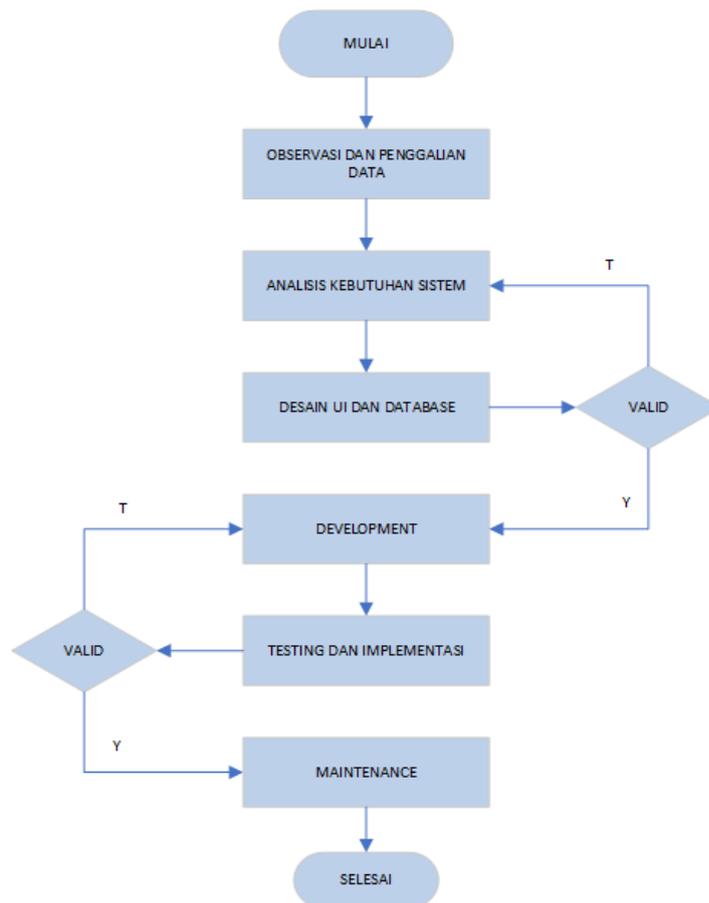


### BAB III METODOLOGI PENELITIAN

#### A. METODOLOGI PENELITIAN

Metodologi penelitian adalah serangkaian langkah-langkah yang sistematis/terstruktur yang dilakukan oleh peneliti untuk menemukan jawaban yang tepat atas pertanyaan pada objek penelitian (Prawiro, 2019).

Dalam penelitian ini metodologi penelitian dijelaskan dalam bentuk diagram alir dibawah ini:



Gambar 3. 1 *Flowchart* Metodologi Penelitian

Penjelasan dari tahapan diatas adalah sebagai berikut:

### 1. **Observasi dan Penggalian Data**

Observasi merupakan salah satu metode pengumpulan data dengan cara mengamati atau meninjau secara cermat dan langsung di lokasi penelitian untuk mengetahui kondisi yang terjadi atau membuktikan kebenaran dari sebuah desain penelitian yang sedang dilakukan. Observasi merupakan pengamatan dan pencatatan secara sistematis terhadap unsur-unsur yang nampak dalam suatu gejala pada objek penelitian (Widoyoko, 2014).

Sedangkan Penggalian Data (*Data Mining*) adalah ekstraksi pola yang menarik dari data dalam jumlah besar. Suatu pola dikatakan menarik apabila pola tersebut tidak sepele, implisit, tidak diketahui sebelumnya, dan berguna. Pola yang disajikan haruslah mudah dipahami, berlaku untuk data yang akan diprediksi dengan derajat kepastian tertentu, berguna, dan baru.

Pada tahap ini penulis terjun langsung ke lapangan yaitu kantor Pondok Pesantren Asaasunnajaah dengan menemui salah satu petugas yang mengurus bagian tabungan santri.

Berdasarkan hasil dari observasi yang dilakukan penulis di Pondok Pesantren Asaasunnajaah dan wawancara yang dilakukan langsung oleh penulis kepada Kang Ma'mun Jamil selaku pengelola tabungan santri, dapat disimpulkan bahwa proses transaksi yang dilakukan masih manual yaitu pencatatan keseluruhan transaksi masih dicatat dibuku yang dimiliki petugas dan buku tabungan yang dipegang oleh santri.

### 2. **Analisis Kebutuhan Sistem**

Analisis kebutuhan sistem sangat diperlukan dalam mendukung kinerja aplikasi, apakah aplikasi yang dibuat telah sesuai dengan kebutuhan atau belum. Karena kebutuhan sistem akan mendukung tercapainya tujuan suatu aplikasi. Tujuan dari tahap analisis adalah

memahami dengan sesungguhnya kebutuhan dari sistem yang baru dan mengembangkan sebuah sistem yang memadahi kebutuhan tersebut atau memutuskan bahwa pengembangan sistem yang baru tidak dibutuhkan. Untuk mempermudah menganalisis sebuah sistem dibutuhkan dua jenis kebutuhan. Kebutuhan fungsional dan kebutuhan nonfungsional.

#### **a. Analisis Kebutuhan Fungsional**

Kebutuhan fungsional adalah jenis kebutuhan yang berisi proses-proses yang dilakukan oleh sistem dan informasi yang dihasilkan sistem (Agustina, 2011). Kebutuhan fungsional menggambarkan fungsionalitas atau layanan sistem dan bergantung pada tipe software, harapan user dan tipe sistem dimana software digunakan . Adapun proses dan informasi yang dihasilkan diantaranya:

- admin dapat membuat laporan dari semua transaksi penyetoran.
- admin dapat membuat laporan dari semua transaksi penarikan.
- sistem dapat melakukan pencatatan transaksi penyetoran dan penarikan.
- sistem dapat mengubah, menyimpan dan menghapus data yang berkaitan dengan tabungan santri.

