

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penyusunan penelitian ini, peneliti mengambil beberapa penelitian terdahulu yang berkaitan dengan penelitian peneliti, dengan tujuan untuk membandingkan penelitian terdahulu dengan penelitian peneliti.

Perangkat Lunak Informasi Orang Hilang Berbasis Mobile pada Kantor Kepolisian Daerah Lampung oleh Furqon Alfaridzi dan chairani (2023) artikel ini membahas tentang permasalahan penerimaan pengaduan orang hilang yang masih bersifat manual dari pihak kepolisian. Untuk mengatasi masalah, peneliti menggunakan algoritma *Fifo* sebagai solusi penyelesain antrian dalam pembuatan laporan pengaduan. Metode yang di gunakan peneliti menggunakan prototype untuk pengembangan perangkat lunak. Pengujian yang di gunakan peneliti menggunakan *Black Box Testing* untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang di butuhkan. Hasil penelitian ini berupa informasi orang hilang berbentuk *Mobile* yang dapat di manfaatkan pihak kepolisian Daerah Lampung untuk meningkatkan kualitas pelayanan penerimaan laporan pengaduan orang hilang.

Penelitian yang dilakukan oleh Heri Maulana dan Kasmawi (2020) dengan judul “Buku Penghubung Berbasis Android Menggunakan Metode Prototyping” latar belakang dari penelitian ini bertujuan untuk membangun sebuah aplikasi buku penghubung berbasis android yang di lengkapi dengan pengumuman, nilai harian, daftar pelajaran, dan kehadiran. Metode yang di gunakan dalam penelitian

ini menggunakan prototyping. Hasil dari penelitian ini yaitu sebuah aplikasi buku yang dapat di gunakan oleh guru dan orang tua.

Ali Fauzan (2023) melakukan penelitian dengan judul "Rancang Bangun Aplikasi Penemuan dan Kehilangan Barang Berbasis Android" penelitian ini bertujuan untuk merancang dan membuat aplikasi pencarian barang hilang berbasis android menggunakan *framework React Native* dan *data base firebase*. Metode *Waterfall* digunakan dalam pengembangan aplikasi untuk memastikan pengembangan perangkat secara sistematis. Hasil dari penelitian ini yaitu sebuah aplikasi pencarian barang hilang berbasis Android menggunakan *framework React Native* dan *database Firebase*, yang memungkinkan pengguna untuk melihat daftar thread kehilangan dan juga mempublikasikan barang yang hilang atau memberikan informasi tentang barang yang ditemukan.

Penelitian yang di lakukan oleh Azel Aldi Saputra¹ dan Saepul Reza Alparizi² (2023) dengan judul "Penerapan Model Prototype Pada Aplikasi Layanan Pengaduan Masyarakat Bantul Berbasis Android" penelitian ini membahas tentang merancang sebuah prototype aplikasi Android yang efektif dan efisien untuk pengaduan masyarakat di daerah bantul. Perancangan prototype aplikasi ini untuk memungkinkan warga mengajukan pengaduan secara mudah, cepat dan akurat, serta memungkinkan pihak berwenang untuk mengelola dan merespon pengaduan dengan lebih baik. Hasil dari penelitian ini diharapkan akan memberikan panduan bagi pengembang dalam merancang aplikasi serupa untuk mendukung partisipasi aktif masyarakat bantul.

Penelitian yang dilakukan oleh Wahida Saleng Apriyanto (2022) dengan

judul “Aplikasi Pelaporan Kendaraan Hilang Berbasis Android Pada Polres Luwu Utara” tujuan dari penelitian ini yaitu merancang dan membuat Aplikasi Pelaporan Kendaraan Hilang Berbasis Android, Pada Polres Luwu Utara yang mudah digunakan untuk membuat pelaporan kendaraan hilang yang dikirimkan oleh pengguna serta memberikan solusi untuk Kantor Kepolisian Luwu Utara untuk mempermudah masyarakat melaporkan kendaraan yang hilang hanya dengan menggunakan smartphone dan dapat diakses menggunakan android dalam laporan pencurian kendaraan. Metode yang digunakan peneliti yaitu metode *waterfall*. Aplikasi ini telah di uji dengan menggunakan metode pengujian *Black Box* dan hasil responden dihitung dengan menggunakan aplikasi *SPPS*.

