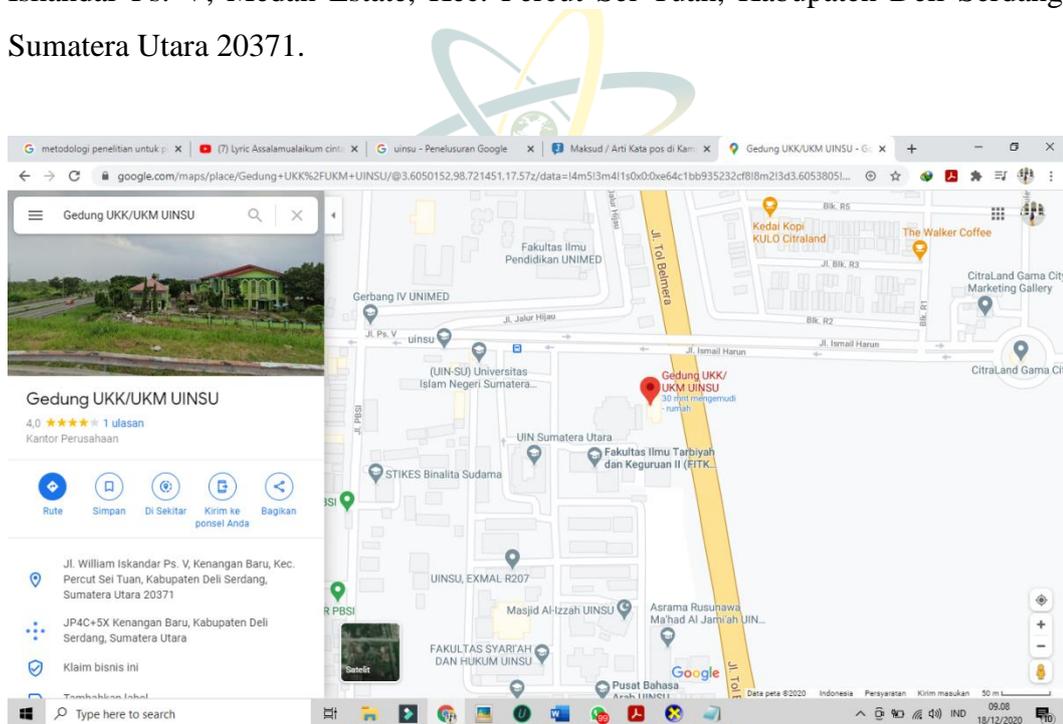


## BAB III METODOLOGI PENELITIAN

### 1.1. Tempat Penelitian

Tempat yang dijadikan studi kasus penelitian oleh penulis yaitu LDK Al-Izzah UIN Sumatera Utara, yang dimana LDK Al-Izzah UIN Sumatera Utara terletak pada gedung UKK/UKM Uin Sumatera Utara yang berlokasi di Jl. William Iskandar Ps. V, Medan Estate, Kec. Percut Sei Tuan, Kabupaten Deli Serdang, Sumatera Utara 20371.



*Gambar 3. 1. Lokasi Gedung UKK/UKM Uin Sumatera Utara*

### 1.2. Waktu Penelitian

Pada penelitian ini penulis membutuhkan jadwal dan tahapan penelitian untuk mencapai batas waktu dalam pembuatan sistem yaitu dari bulan Agustus 2020 sampai Februari 2021. Adapun jadwal tahapan penelitian tersebut adalah sebagai berikut :

Tabel 3. 1. Waktu Penelitian

Tahapan	Agustus				September				Oktober				November				Desember				Januari				Februari			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Riset Penelitian	■	■																										
Pembuatan Proposal			■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
Bimbingan Skripsi					■	■	■	■	■	■	■	■	■	■	■	■												
Seminar Proposal																	■	■	■	■								
Analisis Perancangan Sistem																					■	■	■	■				
Implementasi Sistem																						■	■	■	■	■	■	■
Pengujian Sistem																											■	■
Sidang Skripsi																												■

### 1.3. Kebutuhan Aplikasi

Pada tahap ini, penulis mengkaji persyaratan yang akan digunakan untuk membuat program. Berikut persyaratan yang akan digunakan penulis dalam penelitian ini:

