

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Alat dan Bahan Penelitian**

Dalam penelitian dibutuhkan beberapa alat dan bahan untuk mendukung berjalannya perancangan dan implementasi *aplikasi desktop*.

##### **3.1.2 Alat**

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

- a. Perangkat Keras (*Hardware*)
  1. *Personal Computer (PC)* atau laptop.
    - Intel® Core™ i7 *processor*,
    - 8.00 GB RAM,
    - 64 bit *Operating System*,
    - Windows 10 *Pro*.
  2. *Printer* dokumen untuk mencetak laporan.
- b. Perangkat Lunak (*Software*)
  1. *Visio 2017*
  2. *Netbeans IDE 8.0.2*
  3. *SQL Server 2017*

##### **3.1.3 Bahan**

Dalam penelitian ini bahan yang dibutuhkan adalah :

1. Data yang diperoleh melalui studi literatur berdasarkan penelitian sebelumnya yang masih memiliki keterkaitan dengan *aplikasi desktop* yang dikembangkan. Dari data yang diperoleh, maka didapatkan dari aplikasi pada saat pengembangan aplikasi.

2. Model perekapan keuangan Program Studi Teknologi Informasi Universitas Muhammadiyah Yogyakarta terdahulu.

## **3.2 Teknik Pengumpulan Data**

### **3.2.1 Wawancara**

Wawancara merupakan percakapan yang bertujuan untuk mendapatkan data. Percakapan dilakukan oleh peneliti yang mengajukan pertanyaan dan yang diwawancarai memberikan jawaban atas pertanyaan itu.

Proses wawancara ini dilakukan secara langsung kepada dosen pembimbing 1, Bapak Asroni, S.T., M.Eng dan dosen pembimbing dosen 2, Ibu Aprilia Kurnianti, S.T., M.Eng.. Hal-hal yang dibahas adalah mengenai metode perekapan data keuangan, arsitektur, teknologi yang digunakan serta proses transaksi pada aplikasi desktop rekap keuangan.

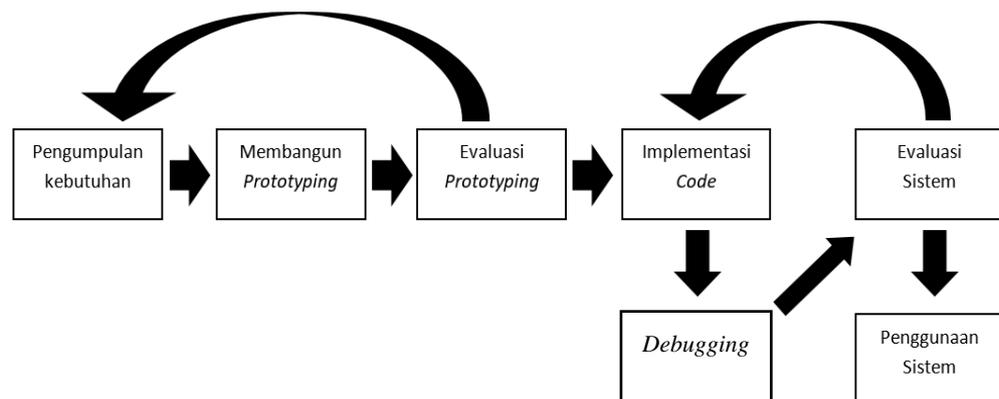
### **3.2.2 Alur Penelitian**

Untuk perancangan dan pembuatan sistem, peneliti menggunakan metode *SDLC* (*Software Development Life Cycle*). Model ini dipilih untuk melakukan proses produksi sistem dan juga proses *maintenance software*.

Model SDLC yang peneliti pakai adalah model *Prototyping*. *Prototyping* perangkat lunak (*software prototyping*) atau siklus hidup menggunakan *prototyping* (*life cycle using prototyping*) adalah salah satu metode siklus hidup sistem yang didasarkan pada konsep model bekerja (*working model*). Model pengembangan *prototyping* mempunyai beberapa kelebihan, diantaranya :

1. Adanya komunikasi pengembang dan *user*. Peran *user* meningkat, karena *user* dapat memberikan masukan setiap saat.
2. *Customize*, maksudnya sistem dibangun dapat berulang kali di evaluasi sehingga perubahan *requirement* dapat ditambah maupun diubah seiring pengembangan dilakukan.

- Implementasi sangat mudah, maksudnya adalah *user* dapat mengenal aplikasi yang akan dikembangkan dan tidak merasa asing saat *user* menggunakannya.



**Gambar 3.1** *Prototype*

Ada beberapa tahapan-tahapan dalam metode *prototyping*. Tahapan-tahapan metode *prototyping* seperti yang terlihat di gambar 3.1. Penjelasan gambar 3.1 adalah sebagai berikut :

1. Pengumpulan Kebutuhan

Tahapan ini adalah tahap awal dimana pengembang dan *user* bersama-sama mendefinisikan format perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun *Prototyping*

Proses ini adalah perancangan sistem (seperti perancangan *database* dan *use case*) dan design program sementara yang berfokus pada penyajian (contohnya adalah tampilan).

3. Evaluasi *Prototyping*

Evaluasi ini dilakukan oleh *user* apakah *prototyping* yang sudah dibuat sudah sesuai dengan kebutuhan dan permintaan. Jika sudah sesuai maka dilanjutkan pada langkah selanjutnya. Jika belum sesuai maka direvisi dengan mengulang pada langkah 1, 2, dan 3.

#### 4. Implementasi *Code*

Setelah disepakati *prototyping* yang sudah dibuat maka kemudian diterjemahkan kedalam bahasa pemograman.

#### 5. *Debugging*

Saat aplikasi sudah selesai, untuk menjadi layak dipakai maka terlebih dahulu harus diuji. Pengujian ini dilakukan agar aplikasi terhindar dari *error* atau *bug*.

#### 6. Evaluasi Sistem

*User* akan mengevaluasi apakah sistem yang sudah selesai dibangun sesuai dengan yang diharapkan. Jika iya lanjut ke langkah selanjutnya, namun jika tidak harus mengulangi pada langkah 4 dan 5.

