Analisis Pengguna

Pengguna	Tanggung jawab	Hak akses	Tingkat pendidikan	Kemampuan yang harus dimiliki
Petugas	Menginputkan data Kas Umum dan Gaji Guru, mengolah data, dan memanipulasi data (*edit*, tambah, hapus, dll)	Mengelola data Kas Umum dan Gaji Guru	1. Minimal S1. 2. Mempunyai sertifikat pelatihan bagi lulusan SMA.	a. Dapat mengoprasikan komputer b. Memiliki ketertarikan pada perangkat lunak misalnya program aplikasi
Admin	Menginputkan data petugas (tambah dan *edit* data petugas, dll)	Hanya mengelola data petugas	a. Minimal S1. b. Mempunyai sertifikat pelatihan admin bagi lulusan SMA.	a. Dapat mengoprasikan komputer b. Memiliki ketertarikan pada perangkat lunak misalnya program aplikasi

3.5 Analisis Fungsional

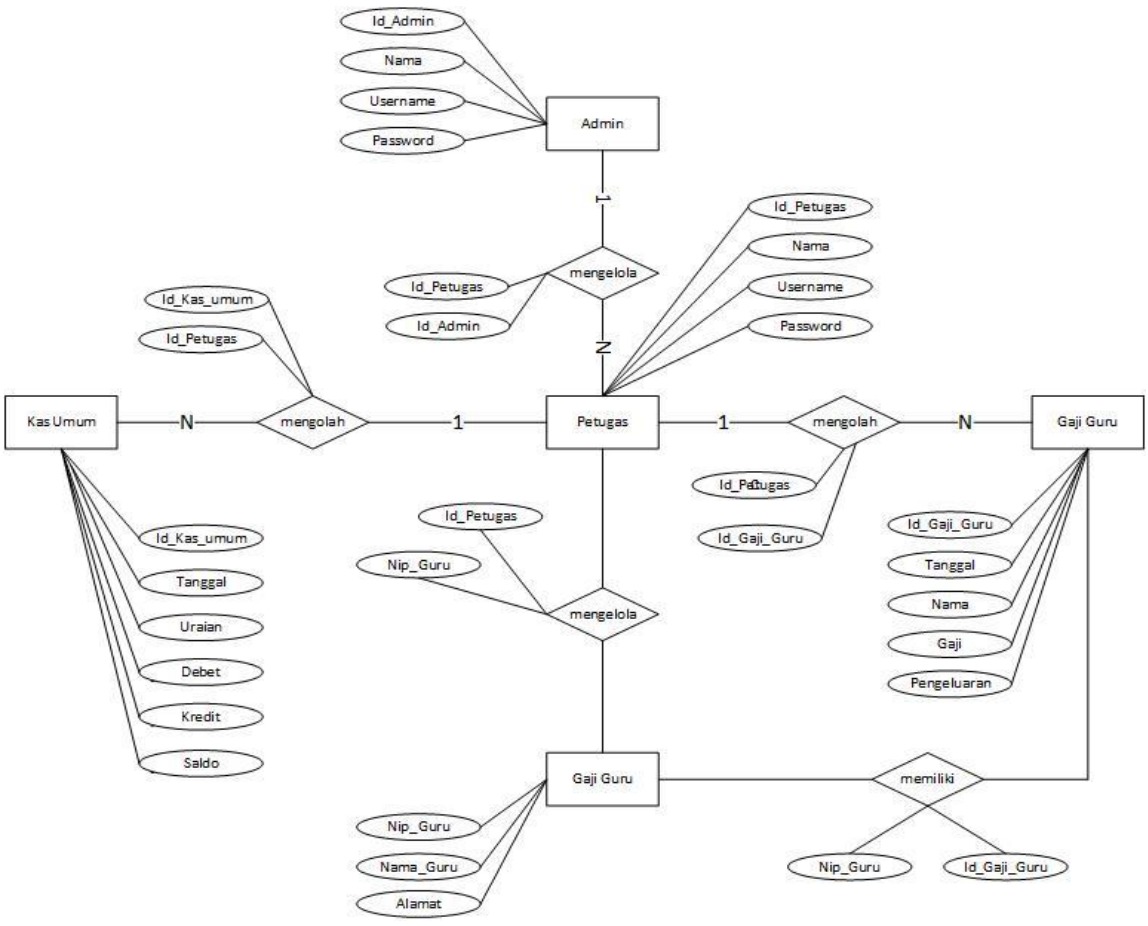
Analisis fungsional dibutuhkan untuk mengembangkan suatu sistem yang berguna bagi proses keuangan. Dimana informasi yang tersedia dapat disajikan dengan menggunakan media berbasis desktop Borland Delphi.

3.5.1 Analisis Perancangan Aliran Informasi

Perancangan aliran informasi ini meliputi perancangan ERD, diagram konteks dan *Data Flow Diagram* (DFD).

3.5.1.1 ERD (Entity Relation Diagram)

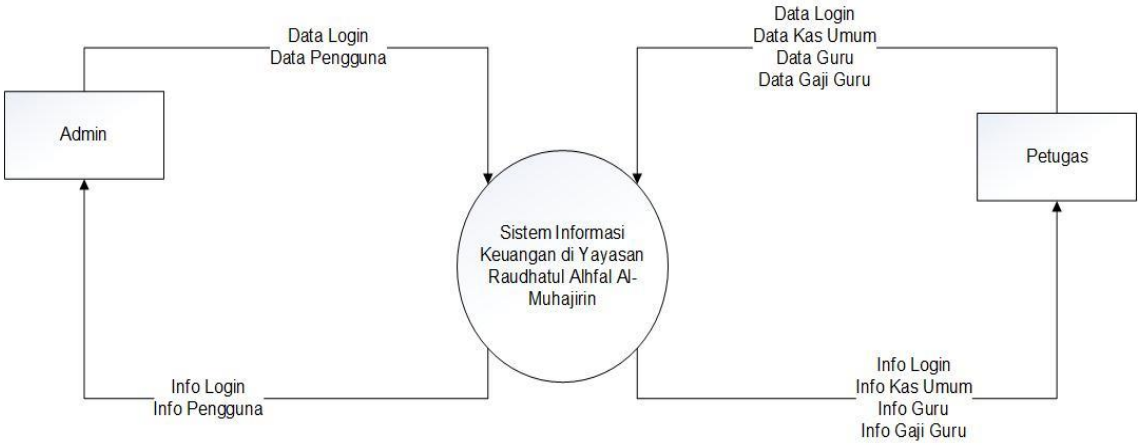
ERD adalah model data yang menggunakan beberapa notasi untuk menggambarkan data dalam konteks entitas dan hubungan yang di deskripsikan oleh data tersebut. Diagram hubungan entitas (ERD) yang telah digunakan dalam pendokumentasian sertifikasi guru ini menggambarkan sistem yang terdiri dari hubungan antar entitas yang sedang berjalan.



Gambar 3. 3 Entity Relationship Diagram

3.5.1.2 Diagram Konteks

Hakekatnya sistem mempunyai keterkaitan dengan sejumlah entitas, baik itu keterkaitan dengan entitas luar sistem maupun keterkaitan dengan entitas dalam sistem. Hubungan antara entitas dengan sistem digambarkan dengan Diagram Konteks. Adapun Diagram Konteks pada Sistem Informasi Berkas dan Kelengkapan Sertifikasi Guru ini adalah sebagai berikut:

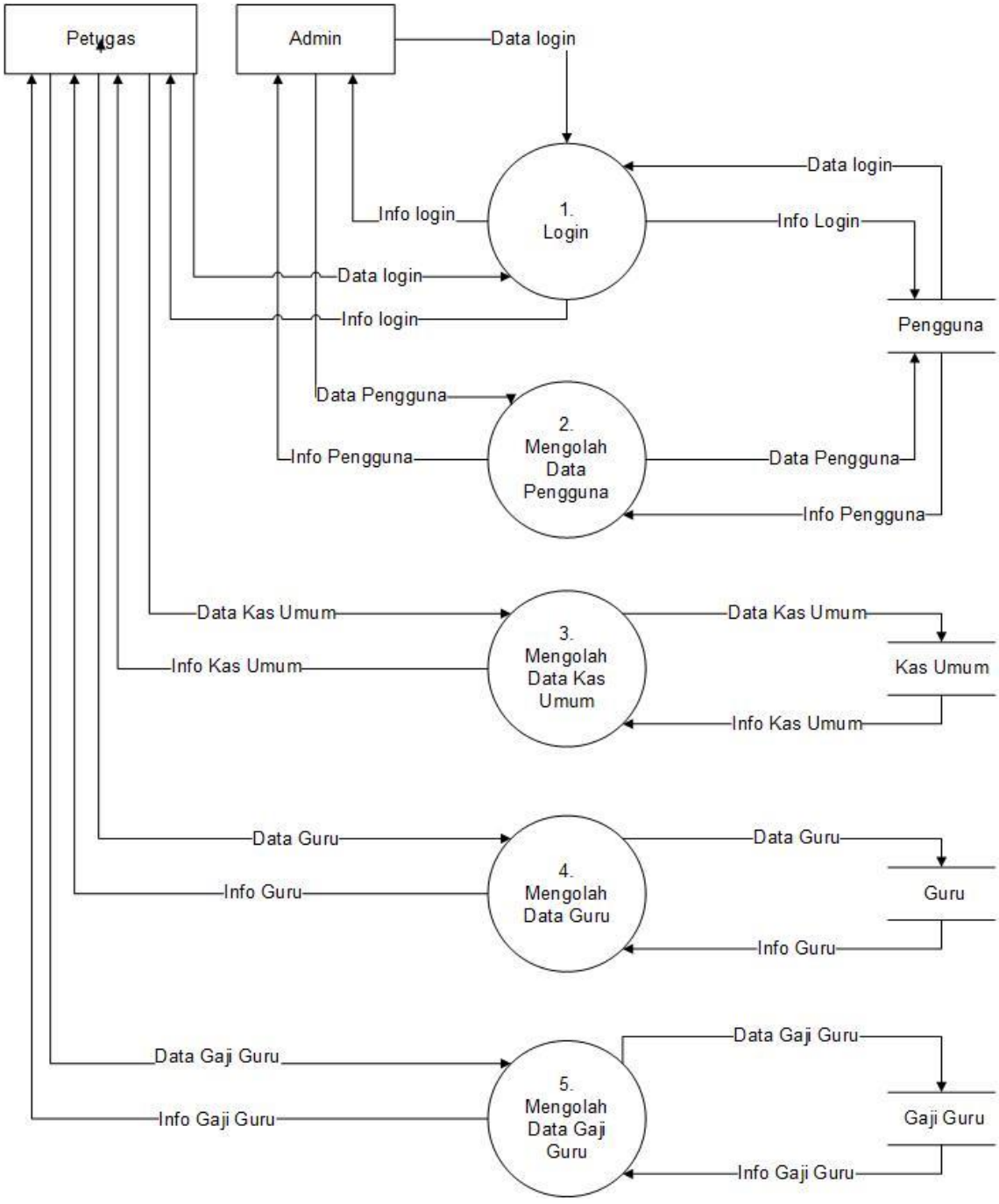


Gambar 3.4 Diagram Konteks

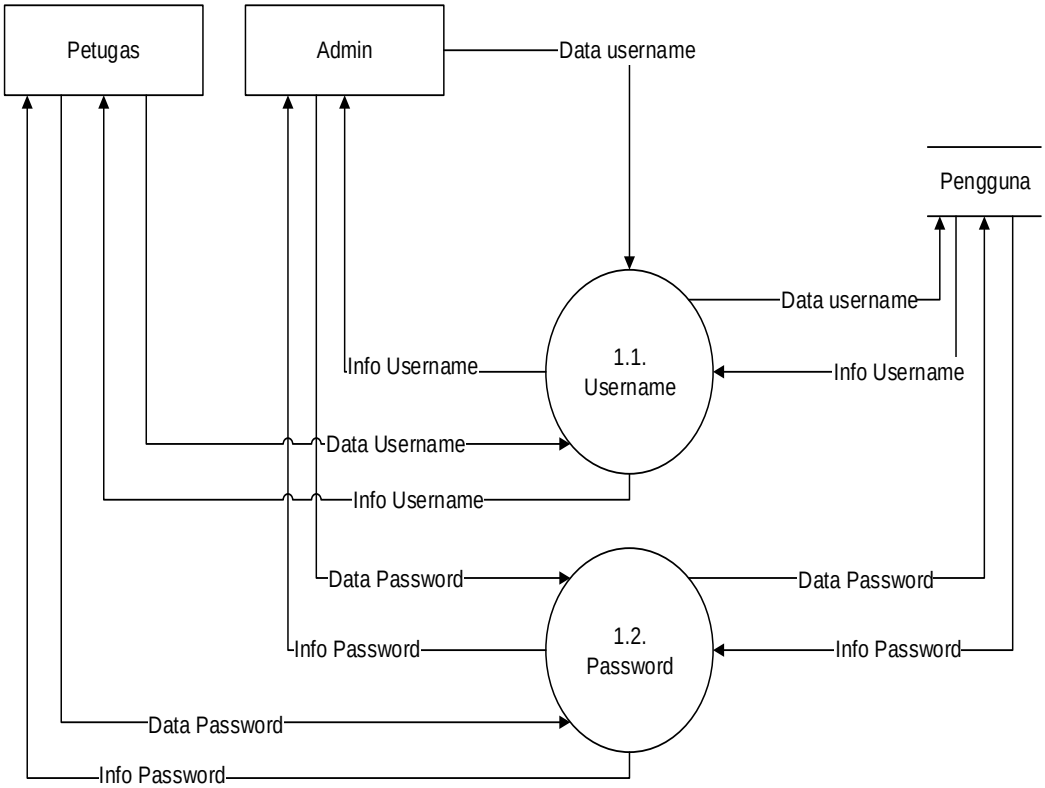
3.5.1.3 DFD *(Data Flow Diagram*)

DFD (*Data Flow Diagram*) merupakan sebuah teknik yang menggambarkan aliran data dan transformasi yang digunakan sebagai perjalanan data dari masukan menuju keluaran.DFD dibuat jika pada Diagram Konteks masih terdapat proses yang mesti dijelaskan lebih rinci.