#### **b. Analisis Kebutuhan Non Fungsional**

Kebutuhan nonfungsional adalah kebutuhan yang menitik beratkan pada property perilaku yang dimiliki oleh sistem. Kebutuhan dalam aspek ini diantaranya:

##### 1) Perangkat Keras

Perangkat keras yang digunakan pada penelitian kali ini diantaranya adalah sebagai berikut.

- Unit laptop dengan spesifikasi

- Intel(R) Core (TM) i3-6006U CPU @ 2.00GHz  
1.99 GHz
- RAM 4,00 GB (3,89 GB usable)
- Hardisk 500 GB
- Flaskdisk
- Printer

## 2) Perangkat Lunak

Perangkat lunak yang digunakan adalah sebagai berikut diantaranya:

- Xampp v.3.2.4 sebagai server
- Figma digunakan sebagai desain antarmuka (interface),
- Visual Studio Code v.1.56.2 sebagai text editor,
- Visio 2016 untuk membuat SDLC, flowchart, dan UML.
- Sistem operasi yang digunakan adalah Kali Linux 64- bit.
- browser yang digunakan adalah google chrome.

## 3. **Desain UI dan Database**

Tahapan desain adalah tahapan untuk memberikan gambaran mengenai aplikasi tabungan santri berbasis WEB yang akan diusulkan. Tahapan perancangan sistem merupakan data yang telah dianalisis ke dalam bentuk yang mudah dan dimengerti oleh pemakai (*User*). Adapun perancangan pada tahap ini meliputi :

### a. *Use Case Diagram*

*Use Case Diagram* adalah proses penggambaran yang dilakukan untuk menunjukkan hubungan antara pengguna dengan sistem yang dirancang. Hasil representasi dari skema tersebut dibuat secara sederhana dan bertujuan untuk memudahkan user dalam membaca informasi yang diberikan (Adani, Mengenal Apa itu Use Case dan Teknik Pembuatannya, 2021). Komponen-komponen pada *use case* diagram di antaranya sebagai berikut.

#### 1) Sistem

Sebuah sistem digambarkan ke dalam bentuk persegi. Fungsinya untuk membatasi *use case* dengan interaksi dari luar sistem.

#### 2) Actor

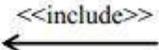
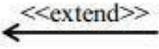
Peran actor sangat penting, tentunya menciptakan use case jadi lebih mudah. Fungsi *Actor* menjelaskan siapa yang berinteraksi dengan sistem. Actor akan memberikan informasi kepada sistem, serta menerima informasi dari sistem. Keduanya bisa terjadi secara bersamaan. Aktor tidak memberikan kontrol terhadap sistem, namun hanya memberikan gambaran mengenai hubungannya dengan sistem.

#### 3) Use Case

Use case adalah komponen gambaran fungsional dalam sebuah sistem. Sehingga konsumen maupun pembuat saling mengenal dan mengerti mengenai alur sistem yang akan dibuat.

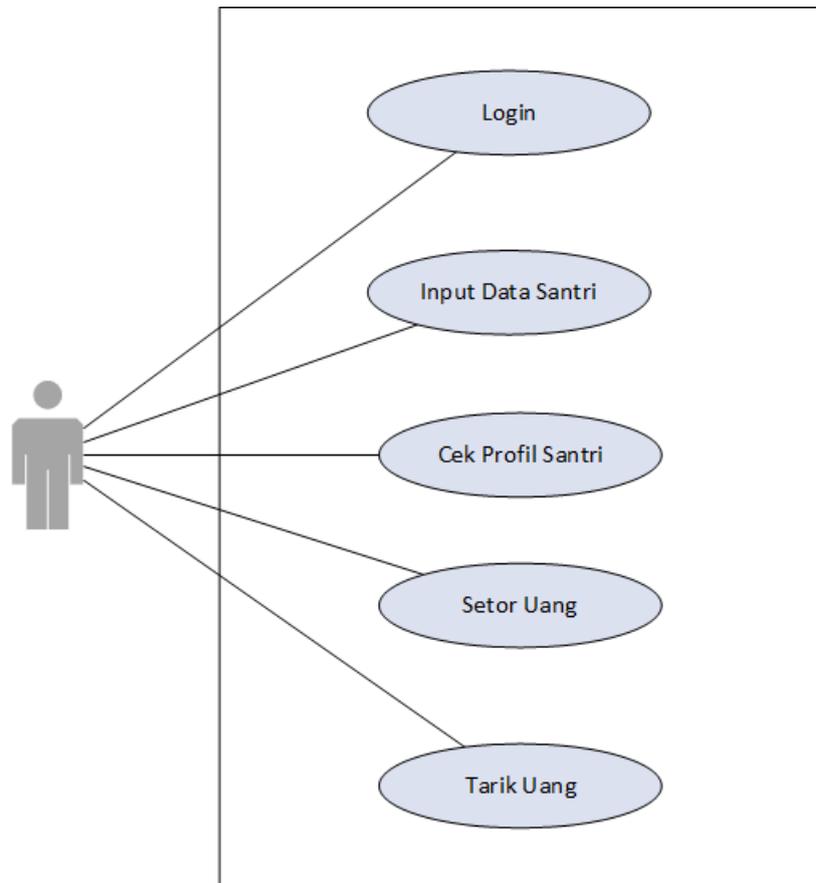
Berikut ini merupakan simbol – simbol dan keterangan dari

*Use Case Diagram*

Simbol	Keterangan
	Aktor : Mewakili peran orang, sistem yang lain, atau alat ketika berkomunikasi dengan <i>use case</i>
	<i>Use case</i> : Abstraksi dan interaksi antara sistem dan aktor
	<i>Association</i> : Abstraksi dari penghubung antara aktor dengan <i>use case</i>
	<i>Generalisasi</i> : Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya
	Menunjukkan bahwa suatu <i>use case</i> merupakan tambahan fungsional dari <i>use case</i> lainnya jika suatu kondisi terpenuhi

Gambar 3. 2 Komponen – Komponen *Use Case Diagram*

Dalam penelitian kali ini penulis membuat *use case diagram* pada aplikasi tabungan santri pada gambar berikut.



