

## **BAB 2**

### **TINJAUAN PUSTAKA**

#### **2.1 Landasan Teori**

Landasan teori adalah penjelasan definisi, konsep, proposisi yang telah disusun rapi, dan sistematis tentang variable-variable dalam sebuah penelitian yang berkaitan dalam pembangunan aplikasi mobile android untuk menyebarkan informasi keadaan sekitar. Landasan teori ini akan menjadi dasar yang kuat dalam penelitian yang akan dilakukan. Serta teori – teori terkait yang mendukung penelitian akan dibahas pada bab ini.

#### **2.2 Aplikasi Mobile**

Aplikasi Mobile adalah 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.

Aplikasi dibuat untuk melayani kebutuhan akan berbagi aktivitas. Jika akan mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman yaitu *language software*, yang dapat berbentuk *assambler*, *complier* atau *interpreter*. Jadi *language software* merupakan bahasanya dan program yang ditulis merupakan program aplikasinya. *Language software* berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkan dalam bahasa mesin supaya bisa dimengerti oleh komputer [4].

Dalam pembuatan aplikasi penulis akan membuat aplikasi ini dalam perangkat mobile, karena lebih cocok diatasi dengan kemampuan perangkat mobile dan dapat dijalankan atau digunakan walaupun pengguna berpindah – pindah dari satu tempat ke tempat lain, serta mempunyai ukuran yang kecil.

### 2.3 Media Sosial

Media sosial adalah sebuah media untuk bersosialisasi satu sama lain dan dilakukan secara online yang memungkinkan manusia untuk saling berinteraksi tanpa dibatasi ruang dan waktu, dengan media sosial manusia dimungkinkan untuk berkomunikasi satu sama lain dimanapun mereka berada dan kapanpun, tidak peduli seberapa jauh jarak mereka, dan tidak peduli siang atau pun malam. Media sosial mengalami perkembangan yang sangat menakjubkan, dimulai dari munculnya Twitter atau Instagram dan media sosial lainnya. Media sosial memiliki banyak manfaat, salah satunya sebagai media untuk menyebarkan informasi.

Para pengguna media sosial bisa melakukan komunikasi atau interaksi, berkiriman pesan, baik pesan teks, gambar, audio hingga video, dan saling berbagi atau *sharing*. Media sosial mengajak siapa saja yang tertarik untuk berpartisipasi dengan memberi kontribusi dan feedback secara terbuka, memberi komentar, serta membagi informasi dalam waktu yang cepat dan tak terbatas. Saat teknologi internet dan mobile phone makin maju maka media sosial pun ikut tumbuh dengan pesat. Kini untuk mengakses Twitter atau Instagram, bisa dilakukan dimana saja dan kapan saja hanya dengan menggunakan sebuah mobile phone. Demikian cepatnya orang bisa mengakses media sosial mengakibatkan terjadinya fenomena besar terhadap arus informasi tidak hanya di negara - negara maju, tetapi juga di Indonesia. Karena kecepatannya media sosial juga mulai tampak menggantikan peranan media massa konvensional dalam menyebarkan berita-berita. Kenapa media sosial? Karena, masyarakat sekarang ini lebih banyak menggunakan media sosial daripada harus membaca koran atau menonton televisi karena dianggap lebih cepat, dan mudah.

Saat ini ada banyak media sosial yang dapat kita temui ketika kita mengakses internet. Untuk itu kita perlu mengetahui jenis - jenis media sosial. Menurut Ron Jones, media sosial terbagi menjadi empat jenis, yaitu :

### 1. *Social News*

Social news menyediakan fitur berita dimana pembacanya dapat memilih topik – topik berita dan memberi komentar pada berita tersebut.

Contoh : Digg, Sphinn, Newsvine dan BallHype.

### 2. *Social Sharing*

Social sharing menyediakan fitur untuk membuat, mengunggah dan membagi foto dan video dengan orang lain.

Contoh : Flickr, Snapfish, YouTube dan Jumpcut.

### 3. *Social Networks*

Social networks memberi kesempatan kepada orang - orang untuk saling terhubung antara satu dengan yang lainnya. Ketika terhubung, kita dapat terus mengikuti up-date informasi dari yang bersangkutan, seperti informasi kontak, kesukaan, posting.

Contoh : Facebook, LinkedIn, MySpace dan Twitter.

### 4. *Social Bookmarking*

Disini kita dapat menemukan site bookmark dan informasi yang menarik. Kita dapat menyimpan bookmarks online dan mengaksesnya dari manapun atau membaginya dengan yang lain.