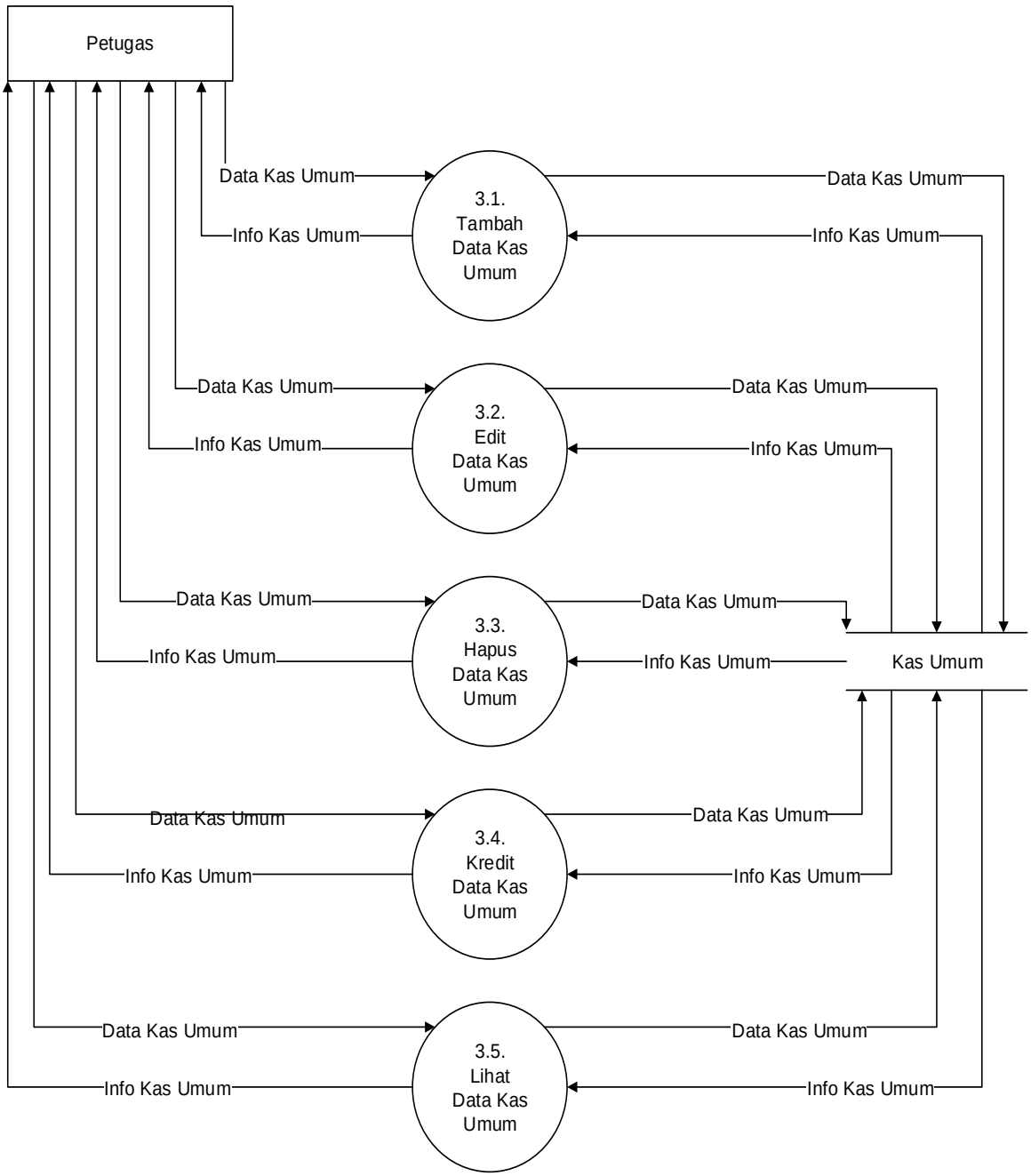
Berikut ini adalah DFD pada Sistem Informasi Keungan di Yayasan Raudhatul Alhfal Al-Muhajirin :



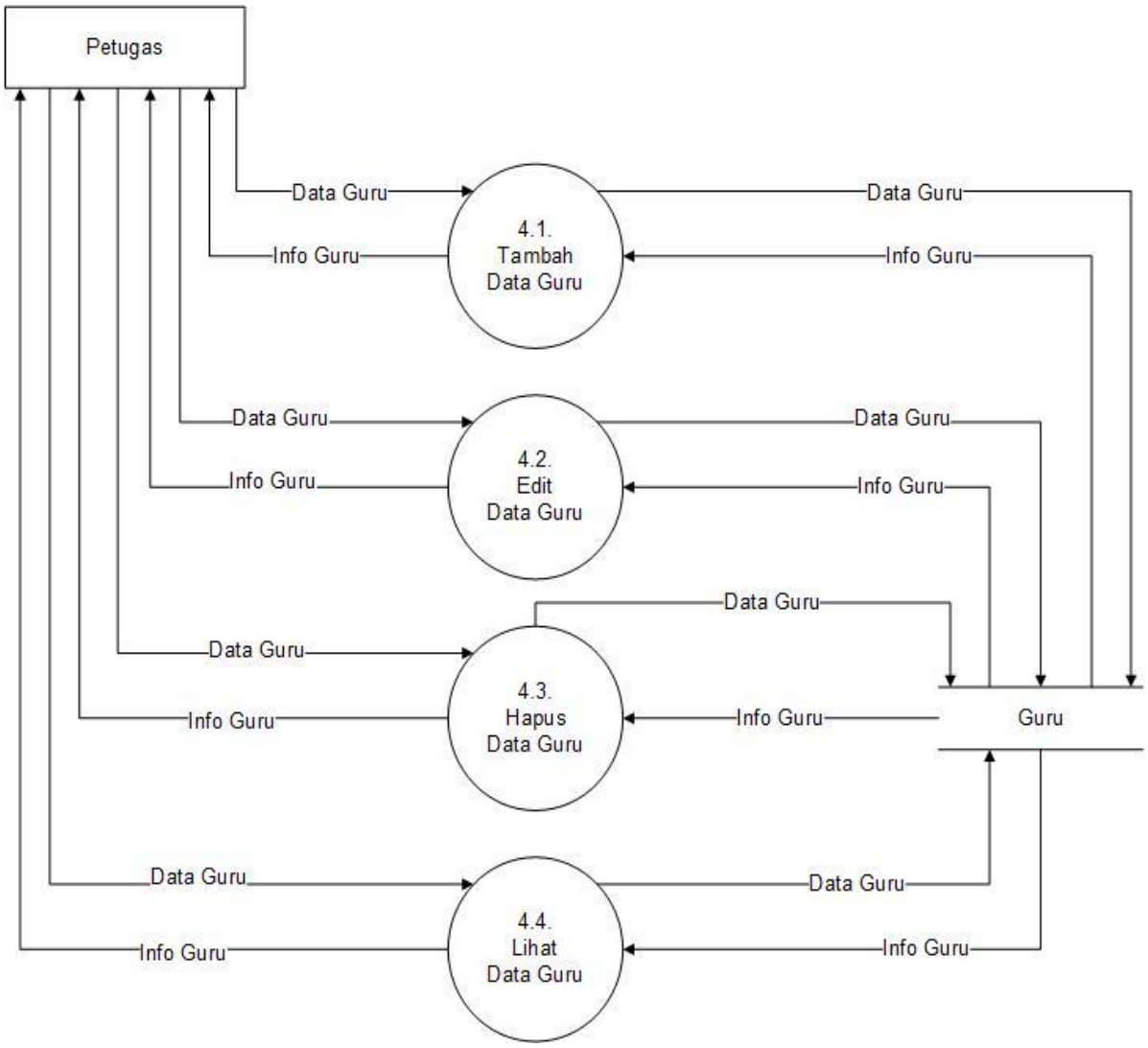
Gambar 3.5 DFD Level 1 Proses Pengolahan Data



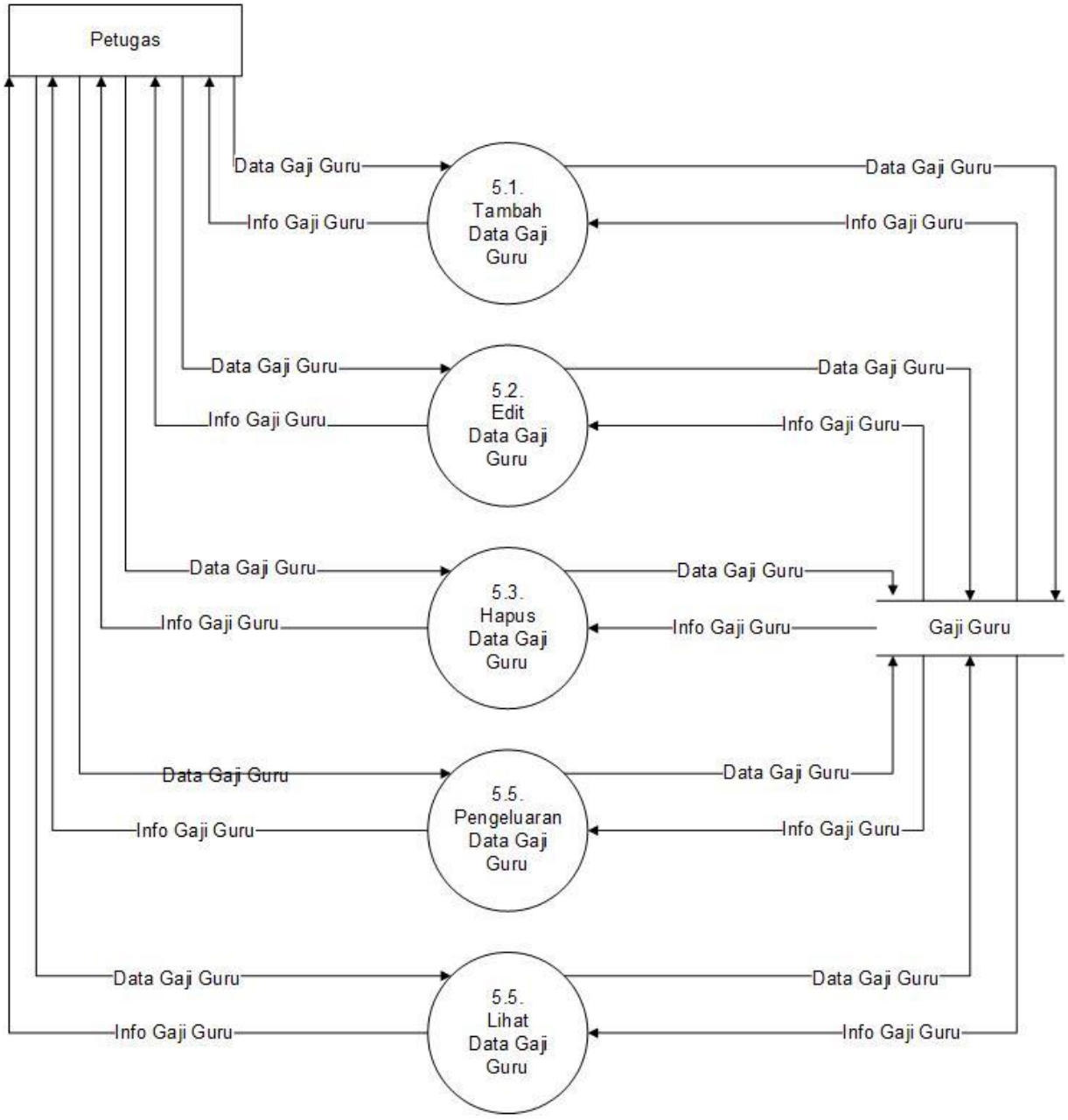
Gambar 3.6 DFD Level 2 Proses 1 Login Pengguna



Gambar 3.7 DFD Level 2 Proses 3 Mengelola Kas Umum



Gambar 3.8 DFD Level 2 Proses 4 Mengelola Data Guru



Gambar 3.9 DFD Level 2 Proses 5 Mengelola Data Gaji Guru

3.5.2. Spesifikasi Proses

Spesifikasi proses digunakan sebagai gambaran proses model aliran pada *Data Flow Diagram*.Untuk menggambarkan proses model aliran pada DFD dijelaskan pada tabel spesifikasi proses di bawah ini. Berikut ini adalah spesifikasi proses dari DFD level 1.

Tabel 3.3 Tabel Spesifikasi Proses

No	Proses	Keterangan
1	No. Proses	1.1
	Nama Proses	Pengecekan Username
	Deskripsi	Untuk mengecek username pada proses verifikasi data login
	Sumber daya	Admin, Petugas
	Tujuan	Admin , Petugas
	Masukan	Data Username
	Keluaran	Info username

	Logika Proses	Begin
		{Admin memasukkan data username ke *database*}
		if username=true
		then tampil info username; pengecekan password
		else username tidak valid
		end;
		end
		Begin
		{Petugas memasukkan data username ke *database*}
		if username=true
		then tampil info username; pengecekan password
		else username tidak valid
		end;
		end

2	No. Proses	1.2
	Nama Proses	Pengecekan Password
	Deskripsi	Untuk mengecek password pada proses verifikasi data login
	Sumber daya	Admin, Petugas
	Tujuan	Admin , petugas
	Masukan	Data password
	Keluaran	Info Password

	Logika Proses	Begin
		{Admin memasukkan data password ke *database*}
		if password=true
		then tampil info password; Masuk Menu Admin
		else password tidak valid
		else;
		else
		Begin
		{Petugas memasukkan data password ke *database*}
		if password=true
		then tampil info password; Masuk Menu Petugas
		else password tidak valid
		else;
		end

3	No. Proses	2.1
	Nama Proses	Tambah pengguna
	Deskripsi	Menambah Pengguna Pada Sistem
	Sumber daya	Admin
	Tujuan	Admin
	Masukan	Data Pengguna
	Keluaran	Info Pengguna

	Logika Proses	Begin {Admin melihat pengguna, Admin menambah Data pengguna, simpan pada *database*} End
4	No. Proses	2.2
	Nama Proses	Edit Data Pengguna
	Deskripsi	Merubah Data pengguna
	Sumber daya	Admin
	Tujuan	Admin
	Masukan	Data Pengguna
	Keluaran	Info Pengguna
	Logika Proses	
5	No. Proses	2.3
	Nama Proses	Hapus Data Pengguna
	Deskripsi	Menghapus Data Pengguna
	Sumber daya	Admin
	Tujuan	Admin
	Masukan	Data Pengguna
	Keluaran	Info Pengguna

	Logika Proses	Begin
		{Admin Menghapus data pengguna}
		if Admin
		then data hapus pengguna, info hapus pengguna
		end;
		End

6	No. Proses	3.1
	Nama Proses	Tambah Kas Umum

	Deskripsi	Menambah Data Kas Umum
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Kas Umum
	Keluaran	Info Kas Umum
	Logika Proses	Begin {Petugas melihat data kas umum, petugas menambah Data kas umum, simpan pada *database*} End
7	No. Proses	3.2
	Nama Proses	Edit Kas Umum
	Deskripsi	Merubah atau mengedit Data Kas Umum
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Kas Umum
	Keluaran	Info Kas Umum
	Logika Proses	
8	No. Proses	**3.3**
	Nama Proses	Hapus Kas Umum
	Deskripsi	Menghapus Data Kas Umum
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Kas Umum
	Keluaran	Info Kas Umum
	Logika Proses	Begin

		{Petugas melihat kas umum, petugas menghapus Data kas umum, simpan pada *database*} End
9	No. Proses	3.4
	Nama Proses	Kredit Data Kas Umum
	Deskripsi	
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Kas umum
	Keluaran	Info Kas Umum
	Logika Proses	
10	No. Proses	3.5
	Nama Proses	Lihat Data Kas Umum
	Deskripsi	Melihat Data Kas Umum
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Kas Umum
	Keluaran	Info Kas Umum

	Logika Proses	Begin
		{Petugas melihat informasi kas umum}
		if petugas
		then tampil semua info kas umum
		end;
		End