#### 7. Menggunakan Sistem

Aplikasi yang telah diuji dan disetujui oleh *user* maka aplikasi sudah siap untuk digunakan.

### 3.3 Analisis Penelitian

#### 3.3.1 Analisis Pengguna

Aplikasi desktop yang dikembangkan dapat digunakan oleh pengguna yaitu admin. Admin memiliki hak akses penuh terhadap aplikasi. Admin dapat melakukan pengolahan data keuangan untuk selanjutnya dapat dijadikan laporan.

#### 3.3.2 Analisis Kebutuhan

Analisis kebutuhan didapat dari diskusi bersama dosen pembimbing proyek ini. Berikut ini adalah hasil analisis kebutuhan pada aplikasi desktop rekap keuangan :

1. Terdapat *Login*, agar hanya admin yang bisa mengakses aplikasi .
2. Terdapat *master data kategori*, untuk mengisi kategori input sesuai kebutuhan.
3. Terdapat *master data tahun ajaran* , untuk mengisi tahun ajaran sesuai kebutuhan.

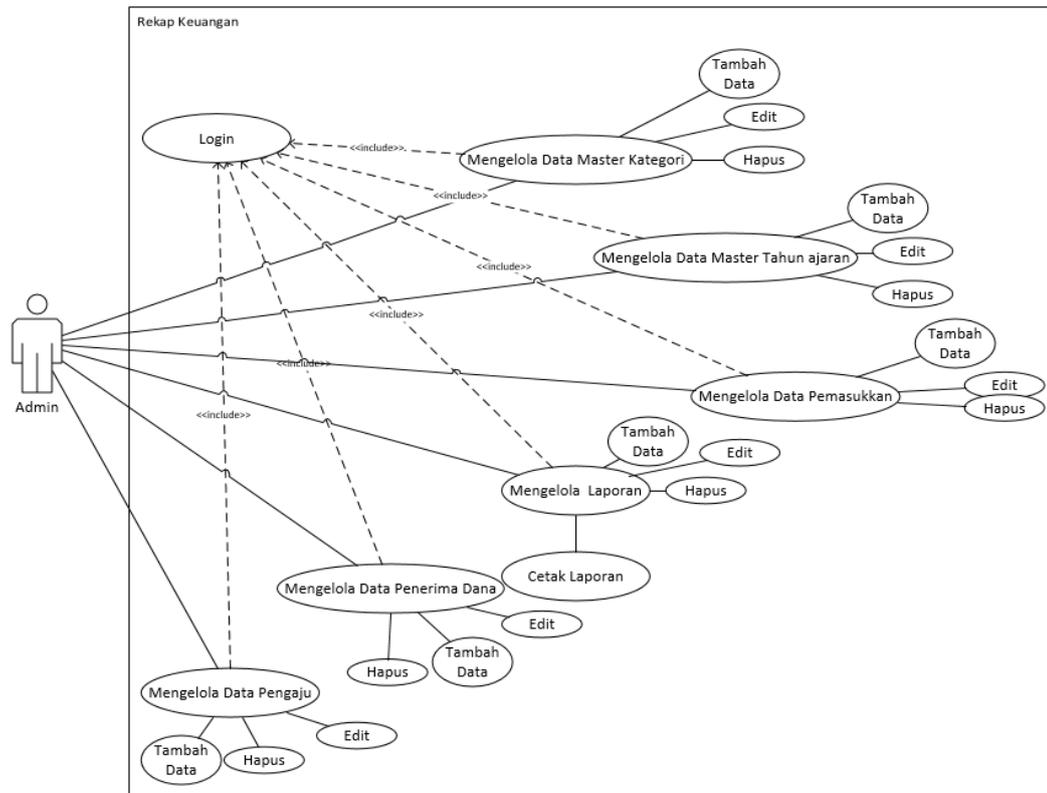
4. Terdapat *data pemasukkan* , untuk mengisi data pemasukkan sesuai kebutuhan.
5. Terdapat *Rekapitulasi* , untuk memasukkan seluruh data pengeluaran yang ada.
6. Terdapat *master data pengaju* , untuk mengisi data pengaju yang akan mengajukan program.
7. Terdapat *master data penerima dana* , untuk mengisi data penerima dana yang masuk ke prodi.
8. Terdapat fitur *cetak*, untuk mencetak hasil pengolahan data dalam bentuk pdf.

### **3.4 Rancangan Sistem dan Basis Data**

Sebelum membangun *aplikasi desktop Rekapitulasi Keuangan Progam Studi Teknologi Informasi* ini, terlebih dahulu dituangkan dalam rancangan diagram – diagram. Diagram ini sebagai patokan jalannya aplikasi, kebutuhan, dan *database*. Metode yang digunakan dalam perancangan *aplikasi desktop* adalah *Unified Modeling Language (UML)*. Metode UML yang dipakai dalam pengembangan *aplikasi web* yaitu model diagram *Use Case Diagram*, *ER Diagram*, *Class Diagram*, dan *Activity Diagram*.