Contoh : Delicious, Faves, StumbleUpon, BlogMarks dan Diigo.

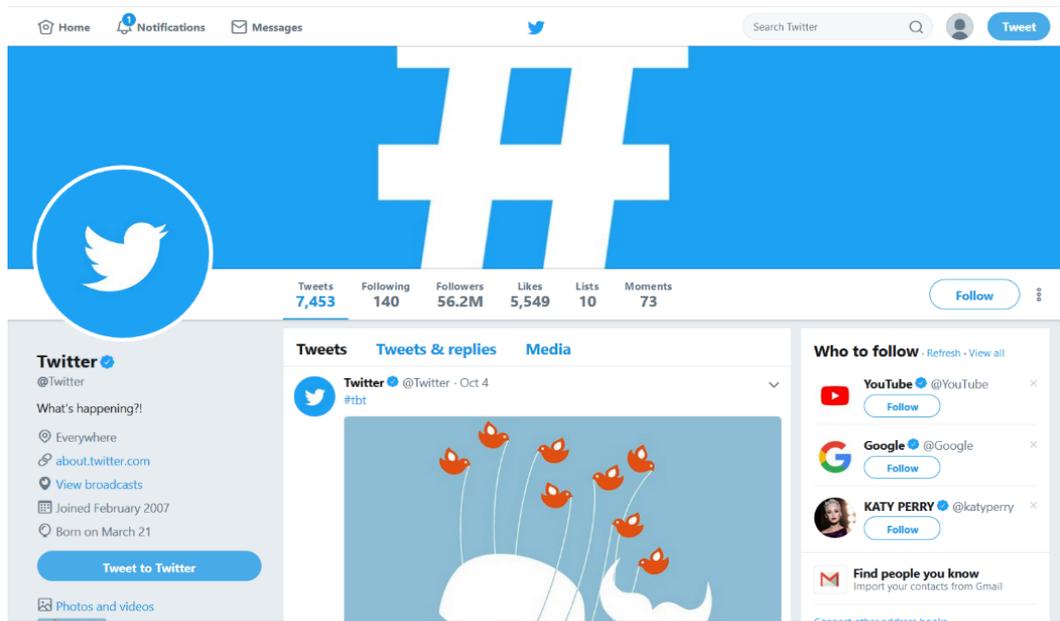
Seperti telah disebutkan di atas, bahwa setidaknya ada empat jenis media sosial. Saat ini media sosial yang paling populer dan banyak digunakan oleh masyarakat Indonesia adalah Twitter, dan Instagram. Masing - masing media sosial tersebut memiliki kekhasan. Salah satu diantaranya akan dibahas disini.

### 2.3.1 Twitter

Twitter adalah sebuah media sosial yang memungkinkan penggunanya untuk mengirim dan membaca pesan berbasis teks yang berkembang pesat karena penggunanya dapat berinteraksi dengan pengguna lainnya baik dari komputer ataupun perangkat mobile mereka, dari manapun dan kapanpun.

Twitter sebagai sebuah media sosial memberikan akses kepada penggunanya untuk mengirimkan sebuah pesan singkat yang terdiri dari 140 karakter, akan tetapi pada tanggal 07 November 2017 bertambah hingga 280 karakter yang disebut dengan (*Tweet*). *Tweet* sendiri bisa terdiri dari pesan teks dan foto. Melalui *Tweet* inilah pengguna Twitter dapat berinteraksi lebih dekat dengan pengguna Twitter lainnya dengan mengirimkan tentang apa yang sedang mereka pikirkan, apa yang sedang dilakukan, tentang kejadian yang baru saja terjadi, tentang berita terkini serta hal lainnya. Apabila dalam *Facebook* dikenal dengan pertemanan, maka di *Twitter* istilah yang dipergunakan adalah *Follower* dan *Following*. Hanya ada dua setting akun untuk twitter, yaitu *Protected* (dikunci) dan tidak dikunci. Apabila sebuah akun disetting *Protected*, ini artinya hanya *Follower* yang di *Follow* kembali atau (*Follow Back*) yang akan dapat melihat *Tweet* pemilik akun. Lain halnya apabila akun tidak dikunci, maka setiap orang baik *Follower* maupun bukan *Follow* dapat melihat semua *Tweet* dari akun tersebut. Selain itu kita juga dapat mengunggah foto dan berbagi link file [5].