Gambar 3. 3 *Use Case Diagram* Aplikasi Tabungan Santri

Skenario *Use Case*:

1) *Login*

Nama *Use Case* : *Login*

*Actor* : Petugas

Deskripsi : Proses ini adalah tahap awal yang dilakukan oleh petugas untuk menggunakan aplikasi.

*Pre – Condition* : Petugas membuka aplikasi lalu mengisi form *login* dengan *username* dan *password* yang benar

*Post – Condition* : Sistem akan memeriksa validasi data yang dimasukan. Apabila valid maka petugas akan masuk sebagai *user* dalam aplikasi

Tabel 3. 1 Skenario *Use Case Login*

<i>Actor</i>	Reaksi Sistem
Petugas membuka aplikasi	
	Menampilkan form <i>login</i>
Memasukan <i>username</i> dan <i>password</i>	
	Validasi <i>User</i>

## 2) Input Data Santri

Nama *Use Case* : Input Data Santri

*Actor* : Petugas

Deskripsi : Proses ini adalah kegiatan yang dilakukan oleh petugas untuk menginput data santri seperti nama dan NIS. Lalu mencari data sesuai dengan data yang dimasukan

*Pre – Condition* : Petugas menginput data santri

*Post – Condition* : Petugas mencari data yang telah dimasukan

Tabel 3. 2 Skenario Input Data Santri

<i>Actor</i>	Reaksi Sistem
Petugas memasukan data santri	
Menekan tombol cari	
	Menampilkan hasil pencarian dari <i>database</i>

3) Cek Profil Santri

Nama *Use Case* : Cek Profil Santri

*Actor* : Petugas

Deskripsi : Petugas mengecek data yang tersimpan santri yang telah dimasukan sebelumnya

*Post – Condition* : Petugas dapat mengecek profil santri.

Jika terjadi kesalahan dalam pengimputan petugas dapat mengeditnya

Tabel 3. 3 Skenario Cek Profil Santri

<i>Actor</i>	Reaksi Sistem
Pilih tombol “lihat”	
	Menampilkan profil santri yang Tersimpan

#### 4) Setor Uang

Nama *Use Case* : Setor Uang

Aktor : Petugas

Deskripsi : Petugas melakukan penambahan nominal uang pada sistem sesuai dengan nominal uang yang diambil

*Pre – Condition* : Petugas melakukan penambahan nominal uang

*Post – Condition* : Santri memberikan uang kepada petugas

Tabel 3. 4 Skenario Setor Uang

<i>Actor</i>	Reaksi Sistem
Pilih tombol “setor”	
	Menampilkan form data uang yang akan disetor
Menginput uang yang akan disetorkan	
Pilih tombol “ok”	
	Menyimpan data
	Menampilkan sisa uang yang tersimpan

### 5) Tarik Uang

Nama *Use Case* : Tarik Uang

Aktor : Petugas

Deskripsi : Petugas melakukan pengurangan nominal uang pada sistem sesuai dengan nominal uang yang diambil

*Pre – Condition* : Petugas melakukan penambahan nominal uang

*Post – Condition* : Santri memberikan uang kepada petugas

Tabel 3. 5 Skenario Tarik Uang

<i>Actor</i>	Reaksi Sistem
Pilih tombol “tarik”	
	Menampilkan form data uang yang akan ditarik
Menginput uang yang akan ditarik	
Pilih tombol “ok”	
	Menyimpan data
	Menampilkan sisa uang yang tersimpan

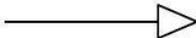
**b. Class Diagram**

*Class diagram* adalah jenis diagram struktur statis dalam UML yang menggambarkan struktur sistem dengan menunjukkan sistem *class*, atributnya, metode, dan hubungan antar objek (Oktriwina, 2021).

Atribut dan operation (metode) dapat memiliki salah satu sifat berikut.

- 1) *Private*, hanya bisa dipanggil dari dlm kelas itu sendiri. metode/atribut diawali “-“.
- 2) *Protected*, hanya dapat dipanggil oleh class yang bersangkutan dan class turunannya. metode diawali dg tanda “#”.
- 3) *Public*, dapat dipanggil dari semua objek. metode/atribut diawali tanda “+”.

Gambar berikut ini penjelasan *symbol relationships* antar *class* yang digunakan pada diagram class.

asosiasi / <i>association</i> 	relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
asosiasi berarah / <i>directed association</i> 	relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
generalisasi 	relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
kebergantungan / <i>dependency</i> 	relasi antar kelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	relasi antar kelas dengan makna semua-bagian ( <i>whole-part</i> )