11	No. Proses	4.1
	Nama Proses	Tambah Data Guru
	Deskripsi	Menambah Data Guru

	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Guru
	Keluaran	Data Guru
	Logika Proses	Begin {Petugas melihat data guru, petugas menambah Data guru, simpan pada *database*} End
12	No. Proses	4.2
	Nama Proses	Edit Data Guru
	Deskripsi	Proses untuk mengolah edit Data Guru
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Guru
	Keluaran	Data Guru
	Logika Proses	
13	No. Proses	4.3
	Nama Proses	Hapus Data Guru
	Deskripsi	Proses untuk menghapus Data Guru
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Guru
	Keluaran	Data Guru
	Logika Proses	Begin {Petugas melihat data guru, petugas menghapus Data guru, simpan pada *database*} End

14	No. Proses	4.5
	Nama Proses	Lihat Data Guru
	Deskripsi	Proses untuk melihat Data guru yanga ada pada database
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Guru
	Keluaran	Data Guru

	Logika Proses	Begin
		{Petugas melihat informasi data guru}
		if petugas
		then tampil semua info data guru
		end;

11	No. Proses	5.1
	Nama Proses	Tambah Gaji Guru
	Deskripsi	Menambah Data Gaji Guru
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Gaji Guru
	Keluaran	Info Data Gaji Guru
	Logika Proses	Begin {Petugas melihat data gaji guru, petugas menambah Data gaji guru, simpan pada *database*} End
12	No. Proses	5.2
	Nama Proses	Edit Data Gaji Guru
	Deskripsi	Merubah Data Gaji Guru
	Sumber daya	Petugas

	Tujuan	Petugas
	Masukan	Data Gaji Guru
	Keluaran	Info Gaji Guru
	Logika Proses	
13	No. Proses	5.3
	Nama Proses	Hapus Gaji guru
	Deskripsi	Menghapus Data Gaji Guru
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Gaji Guru
	Keluaran	Info Gaji Guru
	Logika Proses	Begin {Petugas melihat Gaji Guru, petugas menghapus Data Gaji guru, simpan pada *database*} End
14	No. Proses	5.4
	Nama Proses	Lihat Data Gaji guru
	Deskripsi	Melihat Data Gaji Guru yang ada pada database
	Sumber daya	Petugas
	Tujuan	Petugas
	Masukan	Data Gaji guru
	Keluaran	Info Gaji Guru

	Logika Proses	Begin
		{Petugas melihat informasi gaji guru}
		if petugas
		then tampil semua info gaji guru
		end;

3.5.3. Kamus Data

Berdasarkan arus data pada *Data Flow Diagram* yang telah dibuat, maka untuk mengetahui kebutuhan-kebutuhan informasi pada sistem bisa dilihat pada tabel kamus data di bawah ini.

Tabel 3.4 Tabel Kamus Data

Nama	**Data Login**
Sumber / Tujuan	Admin
Deskripsi	Berisikan data login (username dan password)
Struktur Data	Id_Admin_Nama+@username+password
@username	[0..9], [A..Z],[a..z]
Password	[0..9], [A..Z],[a..z]
Nama	**Data Login**
Sumber / tujuan	Petugas
Deskripsi	Berisikan data login (username dan password)
Struktur Data	Id_Petugas+Nama+@username+password
@username	[0..9], [A..Z],[a..z]
Password	[0..9], [A..Z],[a..z]
Sumber / tujuan	Data Kas Umum
Nama	**Data Kas Umum**
Deskripsi	Berisi data-data
Struktur Data	Id_Kas_umum+Tanggal+Uraian+Debit+Kredit+Saldo
Id_Kas_umum	0..9}a..z|A..Z|ASCI}

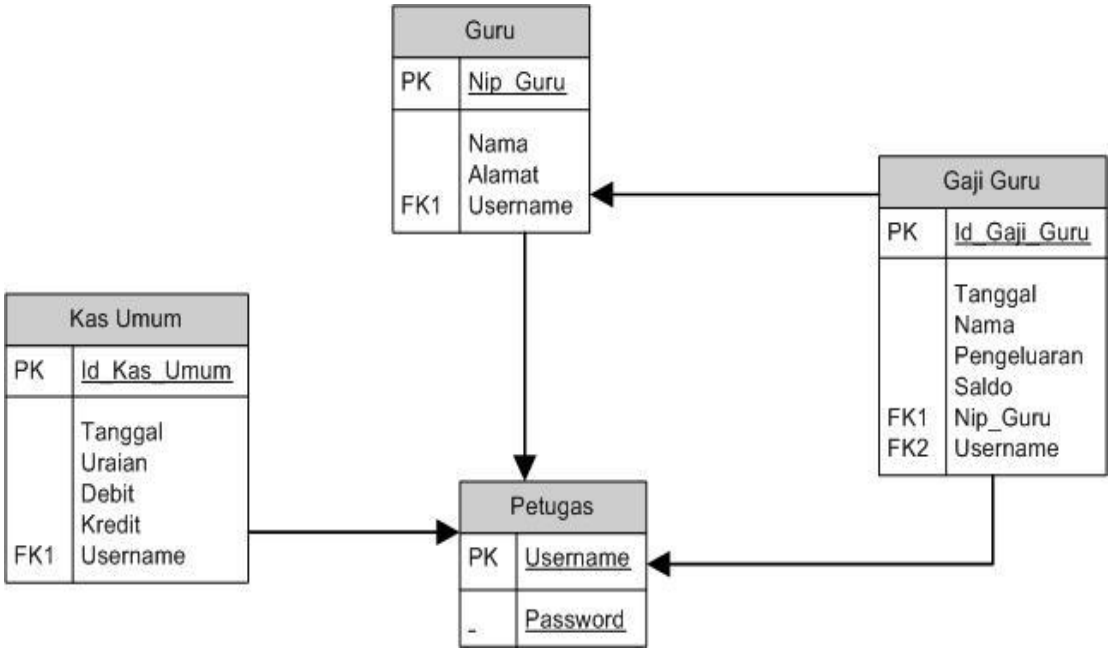
Itanggal	{Date}
Uraian	{0..9}a..z|A..Z|ASCI}
[0..9],	[0..9]
Kredit	[0..9]
Saldo	[0..9]
Nama	**Data Gaji Guru**
Sumber / tujuan	Data Gaji Guru
Deskripsi	
Struktur Data	Id_Gaji_Guru +Tanggal+Nama+Gaji+Pengeluaran+Saldo
Id_Gaji_Guru	0..9}a..z|A..Z|ASCI}
Tanggal	{Date}
Nama	{0..9}a..z|A..Z|ASCI}
Gaji	[0..9]
Pengeluaran	[0..9]
Saldo	[0..9]

3.6 Perancangan Basis Data

Basis data atau *database* adalah kumpulan file-file yang mempunyai kaitan antara satu file dengan file yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu.

3.6.1 Skema Relasi

Skema relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem *database*. Berikut ini adalah skema relasi pada Sistem Informasi Keuangan di Yayasan Raudhatul Alhfal Al-Muhajirin.



Gambar 3.10 Skema Relasi

3.6.2 Struktur Tabel

Berikut ini merupakan struktur tabel dari pembangunan Sistem Informasi Keuangan di Yayasan Raudhatul Alhfal Al-Muhajirin.

Tabel 3. 5 Struktur Tabel Kas Umum

No	Key	Column Name	Data Type	Length	Null	Default
1	**PK**	**Id_Kas_umum**	Varchar	20	No	
2		**Tanggal**	Date		No	
3		**Uraian**	Varchar	100	No	
4		**Debet**	Float		No	
5		**Kredit**	Float			
6		**Saldo**	Float			

Tabel 3.6 Struktur Tabel Guru

No	Key	Column Name	Data Type	Length	Null	Default
1	**PK**	**Nip**	Varchar	20	No	
2		**Nama**	Varchar	50	No	
3		**Alamat**	Varchar	20	No	

Tabel 3.7 Struktur Tabel Gaji Guru

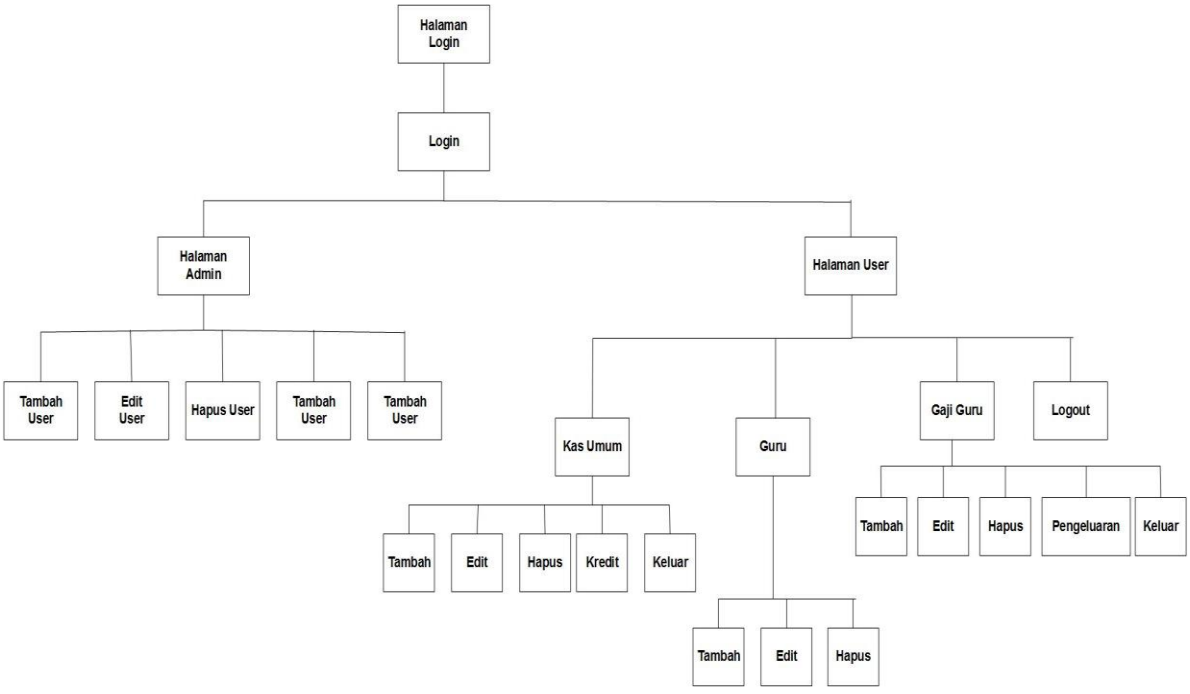
No	Key	Column Name	Data Type	Length	Null	Default
1	**PK**	**Id_Gaji_Guru**	Varchar	20	No	
2		**Tanggal**	Date		No	
3		**Nama**	Varchar	20	No	
4		**Gaji**	Float		No	
5		**Pengeluaran**	Float			
6		**Saldo**	Float			

Tabel 3.9 Struktur Tabel Petugas

No	Key	Column Name	Data Type	Length	Null	Default
1	**PK**	**Username**	Varchar	20	No	
2		**Password**	Varchar	20	No	

3.6.3 Perancangan Struktur Menu

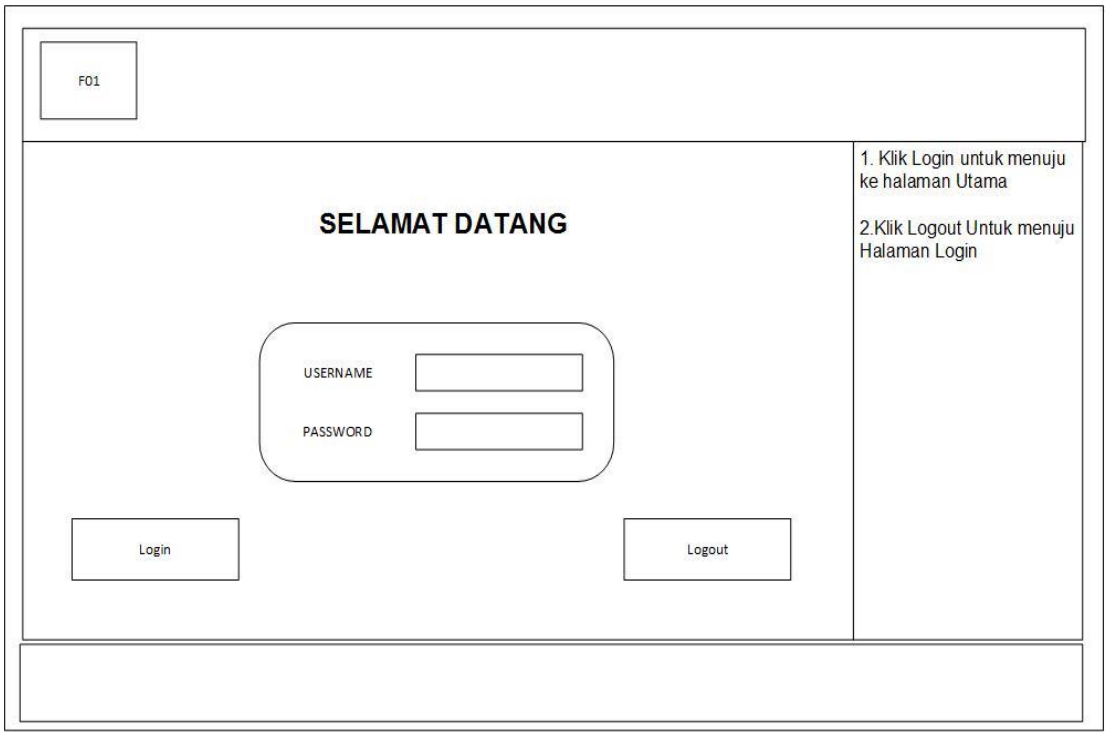
Struktur menu adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer. Sehingga saat menjalankan program,admin dan petugas tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan.Pada perancangan perangkat lunak ini dibuat menu seperti dibawah ini:



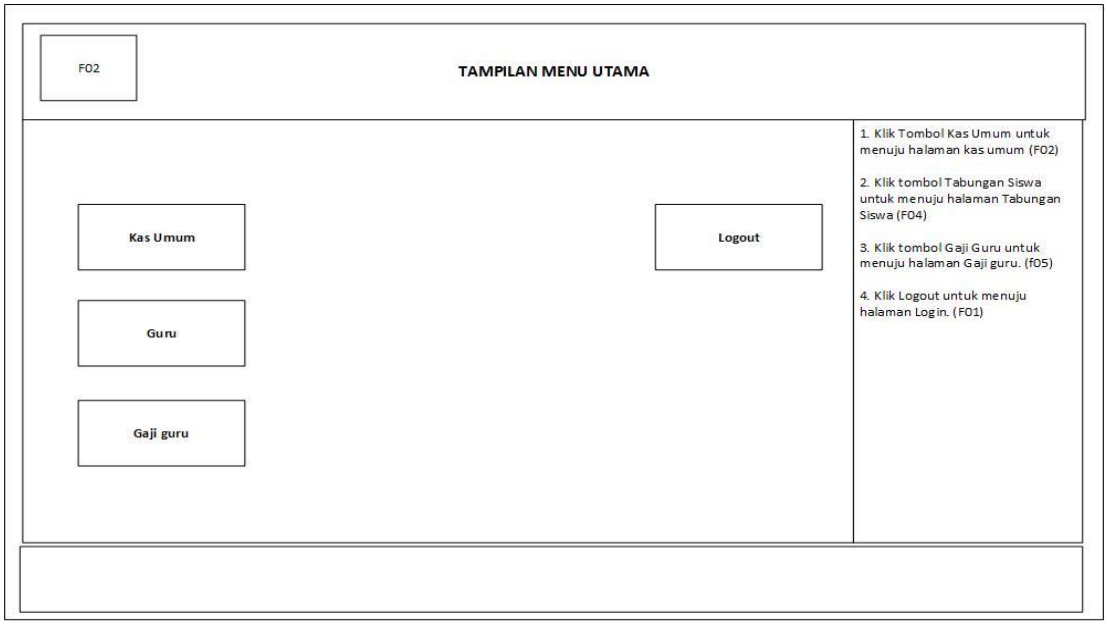
Gambar 3.11 Struktur menu petugas

3.6.4 Perancangan Antarmuka

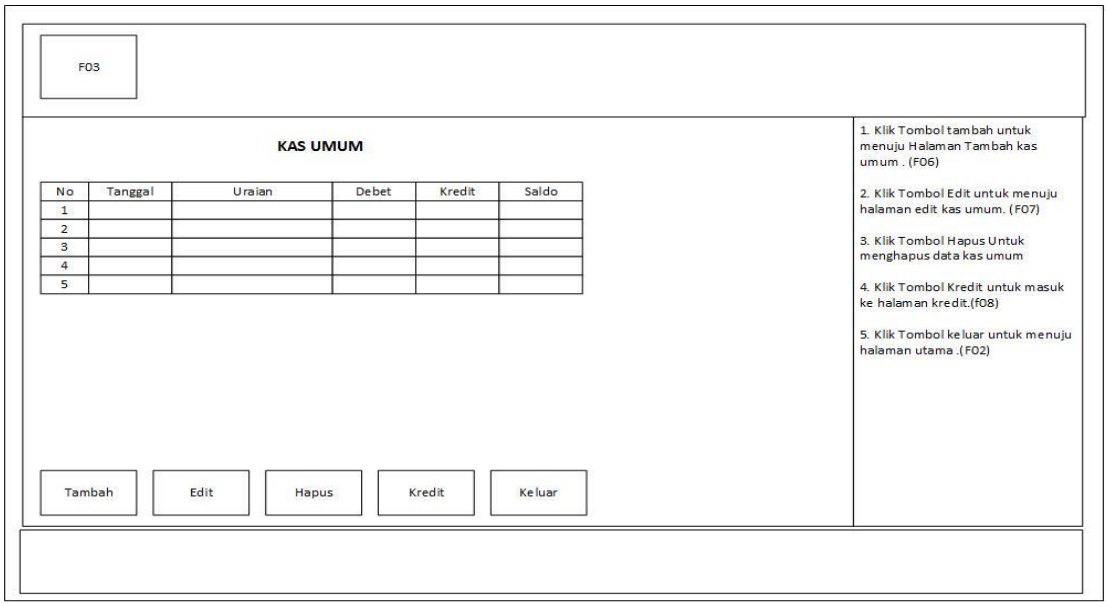
Tahap perancangan antarmuka dilakukan untuk merancang antarmuka agar dapat digunakan oleh pengguna baik admin maupun petugas yang menggunakan Sistem Informasi ini.



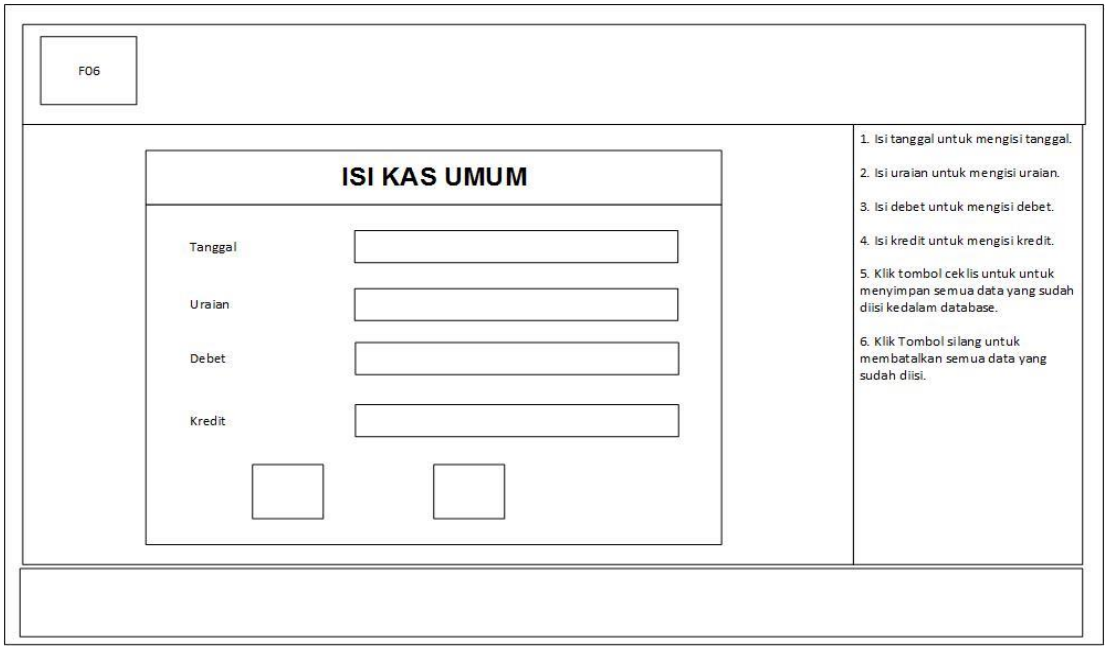
Gambar 3. 12 Perancangan Antarmuka Login Admin dan Petugas



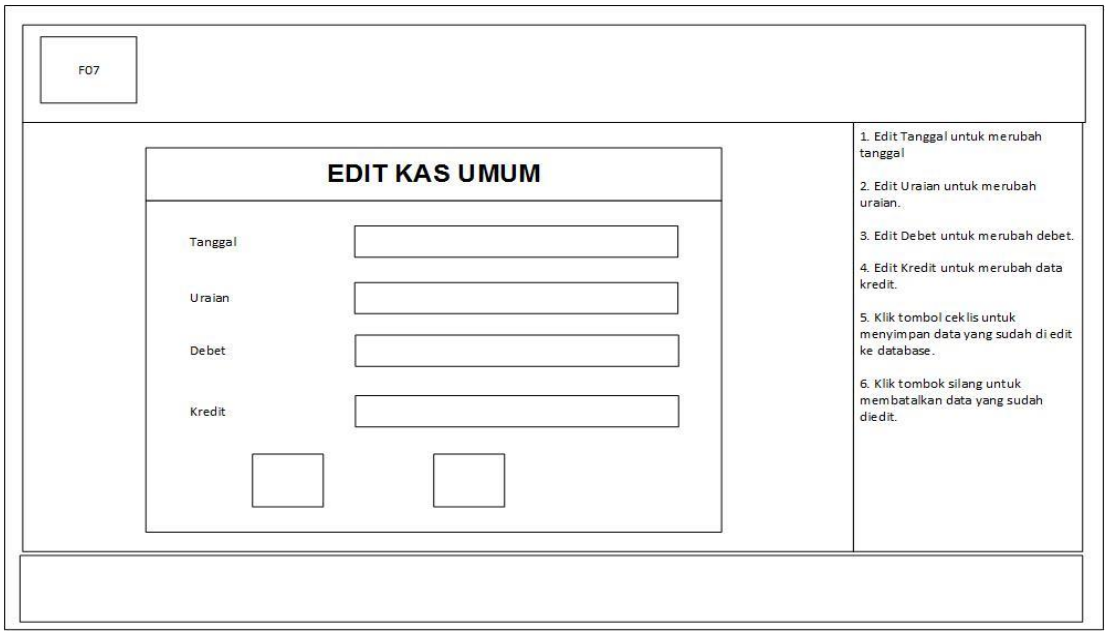
Gambar 3.13 Perancangan Antarmuka Halaman Utama Petugas



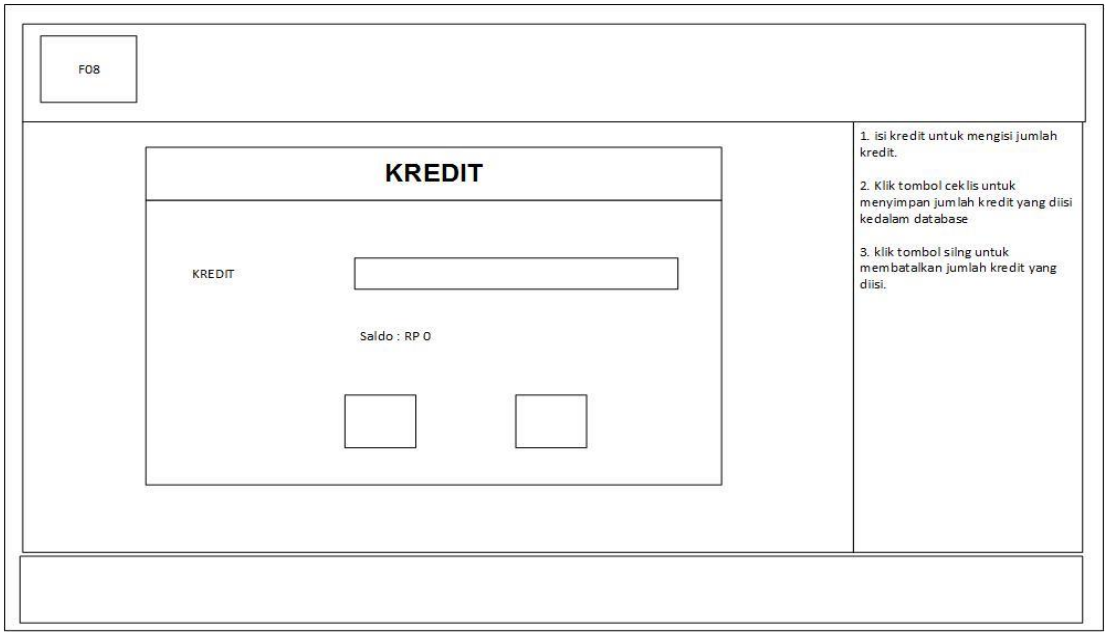
Gambar 3. 14 Perancangan Antarmuka Halaman Kas Umum



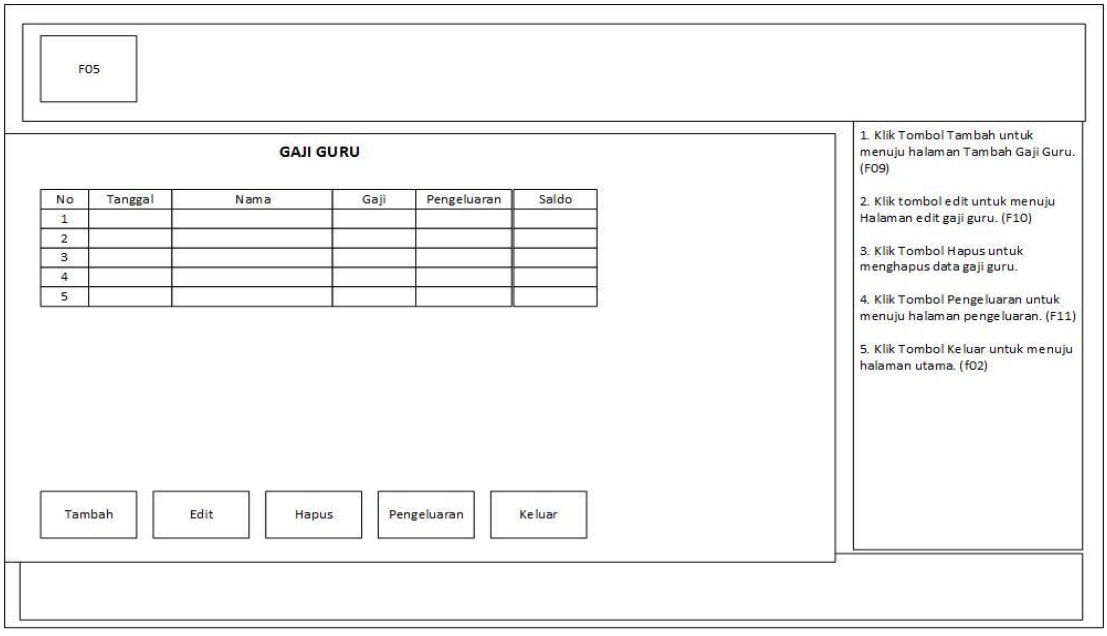
Gambar 3.15 Perancangan Antarmuka Halaman Tambah Kas Umum



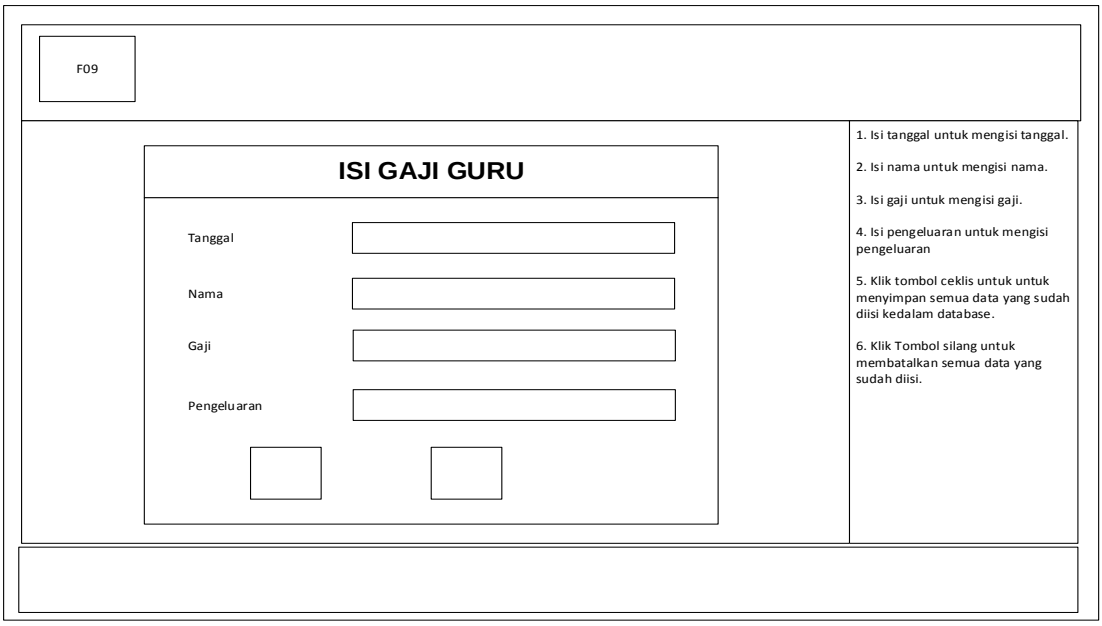
Gambar 3.16 Perancangan Antarmuka Halaman Edit Kas Umum