Digunakannya media sosial *Twitter* dalam menyebarkan informasi keadaan sekitar karena media sosial *Twitter* mempunyai fitur mencari menggunakan hastag (#) yang berguna untuk mempermudah dalam mencari suatu tema, kegiatan atau informasi, sehingga masyarakat yang ingin mencari informasi dapat lebih mudah dalam mendapatkan suatu informasi.



**Gambar 2. 1 Media Sosial Twitter**

## 2.4 Android



**Gambar 2. 2 Logo Android**

Android adalah sistem operasi yang digunakan di smartphone dan juga tablet PC berbasis linux yang terdiri dari sistem operasi, android berfungsi sebagai penghubung (*device*) antara pengguna dan perangkat keras pada smartphone atau alat elektronik tertentu. Sehingga, hal tersebut memungkinkan pengguna dapat berinteraksi dengan *device* dan menjalankan berbagai aplikasi mobile. sebuah aplikasi dengan aplikasi lain. Android tidak terikat ke satu merek Handphone saja, hal tersebut tidak terlepas dari adanya ikatan kerjasama antara pihak Android dengan berbagai macam perusahaan teknologi raksasa seperti Samsung, ASUS, Xiaomi, Cross, HTC, Sony, Oppo, Nokia, Coolpad, Leovo, dan Motorola yang kini menggunakan Android sebagai platform ponsel mereka [6]. Fungsinya pun sama seperti sistem operasi pada umumnya yaitu seperti sistem operasi iOS, BlackBerry OS, Windows Phone, dan Symbian OS. Pada awalnya Android baru dirilis pada bulan Oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White dibawah sebuah perusahaan bernama Android Inc di Palo Antom, California. Sebelum akhirnya diakuisisi oleh Google pada tahun 2005.

Android dibuat dengan basis Linux, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan. Kelebihan Android yaitu *open source*. Hal tersebut tentunya memberikan ruang bagi siapapun untuk mengembangkan sistem operasi yang satu ini. *User Friendly*. Mulai dari tampilan dan cara kerja,

Android tergolong sistem operasi yang mudah digunakan oleh siapapun. Dengan kata lain, seorang pengguna Android tidak membutuhkan keahlian khusus untuk melakukan pengoprasian [6].

Android telah mengalami beberapa pembaruan sejak pertama kali rilis, berikut ini jenis - jenis versi dan *Codename* Android [7] :

1. Android Donut (1.6) - (15 September 2009)

Versi ini memperkenalkan fitur *quick search box*, dimana pengguna dapat mencari konten di website dan dalam ponsel dalam satu tempat. Sebelum ada Google Play, ada Android Market. Dan di versi Donut, Android Market mendapatkan rancangan desain baru yang menampilkan aplikasi gratis dan berbayar terbaik. Tidak hanya itu, Donut juga mempelopori kemampuan agar Android bisa digunakan pada berbagai ponsel dengan ukuran layar berbeda.

2. Android Eclair (2.0 / 2.1) – (12 Januari 2010)

Versi Android awal yang mulai dipakai oleh banyak smartphone, fitur utama Eclair yaitu perubahan total struktur dan tampilan user interface dan merupakan versi Android yang pertama kali mendukung format HTML5.

3. Android Froyo / Frozen Yogurt (2.2) – (20 Mei 2010)

Android 2.2 dirilis dengan 20 fitur baru, antara lain peningkatan kecepatan, fitur Wi-Fi hotspot tethering dan dukungan terhadap Adobe Flash.

4. Android Gingerbread (2.3) – (6 Desember 2010)

Perubahan utama di versi 2.3 ini termasuk update UI, peningkatan fitur *soft keyboard & copy/paste*, *power management*, dan support *Near Field Communication* (NFC), memungkinkan pengguna untuk membaca tag NFC yang tertanam dalam poster, stiker, atau iklan

5. Android Honey Comb (3.0, 3.1, dan 3.2) – (15 Juli 2011)

Merupakan versi Android yang ditujukan untuk gadget / device dengan layar besar seperti Tablet PC; Fitur baru Honeycomb yaitu dukungan terhadap prosesor multicore dan grafis dengan hardware acceleration, Quick Settings (Pengaturan Cepat) yang memberikan pengguna kemudahan untuk mengecek informasi penting seperti tanggal, waktu, daya baterai, dan status koneksi perangkat dalam satu tempat.