Penelitian yang di lakukan oleh Alfin Riyadi dan Savitri Galih (2019) dengan judul “Perancangan Aplikasi Tracking Dalam Studi Kasus Maraknya Orang Hilang Pada Saat Naik Haji Berbasis Android” penelitian ini membahas tentang pengembangan aplikasi pencarian terhadap orang hilang dengan memanfaatkan kecanggihan teknologi yang diterapkan pada pengembangan 22arker berbasis mobile. Pembangunan aplikasi ini menggunakan 22arker operasi android dengan memanfaatkan pelacakan longitude dan latitude dari keberadaan device yang terhubung dengan koneksi internet yang berguna memperbaharui lokasi dalam GPS dan data dari lokasi disimpan di firebase database. Metode yang di gunakan dalam proses pengembangan 22arker adalah *SDLC (Sytem Developmen Life Cycle)*. Aplikasi tracking ini dapat menyelesaikan masalah dengan mengandalkan 3 role yaitu Admin, Jemaah, dan Muthawif.

2.2. Landasan Teori

2.2.1 Defenisi *Android*

Android merupakan *Operation System* (OS) yang berbasis JAVA, dan berjalan di kernel LINUX, yang bersifat *Open Source*, Android dimiliki oleh Google. Karena bersifat *Open Source*, maka pengembang dapat membuat berbagai aplikasi *android*. Selain itu media komunikasi *smartphone* memiliki berbagai fitur yang menggunakan layar sentuh, User Interface atau tampilan antarmuka yang bagus dan baik, kamera, *Internet*, Wi-Fi, GPS, *Bluetooth*, aplikasi aplikasi canggih, RAM dan Prosesor yang mumpuni serta penyimpanan Internal yang besar. *Android* menggunakan Bahasa pemrograman baru Java pada umumnya. Sedangkan itu, Java merupakan sebuah bahasa pemrograman yang banyak sekali dipakai karena berorientasi objek dan tersusun oleh *class* serta mempunyai *variable* dan *method* (OOP), dan Java bersifat gratis.[5].



Gambar 2.1. Logo Sistem Operasi Android

2.2.2 Defenisi *Android Studio*

Android Studio adalah sebuah *Software Tools Integrated Development Environment* (IDE) untuk *platform Android*. *Android Studio* diluncurkan pada tanggal 16 mei 2013 pada konferensi Google I/O oleh produk manajer Google, *Ellie Powers*. *Android Studio* bersifat *free* di bawah *Apache License 2.0*. *Android*

Studio berbasis *JetBrains' IntelliJ IDEA*, *studio* didesain khusus untuk *android studio*. [6]

Android Studio merupakan salah satu IDE (Integrated Development Environment) untuk membuat Aplikasi Android, android studio adalah lingkungan pengembangan Android baru berdasarkan IntelliJ IDEA. Mirip dengan Eclipse dengan ADT Plugin, Android Studio menyediakan alat pengembang Android terintegrasi untuk pengembangan dan debugging. [7]



Gambar 2.2 Logo Android Studio

2.2.1 Defenisi Kotlin

Pada Google I/O 2017, Google mengumumkan Kotlin, bahasa pemrograman baru yang resmi digunakan untuk mengembangkan aplikasi Android. Kotlin merupakan bahasa pemrograman yang dikembangkan oleh Jet Brains, serta mengembangkan Android Studio IDE. Bahasa Kotlin merupakan pengembangan dari bahasa Java yang sebelumnya. Bahasa Kotlin memiliki fitur bahasa yang lebih baru daripada bahasa Jafa. Sejak 2017, pengembang Android dapat mulai menulis aplikasi Android dari awal menggunakan Kotlin, mengembangkan aplikasi Android mereka yang sudah ada yang ditulis dalam Java

dengan menambahkan kode Kotlin (mungkin berkat interoperabilitas antara dua bahasa), atau memigrasikan aplikasi Android merekai Java ke Kotli. [8]

Dengan migrasi kode Java ke Kotlin, pengembang dapat menggunakan fitur bahasa pemrograman (misalnya extension functions, lambdas, smart casts) yang tidak tersedia dalam pengembangan Java untuk Android, dan untuk mendapatkan kode yang lebih aman (menghindari Null Pointer Exceptions).