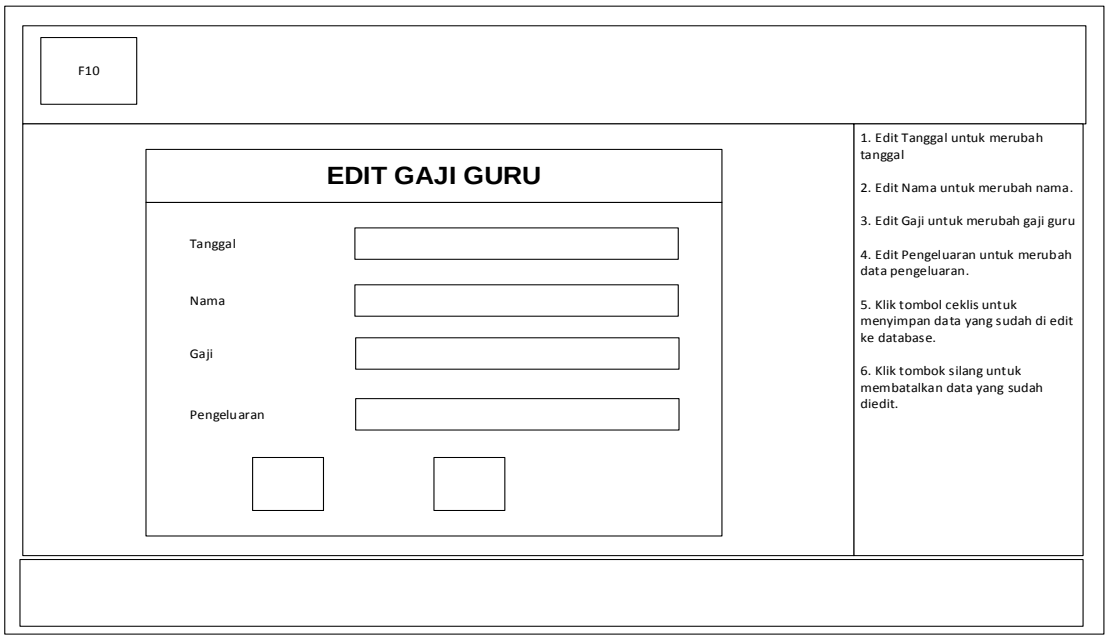
Gambar 3.17 Perancangan Antarmuka Halaman Kredit Kas Umum



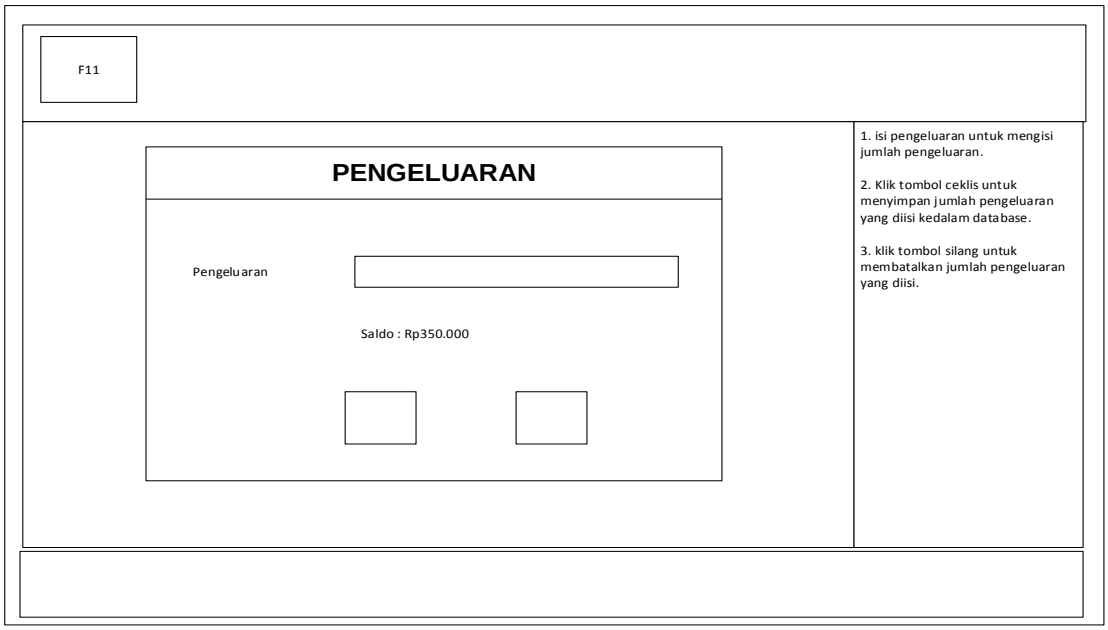
Gambar 3.18 Perancangan Antarmuka Halaman Gaji Guru



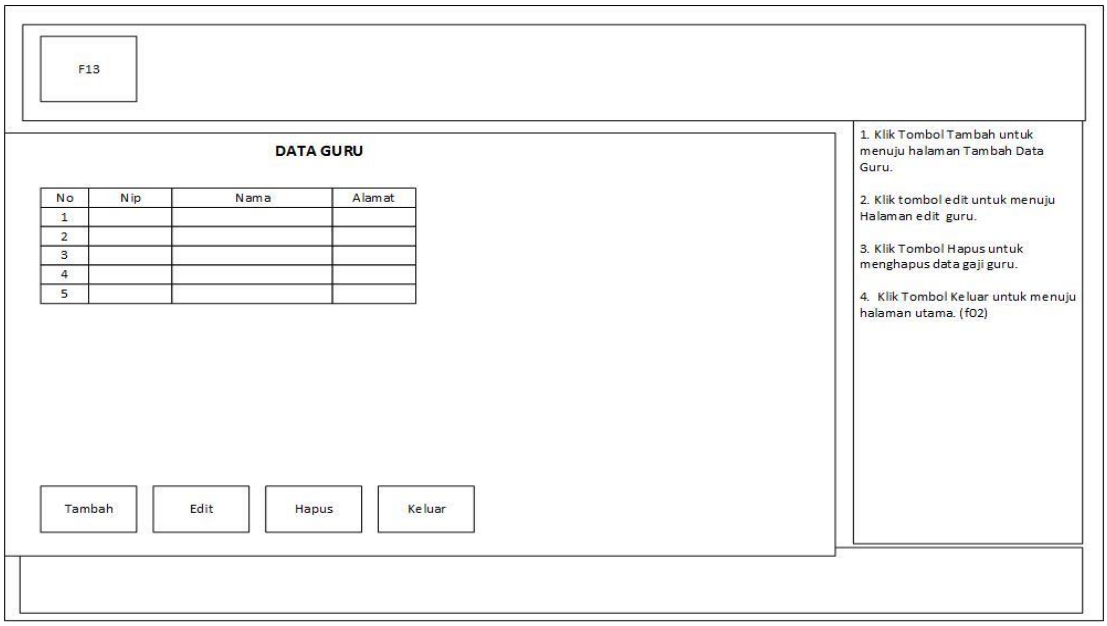
Gambar 3.19 Perancangan Antarmuka Halaman Tambah Gaji Guru



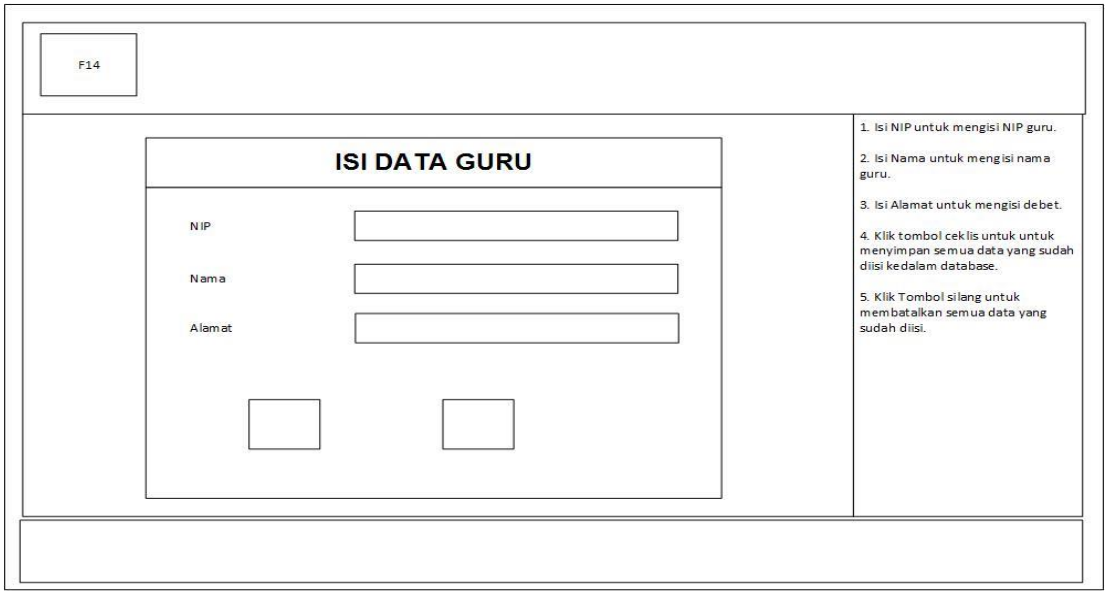
Gambar 3.20 Perancangan Antarmuka Halaman Tambah Edit Gaji Guru



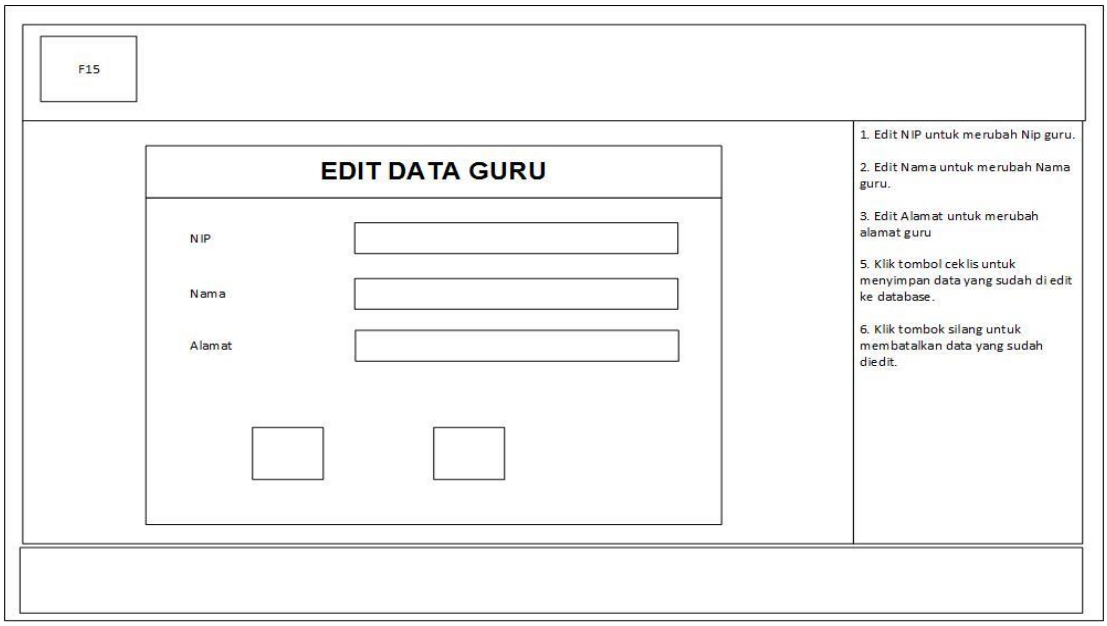
Gambar 3.21 Perancangan Antarmuka Halaman Pengeluaran Gaji Guru



Gambar 3.22 Perancangan Antarmuka Halaman Data Guru



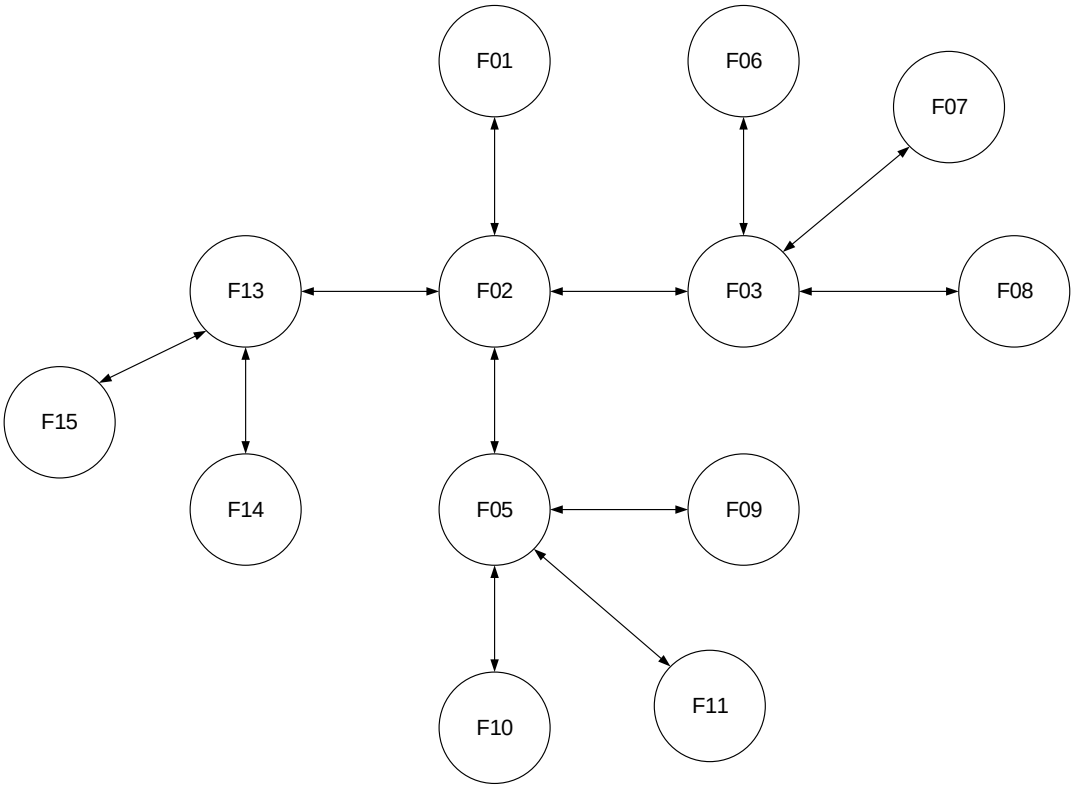
Gambar 3.23 Perancangan Antarmuka Halaman Tambah Data Guru



Gambar 3.24 Perancangan Antarmuka Halaman Edit Guru

3.6.5 . Jaringan Semantik

Jaringan semantik merupakan jaringan data dan informasi, yang menunjukan hubungan antar berbagai objek.



Gambar 3.25 Jaringan Semantik

3.7 Implementasi

Tujuan Implementasi adalah untuk mengkonfirmasikan modul program perancangan pada para pelaku sistem sehingga user dapat memberi masukan kepada pembangun sistem.