#### **3.4.1 Use Case Diagram**

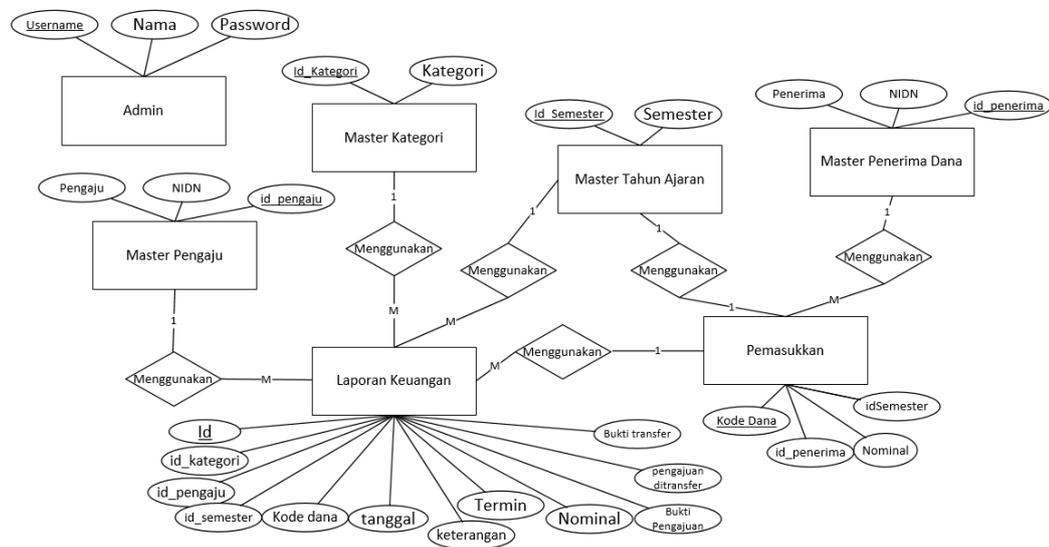
Gambaran *Use Case Diagram* yang digunakan dalam aplikasi web dapat dilihat pada gambar 3.2.



**Gambar 3. 2** Use Case Diagram

Gambar 3.2 diatas menunjukkan hubungan antara *actor* dan *use case*. Pada gambar diatas menjelaskan pada *aplikasi desktop*, hanya admin yang bisa mengakses aplikasi ini. Admin memiliki hak akses penuh untuk mengolah semua data yang ada, baik tambah data master kategori, tambah data master tahun ajaran, maupun input laporan rekapitulasi dan lain - lain. Dalam kegiatannya, pengelola harus terlebih dahulu melakukan *login*.

### 3.4.2 ER Diagram



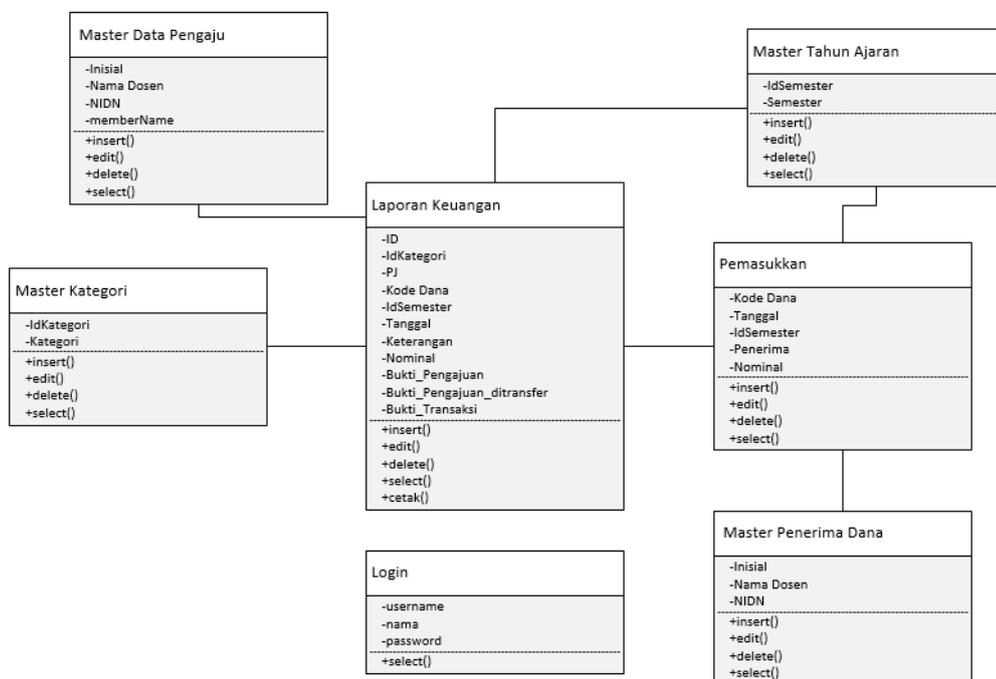
**Gambar 3. 3** ER Diagram

Gambar 3.3 diatas adalah *ER Diagram* yang menunjukkan entitas dengan atribut dan relasinya. Entitas ini yang nantinya akan menjadi rancangan tabel pada *database* program. Entitas diatas memiliki relasi dari satu dengan yang lain. Beberapa relasi akan dijelaskan sebagai berikut:

1. *Master Kategori* memiliki relasi *one-to-many* dengan *Rekapitulasi*. Satu data kategori dapat digunakan untuk membuat banyak rincian pengeluaran.
2. *Master Tahun Ajaran* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu data tahun ajaran dapat digunakan untuk membuat banyak rincian pengeluaran.
3. *Pemasukkan* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu data pemasukkan dapat dirinci lagi akan digunakan untuk keperluan program studi.
4. *Master Pengaju* memiliki relasi *one-to-many* dengan *Laporan Keuangan*. Satu pengaju dapat mengajukan banyak pengajuan. Pengajuan tersebut akan dicatat dalam rincian pengeluaran.

5. *Master Tahun Ajaran* memiliki relasi *one-to-one* dengan *Pemasukkan*. Satu data data tahun ajaran hanya bisa digunakan untuk membuat satu data pemasukkan.
6. *Master Penerima Dana* memiliki relasi *one-to-many* dengan *Pemasukkan*. Satu penerima dana bisa mendapatkan banyak pemasukkan.

### 3.4.3 Class Diagram



**Gambar 3. 4** Class Diagram

*Class diagram* diatas menunjukkan *class-class* yang ada didalam sistem. Setiap *class* memiliki metode yang hampir sama yaitu:

1. *Create* untuk menambah data ke database.
2. *Read* untuk membaca data dari database.
3. *Edit* untuk mengubah data yang sudah ada dalam database.
4. *Delete* untuk menghapus data dalam database.
5. *Cetak* untuk mencetak laporan.

6. *Search* untuk mencari data dalam database berdasarkan data tertentu.