Gambar 2.3 Logo Kotlin

2.2.2 Defenisi Aplikasi Mobile

Menurut Pressman dan Bruce aplikasi *mobile* merupakan aplikasi yang telah dirancang khusus untuk *platform mobile* (misalnya *Ios*, *android*, atau *windows mobile*). Dalam banyak kasus, aplikasi *mobile* memiliki *user interface* dengan mekanisme interaksi unik yang disediakan oleh *platform mobile*, *interoperabilitas* dengan sumber daya berbasis *web* yang menyediakan akses ke beragam informasi yang relevan dengan aplikasi, dan kemampuan pemrosesan

sistem untuk pengumpulan, analisis, dan format informasi dengan cara yang paling cocok untuk *platform mobile*. Selain itu aplikasi *mobile* menyediakan kemampuan penyimpanan persisten dalam *platform*. [9]

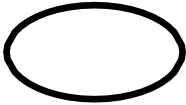
2.2.3 Defenisi Unified Modeling Language (UML)

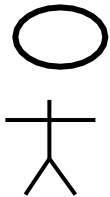
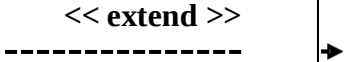
Unified Modeling Language (UML) adalah standarisasi internasional untuk notasi yang berbentuk grafik, ULM dikemukakan oleh Grady Booch, Ivar Jacobson dan James Rumbaugh. UML Versi terbaru yaitu 2.5, terdiri dari lima belas diagram. Diagram-diagram tersebut dibagi kedalam dua kelompok, yaitu structure diagram dan behavior diagram. Structure diagram terdiri dari package, object, component, class, deployment, composite structure dan profile diagram. Behavior diagram menggambarkan hubungan dinamis antara objek serta mewakili 26 informasi bisnis. Dengan demikian UML (*Unified Modeling Language*) adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah dengan menggunakan diagram dan teks-teks pendukung. [10]

1. Use Case Diagram

Use Case digunakan untuk mendeskripsikan sebuah interaksi antara satu atau lebih dengan informasi yang akan dibuat. Berikut symbol-simbol dari *use case diagram* dapat dilihat pada tabel 2.

Tabel. 2.1 *Use Case Diagram*

Simbol	Nama	Keterangan
	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan

		antar unit dan sistem.
	<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.
	<i>Association</i>	Komunikasi antar sistem dan <i>Use Case</i> yang berpartisipasi.
	<i>Extend</i>	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambah dapat berdiri sendiriwalau tanpa <i>Use Case</i> tambahan.
	<i>Include</i>	Relasi <i>Use Case</i> tambahan ke sebuah <i>Use Case</i> dimana <i>Use Case</i> yang ditambahkan memerlukan <i>Use Case</i> ini untuk menjalankan fungsinya.
		
	<i>Generalization</i>	Hubungan generalisasi dan spesialisasi antara dua buah <i>Use Case</i> yang mana fungsi yang satulebih umum dari yang lainnya.

2. Activity Diagram

Activity Diagram menggambarkan *workflow* (aliran kerja) dari sebuah 28arker atau proses bisnis atau menu yang ada pada perangkat lunak.

Berikut Simbol-simbol dari *activity diagram* dapat dilihat pada tabel 2.2

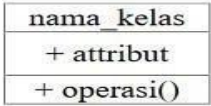
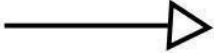
Tabel 2.2 Activitiy Diagram

Simbol	Nama	Keterangan
	Status awal	Status awal aktivitas pada sebuah diagram aktivitas memiliki sebuah status awal.
	Aktivitas	Aktivitas yang dilakukan 28arker, aktivitas biasanya diawali dengan kata kerja.
	Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Pengabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
	Status akhir	Status akhir yang dilakukan 28arker, sebuah diagram aktivitas memiliki sebuah status akhir.

3. Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan di buat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut Simbol-simbol dari class diagram

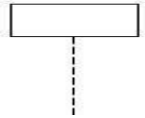
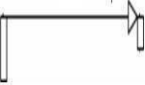
Tabel 2.3 Class Diagram

Simbol	Nama	Keterangan
	<i>Class</i>	Kelas pada struktur 29arker.
	<i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i> .
	<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> .
	<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi/spesialisasi (umum khusus).
	<i>Depedency</i>	Relasi antar kelas dengan makna kebergantung.
	<i>Aggregation</i>	Relasi antarkelas dengan makna semua bagian.

4. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Berikut simbol-simbol dari *Sequence* dapat dilihat pada table 2.