6. Android Ice Cream Sandwich (4.0) – (19 Oktober 2011)

Android Ice Cream Sandwich diumumkan pada 10 Mei 2011 di ajang *Google I/O Developer Conference* (San Francisco) dan resmi dirilis pada tanggal 19 Oktober 2011 di Hongkong. “Android Ice Cream Sandwich” dapat digunakan baik di smartphone ataupun tablet. Fitur utama Android ICS 4.0 ialah *Face Unlock*, *Android Beam* fitur komunikasi area dekat yang memungkinkan dilakukannya pertukaran jarak pendek bookmark web, info kontak, arah, video YouTube, dan data lainnya, perubahan major *User Interface*, dan ukuran layar standar (native screen) beresolusi 720p (*high definition*).

7. Android Jelly Bean (4.1, 4.2, dan 4.3) – (24 Juli 2013)

Android Jelly Bean lebih menfokuskan fiturnya ke peningkatan User Interface yang lebih lancar dan responsif. Di versi ini juga menandai hadirnya fitur *Google Now* dimana pengguna bisa mendapatkan laporan cuaca serta waktu perjalanan sebelum keluar rumah. yang memberikan saran dan rekomendasi berdasarkan data-data yang tersimpan (kontak, kalender, lokasi) di handphone.

8. Android KitKat (4.4) – (31 Oktober 2013)

Android KitKat menfokuskan pada manajemen memory sehingga device low-end yang memiliki memoery rendah tetap dapat berjalan lancar dan minim lagg. Selain itu hadir beberapa fitur lain seperti kemampuan untuk

menggunakan semua tampilan layar (immersive mode), dukungan Mobile Printing, penambahan dukungan sensor baru.

9. Android Lollipop (5.0) – (12 November 2014)

Perubahan desain User interface menggunakan material design, perubahan runtime dari dalvik menjadi Android Runtime (ART), dan hadirnya fitur penghemat baterai.

10. Android Marshmallow (6.0) – (5 Oktober 2015)

Perubahan permission (izin aplikasi), manajemen daya untuk baterai bernama *Doze*, mendukung secara native sensor sidik jari, dan USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB

11. Android Nougat (7.0) – (9 Maret 2016)

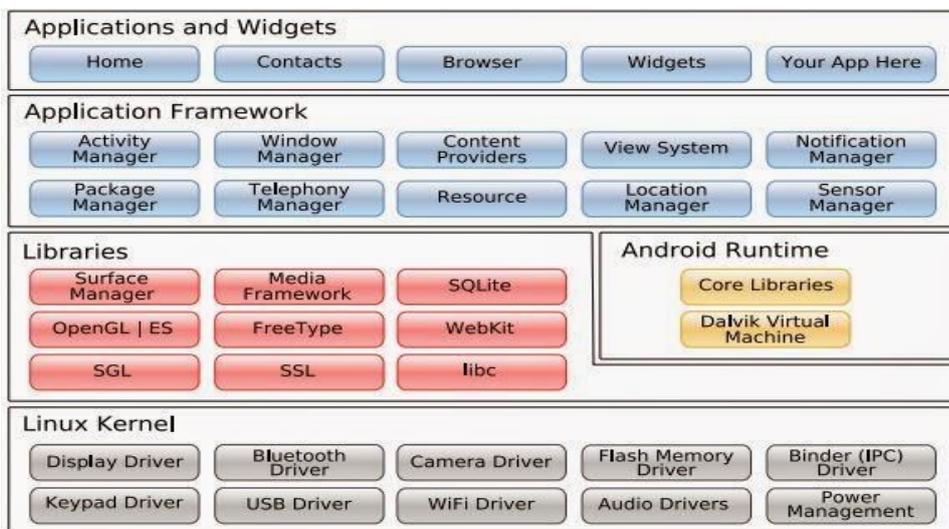
Kehadiran split-screen multi-window mode yang mendukung aplikasi apun di tata letak desktop, seamless updates, penghemat baterai Doze, data saver penghemat kuota internet, penambahan emoji baru.

12. Android Oreo (8.0) – (21 Maret 2017)

Android Oreo akan meminimalkan aktivitas di latar belakang pada aplikasi yang jarang digunakan. Hal ini dilakukan agar bermacam aktivitas di latar belakang itu tak menyedot daya baterai yang berlebihan. Fitur Android 8.0 Oreo selanjutnya ialah **Picture-in-picture**. Fitur ini yaitu dapat nonton video sambil menjalankan aplikasi lainnya, tentunya menjadikan kita lebih produktif, bisa nonton video dan tetap sambil kerja.