#### 1.3.1. Perangkat Keras

Adapun spesifikasi perangkat keras (*hardware*) dalam pembuatan aplikasi android terdiri dari sebagai berikut:

1. Laptop Lenovo, Intel(R) Core(TM) i3-4030U CPU @ 1.90GHz (4 CPUs), ~1.9GHz
2. RAM 10 GB
3. HDD 500 GB
4. *Smartphone* Android

#### 1.3.2. Perangkat Lunak

Adapun perangkat lunak yang digunakan dalam pembuatan skripsi dan pengembangan sistem ini adalah sebagai berikut:

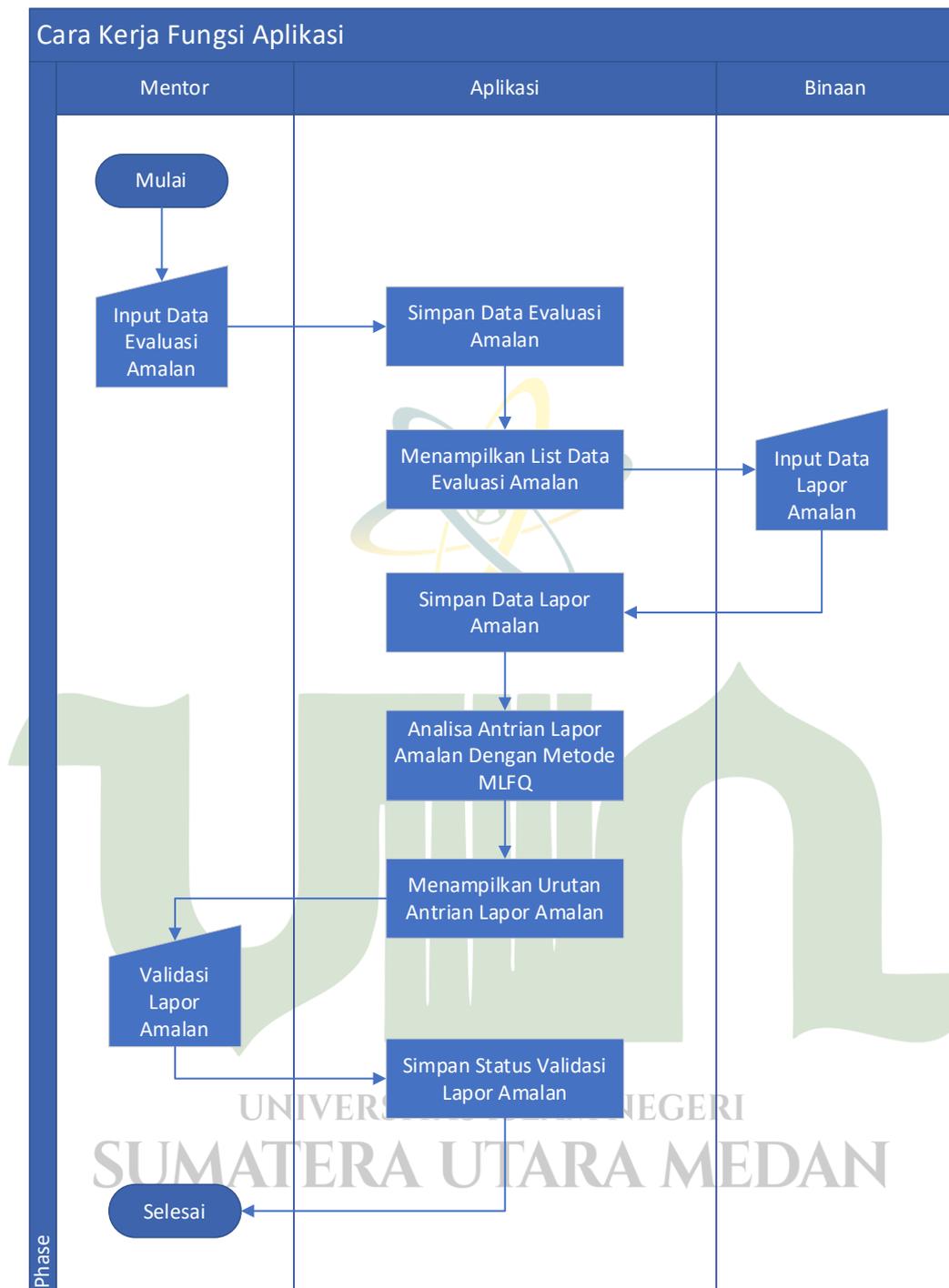
1. Windows 10 Home Single Language 64-bit (10.0, Build 18363)