Tabel 2.4 Sequence diagram

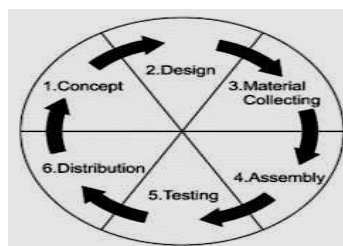
Simbol	Nama	Keterangan
	<i>Lifeline</i>	Objek entity, antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

2.2.4 Multimedia Development Life Cycle (MDLC)

Metode MDLC merupakan metode yang dirancang dan dikembangkan pada suatu aplikasi yang merupakan gabungan dari beberapa media seperti gambar, suara, video, animasi dan lainnya.[11] Penelitian ini menggunakan pendekatan *Multimedia development life cycle* sehingga fase penelitian yang dilakukan adalah sebagai berikut:

1. Metode Pengonsepan (*concept*). Tahapan ini merupakan tahap untuk menentukan tujuan dan target pengguna dari aplikasi yang akan dibuat. Penelitian ini bertujuan untuk membuat aplikasi orang hilang sebagai media informasi pada Kapolsek Rantepao berbasis Android.
2. Metode Perancangan (*design*). Pererancangan 30arker ini menggunakan UML (*Unified Modeling Language*) sehingga diagram yang digunakan ialah *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, *Class Diagram* dan *User Interface* pada Aplikasi.
3. Metode Pengumpulan Materi (*material collecting*). Pada tahap pengumpulan materi digunakan untuk mengumpulkan bahanbahan yang nantinya akan digunakan pada aplikasi. Diantaranya seperti desain gambar

4. yang akan dijadikan objek 3D, desain objek 3D sebagai informasi pada aplikasi.
5. Metode Pembuatan (*assembly*). Tahap selanjutnya adalah pembuatan, aplikasi yang akan dibuat didasarkan pada tahap perancangan agar hasilnya sesuai dengan yang diharapkan.
6. Metode Pengujian (*testing*). Pada aplikasi ini dilakukan pengujian dengan *Black Box Testing*, hasil dari pengujian ini bertujuan untuk mengetahui letak kesalahan yang mungkin terjadi saat aplikasi berjalan dan memastikan bahwa segala komponen ataupun tombol pada aplikasi telah berfungsi sebagaimana fungsinya.
7. Metode Distribusi (*distribution*). Tahap ini merupakan tahap terakhir dalam pembuatan aplikasi. Pendistribusian dilakukan apabila aplikasi telah layak pakai dan sesuai dengan tujuan dibuatnya aplikasi. Hasil akhir dari aplikasi ini akan berbentuk file APK, file tersebut akan disimpan pada media penyimpanan online sehingga dapat dengan mudah diunduh atau diakses oleh masyarakat.



Gambar 2.4 Tahapan metode MDLC

2.2.5 Model Prototype

Prototype merupakan versi awal dari 32arker perangkat lunak yang digunakan untuk mendemonstrasikan konsep-konsep, percobaan rancangan, dan menentukan lebih banyak masalah dan solusi yang memungkinkan. Sistem *prototype* memperbolehkan pengguna untuk mengetahui bagaimana 32arker berjalan dengan baik. [12]

Pengguna metode *prototype* didalam penelitian ini bertujuan agar peneliti mendapatkan gambaran aplikasi yang akan dibangun melalui tahap pembangunan aplikasi *prototype* terlebih dahulu yang akan dievaluasi oleh user. Aplikasi *prototype* yang telah dievaluasi oleh user selanjutnya akan dijadikan acuan untuk membuat aplikasi yang dijadikan produk akhir sebagai output dari penelitian. Berikut Gambaran dari metode yang di gunakan. Tujuan utama dari *prototype* secara garis besar dapat dikelompokkan ke dalam tiga bagian, yaitu:

- 1) Membantu pengembangan persyaratan, jika tidak dapat ditentukan dengan mudah.
- 2) Mengesahkan pengembangan, khususnya dengan customer atau langganan dan user yang potensial. Menyajikan sebagian tempat pengembangan jika menggunakan strategi penelitian evolusi *prototype*. Bagi Perangkat lunak yang telah di uji dan diterima pelanggan siap untuk digunakan.

1. Jenis-Jenis *prototype*

- a. *Feasibility prototype*: Digunakan untuk menguji kelayakan dari teknologi yang akan digunakan untuk system informasi yang akan disusun.