3.7.1 Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan untuk mengimplementasikan perangkat lunak, antara lain:

1. Processor Intel Pentium IV 2.0 Ghz
2. Memory RAM DDR 512 MB
3. Harddisk 120 GB
4. VGA 128 MB
5. Monitor 17" LCD
6. Keyboard dan Mouse Standar

3.7.2. Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk mengimplementasikan sistem adalah, sebagai berikut:

1. Microsoft Windows XP / Microsoft Windows 7
2. Borland Delphi
3. Adobe Photoshop

3.7.3. Implementasi Basis Data

Pembuatan basis data dilakukan dengan menggunakan Delphi adalah sebagai berikut:

Admin

```
unit ULogin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, jpeg, ExtCtrls;

type
```

```
  TFLogin = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    ProgressBar1: TProgressBar;
    eusername: TEdit;
    epassword: TEdit;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLogin: TFLogin;

implementation

uses UAdmin, UDM, UUser;

{$R *.dfm}

procedure TFLogin.BitBtn1Click(Sender: TObject);

label A;
var
  i : integer;
begin
  ProgressBar1.Position := 0;
  ProgressBar1.Max := 100;

  for i := 0 to 100 do
  begin
    ProgressBar1.Position := i;
    Sleep(3);
    Application.ProcessMessages;
  end;
begin
  if (Eusername.Text='') or (Epassword.Text='')then
  begin
```

```
    Application.MessageBox('Username dan Password Harus diisi
!!!!','Konfirmasi',MB_OK or
    MB_ICONERROR);
     if EUsername.Text='' then
       EUsername.SetFocus
       else
       Epassword.SetFocus;
  end
   else
  begin
   if (Eusername.Text='admin') then
     begin
       if (Epassword.Text='admin') then
         begin
         FLogin.Hide;
         Fadmin.LAdmin.Caption:='Admin';
         Fadmin.show;
         end
       else
           begin
             Application.MessageBox('Maaf, Password Yang anda masukan Tidak cocok
!!!!','Konfirmasi',MB_OK or
             MB_ICONERROR);
             EPassword.SetFocus;
           end;
     end
   else
   begin
     DM.TLogin.First;
     A:
     if EUsername.Text=DM.TLoginUsername.Value then
       begin
       If Epassword.Text=DM.TloginPassword.Value then
     begin
         FLogin.Hide;
         FUser.LUser.Caption:=DM.TLoginUsername.Value;
         FUser.Show;
       end else
       begin
         Application.MessageBox('Maaf, Password Yang anda masukan tidak cocok
!!!!','Konfirmasi',MB_OK or
     MB_ICONERROR);
       Epassword.SetFocus;
```

```
        end;
      end
      else
      begin
    if DM.TLogin.Eof then
        begin
          Application.MessageBox('Maaf, Username Yang anda masukan tidak Tidak
terdaftar !!!!','Konfirmasi',MB_OK or
          MB_ICONERROR);
          EUsername.setfocus;
        end
      else
        begin
        DM.TLogin.Next;
        goto A;
        end;
     end;
    end;
  end;
end;
end;
procedure TFLogin.BitBtn2Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    FLogin.Close;
      exit;
end;
    end;
end.
```

LOGIN

```
unit ULogin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, jpeg, ExtCtrls;

type
```

```
  TFLogin = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    ProgressBar1: TProgressBar;
    eusername: TEdit;
    epassword: TEdit;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLogin: TFLogin;

implementation

uses UAdmin, UDM, UUser;

{$R *.dfm}

procedure TFLogin.BitBtn1Click(Sender: TObject);

label A;
var
  i : integer;
begin
  ProgressBar1.Position := 0;
  ProgressBar1.Max := 100;

  for i := 0 to 100 do
  begin
    ProgressBar1.Position := i;
    Sleep(3);
    Application.ProcessMessages;
  end;
begin
  if (Eusername.Text='') or (Epassword.Text='')then
  begin
```

```
    Application.MessageBox('Username dan Password Harus diisi
!!!!','Konfirmasi',MB_OK or
    MB_ICONERROR);
     if EUsername.Text='' then
       EUsername.SetFocus
       else
       Epassword.SetFocus;
  end
   else
  begin
   if (Eusername.Text='admin') then
     begin
       if (Epassword.Text='admin') then
         begin
         FLogin.Hide;
         Fadmin.LAdmin.Caption:='Admin';
         Fadmin.show;
         end
       else
           begin
             Application.MessageBox('Maaf, Password Yang anda masukan Tidak cocok
!!!!','Konfirmasi',MB_OK or
             MB_ICONERROR);
             EPassword.SetFocus;
           end;
     end
   else
   begin
     DM.TLogin.First;
     A:
     if EUsername.Text=DM.TLoginUsername.Value then
       begin
       If Epassword.Text=DM.TloginPassword.Value then
     begin
         FLogin.Hide;
         FUser.LUser.Caption:=DM.TLoginUsername.Value;
         FUser.Show;
       end else
       begin
         Application.MessageBox('Maaf, Password Yang anda masukan tidak cocok
!!!!','Konfirmasi',MB_OK or
     MB_ICONERROR);
       Epassword.SetFocus;
```

```
        end;
      end
      else
      begin
    if DM.TLogin.Eof then
        begin
          Application.MessageBox('Maaf, Username Yang anda masukan tidak Tidak
terdaftar !!!!','Konfirmasi',MB_OK or
          MB_ICONERROR);
          EUsername.setfocus;
        end
      else
        begin
        DM.TLogin.Next;
        goto A;
        end;
     end;
    end;
  end;
end;
end;
procedure TFLogin.BitBtn2Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    FLogin.Close;
      exit;
end;
    end;
end.
```

USER

```
unit ULogin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, jpeg, ExtCtrls;

type
```

```
  TFLogin = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    ProgressBar1: TProgressBar;
    eusername: TEdit;
    epassword: TEdit;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLogin: TFLogin;

implementation

uses UAdmin, UDM, UUser;

{$R *.dfm}

procedure TFLogin.BitBtn1Click(Sender: TObject);

label A;
var
  i : integer;
begin
  ProgressBar1.Position := 0;
  ProgressBar1.Max := 100;

  for i := 0 to 100 do
  begin
    ProgressBar1.Position := i;
    Sleep(3);
    Application.ProcessMessages;
  end;
begin
  if (Eusername.Text='') or (Epassword.Text='')then
  begin
```

```
    Application.MessageBox('Username dan Password Harus diisi
!!!!','Konfirmasi',MB_OK or
    MB_ICONERROR);
     if EUsername.Text='' then
       EUsername.SetFocus
       else
       Epassword.SetFocus;
  end
   else
  begin
   if (Eusername.Text='admin') then
     begin
       if (Epassword.Text='admin') then
         begin
         FLogin.Hide;
         Fadmin.LAdmin.Caption:='Admin';
         Fadmin.show;
         end
       else
           begin
             Application.MessageBox('Maaf, Password Yang anda masukan Tidak cocok
!!!!','Konfirmasi',MB_OK or
             MB_ICONERROR);
             EPassword.SetFocus;
           end;
     end
   else
   begin
     DM.TLogin.First;
     A:
     if EUsername.Text=DM.TLoginUsername.Value then
       begin
       If Epassword.Text=DM.TloginPassword.Value then
     begin
         FLogin.Hide;
         FUser.LUser.Caption:=DM.TLoginUsername.Value;
         FUser.Show;
       end else
       begin
         Application.MessageBox('Maaf, Password Yang anda masukan tidak cocok
!!!!','Konfirmasi',MB_OK or
     MB_ICONERROR);
       Epassword.SetFocus;
```

```
        end;
      end
      else
      begin
    if DM.TLogin.Eof then
        begin
          Application.MessageBox('Maaf, Username Yang anda masukan tidak Tidak
terdaftar !!!!','Konfirmasi',MB_OK or
          MB_ICONERROR);
          EUsername.setfocus;
        end
      else
        begin
        DM.TLogin.Next;
        goto A;
        end;
     end;
    end;
  end;
end;
end;
procedure TFLogin.BitBtn2Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    FLogin.Close;
      exit;
end;
    end;
end.
```

UNIT ADMIN

```
unit UAdmin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Buttons, Menus, jpeg;

type
  TFAdmin = class(TForm)
```

```
    Panel1: TPanel;
    Image1: TImage;
    MainMenu1: TMainMenu;
    Edit1: TMenuItem;
    ambahUser1: TMenuItem;
    EditUser1: TMenuItem;
    HapusUser1: TMenuItem;
    Out1: TMenuItem;
    Logout1: TMenuItem;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    Shape1: TShape;
    Label1: TLabel;
    LAdmin: TLabel;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure ambahUser1Click(Sender: TObject);
    procedure EditUser1Click(Sender: TObject);
    procedure Logout1Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure HapusUser1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FAdmin: TFAdmin;

implementation

uses UTambahUser, UEditUser, ULogin, UHapusUser, UDM;

{$R *.dfm}

procedure TFAdmin.SpeedButton1Click(Sender: TObject);
begin
```

```
DM.TLogin.Insert;
FTambahUser.ShowModal;
end;

procedure TFAdmin.SpeedButton2Click(Sender: TObject);
begin
FEditUser.ShowModal;
end;

procedure TFAdmin.SpeedButton4Click(Sender: TObject);
begin
  if Application.MessageBox('Anda ingin Log Out ?','Konfimasi',MB_YESNO or
MB_ICONQUESTION)=id_Yes then
  begin
  Fadmin.Hide;
  FLogin.Show;
  FLogin.EUsername.Clear;
  Flogin.EPassword.Clear;
  Flogin.EUsername.SetFocus;
  end
  else
  begin
  Focused;
  refresh;
  end;
   end;

procedure TFAdmin.SpeedButton5Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    Fadmin.close;
      exit;
end;
  end;

procedure TFAdmin.ambahUser1Click(Sender: TObject);
begin
FTambahUser.ShowModal;
end;

procedure TFAdmin.EditUser1Click(Sender: TObject);
```

```
begin
FEditUser.ShowModal;
end;

procedure TFAdmin.Logout1Click(Sender: TObject);

begin
  if Application.MessageBox('Anda ingin Log Out ?','Konfimasi',MB_YESNO or
MB_ICONQUESTION)=id_Yes then
  begin
  Fadmin.Hide;
  FLogin.Show;
  FLogin.EUsername.Clear;
  Flogin.EPassword.Clear;
  Flogin.EUsername.SetFocus;
  end
  else
  begin
  Focused;
  refresh;
  end;
   end;


procedure TFAdmin.Exit1Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    Fadmin.close;
      exit;
    end;
    end;
procedure TFAdmin.SpeedButton3Click(Sender: TObject);
begin
FHapusUser.ShowModal;
end;

procedure TFAdmin.HapusUser1Click(Sender: TObject);
begin
FHapusUser.ShowModal;
end;
end.
```

UDM

```
unit UDM;

interface

uses
  SysUtils, Classes, DB, DBTables;

type
  TDM = class(TDataModule)
    Login: TDataSource;
    TLogin: TTable;
    TLoginUsername: TStringField;
    TLoginPassword: TStringField;
    DSKas: TDataSource;
    TKas: TTable;
    TKasNo: TAutoIncField;
    TKasTanggal: TDateField;
    TKasUraian: TStringField;
    TKasDebet: TCurrencyField;
    TKasKredit: TCurrencyField;
    TKasSaldo: TCurrencyField;
    DSGuru: TDataSource;
    TGuru: TTable;
    TGuruNIP: TStringField;
    TGuruGaji: TCurrencyField;
    TGuruPengeluaran: TCurrencyField;
    TGuruTanggal: TDateField;
    TDGuru: TTable;
    DSDGuru: TDataSource;
    TDGuruNo: TAutoIncField;
    TDGuruNIP: TStringField;
    TDGuruNama: TStringField;
    TDGuruAlamat: TStringField;
    TGuruSaldo: TCurrencyField;
    TGuruNama: TStringField;
    procedure TKasCalcFields(DataSet: TDataSet);
    procedure TGuruCalcFields(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  DM: TDM;

implementation

uses ULogin;

{$R *.dfm}

procedure TDM.TKasCalcFields(DataSet: TDataSet);
begin
DM.TKasSaldo.Value:=DM.TKasDebet.Value-DM.TKasKredit.Value;
end;

procedure TDM.TGuruCalcFields(DataSet: TDataSet);
begin
DM.TGuruSaldo.Value:=DM.TGuruGaji.Value-DM.TGuruPengeluaran.Value;
end;

end.
```

EDIT USER

```
unit UEditUser;

interface

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

type
  TFEditUser = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

```
  end;

var
  FEditUser: TFEditUser;

implementation

uses UDM, UIsiEditUser;

{$R *.dfm}

procedure TFEditUser.BitBtn1Click(Sender: TObject);
begin
DM.TLogin.Edit;
Fisiedituser.Show;
end;

end.
```

GAJI GURU

```
unit UGajiGuru;

interface

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

type
  TFGajiGuru = class(TForm)
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    SpeedButton6: TSpeedButton;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
```

```
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FGajiGuru: TFGajiGuru;

implementation

uses UDM, UIsiGajiGuru, UPengeluaran, ULapGajiGuru;

{$R *.dfm}

procedure TFGajiGuru.SpeedButton1Click(Sender: TObject);
begin
DM.TGuru.Insert;
FIsiGajiGuru.ShowModal;
end;

procedure TFGajiGuru.SpeedButton2Click(Sender: TObject);
begin
DM.TGuru.Edit;
FIsiGajiGuru.ShowModal;
end;

procedure TFGajiGuru.SpeedButton3Click(Sender: TObject);
var
  Pesan:String;
begin
   Pesan:='Anda yakin data berikut : '+#13+#13+
         'Nama : '+DM.TGuruNama.Value+#13+
         'akan dihapus?';
if messageDlg (Pesan,mtConfirmation,[mbYes,mbNo],0)=mrYes then
begin
DM.TGuru.Delete;
end;
```

```
end;

procedure TFGajiGuru.SpeedButton4Click(Sender: TObject);
begin
DM.TGuru.Edit;
FPengeluaran.Showmodal;
end;

procedure TFGajiGuru.SpeedButton5Click(Sender: TObject);
begin
close;
end;

procedure TFGajiGuru.SpeedButton6Click(Sender: TObject);
begin
LapGajiGuru.Preview;
end;

end.
```

DATA GURU

```
unit UGuru;

interface

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

type
  TFGuru = class(TForm)
    Panel1: TPanel;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton5: TSpeedButton;
    Panel2: TPanel;
    DBGrid1: TDBGrid;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FGuru: TFGuru;

implementation

uses UDM, UIsiGuru;

{$R *.dfm}

procedure TFGuru.SpeedButton1Click(Sender: TObject);
begin
  DM.TDGuru.Insert;
  FIsiGuru.ShowModal;
end;

procedure TFGuru.SpeedButton2Click(Sender: TObject);
begin
DM.TDGuru.Edit;
FIsiGuru.ShowModal;
end;

procedure TFGuru.SpeedButton3Click(Sender: TObject);
var
  Pesan:String;
begin
   Pesan:='Anda yakin data berikut : '+#13+#13+
         'Nama : '+DM.TDGuruNama.Value+#13+
         'akan dihapus?';
if messageDlg (Pesan,mtConfirmation,[mbYes,mbNo],0)=mrYes then
begin
DM.TDGuru.Delete;
end;
end;
procedure TFGuru.SpeedButton5Click(Sender: TObject);
begin
close;
end;
```

KAS UMUM