2. Microsoft Visio
3. Mendeley Desktop Versi 1.19.3
4. Visual Studio Code
5. Visual Paradigma
6. Node.Js
7. Firebise SDK
8. Android SDK
9. Balsamiq Wireframes

#### 1.4.Cara Kerja

##### 1.4.1. Cara Kerja Fitur Sistem

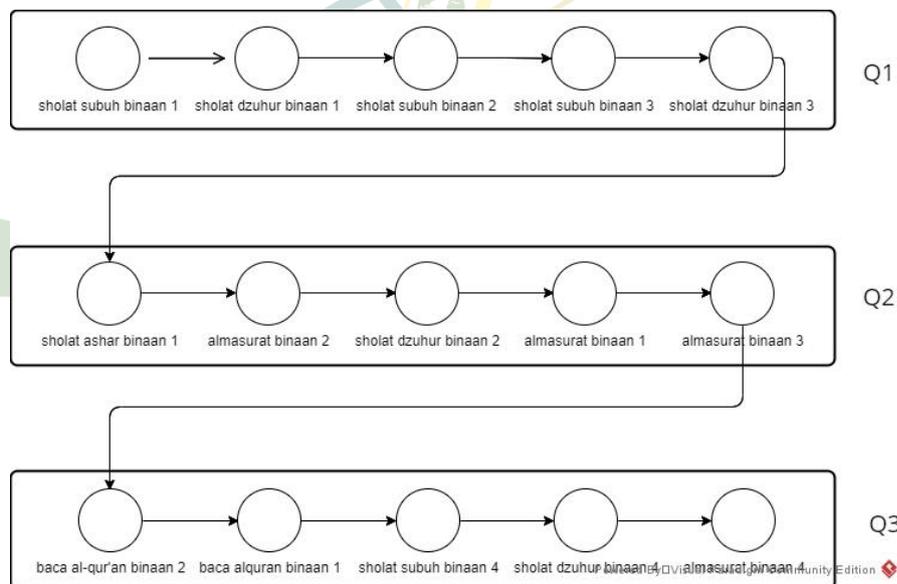
Mentor pada aplikasi dapat memberikan tugas lapor amalan. Binaan yang telah siap untuk melakukan lapor amalan dapat menginputkan data lapor amalan. Data lapor amalan binaan dianalisa oleh sistem dengan metode *MLFQ* sehingga mendapatkan jadwal lapor amalan yang optimal. Mentor mendapatkan list data antrian yang optimal dari sistem, sehingga mentor dapat langsung memanggil binaan untuk melakukan lapor amalan. Setiap binaan yang sudah lapor amalan divalidasi oleh mentor melalui aplikasi dengan tekan tombol “Approve” apabila sudah melaporkan amalan dengan benar. Namun jika binaan tidak sesuai dalam melaporkan amalan, mentor dapat tekan tombol “Reject”. Dengan adanya aplikasi ini, mentor dapat langsung laporan lapor amalan dari sistem, sehingga tidak perlu lagi membuat rekap lapor amalan secara manual. Gambar 3.2 merupakan alur fitur aplikasi.