Requirement Prototype : Digunakan untuk mengetahui kebutuhan aktivitas bisnis user.

- b. *Desain Prototype*: Digunakan untuk mendorong perancangan 33arker informasi yang akan digunakan.
- c. *Implementation prototype* adalah lanjutan dari rancangan protipe, prototype tersebut langsung disusun sebagai suatu *system* informasi yang akan digunakan

2. Teknik –teknik *prototype*

- a. Perancangan mode
- b. Perancangandialog
- c. Simulas

3. Kelebihan dan kekurangan *prototype*

Adapun kelebihan dari *prototype* adalah:

- 1. End user dapat berpartisipasi aktif
- 2. Penentuan kebutuhan lebih mudah diwujudkan
- 3. Mempersingkat waktu pengembangan SI
- 4. Adanya komunikasi yang baik antara pengembang danpelanggan
- 5. Pengembang dapat bekerja lebih baik dalam menentukankebutuhan pelenggan.
- 6. Pelanggan berperan aktif dalam pengembangan system
- 7. Lebih menghemat waktu dalam dalam pengembagan system

8. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkan.

Adapun kelemahan dari *prototype* adalah:

1. Proses analisis dan perancangan terlalu singkat
2. Mengesampingkan kreatif pemecahan masalah
3. Kurang fleksibel dalam menghadapi perubahan
4. Prototype yang di hasilkan tidak selamanya mudah dirubah
5. Prototype terlalu cepat selesai

2.2.6 Basis Data

Basis data atau *database*, berasal dari kata yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat berkumpul, data adalah wujud dari dunia nyata yang mewakili suatu objek. Basis data adalah kumpulan file-file yang saling berhubungan yang biasa ditunjukkan dengan kunci dari tiap file yang ada. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. [13]

A. Konsep Dasar Basis Data

Basis data adalah kumpulan informasi yang disimpan didalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut.

Sistem basis data adalah kumpulan dari program aplikasi yang berinteraksi dengan basis data bersama dengan *Database Management System* (DBMS) dan

basis data itu sendiri.

B. Komponen Basis Data

Sistem basis data terdiri dari 4 komponen pokok, yaitu:

1. Basis data (*database*)

Basis data dengan ciri-ciri:

- a. Data disimpan secara terintegrasi (*integrated*) Terintegrasi maksudnya adalah data terkumpul dari berbagai macam *file* yang disusun dengan cara menghilangkan bagian- bagian yang data yang berulang (*redundansi*).
- b. Data dapat dipakai secara bersama-sama (*shared*) *Shared* yaitu masing-masing bagian dari basis data dapat diakses oleh *user* dalam waktu yang bersamaan, untuk aplikasi yang berbeda.

2. perangkat keras (*hardware*)

Terdiri dari semua perangkat keras komputer yang digunakan untuk pengelolaan basis data tersebut. Adapun perangkat keras yang terdapat dalam sebuah basis data, terdiri dari :

- a. Komputer (satu perangkat untuk system stand alone atau beberapa computer untuk jaringan).
- b. Perangkat untuk menyimpan data permanen (*harddisk*).
- c. Perangkat komunikasi (untuk sistem jaringan).

3. Perangkat lunak (*software*)

Berfungsi sebagai perantara (*interface*) antara *user* dengan data fisik pada basis data, dapat berupa:

- a. *Data base Managemen System (DBMS)*
- b. Program-program aplikasi
- c. Pemakai (*user*)

Merupakan pengguna basis data yang berinteraksi secara tidak langsung dengan basis data melalui program aplikasi basis data (DBMS). *User* terbagi menjadi 3 klasifikasi yaitu:

1. *Database administrator (DBA)*, yang membuat basis data dan mengontrol akses ke basis data.
2. *Programmer*, yang membuat aplikasi basis data yang digunakan oleh DBA dan pemakai akhir.
3. Pemakai akhir (*end user*), yang melakukan penambahan, penghapusan, pengubahan, dan pengaksesan data