7. *Login* berfungsi untuk masuk kedalam aplikasi.

### 3.4.4 Kamus Data Base

Berikut ini merupakan tabel dari kamus basis data (*database*) pada *aplikasi desktop Rekapitulasi Keuangan Program Studi Teknologi Informasi*.

**Tabel 3. 1** Kamus Data Admin

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Username	Varchar (10)	<i>Primary key</i>	Username admin
Nama	Varchar (50)	-	Nama lengkap admin
Password	Varchar (10)	-	Password admin

**Tabel 3. 2** Kamus Basis Data Master Kategori

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Id_Kategori	Varchar (4)	<i>Primary key</i>	Id Kategori
Kategori	Varchar (50)	<i>Unique</i>	Jenis Kategori

**Tabel 3. 3** Kamus Basis Dasta Master Tahun Ajaran

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Id_ta	Varchar (7)	<i>Primary key</i>	Id note
Semester	Varchar (5)	<i>Unique</i>	Semester

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
TA	Varchar (9)	-	Tahun ajaran

**Tabel 3. 4** Kamus Basis Data Laporan Keuangan

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
ID	Varchar (8)	<i>Primary key</i>	Id laporan
Id_Kategori	Varchar (4)	<i>Foreign key</i>	Jenis kategori
PJ	Varchar (4)	<i>Foreign key</i>	Penanggung jawab
Kode Dana	Varchar (6)	<i>Foreign key</i>	Kode dana pemasukkan
Id_ta	Varchar (7)	<i>Foreign key</i>	Tahun ajaran
Tanggal	Date	-	Tanggal pengisian laporan
Keterangan	Varchar (MAX)	-	Keterangan laporan
Termin	Varchar (8)	-	Periode laporan
Bukti_Pengajuan	Varchar(MAX)	-	Bukti Pengajuan

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Bukti_Pengajuan_Transfer	Varchar(MAX)	-	Bukti pengajuan telah di transfer
Bukti_Transaksi	Varchar(MAX)	-	Bukti transaksi
Nominal	Decimal (18,0)	-	Nominal pengeluaran

**Tabel 3. 5** Tabel Pemasukkan

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Id_pemasukkan	Varchar (6)	<i>Primary key</i>	Id data pemasukkan
Tanggal	date	-	Tanggal <i>input</i> pemasukkan
Id_ta	Varchar (7)	<i>Unique</i>	Menunjukkan untuk semester berapa dana akan digunakan
Penerima	Varchar (4)	-	Penerima dana
Nominal	Decimal (18,0)	-	Nominal dana

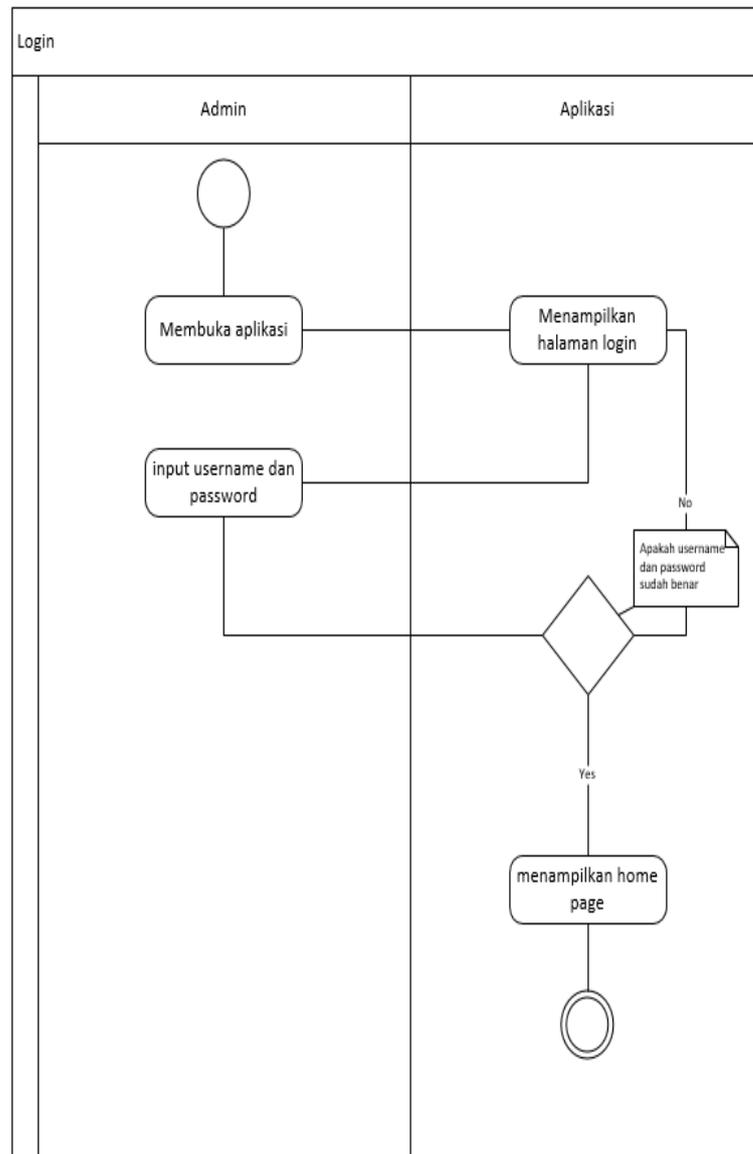
**Tabel 3. 6** Kamus Basis Data Pengaju

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Inisial	Varchar (10)	<i>Primary key</i>	Kode Dosen
Nama Dosen	Varchar (50)	-	Nama Dosen
NIDN	Varchar (15)	-	Nomor Induk Dosen

**Tabel 3. 7** Kamus Basis Data Penerima Dana Fakultas

<b>Nama Kolom</b>	<b>Tipe Data</b>	<b>Key</b>	<b>Keterangan</b>
Inisial	Varchar (4)	<i>Primary key</i>	Kode Dosen
Nama Dosen	Varchar (50)	-	Nama Dosen
NIDN	Varchar (12)	-	Nomor Induk Dosen