Gambar 3. 2. Cara Kerja Fitur Sistem

#### 1.4.2. Cara Kerja Metode Multilevel Feedback Queue (MLFQ) pada Aplikasi Evaluasi Ibadah Harian

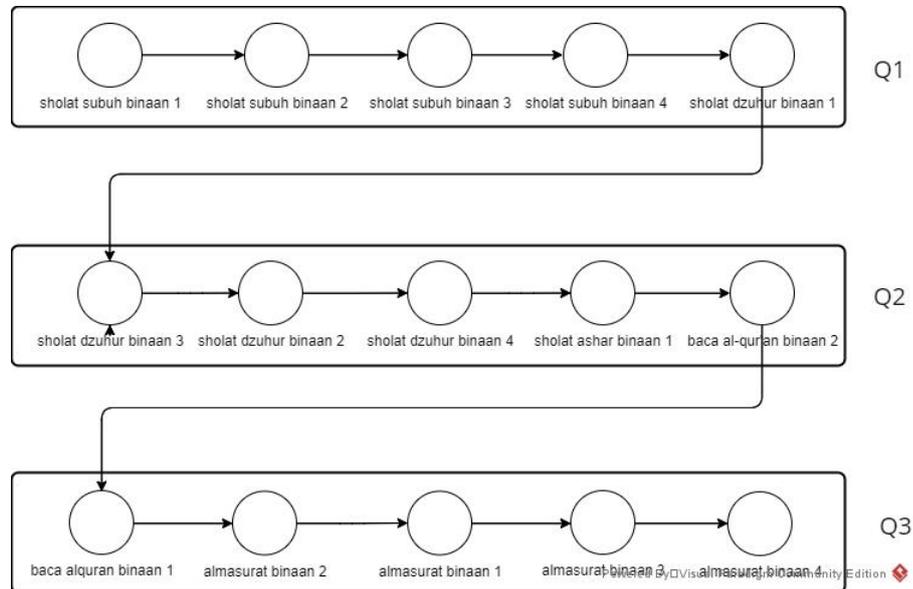
Langkah awal pengerjaan algoritma MLFQ pada aplikasi evaluasi amalan harian menggunakan *dynamic time quantum* yaitu menentukan jumlah queue. Pada aplikasi ini dipakai 3 queue (Q1, Q2, dan Q3), masing-masing queue memiliki *quantum time*, pertama sebelum diproses oleh algoritma *MLFQ* semua antrian amalan binaan yang sudah memiliki burst time masing-masing masuk semua ke Q1, Q2, dan Q3 sesuai kapasitas *dynamic time quantum* pada queue itu sendiri. Berikut adalah contoh antrian sebelum di analisa *MLFQ* pada aplikasi evaluasi ibadah harian yaitu :



Gambar 3. 3. Antrian Sebelum di Analisa MLFQ

Setelah antrian amalan binaan semuanya masuk maka proses analisa *MLFQ* dimulai dengan menyatukan terlebih dahulu amalan amalan yang sama, lalu akan dihitung setiap burst time amalan dan banyak banyak amalan yang dilaporkan berdasarkan algoritma *mlfq* yang menghasilkan nomor urut antrian dengan waktu yang efisien. Semua proses dikerjakan berurutan berdasarkan level prioritas paling tinggi yaitu Q1. Ketika mentor melakukan validasi lapor amalan binaan tetapi *quantum time* pada Q1 habis dan antrian amalan binaan masih ada maka proses tersebut melakukan perpindahan antrian ke Q2. Jika mentor juga belum selesai validasi lapor amalan binaan maka mencapai *queue* paling rendah yaitu Q3. Berikut

adalah contoh antrian amalan berdasarkan analisa *MLFQ* pada aplikasi evaluasi ibadah harian yaitu :



Gambar 3. 4. Antrian Berdasarkan Analisa *MLFQ*

#### 1.4.3. Metode Pengumpulan Data

Dalam penelitian ini, penulis akan menggunakan metode kualitatif yaitu dengan melakukan observasi dan wawancara yang dijelaskan sebagai berikut:

##### 1. Observasi

*Observasi* merupakan pengamatan secara langsung ke suatu tempat yang ingin diteliti. Penulis melakukan observasi pada pihak LDK Al-Izzah UIN Sumatera Utara yaitu Ukhti Niswatun Mujahidah selaku Sekretaris Departemen Kaderisasi LDK AL-Izzah UINSU, yang dimana Departemen Kaderisasi yang mengontrol para kader LDK Al-Izzah, lebih banyak mengetahui dan memiliki data LDK AL-Izzah UIN Sumatera Utara seperti profil, serta data pementor dan binaan.