```
unit UIsiKasUmum;

interface

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

type
  TFIsiKasUmum = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Label5: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FIsiKasUmum: TFIsiKasUmum;

implementation

uses UDM;

{$R *.dfm}

procedure TFIsiKasUmum.BitBtn1Click(Sender: TObject);
begin
DM.TKasTanggal.Value;
```

```
if DM.TKasTanggal.IsNull then
 begin
   ShowMessage('Tanggal Tidak Boleh Kosong Dan Diisi Dengan Benar
(dd/mm/yyyy) !!!');
   dbEDIT1.SetFocus;
   Exit;
   end;
begin
DM.TKasUraian.Value:=Trim(DM.TKasUraian.Value);
if DM.TKasUraian.IsNull then
 begin
   ShowMessage('Uraian Tidak Boleh Kosong!');
   dbEDIT2.SetFocus;
   Exit;
   end;
begin
DM.TKasDebet.Value;
if DM.TKasDebet.IsNull then
 begin
   ShowMessage('Debet Tidak Boleh Kosong!');
   dbEDIT3.SetFocus;
   Exit;
   end;
DM.TKas.Post;
Close;
end;
end;
end;

procedure TFIsiKasUmum.BitBtn2Click(Sender: TObject);
begin
if messageDlg('Pengisian data dibatalkan ?',mtConfirmation,
          [mbYes,mbNo],0)=mrYes then
begin
DM.TKas.Cancel;
Close;
end;
end;
end.
```

KREDIT

```
unit UKredit;

interface

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

type
  TFKredit = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    DBEdit1: TDBEdit;
    Label2: TLabel;
    DBText1: TDBText;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FKredit: TFKredit;

implementation

uses UDM;

{$R *.dfm}

procedure TFKredit.BitBtn1Click(Sender: TObject);
begin
DM.TKas.Post;
Close;
end;

procedure TFKredit.BitBtn2Click(Sender: TObject);
```

```
begin
if messageDlg('Pengisian data dibatalkan ?',mtConfirmation,
          [mbYes,mbNo],0)=mrYes then
begin
DM.TKas.Cancel;
Close;
end;
end;

end.
```

LAPORAN GAJI GURU

```
unit ULapGajiGuru;

interface

uses Windows, SysUtils, Messages, Classes, Graphics, Controls,
  StdCtrls, ExtCtrls, Forms, QuickRpt, QRCtrls;

type
  TLapGajiGuru = class(TQuickRep)
    PageHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    ColumnHeaderBand1: TQRBand;
    DetailBand1: TQRBand;
    SummaryBand1: TQRBand;
    PageFooterBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRLabel9: TQRLabel;
```

```
    QRSysData1: TQRSysData;
    QRLabel10: TQRLabel;
    QRSysData2: TQRSysData;
  private

  public

  end;

var
  LapGajiGuru: TLapGajiGuru;

implementation

uses UDM;

{$R *.DFM}

end.
```

LAPORAN KAS UMUM

```
unit ULapKasUmum;

interface

uses Windows, SysUtils, Messages, Classes, Graphics, Controls,
  StdCtrls, ExtCtrls, Forms, QuickRpt, QRCtrls;

type
  TLapKasUmum = class(TQuickRep)
    PageHeaderBand1: TQRBand;
    TitleBand1: TQRBand;
    DetailBand1: TQRBand;
    SummaryBand1: TQRBand;
    PageFooterBand1: TQRBand;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRBand1: TQRBand;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRLabel9: TQRLabel;
    QRLabel10: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
```

```
    QRDBText1: TQRDBText;
    QRDBText2: TQRDBText;
    QRDBText3: TQRDBText;
    QRDBText4: TQRDBText;
    QRDBText5: TQRDBText;
    QRDBText6: TQRDBText;
    QRLabel5: TQRLabel;
    QRSysData1: TQRSysData;
    QRLabel6: TQRLabel;
    QRSysData2: TQRSysData;
  private

  public

  end;

var
  LapKasUmum: TLapKasUmum;

implementation

uses UDM, UKasUmum;

{$R *.DFM}

end.
```

LOGIN

```
unit ULogin;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ComCtrls, jpeg, ExtCtrls;

type
  TFLogin = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    ProgressBar1: TProgressBar;
    eusername: TEdit;
    epassword: TEdit;
```

```
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FLogin: TFLogin;

implementation

uses UAdmin, UDM, UUser;

{$R *.dfm}

procedure TFLogin.BitBtn1Click(Sender: TObject);

label A;
var
  i : integer;
begin
  ProgressBar1.Position := 0;
  ProgressBar1.Max := 100;

  for i := 0 to 100 do
  begin
    ProgressBar1.Position := i;
    Sleep(3);
    Application.ProcessMessages;
  end;
begin
  if (Eusername.Text='') or (Epassword.Text='')then
  begin
     Application.MessageBox('Username dan Password Harus diisi
!!!!','Konfirmasi',MB_OK or
     MB_ICONERROR);
      if EUsername.Text='' then
        EUsername.SetFocus
        else
```

```
      Epassword.SetFocus;
  end
   else
  begin
   if (Eusername.Text='admin') then
    begin
     if (Epassword.Text='admin') then
      begin
      FLogin.Hide;
      Fadmin.LAdmin.Caption:='Admin';
      Fadmin.show;
      end
     else
        begin
          Application.MessageBox('Maaf, Password Yang anda masukan Tidak cocok
!!!!','Konfirmasi',MB_OK or
          MB_ICONERROR);
          EPassword.SetFocus;
        end;
    end
   else
   begin
    DM.TLogin.First;
    A:
    if EUsername.Text=DM.TLoginUsername.Value then
     begin
     If Epassword.Text=DM.TloginPassword.Value then
    begin
      FLogin.Hide;
      FUser.LUser.Caption:=DM.TLoginUsername.Value;
      FUser.Show;
     end else
     begin
       Application.MessageBox('Maaf, Password Yang anda masukan tidak cocok
!!!!','Konfirmasi',MB_OK or
    MB_ICONERROR);
     Epassword.SetFocus;
     end;
    end
    else
    begin
   if DM.TLogin.Eof then
     begin
```

```
        Application.MessageBox('Maaf, Username Yang anda masukan tidak Tidak
terdaftar !!!!','Konfirmasi',MB_OK or
        MB_ICONERROR);
        EUsername.setfocus;
       end
     else
       begin
       DM.TLogin.Next;
       goto A;
       end;
    end;
   end;
  end;
end;
end;
procedure TFLogin.BitBtn2Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
   FLogin.Close;
     exit;
end;
   end;
end.
```

PENGELUARAN

```
unit UPengeluaran;

interface

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

type
  TFPengeluaran = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    DBText1: TDBText;
    Panel2: TPanel;
```

```
    DBEdit1: TDBEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FPengeluaran: TFPengeluaran;

implementation

uses UDM;

{$R *.dfm}

procedure TFPengeluaran.BitBtn1Click(Sender: TObject);
begin
DM.TGuru.Post;
Close;
end;

procedure TFPengeluaran.BitBtn2Click(Sender: TObject);
begin
if messageDlg('Pengisian data dibatalkan ?',mtConfirmation,
            [mbYes,mbNo],0)=mrYes then
begin
DM.TGuru.Cancel;
Close;
end;
   end;
end.
```

TAMBAH USER

```
unit UTambahUser;

interface

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

type
  TFTambahUser = class(TForm)
    Panel1: TPanel;
    Label1: TLabel;
    Label2: TLabel;
    DBEdit1: TDBEdit;
    DBEdit2: TDBEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    Panel2: TPanel;
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FTambahUser: TFTambahUser;

implementation

uses UDM;

{$R *.dfm}

procedure TFTambahUser.BitBtn1Click(Sender: TObject);
begin
DM.TLoginUsername.Value:=Trim(DM.TLoginUsername.Value);
  if DM.TLoginUsername.IsNull then
  begin
    ShowMessage('Username Tidak Boleh Kosong !!!');
```

```
    dbEDIT1.SetFocus;
    Exit;
    end;
    begin
 DM.TLoginPassword.Value:=Trim(DM.TLoginPassword.Value);
  if DM.TLoginPassword.IsNull then
  begin
    ShowMessage('Password Tidak Boleh Kosong !!!');
    dbEDIT2.SetFocus;
    Exit;
    end;
    DM.TLogin.Post;
    close;
end;
  end;

procedure TFTambahUser.BitBtn2Click(Sender: TObject);
begin
if MessageDlg('Pengisisan data
dibatalkan?',mtConfirmation,[mbYes,mbNo],0)=mrYes then
  begin
  dm.TLogin.Cancel;
  close;
  end;
    end;
end.
```

VALIDASI USER

```
unit UUser;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, Menus, Buttons, jpeg;

type
  TFUser = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
```

```
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    MainMenu1: TMainMenu;
    Edit1: TMenuItem;
    ambahUser1: TMenuItem;
    EditUser1: TMenuItem;
    HapusUser1: TMenuItem;
    Out1: TMenuItem;
    Logout1: TMenuItem;
    Shape1: TShape;
    Label1: TLabel;
    LUser: TLabel;
    procedure SpeedButton1Click(Sender: TObject);
    procedure LaporanKasUmum1Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure Exit1Click(Sender: TObject);
    procedure Logout1Click(Sender: TObject);
    procedure LaporanGajiGuru1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure HapusUser1Click(Sender: TObject);
    procedure EditUser1Click(Sender: TObject);
    procedure ambahUser1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  FUser: TFUser;

implementation

uses UKasUmum, ULapKasUmum, UGajiGuru, ULogin, ULapGajiGuru, UGuru;

{$R *.dfm}

procedure TFUser.SpeedButton1Click(Sender: TObject);
begin
FKasUmum.ShowModal;
end;
```

```
procedure TFUser.LaporanKasUmum1Click(Sender: TObject);
begin
LapKasUmum.Preview;
end;

procedure TFUser.SpeedButton3Click(Sender: TObject);
begin
FGajiGuru.Showmodal;
end;

procedure TFUser.SpeedButton4Click(Sender: TObject);
begin
  if Application.MessageBox('Anda ingin Log Out ?','Konfimasi',MB_YESNO or
MB_ICONQUESTION)=id_Yes then
  begin
  FUser.Hide;
  FLogin.Show;
  FLogin.EUsername.Clear;
  Flogin.EPassword.Clear;
  Flogin.EUsername.SetFocus;
  end
  else
  begin
  Focused;
  refresh;
  end;
   end;

procedure TFUser.SpeedButton5Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
begin
    FUser.close;
      exit;
end;
  end;

procedure TFUser.Exit1Click(Sender: TObject);
begin
if MessageDlg('Anda Yakin Ingin Keluar ?',mtConfirmation,
[mbYes,mbNo],0)=mrYes then
```

```
begin
   FUser.close;
     exit;
end;
  end;


procedure TFUser.Logout1Click(Sender: TObject);
begin
  if Application.MessageBox('Anda ingin Log Out ?','Konfimasi',MB_YESNO or
MB_ICONQUESTION)=id_Yes then
  begin
  FUser.Hide;
  FLogin.Show;
  FLogin.EUsername.Clear;
  Flogin.EPassword.Clear;
  Flogin.EUsername.SetFocus;
  end
  else
  begin
  Focused;
  refresh;
  end;
    end;
procedure TFUser.LaporanGajiGuru1Click(Sender: TObject);
begin
LapGajiGuru.Preview;
end;

procedure TFUser.SpeedButton2Click(Sender: TObject);
begin
FGuru.ShowModal;
end;

procedure TFUser.HapusUser1Click(Sender: TObject);
begin
FGajiGuru.Showmodal;
end;

procedure TFUser.EditUser1Click(Sender: TObject);
begin
FGuru.Showmodal;
end;
```

```
procedure TFUser.ambahUser1Click(Sender: TObject);
begin
FKasUmum.Showmodal;
end;