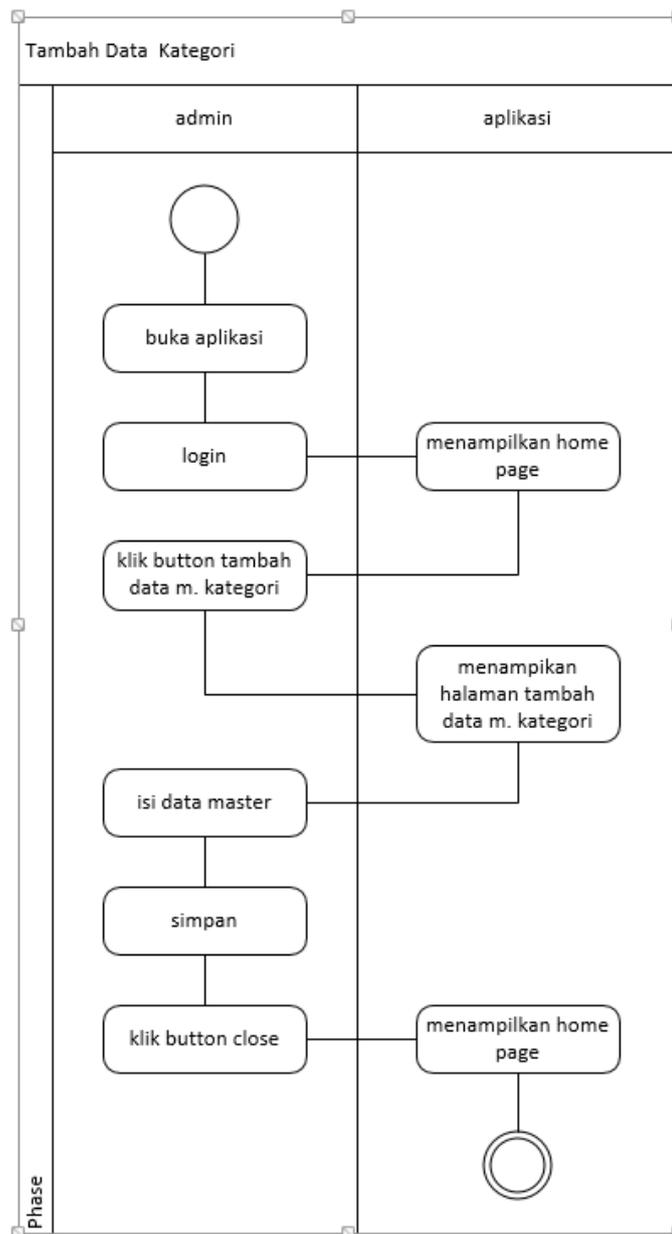
### 3.4.5 Activity Diagram Login



**Gambar 3. 5** Activity Diagram Login

Gambar 3.5 diatas menunjukkan *activity diagram* untuk login. Sebelum masuk ke *aplikasi desktop* admin harus melakukan login terlebih dahulu. Username dan password akan di identifikasi apakah sudah sesuai. Jika username dan password sesuai maka akan diteruskan ke *home page*. Jika salah akan di kembalikan ke form login.

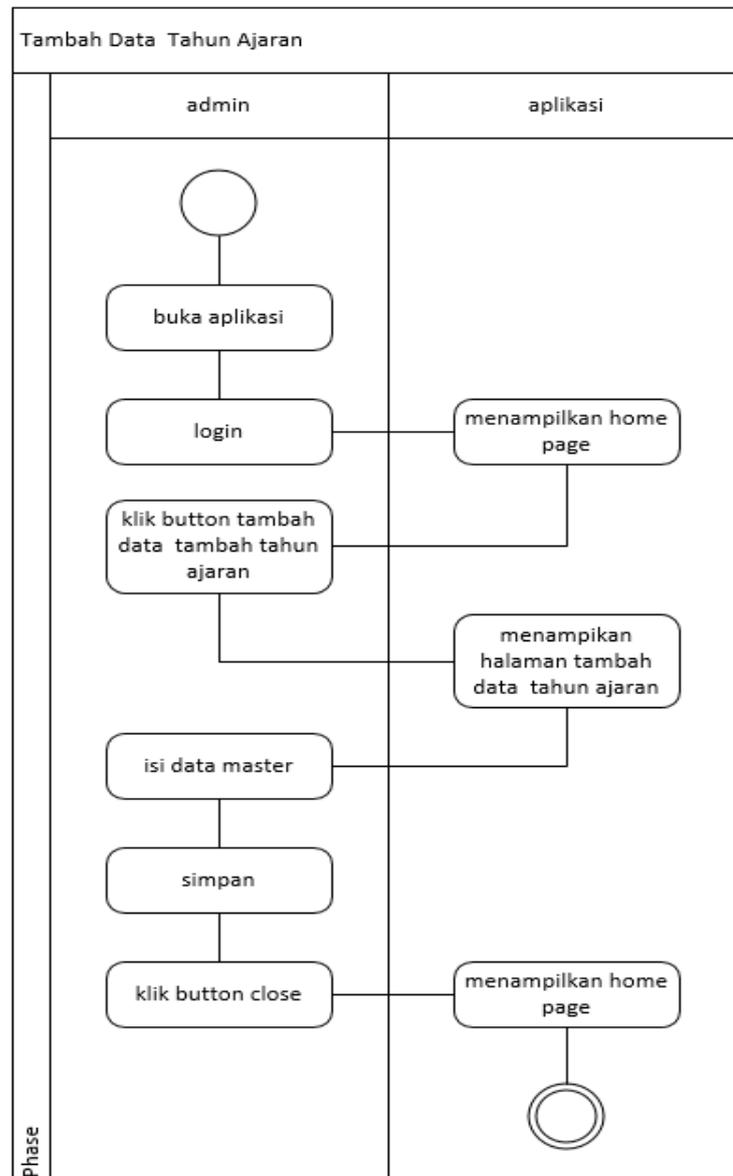
### 3.4.6 Activity Diagram Tambah Data Kategori



**Gambar 3. 6** Activity Diagram Tambah Data Master Kategori

Gambar 3.6 diatas menunjukkan *activity diagram* untuk tambah data master kategori. Data master kategori ini dibuat untuk nantinya di pakai di *form* laporan rekapitulasi. Admin harus mengisi *idKategori* dan *kategori* sesuai dengan kebutuhan.