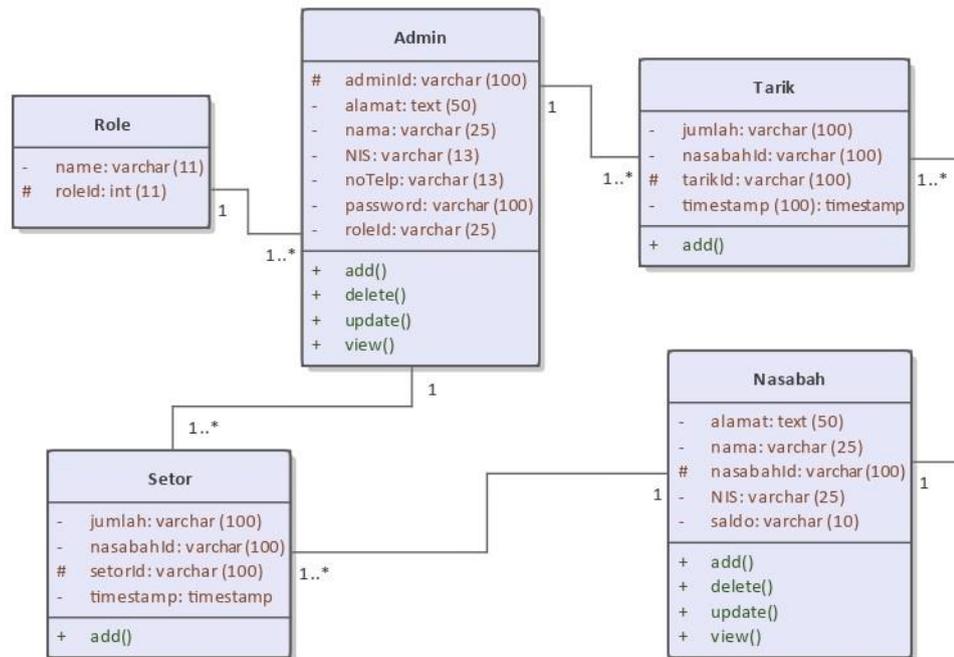
Gambar 3. 4 Symbol Relationships antar class

Relationship Multiplicity menunjukkan jumlah suatu objek yang bisa berhubungan dengan objek lain.

Nilai Kardinalitas	Arti	Contoh
 0..1	Nol atau satu	karyawan — 0..1 istri
1	Hanya satu	negara — 1 presiden
0..*	Nol atau lebih	karyawan — 0..* anak
1..*	Satu atau lebih	bos — 1..* bawahan
n	Hanya n (dengan n > 1)	karyawan — n cek up
0..n	Nol sampai n (dengan n > 1)	karyawan — 0..n sim
1..n	Satu sampai n (dengan n > 1)	kereta api — 1..n gerbong

Gambar 3. 5 Angka Kardinalitas

Dibawah ini merupakan *class diagram* dari aplikasi tabungan santri.



Gambar 3. 6 Class Diagram Aplikasi Tabungan Santri

### c. *Sequence Diagram*

*Sequence diagram* atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci (Setiawan, 2021). Selain itu *sequence diagram* juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya. Objek-objek yang berhubungan dengan berjalannya proses operasi biasanya diurutkan dari kiri ke kanan.

Berikut beberapa komponen utama yang sering digunakan:

#### 1) *Actor*

Komponen yang pertama adalah *actor*. Komponen ini menggambarkan seorang pengguna (*user*) yang berada di

luar sistem dan sedang berinteraksi dengan sistem. Dalam *sequence diagram*, aktor biasanya digambarkan dengan simbol *stick figure*.

## 2) *Activation box*

Selanjutnya ada *activation box*. Komponen *activation box* ini merepresentasikan waktu yang dibutuhkan suatu objek untuk menyelesaikan tugasnya. Semakin lama waktu yang diperlukan, maka secara otomatis *activation box* nya juga akan menjadi lebih panjang. Komponen ini digambarkan dengan bentuk persegi panjang.

## 3) *Lifeline*

Berikutnya adalah *lifeline*. Komponen ini digambarkan dengan bentuk garis putus-putus. *Lifeline* ini biasanya memiliki kotak yang berisi objek yang memiliki fungsi untuk menggambarkan aktifitas dari objek.

## 4) Objek

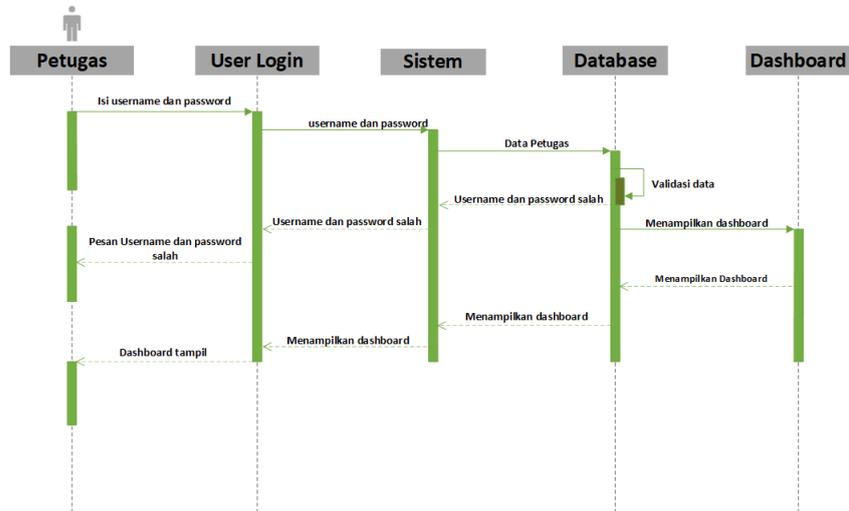
Komponen berikutnya adalah objek. Komponen objek ini digambarkan memiliki bentuk kotak yang berisikan nama dari objek dengan garis bawah. Biasanya objek berfungsi untuk mendokumentasikan perilaku sebuah objek pada sebuah sistem.

## 5) *Messages*

Terakhir ada *messages* atau pesan. Komponen ini untuk menggambarkan komunikasi antar objek. *Messages* biasanya muncul secara berurutan pada *lifeline*. Komponen *messages* ini direpresentasikan dengan anak panah. Inti dari sebuah diagram urutan terdapat pada komponen *lifeline* dan *messages* ini.