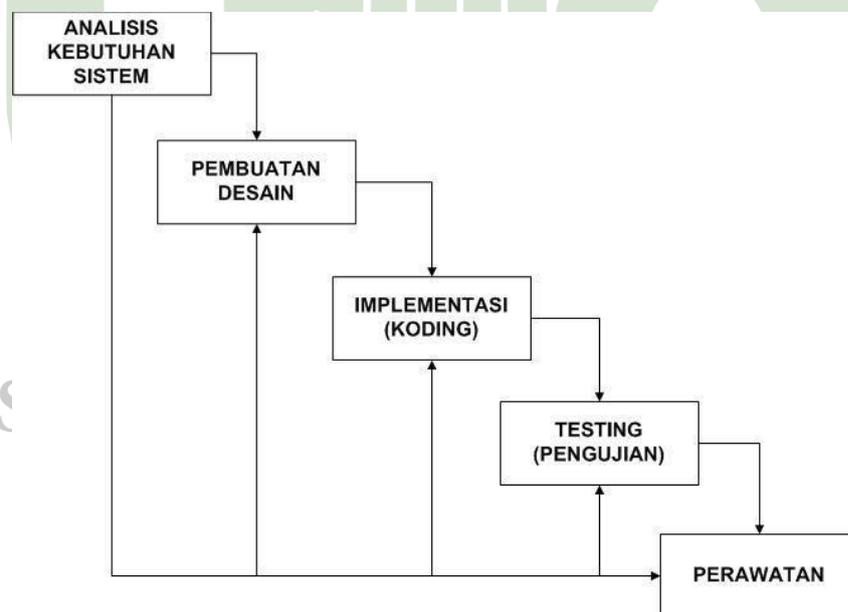
##### 2. Wawancara

Wawancara merupakan suatu metode pengumpulan data dengan berbicara kepada sumber data; Percakapan ini dilakukan baik secara langsung maupun tidak langsung melalui dialog verbal (tanya jawab). Wawancara ini ditujukan kepada bagian Kaderisasi LDK Al-Izzah UIN

Sumatera Utara yaitu Ukhti Niswatun Mujahidah serta beberapa pementor dan binaan yaitu Ukhti Bela Putriyani dan Ukhti Nurhadizah Saragih. Pada tahapan ini penulis mengajukan beberapa pertanyaan terkait kendala dalam melakukan proses pelaporan ibadah harian binaan, serta kendala dalam kegiatan *mentoring* itu sendiri.

#### 1.4.4. Metode Pengembangan Sistem

Dalam penelitian ini penulis mengembangkan sistem menggunakan metode *Software Development Live Cycle (SDLC)* model *waterfall*. Menurut Pressman (2010) dalam (Topan, Wowor, and Najooan 2015), Salah satu model desain perangkat lunak adalah model *waterfall*. Metode perencanaan yang mengharuskan semua operasi proses direncanakan dan dijadwalkan sebelum dijalankan adalah model *waterfall*. Analisis kebutuhan sistem, pengembangan desain, implementasi (coding), pengujian, dan pemeliharaan merupakan langkah-langkah dalam proses model *waterfall*.



Gambar 3. 5. Model *Waterfall*

### 1. Analisa Kebutuhan Sistem

Pada langkah ini, penulis menganalisis kebutuhan perangkat lunak dan melakukan wawancara dan observasi dengan calon pengguna. Pada langkah ini, dokumen kebutuhan pengguna atau, mungkin lebih akurat, informasi tentang preferensi pengguna terhadap sistem akan dibuat. Penulis akan menggunakan dokumen ini sebagai panduan saat menerjemahkannya ke dalam bahasa pemrograman.

### 2. Pembuatan Desain Sistem

Pada tahap ini, kebutuhan pengguna menjadi dasar proses pembuatan desain sistem. Persyaratan akan diubah menjadi desain perangkat lunak selama proses desain sehingga dapat diperkirakan sebelum pengkodean dilakukan. Penulis menggunakan pemodelan UML (Unified Modeling Language), yang memanfaatkan diagram aktivitas, kelas, dan use case, untuk membuat sistem. Selain itu, penulis akan mengembangkan antarmuka pengguna. Sebuah dokumen yang dikenal sebagai persyaratan perangkat lunak akan menjadi hasil dari langkah ini.

### 3. Implementasi (*Coding*)

*Coding* atau pengkodean merupakan konversi desain ke bahasa pemrograman. Transaksi yang diminta pengguna akan diterjemahkan oleh penulis. Karena ini adalah tahap pengembangan perangkat lunak yang sebenarnya, penggunaan komputer akan mencapai titik tertinggi selama tahap ini.