end.
```

3.7.4. Implementasi Antar Muka

1. Implementasi Login

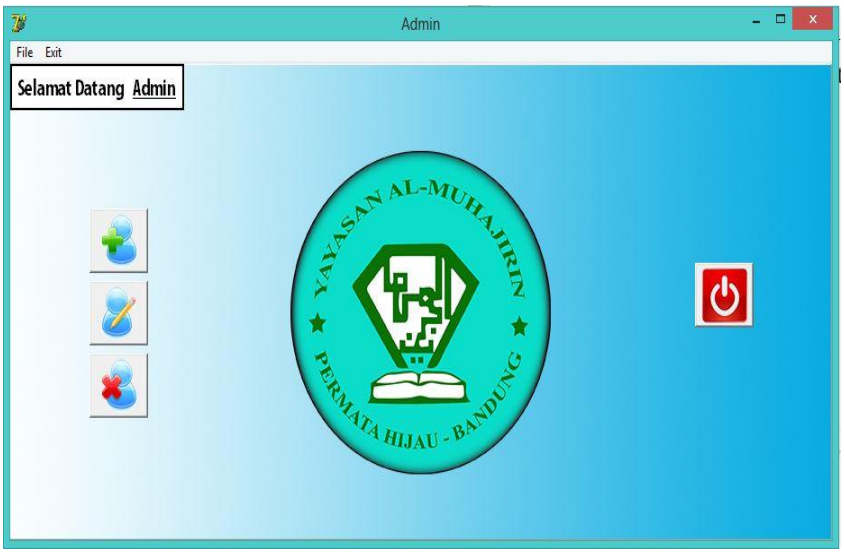
Implementasi halam login digunakan untuk masuk kedalam sistem atau aplikasi. Tampilan ini merupakan tampilan utama ketika program dijalankan, pengguna harus mengisi *username* dan *password* sebelum masuk ke menu utama. Jika *username* atau *password* tidak sesuai, maka akan muncul pesan dihalaman tersebut.



Gambar 3.26 Implementasi Antarmuka Login

2. Implementasi Menu Admin

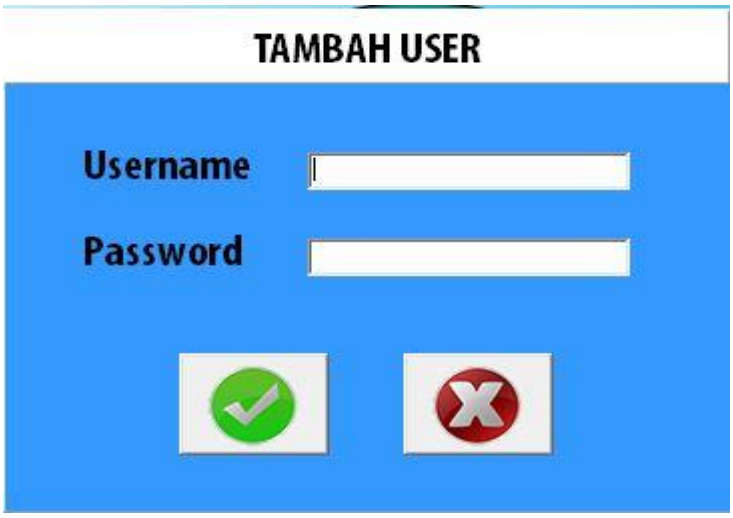
Implementasi berikut merupakan halaman menu admin setelah admin melakukan login, di halaman ini admin dapat memilih menu tambah *user*, edit *user* dan hapus *user* serta keluar.



Gambar 3.27 Implementasi Antarmuka Menu Admin

3. Implementasi Antarmuka Tambah User

Implementasi halaman ini adalah menu halaman tambah data User yang akan memproses penambahan data User dengan mengisi data yang ada pada halaman ini. Jika ada data yang tidak terisi maka akan muncul pesan pada halaman ini untuk tidak mengkosongkan form dan data tidak langsung tersimpan, jika ingin membatalkan pengisian terdapat tombol batal untuk pembatalan.



Gambar 3.28 Implementasi Antarmuka Tambah User

4. Implementasi Antarmuka Menu Utama

Implementasi halaman berikut merupakan halaman tampilan menu utama dari petugas ketika petugas telah melakukan login. Pada halaman ini petugas dapat memilihi menu Kas Umum, guru, dan Gaji Guru.

5. Implementasi Antarmuka Pengolahan Data Kas Umum

Implementasi halaman ini adalah menu halaman dari pengolahan data Kas Umum, pada halaman ini ada beberapa menu yang bisa dioperasikan seperti tambah data, *edit* data, hapus data, kredit, dan laporan .



Gambar 3.29 Implementasi Antarmuka Pengolahan Data Kas Umum

6. Implementasi Antarmuka Tambah Kas Umum

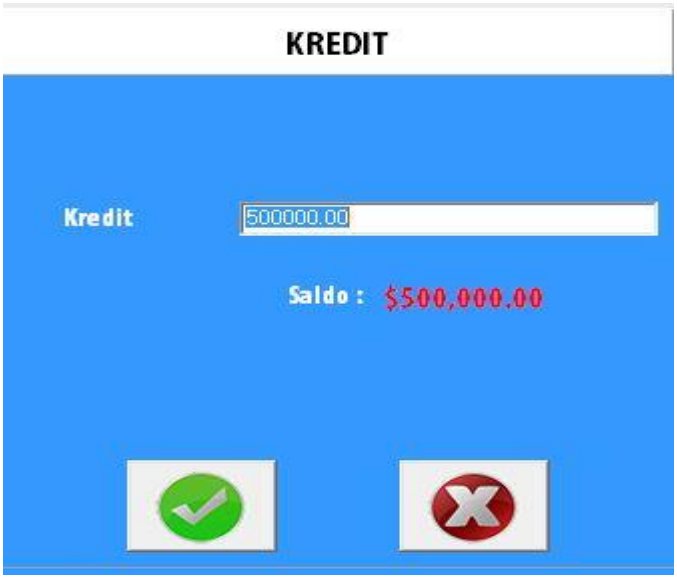
Implementasi halaman ini adalah menu halaman tambah data Kas Umum yang akan memproses penambahan data Kas Umum dengan mengisi data yang ada pada halaman ini. Jika ada data yang tidak terisi maka akan muncul pesan pada halaman ini untuk tidak mengkosongkan form dan data tidak langsung tersimpan, jika ingin membatalkan pengisian terdapat tombol batal untuk pembatalan.



Gambar 3.30 Implementasi Antarmuka Tambah Kas Umum

7. Implementasi Antarmuka Kredit Kas Umum

Implementasi halaman ini adalah menu halaman kredit yang ada pada Kas Umum. Halaman ini akan memproses jumlah kredit pada Kas Umum dengan mengisi data yang ada pada halaman ini. Jika ada data yang tidak terisi maka akan muncul pesan pada halaman ini untuk tidak mengkosongkan form dan data tidak langsung tersimpan, jika ingin membatalkan pengisian terdapat tombol batal untuk pembatalan.



Gambar 3.31 Implementasi Antarmuka Kredit Kas Umum

8. Implementasi Antarmuka Pengolahan Data Guru

Implementasi halaman ini adalah menu halaman dari pengolahan data Guru, pada halaman ini ada beberapa menu yang bisa dioperasikan seperti tambah data, *edit* data, hapus data.

9. Implementasi Pengolahan Data Gaji Guru

Implementasi halaman ini adalah menu halaman dari pengolahan data Gaji Guru, pada halaman ini ada beberapa menu yang bisa dioperasikan seperti tambah data, *edit* data, hapus data, pengeluaran, dan laporan.



Gambar 3.32 Implementasi Antarmuka Gaji Guru

10. Implementasi Antarmuka Tambah Gaji Guru

Implementasi halaman ini adalah menu halaman tambah data Gaji Guru yang akan memproses penambahan data Gaji Guru dengan mengisi data yang ada pada halaman ini. Jika ada data yang tidak terisi maka akan muncul pesan pada halaman ini untuk tidak mengkosongkan form dan data tidak langsung tersimpan, jika ingin membatalkan pengisian terdapat tombol batal untuk pembatalan.



Gambar 3.33 Implementasi Antarmuka Tambah Gaji Guru

11. Implementasi Antarmuka Pengeluaran Gaji Guru

Implementasi halaman ini adalah menu halaman pengeluaran yang ada pada menu Gaji Guru. Halaman ini akan memproses jumlah pengeluaran pada Gaji Guru dengan mengisi data yang ada pada halaman ini. Jika ada data yang tidak terisi maka akan muncul pesan pada halaman ini untuk tidak mengkosongkan form dan data tidak langsung tersimpan, jika ingin membatalkan pengisian terdapat tombol batal untuk pembatalan.



Gambar 3. 4 Implementasi Antarmuka Pengeluaran Gaji Guru

3.8 Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Pengujian perangkat lunak ini merupakan tahapan untuk menemukan kesalahan-kesalahan dan kekurangan-kekurangan pada perangkat lunak yang dibangun sehingga bisa diketahui apakah perangkat lunak tersebut telah memenuhi kreteria sesuai dengan tujuan atau tidak. Adapun metode pengujian yang digunakan pada perangkat lunak ini adalah metode pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Metode pengujian *black box* ini terdiri dari dua tahapan pengujian, yaitu tahapan pengujian alpha dan tahapan pengujian beta.

3.8.1 Pengujian Alpha

Pengujian Alpha merupakan pengujian fungsional diadakan di lingkungan pembangun oleh sekumpulan pengguna yang akan menggunakan perangkat lunaknya. Pihak pembangun mendampingi serta mencatat kesalahan-kesalahan maupun permasalahan yang dirasakan oleh pengguna. Pengujian alpha yang dilakukan pada Aplikasi ini dibagi menjadi tiga bagian, yaitu skenario pengujian, kasus dan hasil pengujian, dan kesimpulan pengujian.

3.8.1.1 Skenario Pengujian

Skenario pengujian memaparkan urutan dan hal yang diuji pada pengujian yang dilakukan pada Aplikasi ini. Adapun skenario pengujian yang dilakukan dapat dilihat pada Tabel 3.

Tabel 3.10 Rencana Pengujian

Item Pengujian	**Detail Pengujian**	**Jenis Pengujian**
Login	Username	*Black box*
	Password	*Black box*
Tambah Data Kas Umum	Id_Kas_Umum	*Black box*
	Tanggal	*Black box*
	Uraian	*Black box*

	Kredit	*Black box*
Edit Data Kas Umum	Id_Kas_Umum	*Black box*
	Tanggal	*Black box*
	Uraian	*Black box*
Hapus Data Kas Umum	Id_Kas_Umum	*Black box*
	Tanggal	*Black box*
	Uraian	*Black box*
	Kredit	*Black box*
Tambah Data Guru	Nip_Guru	*Black box*
	Nama	*Black box*
	Alamat	*Black box*
Edit Data Guru	Nip_Guru	*Black box*
	Nama	*Black box*
	Alamat	*Black box*
Hapus Data Guru	Nip_Guru	*Black box*
	Nama	*Black box*
	Alamat	*Black box*
Tambah Gaji Guru	Id_Gaji_Guru	*Black box*
	Tanggal	*Black box*
	Nama	*Black box*
	Gaji	*Black box*
	Pengeluaran	*Black box*
Edit Data Gaji Guru	Id_Gaji_Guru	*Black box*
	Tanggal	*Black box*
	Nama	*Black box*
	Gaji	*Black box*
Hapus Data Gaji Guru	Id_Gaji_Guru	*Black box*
	Tanggal	*Black box*
	Nama	*Black box*
	Gaji	*Black box*
	Pengeluaran	*Black box*

3.8.1.2 Kasus Dan Hasil Pengujian Data

Kasus dan hasil pengujian berisi pemaparan dari rencana pengujian yang telah disusun pada skenario pengujian. Pengujian ini dilakukan secara *black box*

dengan hanya memperhatikan masukan ke dalam sistem dan keluaran dari masukan tersebut. Berikut ini pemaparan dari setiap butir pengujian yang terdapat pada skenario pengujian :

a. Pengujian Login Admin

Tabel 3.11 Pengujian Login Admin

Kasus Data Hasil Uji (data Normal)	
Data Masukan	*username* : admin, *Password*:admin
Yang Diharapkan	Data yang dimasukkan benar lalu masuk ke tampilan menu Admin
Pengamatan	Menampilkan menu utama pengolahan data User
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	*Username* dan *Password* salah atau kosong
Yang Diharapkan	Apabila salah atau kosong dalam memasukan username dan password maka tidak dapat login dan Muncul pesan kesalahan gagal login
Pengamatan	Muncul pesan gagal login
Kesimpulan	Diterima

b. Pengujian Login Petugas

Tabel 3.12 Pengujian Login Petugas

Kasus Data Hasil Uji (data Normal)	
Data Masukan	*username* : petugas, *Password*:petugas
Yang Diharapkan	Data yang dimasukkan benar lalu masuk ke tampilan menu data petugas.
Pengamatan	Menampilkan menu utama pengolahan data petugas
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	*Username* dan *Password* salah atau kosong
Yang Diharapkan	Apabila salah atau kosong dalam memasukan username dan password maka tidak dapat login dan Muncul pesan kesalahan gagal login

Pengamatan	Muncul pesan gagal login
Kesimpulan	Diterima

c. Pengujian Tambah Kas Umum

Tabel 3.13 Pengujian Tambah Kas Umum

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Tanggal, Uraian , Kredit.
Yang Diharapkan	Menampilkan form tambah data kas umum
Pengamatan	Menampilkan form penambahan kas umum
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Tanggal, Uraian , dan Kredit kosong.
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian data dibatalkan?" jika proses dibatalkan.
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

d. Pengujian Edit Data Kas Umum

Tabel 3.14 Pengujian Edit Data Kas Umum

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Tanggal, Uraian , Kredit.
Yang Diharapkan	Data kas umum yang diubah muncul sesuai yang diharapkan
Pengamatan	Menampilkan form pengeditan data kas umum
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Tanggal, Uraian , dan Kredit kosong.
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian

	data dibatalkan?" jika proses dibatalkan.
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

e. Pengujian Hapus Data Kas Umum

Tabel 3.15 Pengujian Hapus Data Kas Umum

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Tanggal, Uraian , Kredit.
Yang Diharapkan	Menghapus data kas umum dari form pengolahan data kas umum
Pengamatan	Menghapus data kas umum
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Tanggal, Uraian , dan Kredit .
Yang Diharapkan	Menampilkan pesan kesalahan
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

f. Pengujian Tambah Data Guru

Tabel 3.16 Pengujian Tambah Data Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nip_Guru, Nama, Alamat
Yang Diharapkan	Menampilkan form tambah data guru
Pengamatan	Menampilkan form penambahan guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nip_Guru, Nama, Alamat
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian data dibatalkan?" jika proses dibatalkan.

Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

g. Pengujian Edit Data Guru

Tabel 3.17 Pengujian Edit Data Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nip_Guru, Nama, Alamat
Yang Diharapkan	Data guru yang diubah muncul sesuai yang diharapkan
Pengamatan	Menampilkan form pengeditan data guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nip_Guru, Nama, Alamat
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian data dibatalkan?" jika proses dibatalkan.
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

h. Pengujian Hapus Data Guru

Tabel 3.18 Pengujian Hapus Data Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nip_Guru, Nama, Alamat
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian data dibatalkan?" jika proses dibatalkan.
Pengamatan	Menghapus data guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nip_Guru, Nama, Alamat

Yang Diharapkan	Menampilkan pesan kesalahan
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

i. Pengujian Tambah Data Gaji Guru

Tabel 3.19 Pengujian Tambah Data Gaji Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Menampilkan form tambah data gaji guru
Pengamatan	Menampilkan form penambahan gaji guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan "data tidak boleh kosong!" atau "pengisian data dibatalkan?" jika proses dibatalkan.
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

j. Pengujian Edit Data Gaji Guru

Tabel 3.20 Pengujian Edit data Gaji Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Data gaji guru yang diubah muncul sesuai yang diharapkan
Pengamatan	Menampilkan form pengeditan data gaji guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau

	pesan “data tidak boleh kosong!” atau “pengisian data dibatalkan?” jika proses dibatalkan.
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

k. Pengujian Hapus Data Gaji Guru

Tabel 3.21 Pengujian Hapus Data Gaji Guru

Kasus Data Hasil Uji (data Normal)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Menampilkan pesan kesalahan bahwa data tidak boleh kosong jika menekan tombol simpan atau pesan “data tidak boleh kosong!” atau “pengisian data dibatalkan?” jika proses dibatalkan.
Pengamatan	Menghapus data gaji guru
Kesimpulan	Diterima
Kasus Data Dan Hasil Uji (Data Salah)	
Data Masukan	Nama, Gaji, Pengeluaran
Yang Diharapkan	Menampilkan pesan kesalahan
Pengamatan	Menampilkan pesan kesalahan
Kesimpulan	Diterima

3.8.1.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian *alpha* yang telah dilakukan dapat ditarik kesimpulan bahwa aplikasi sudah berjalan cukup baik sesuai dengan yang diharapkan, baik itu dari segi validasi maupun proses penanganan kesalahan.

3.8.2 Pengujian Beta

Pengujian beta dilakukan di lingkungan pengguna tanpa kehadiran pihak pembangun aplikasi. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya. Pengguna melakukan penilaian terhadap perangkat lunak dengan melakukan wawancara kepada petugas yang menggunakan sistem ini.
Berikut hasil pengujian *betha* :

3.1. Apakah aplikasi ini mempermudah dalam penyimpanan data?
Jawab : YA

3.2. Apakah aplikasi ini mudah untuk dipahami?
Jawab : YA

3.3. Apakah aplikasi ini membantu para pegawai dalam pengolahan data Kas Umum, Data Guru, dan Data Gaji Guru?
Jawab : YA

3.4. Apakah aplikasi ini perlu dikembangkan lagi?
Jawab : YA

3.8.2.1 Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil wawancara diatas didapatkan dari pengujian *betha*, dengan mewawancarai petugas yang menggunakan Aplikasi Sistem Informasi di Yayasan Raudhatul Alhfal Al-Muhajirin maka dapat disimpulkan bahwa pembangunan Aplikasi sudah cukup mengatasi masalah yang ada, mempermudah penyimpanan data dan pengolahan data, namun aplikasi aplikasi ini masih perlu untuk dikembangkan lagi.