Berikut ini merupakan *sequence diagram* pada aplikasi tabungan santri

a) *Login*

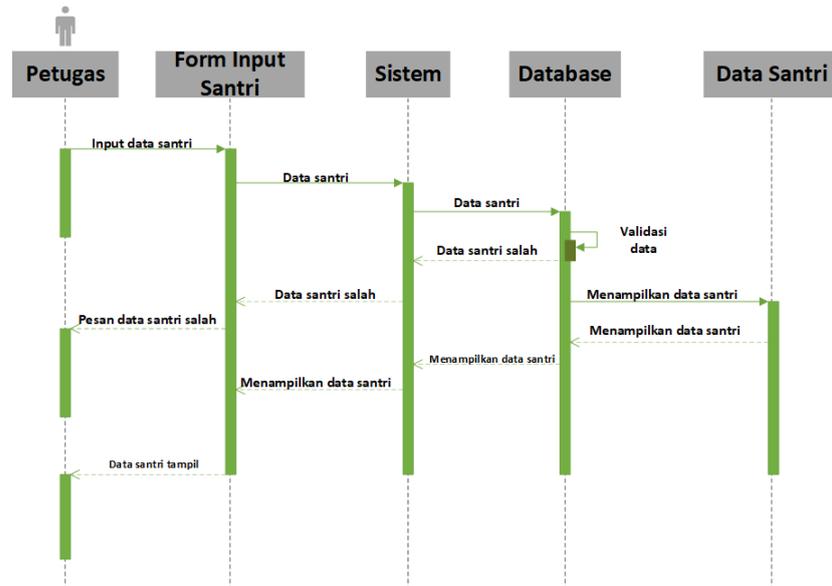


Gambar 3. 7 *Sequence Diagram Login*

Keterangan :

Pada diagram diatas terdapat satu aktor (petugas) dan empat objek, yaitu *user login*, sistem, database, dan dashboard. Pertama-tama petugas akan masuk ke tampilan *user login* dengan menggunakan *username* dan *password*. Lalu, sistem akan mengirimkan data tersebut ke *database* untuk divalidasi. Di dalam *database* data petugas akan diperiksa dan divalidasi. Jika data yang dimasukan salah dan tidak valid, maka akan menampilkan pesan bahwa ID atau *Password* salah. Sedangkan jika data yang dimasukan benar dan valid, maka sistem akan menampilkan dashboard aplikasi.

## b) Input Data Santri

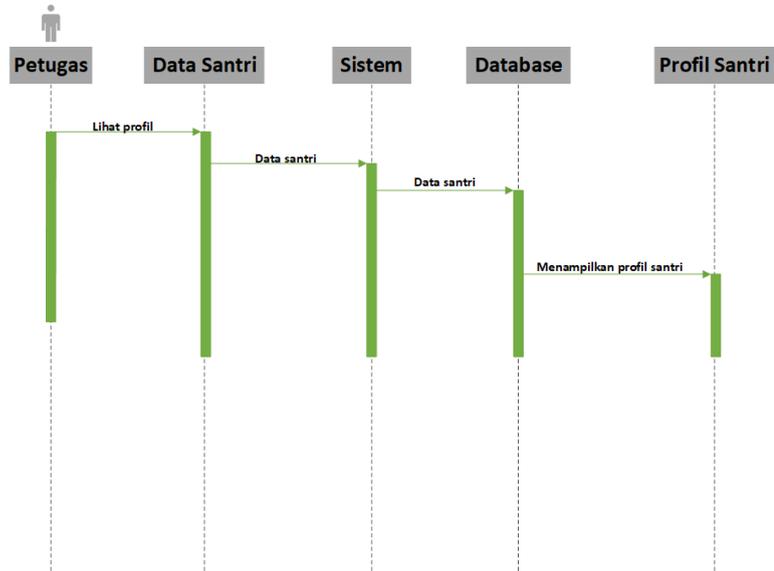


Gambar 3. 8 *Sequence Diagram* Input Data Santri

### Keterangan :

Pada diagram diatas terdapat satu aktor (petugas) dan empat objek, yaitu form input santri, sistem, database, dan data santri. Pertama-tama petugas akan masuk ke tampilan form input santri dengan klik tombol tambah. Lalu, sistem akan mengirimkan data tersebut ke database untuk divalidasi. Di dalam database data santri akan diperiksa dan divalidasi. Jika data yang dimasukan salah dan tidak valid, maka akan menampilkan pesan bahwa data santri salah. Sedangkan jika data yang dimasukan benar dan valid, maka sistem akan menampilkan data santri.

### c) Cek Profil Santri

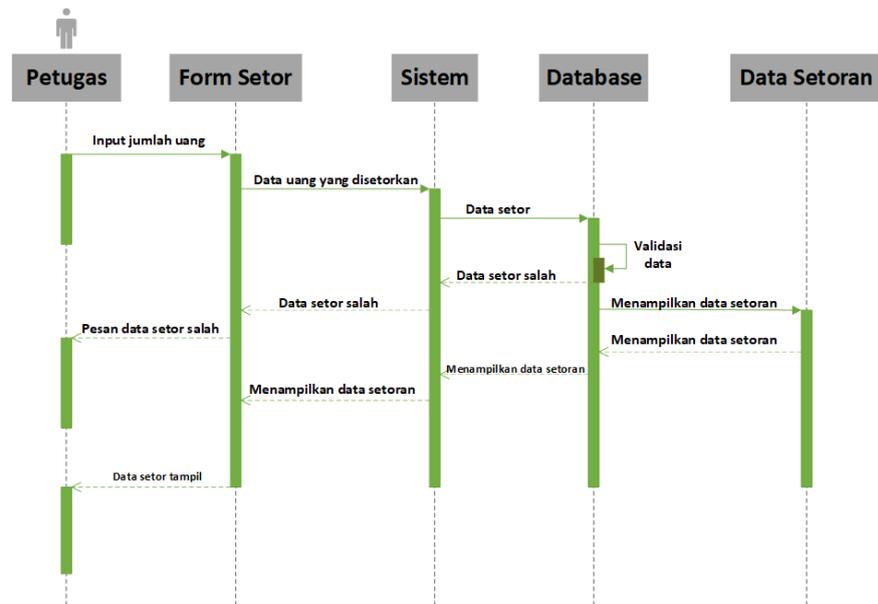


Gambar 3. 9 *Sequence Diagram* Cek Profil Santri

#### Keterangan :

Pada diagram diatas terdapat satu aktor (petugas) dan empat objek, yaitu data santri, sistem, database, dan profil santri. Pertama-tama petugas akan klik tombol “lihat”. Sistem akan mengirimkan data tersebut ke *database*. Selanjutnya sistem akan menampilkan Profil santri.