### 4. Pengujian (*Testing*)

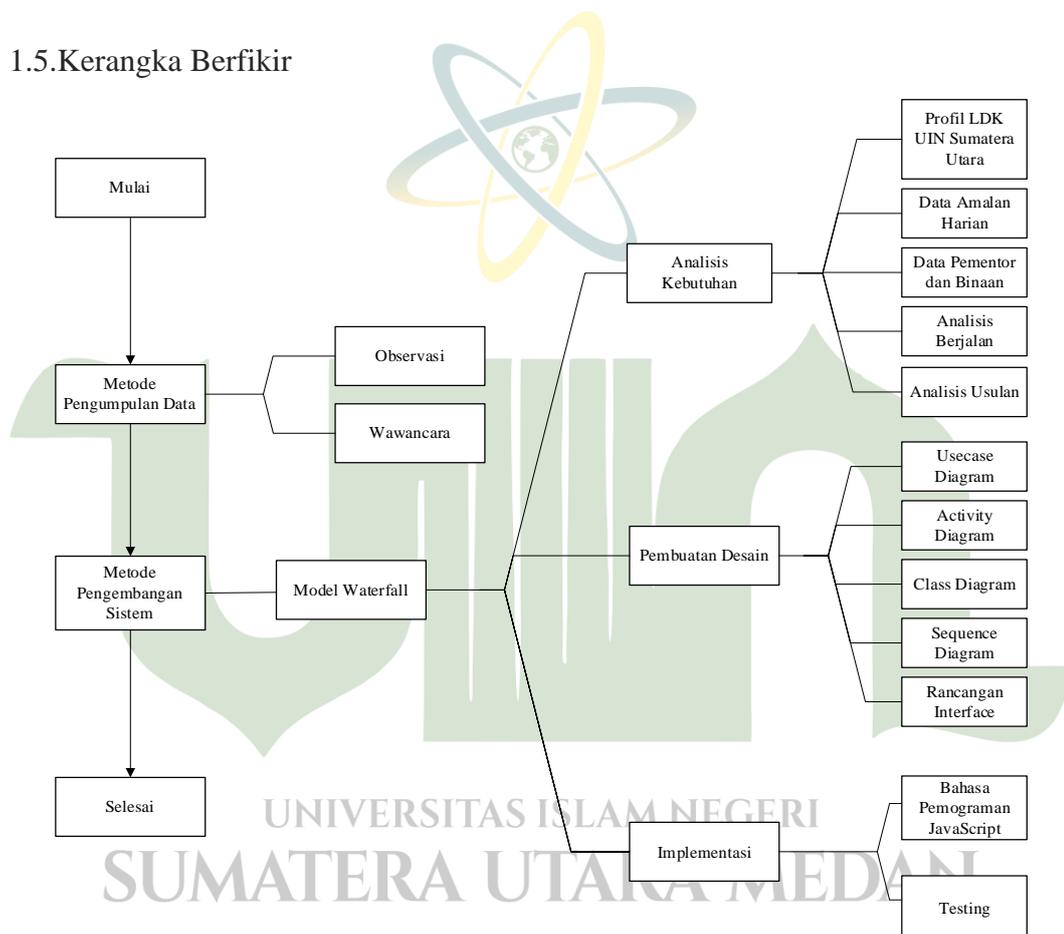
Fase ini dapat dianggap sebagai fase terakhir dalam pengembangan sistem atau perangkat lunak. Pengguna akan menggunakan sistem akhir setelah penulis menyelesaikan analisis, desain, dan pengkodean. Untuk memastikan bahwa setiap komponen telah dievaluasi, pengujian terfokus dilakukan pada perangkat lunak dari sudut pandang fungsional. Dengan melakukan ini, kesalahan berkurang dan hasilnya dijamin sesuai dengan yang dibuat. Penulis melakukan pengujian menggunakan Postman dan emulator Android. Hanya program prototipe yang digunakan untuk

pengujian, dan aplikasi dievaluasi menurut fungsionalitas dan alur programnya.

#### 5. Perawatan/Pemeliharaan

Karena *software* yang dikembangkan tidak selalu seperti itu, maka diperlukan pemeliharaan *software* termasuk pengembangannya. Mungkin masih ada bug kecil yang terlewatkan pertama kali, atau program mungkin mempunyai fitur baru yang belum ditambahkan.

#### 1.5. Kerangka Berfikir



Gambar 3. 6. Kerangka Berfikir