2.2.9 Firebase

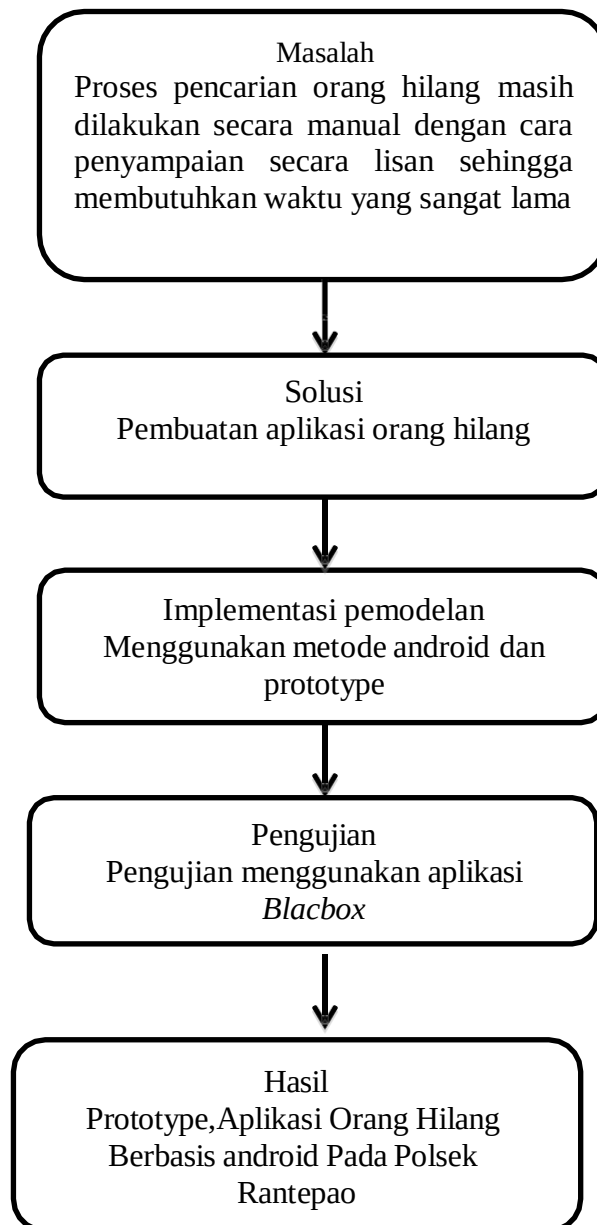
Firebase adalah penyedia layanan cloud dengan back-end sebagai servis yang berbasis di San Fransisco, California. Firebase membuat sejumlah produk untuk pengembangan aplikasi mobile ataupun web. Firebase didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan cloud database secara realtime di tahun 2012. Menurut B. Octavianus dalam bukunya yang berjudul *Android* apa itu *Firebase* ? *Firebase* adalah *Backend as a service (BaaS)* yang saat ini yang dimiliki oleh *Google*. *Firebase* merupakan solusi yang ditawarkan oleh *Google* untuk mempermudah pengembangan aplikasi *mobile*. *Firebase* merupakan sebuah solusi perpesanan lintas-*platform* yang memungkinkan mengirimkan pesan dan pemberitahuan dengan terpercaya tanpa

biaya. *Firebase* memiliki produk utama, yaitu menyediakan database *realtime* dan *backend* sebagai layanan (*Backend as a Service*). Layanan ini menyediakan pengembang aplikasi API yang memungkinkan aplikasi data yang akan disinkronisasi di klien dan disimpan di *cloud Firebase* ini. *Firebase* menyediakan library untuk berbagai client platform yang memungkinkan integrasi dengan *Android*, *Ios*, *JavaScript*, *Java*, *Objective- C* dan *Node* aplikasi *Js* dan dapat juga disebut sebagai layanan *DbaaS(Database as a Service)* dengan konsep *realtime*. [14]

2.2.10 Metode Pengujian *Blackbox*

Metode *Black box Testing* merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengujian pada spesifikasi fungsional program. Proses *Black Box Testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap formnya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan. Pengujian dengan metode *Blackbox Testing* memiliki dua teknik yaitu *Equivalence Partitioning* dan *Boundary Value Analysis*. Pada penelitian ini menggunakan teknik *equivalence partitioning*. Pengujian *equivalence partitioning* merupakan tahapan pengujian melalui klasifikasi dan pengelompokan berdasarkan fungsinya. Dalam pengujian ini tahapan yang akan dilakukan adalah diawali dengan melakukan penentuan *test case* perangkat lunak untuk menentukan perangkat lunak informasi sebagai *test case* kemudian diuji dengan metode *equivalence partitioning*. [15]

2.3. Kerangka Pikir



Gambar 2.1 Kerangka Pikir