#### d) Setor Uang

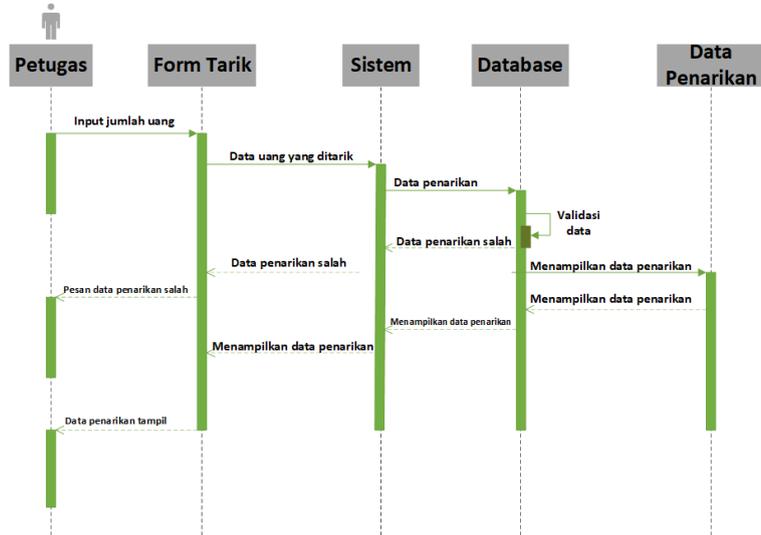


Gambar 3. 10 *Sequence Diagram* Setor Uang

#### Keterangan :

Pada diagram diatas terdapat satu aktor (petugas) dan empat objek, yaitu form setor, sistem, *database*, dan data santri. Pertama-tama petugas akan masuk ke tampilan form setor dengan klik tombol “setor”. Lalu, sistem akan mengirimkan data tersebut ke *database* untuk divalidasi. Di dalam *database* data penyetoran akan diperiksa dan divalidasi. Jika data yang dimasukan salah dan tidak valid, maka akan menampilkan pesan bahwa data setoran salah. Sedangkan jika data yang dimasukan benar dan valid, maka sistem akan menampilkan data.

e) Tarik Uang



Gambar 3. 11 *Sequence Diagram* Tarik Uang

Keterangan :

Pada diagram diatas terdapat satu aktor (petugas) dan empat objek, yaitu form tarik, sistem, *database*, dan data santri. Pertama-tama petugas akan masuk ke tampilan form tarik dengan klik tombol “tarik”. Lalu, sistem akan mengirimkan data tersebut ke *database* untuk divalidasi. Di dalam *database* data penarikan akan diperiksa dan divalidasi. Jika data yang dimasukan salah dan tidak valid, maka akan menampilkan pesan bahwa data penarikan salah. Sedangkan jika data yang dimasukan benar dan valid, maka sistem akan menampilkan data penarikan.

#### d. Perancangan *User Interface*

Pada tahap ini perancangan *interface* atau antarmuka dengan menggunakan figma dengan tujuan untuk menggambarkan rancangan tampilan dari sistem yang akan dibuat. Berikut ini merupakan hasil *mockup* Aplikasi Tabungan Santri.

##### 1) Halaman *Login*



Gambar 3. 12 *User Interface Login*

Gambar diatas merupakan *form login* yang digunakan untuk *login* pengguna, admin agar dapat mengakses halaman selanjutnya yaitu *dashboard*

##### 2) Halaman *Dashboard*



Gambar 3. 13 *User Interface Dashboard*

Gambar diatas merupakan *interface* pada halaman dashboard berfungsi untuk memberikan informasi. Terdapat juga beberapa fitur dibagian *sidebar*.

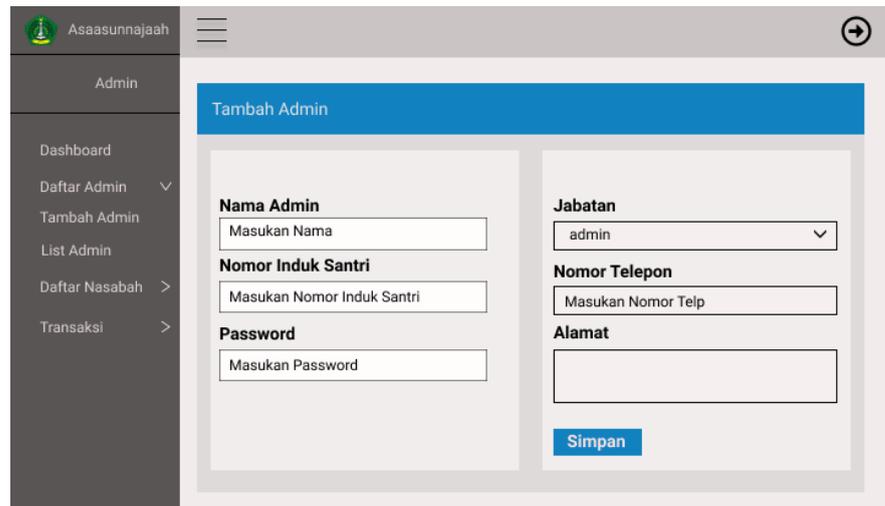
### 3) Halaman List Admin



Gambar 3. 14 *User Interface List Admin*

Pada gambar diatas merupakan tampilan list admin. Halaman tersebut berfungsi untuk menampilkan seluruh admin.