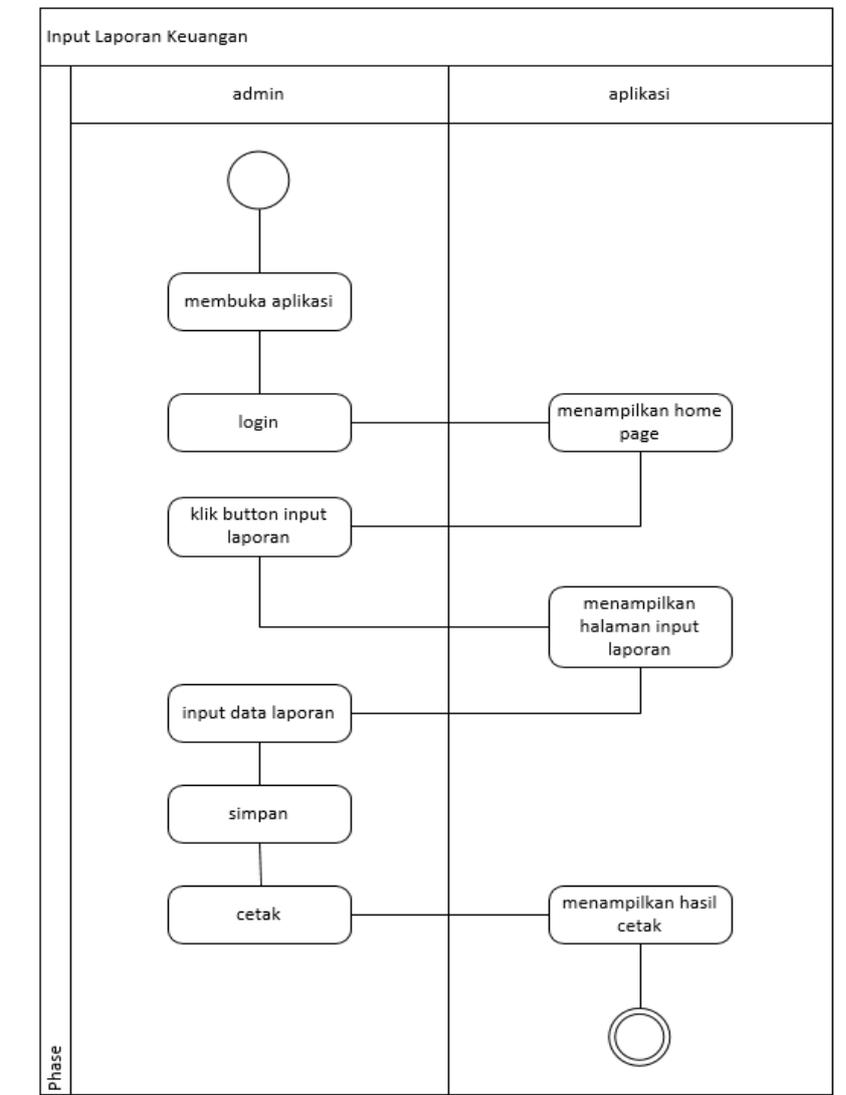
### 3.4.7 Activity Diagram Tambah Data Tahun Ajaran



**Gambar 3. 7** Activity Diagram Tambah Data Master Tahun Ajaran

Gambar 3.7 diatas menunjukkan *activity diagram* untuk tambah data master tahun ajaran. Fitur ini sama dengan data master kategori. Data master tahun ajaran ini nantinya akan di pakai di *form* laporan rekapitulasi sebagai salah satu inputan.

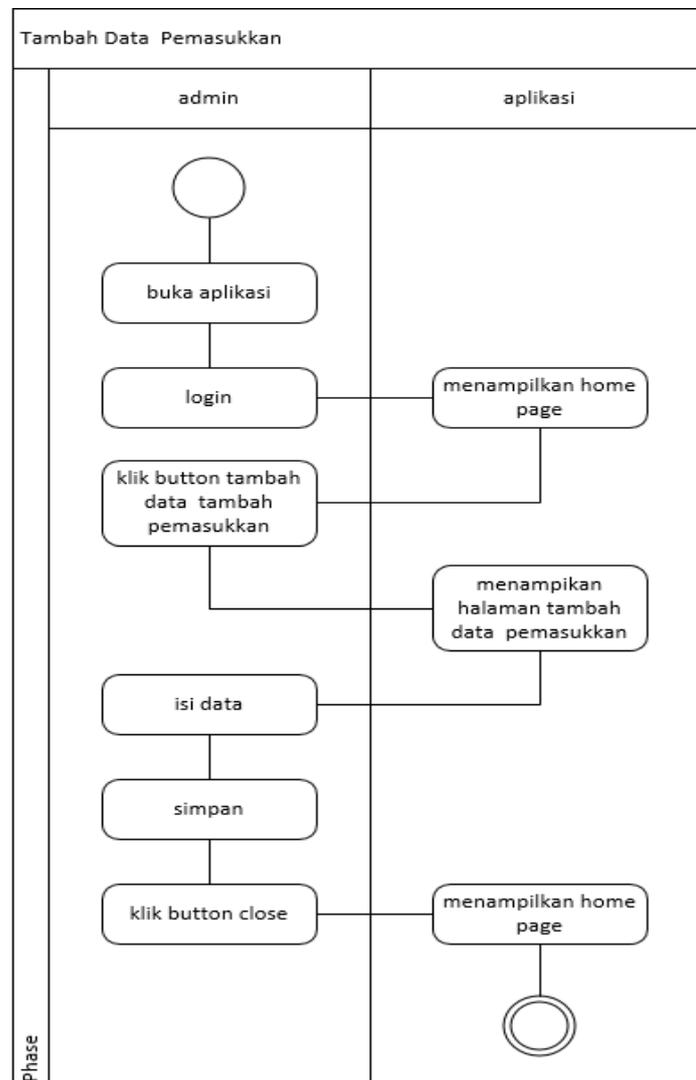
### 3.4.8 Activity Diagram Laporan Keuangan



**Gambar 3. 8** Activity Diagram Laporan Keuangan

Gambar 3.8 diatas menunjukkan *activity diagram* laporan rekapitulasi. Disini admin menginputkan laporan - laporan terkait keuangan. Di sini admin juga dapat mengedit data, menghapus data, *search* data, dan mencetak hasil laporan tersebut.