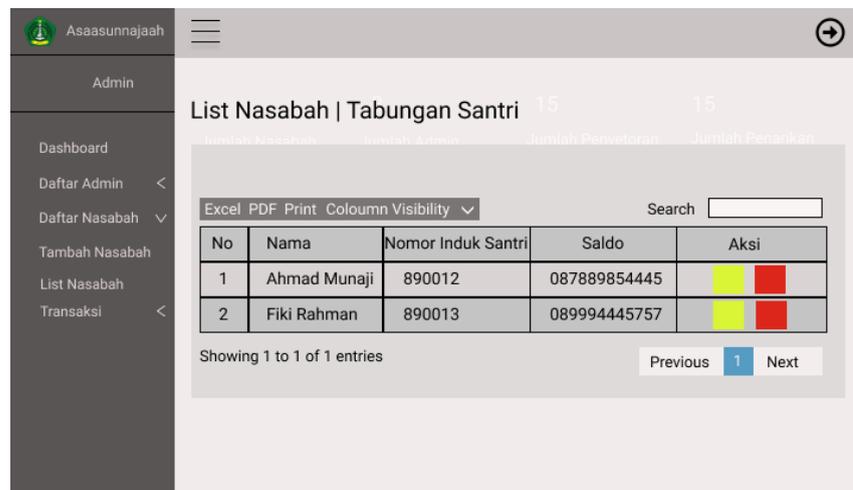
### 4) Halaman Tambah Admin



Gambar 3. 15 *User Interface* Tambah Admin

Pada gambar merupakan halaman untuk menambah admin. Pada halaman ini juga terjadi enkripsi pada *password*.

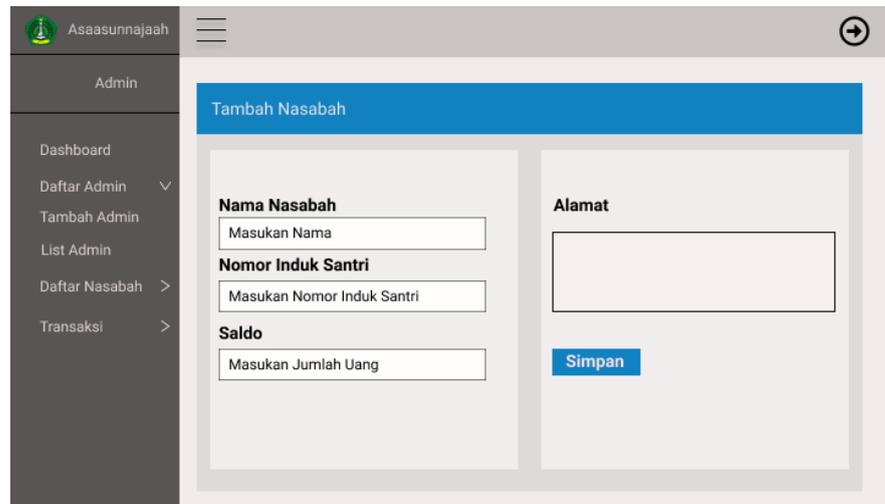
#### 5) Halaman List Nasabah



Gambar 3. 16 *User Interface* List Nasabah

Pada gambar diatas merupakan tampilan list nasabah. Halaman tersebut berfungsi untuk menampilkan seluruh nasabah.

#### 6) Halaman Tambah Nasabah



Gambar 3. 17 *User Interface* Tambah Nasabah

Pada gambar diatas merupakan halaman untuk menambah nasabah yang akan menabung.

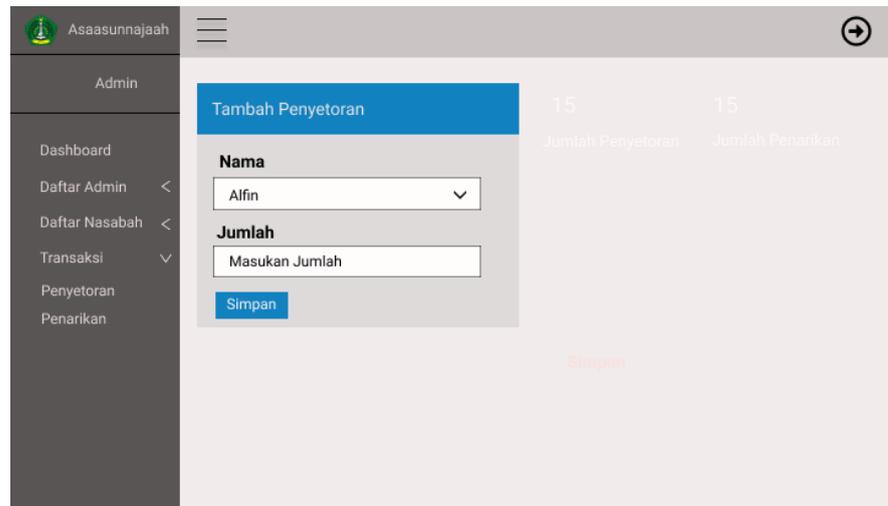
#### 7) Halaman List Penyetoran



Gambar 3. 18 *User Interface* List Penyetoran

Pada gambar diatas merupakan tampilan list penyetoran. Halaman tersebut berfungsi untuk menampilkan seluruh transaksi berupa setor.

#### 8) Halaman Tambah Penyetoran



Gambar 3. 19 *User Interface* Tambah Penyetoran

Pada gambar berikut merupakan halaman untuk menambah penyetoran jika ada nasabah yang akan menabung.

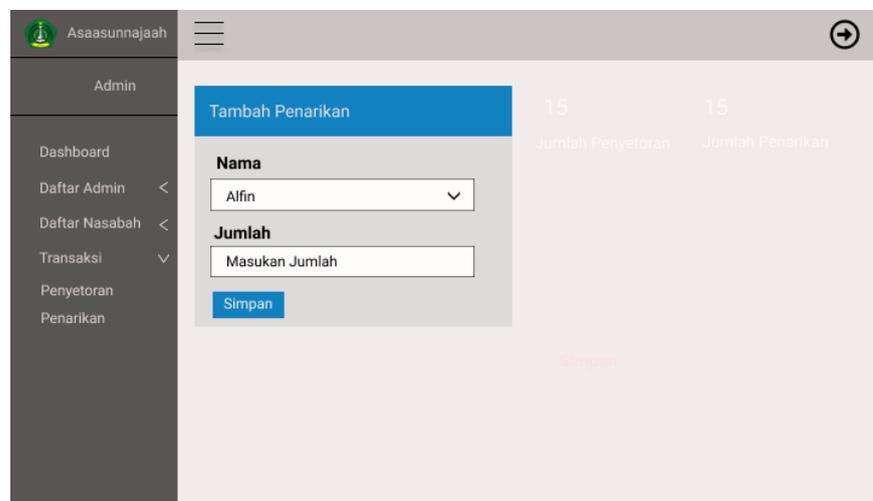
## 9) Halaman List Penarikan



Gambar 3. 20 *User Interface* List Penarikan

Pada gambar berikut merupakan tampilan list penarikan. Halaman tersebut berfungsi untuk menampilkan seluruh transaksi berupa tarik.

## 10) Halaman Tambah Penarikan



Gambar 3. 21 *User Interface* Tambah Penarikan

Pada gambar berikut merupakan halaman untuk menambah penarikan jika ada nasabah yang akan mengambil uang.

**e. Perancangan Database**

Pada tahap ini perancangan *database* dengan menggunakan phpMyAdmin. Berikut ini merupakan *database* Aplikasi Tabungan Santri.

1) Tabel Admin

Nama Tabel : tb\_admin

Tabel 3. 6 Tabel Admin

Nama	Type	Length	Ket
adminId	Integer	11	Primary key
Nama	Varchar	100	-
NIS	Varchar	100	-
roleId	Integer	11	Foreign key
Password	Varchar	100	-
nomorTelp	Varchar	100	-
Alamat	text	-	-

2) Tabel Nasabah

Nama Tabel : tb\_nasabah

Tabel 3. 7 Tabel Nasabah

Nama	Type	Length	Ket
nasabahId	Integer	11	Primary key
nama	Varchar	100	-
NIS	Varchar	100	-
alamat	Text	-	-
Saldo	Varchar	100	-

3) Tabel Role

Nama Tabel : tb\_role

Tabel 3. 8 Tabel Role

Nama	Type	Length	Ket
roleId	Integer	11	Primary key
Name	Varchar	25	-

4) Tabel Penyetoran

Nama Tabel : tb\_penyetoran

Tabel 3. 9 Tabel Penyetoran

Nama	Type	Length	Ket
setorId	Integer	11	Primary key
nasabahId	Integer	25	Foreign key
jumlah	varchar	100	-
timestamp	timestamp	-	-

5) Tabel Penarikan

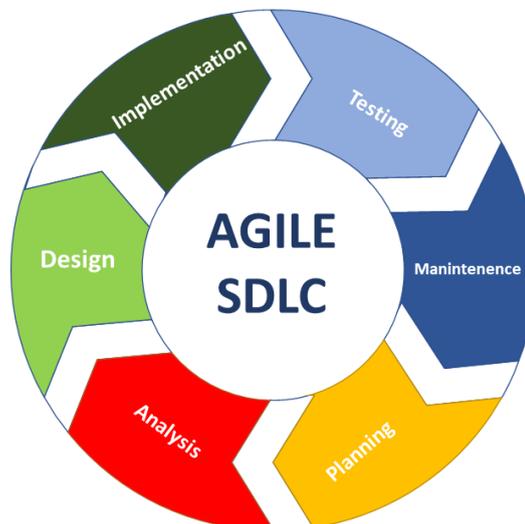
Nama Tabel : tb\_penarikan

Tabel 3. 10 Tabel Penarikan

Nama	Type	Length	Ket
tarikId	Integer	11	Primary key
nasabahId	Integer	25	Foreign key
Jumlah	Varchar	100	-
timestamp	timestamp	-	-

## B. METODE PENGEMBANGAN SISTEM

Dalam penelitian ini metode dalam pengembangan perangkat lunak SDLC (*Software Development Life Cycle*) menggunakan *Agile Software Development*. *Agile Software Development* adalah metodologi pengembangan *software* yang didasarkan pada proses pengerjaan yang dilakukan berulang dimana, aturan dan solusi yang disepakati dilakukan dengan kolaborasi antar tiap tim secara terorganisir dan terstruktur (Adani, Metodologi Agile, 2020). Untuk metodenya dijelaskan pada gambar dibawah ini.



Gambar 3. 22 Metode Agile SDLC

### **C. METODE KEAMANAN SISTEM**

*Metode keamanan sistem dengan menggunakan algoritma AES 256 bertujuan untuk meningkatkan keamanan sistem dari serangan kejahatan cyber. Dalam hal ini untuk mengantisipasi terjadinya kejahatan cyber perlu dilakukan pengujian keamanan yang bertujuan menguji tingkat keamanan database system.*

Pengujian keamanan dengan menggunakan AES dilakukan pengujian sistem dengan cara sebagai berikut:

1. Pengujian *database server*
2. Pengujian *brute force*

#### D. JADWAL PENELITIAN

Dalam penelitian ini jadwal penelitian dapat dilihat dalam bentuk tabel jadwal penelitian sebagai berikut.

Tabel 3. 11 Jadwal Penelitian

Kegiatan	Bulan September				Bulan Oktober				Bulan November				Bulan Desember			
	Minggu Ke															
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Observasi dan Penggalian Data	■	■														
Analisis Kebutuhan Sistem			■	■												
Desain UI dan Database					■	■										
Development							■	■	■	■	■	■				
Testing dan Implementasi													■	■		
Maintenance															■	■