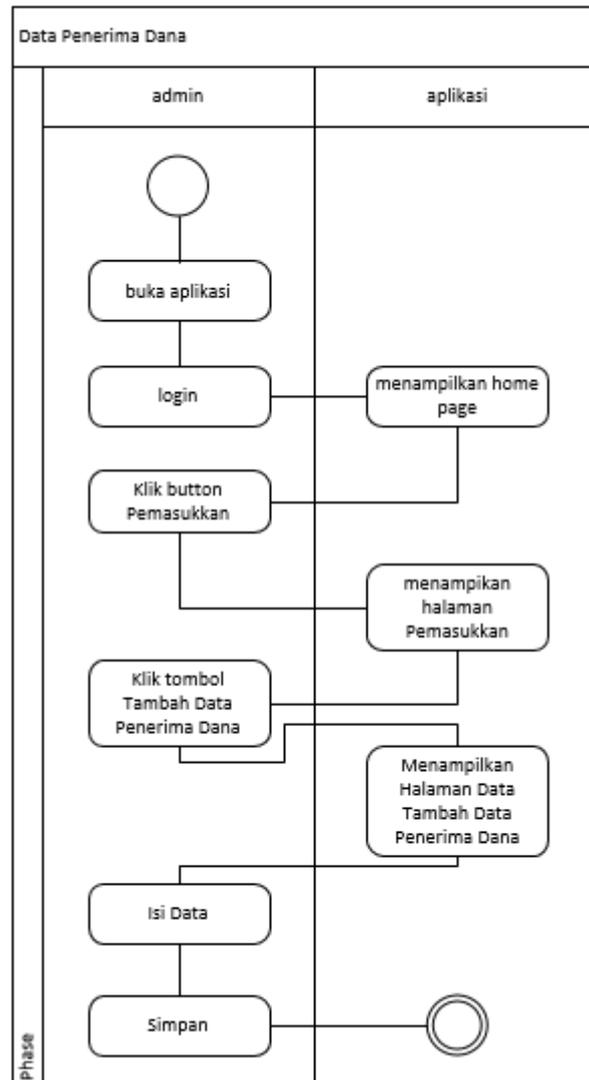
### 3.4.9 Activity Diagram Pemasukkan



**Gambar 3. 9** Activity Diagram Pemasukkan

Gambar 3.9 diatas menunjukkan *activity diagram* pemasukkan. Admin dapat menambahkan data pemasukkan dana yang diberikan pada pihak program studi. Admin dapat menambah, mengubah, dan menghapus data pemasukkan.

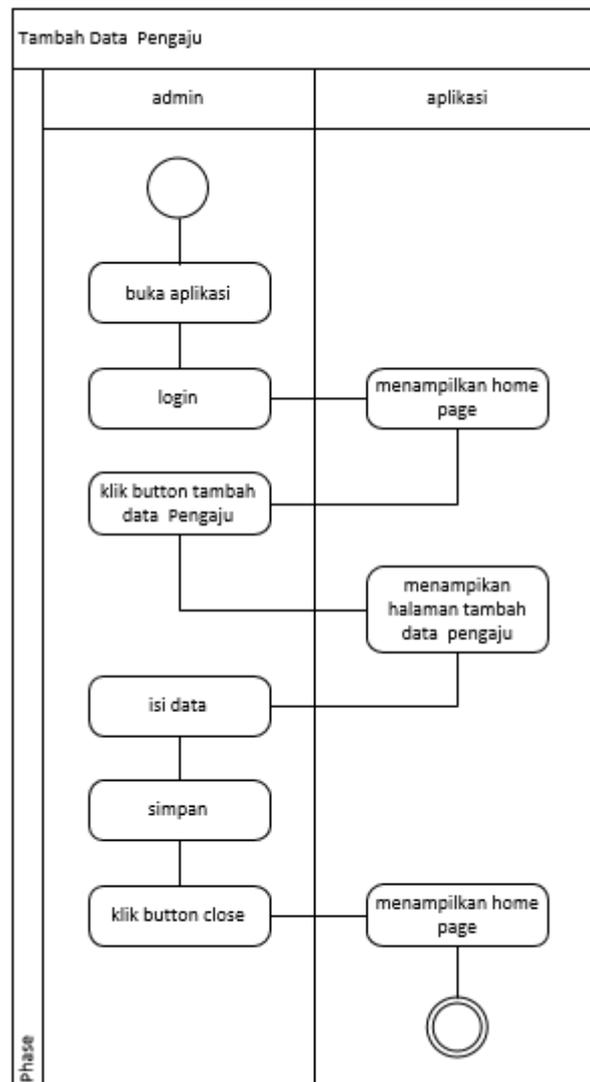
### 3.4.10 Activity Diagram Data Penerima Dana



**Gambar 3. 10** *Activity Diagram* Data Penerima Dana

Gambar 3.10 menunjukkan *activity diagram* data penerima dana. Admin dapat menambahkan data penerima dana yang diberikan ke pihak prodi. Admin dapat menambah, mengubah, serta menghapus data penerima dana.

### 3.4.11 Activity Diagram Tambah Data Pengaju



**Gambar 3. 11** Activity Diagram Tambah Data Pengaju

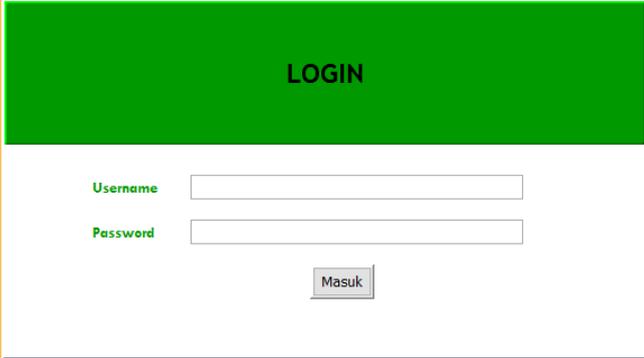
Gambar 3.11 menunjukkan *activity diagram* tambah data pengaju. Admin dapat menambahkan data pengaju yang akan menggunakan dana yang tersedia berdasarkan kategori tertentu. Admin dapat menambah, menghapus, serta mengubah data.

### 3.5 Rancangan Antarmuka

Dalam membangun sebuah *aplikasi desktop* rancangan antarmuka sangatlah penting. Dengan merancang antarmuka, akan memberikan kemudahan dalam mendesain *aplikasi desktop*. Antarmuka akan digunakan sebagai penghubung antara *user* dengan sistem *aplikasi desktop*. *Aplikasi desktop* akan memberikan *feedback* berupa antarmuka pada setiap kegiatan *user*. Komunikasi ini yang akan membuat sistem *aplikasi desktop* berjalan.

#### 3.5.1 Rancangan Antarmuka *Login*

*Login* adalah autentikasi awal untuk dapat masuk ke dalam aplikasi. Rancangan antarmuka *login* dapat dilihat di gambar 3.12. Dalam halaman *login* terdapat *form login* dan tombol tambah admin. Tombol tambah admin akan mengantarkan admin ke *form* tambah admin.



The image shows a login interface. At the top, there is a green rectangular header with the word "LOGIN" centered in white capital letters. Below the header, the background is white. On the left side, the labels "Username" and "Password" are written in green text. To the right of each label is a white rectangular input field with a thin grey border. Below the "Password" field, there is a grey rectangular button with the word "Masuk" written in white text.

**Gambar 3. 12** Rancangan Antarmuka Login

#### 3.5.3 Rancangan Antarmuka *Homepage*

Setelah melakukan *login*, maka *homepage* akan tampil. Disini admin dapat memilih menu yang ada. Menu – menu tersebut adalah *form* input laporan, *form* tambah data master kategori, dan *form* tambah data master tahun ajaran. Rancangan antarmuka halaman *homepage* dapat dilihat digambar 3.13.



**Gambar 3. 13** Rancangan Antarmuka Homepage

### 3.5.4 Rancangan Antarmuka Tambah Data Master Kategori

Halaman ini adalah halaman dimana admin dapat menambah kategori laporan yang nantinya akan digunakan di *form* input laporan. Rancangan antarmuka tambah data master kategori dapat dilihat di gambar 3.14

**INPUT KATEGORI**

ID Kategori

Kategori

Title 1	Title 2

**Gambar 3. 14** Rancangan Antarmuka Tambah Data Master Kategori

### 3.5.5 Rancangan Antarmuka Tambah Data Master Tahun Ajaran

Halaman ini adalah halaman dimana admin dapat menambah data tahun ajaran. Data tahun ajaran ini akan digunakan di *form* input laporan. Rancangan antarmuka tambah data master tahun ajaran dapat dilihat di gambar 3.15.

The screenshot shows a web form titled "INPUT NOTE". On the left side, there are three input fields: "ID Note" (text), "Semester" (dropdown menu with "Gasal" selected), and "Tahun Ajaran" (text). Below these fields are four buttons: "Input", "Edit", "Delete", and "Close". On the right side, there is a table with two columns, "Title 1" and "Title 2". The table has three empty rows below the header. Below the table is a large, empty rectangular area.

**Gambar 3. 15** Rancangan Antarmuka Tambah Data Master Tahun Ajaran

### 3.5.6 Rancangan Antarmuka Laporan Keuangan

Halaman ini adalah halaman dimana *admin* dapat menambah, mengubah, menghapus, dan mencetak laporan. Rancangan antarmuka *form* input rincian pengeluaran dapat dilihat di gambar 3.16

## Laporan Rekapitulasi Keuangan Program Studi Teknologi Informasi UMY



No

Kategori

Note

Tanggal

Keterangan

Termin

Nominal

ATK

ID	Kategori	Note	Tanggal	Keterangan	Termin	Nominal

**Gambar 3. 16** Rancangan Antarmuka Laporan Keuangan

### 3.5.7 Rancangan Antarmuka Data Pemasukkan

Halaman ini adalah halaman dimana admin dapat menambahkan data pemasukkan, mengubah, serta menghapus data. Rancangan antarmuka data pemasukkan dapat dilihat pada gambar 3.17

**PEMASUKKAN**

ID

Tanggal

Th. Ajeran

Penerima

Nominal

Title 1	Title 2

**Gambar 3. 17** Rancangan Antarmuka Data Pemasukkan

### 3.5.8 Rancangan Antarmuka Data Penerima Dana

Halaman ini adalah halaman dimana admin dapat menambah, mebgubah, dan menghapus data penerima dana yang diberikan ke pihak program studi. Rancangan antarmuka data penerima dana dapat dilihat pada gambar 3.18

**Gambar 3. 18** Rancangan Antarmuka Tambah Data Penerima Dana

### 3.5.9 Rancangan Antarmuka Data Pengaju

Halaman ini adalah halaman dimana admin dapat menambah, mengubah, serta menghapus data pengaju yang akan menggunakan dana yang tersedia. Rancangan antarmuka data pengaju dapat dilihat pada gambar 3.19

**Gambar 3. 19** Rancangan Antarmuka Tambah Data Pengaju