## 2.5 Arsitektur Android

Secara umum, arsitektur Android terdiri dari beberapa lapisan software, yaitu *Applications and Widgets*, *Application Framework*, *Libraries*, *Android Run-time* sebagai lapisan perantara (*middleware*), serta *Linux Kernel* yang membungkus di bawahnya. Untuk penjelasan setiap lapisan tersebut secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut [7]. :



N. Safaat, "Android," in *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*, Bandung, Informatika, 2012, p. 7.

**Gambar 2. 3 Arsitektur Sistem**

### 1. *Application dan Widgets*

Application dan Widgets ini adalah layer di mana kita berhubungan dengan aplikasi saja, di mana biasanya *user* men-download aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut.

### 2. *Application Frameworks*

Android adalah "*Open Development Platform*" yaitu Android menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, menambahkan status notifications, dan sebagainya.

### 3. *Libraries*

Libraries ini adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses libraries untuk menjalankan aplikasinya.

### 4. *Android Runtime*

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi Linux yang terbagi menjadi dua bagian yaitu *Core Libraries* dan *Dalvik Virtual Machine*.

### 5. *Linux Kernel*

Linux Kernel adalah *layer* dimana inti dari *operating system* dari Android itu berada.

Alasan digunakannya Android pada penelitian ini karena Android menggunakan sistem open source alias tidak berbayar, cepat dan responsive, user friendly karena android tergolong sistem operasi yang mudah digunakan oleh orang awam sekalipun. Dengan kata lain, seorang pengguna android tidak membutuhkan keahlian khusus untuk melakukan pengoperasian dan menjadi sebuah sistem operasi perangkat bergerak (*mobile device*) yang paling banyak digunakan di dunia, serta android Dari sample 100 orang yang sudah dilakukan didapat data 83,4% orang pengguna perangkat Android. Teknologi yang terdapat pada perangkat Android bisa dimanfaatkan untuk membantu permasalahan yang sedang diteliti saat ini.

## **2.6 Android SDK (*Software Development Kit*)**

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java [7].

## **2.7 Firebase Cloud Messaging**

Firestore Cloud Messaging (FCM) adalah solusi pengiriman pesan lintas platform yang memungkinkan kita mengirimkan pesan dengan tepercaya tanpa biaya. Dengan FCM, kita dapat memberi tahu aplikasi klien bahwa email baru atau data lainnya tersedia untuk disinkronkan. Kita dapat mengirim pesan notifikasi untuk mendorong interaksi kembali dan retensi pengguna. Untuk kasus penggunaan seperti messaging instan, pesan dapat mentransfer payload hingga 4 KB ke aplikasi klien.

Pada aplikasi yang akan dibangun akan terdapat fitur yang memanfaatkan Firestore Cloud Messaging untuk mengirimkan pesan notifikasi yang berisi informasi baru.

### **2.7.1 Firestore Realtime Database**

Firestore Realtime Database adalah database NoSQL yang di-host di cloud dan dapat digunakan untuk menyimpan dan menyinkronkan data antarpengguna secara real-time. Realtime Database dikirimkan dengan SDK seluler dan web sehingga dapat membuat aplikasi tanpa memerlukan server. Realtime Database juga dapat menjalankan kode backend yang merespons peristiwa yang dipicu oleh database menggunakan Cloud Function for Firestore.

Firestore Realtime Database memungkinkan developer untuk membuat aplikasi kolaboratif dan kaya fitur dengan menyediakan akses yang aman ke database, langsung dari kode sisi klien. Data disimpan di drive lokal. Bahkan saat offline sekalipun, peristiwa realtime terus berlangsung, sehingga pengguna akhir akan merasakan pengalaman yang responsif. Ketika koneksi perangkat pulih kembali, Realtime Database akan menyinkronkan perubahan data lokal dengan update jarak jauh yang terjadi selama klien offline, sehingga setiap perbedaan akan otomatis digabungkan.

Realtime Database menyediakan bahasa aturan berbasis ekspresi yang fleksibel, atau disebut juga Aturan Keamanan Firestore Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firestore Authentication, developer dapat

menentukan siapa yang memiliki akses ke data tertentu dan bagaimana mereka dapat mengaksesnya.

Realtime Database adalah database NoSQL, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait. API Realtime Database dirancang agar hanya mengizinkan operasi yang dapat dijalankan dengan cepat. Hal ini memungkinkan Anda untuk membangun pengalaman realtime yang luar biasa dan dapat melayani jutaan pengguna tanpa mengorbankan kemampuan respons. Oleh karena itu, perlu dipikirkan bagaimana pengguna mengakses data, kemudian buat struktur data sesuai dengan kebutuhan tersebut.

## **2.8 Android Studio**

Android studio merupakan *Integrated Development Environment (IDE)* atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang dirancang khusus untuk pengembangan sistem operasi Google Android. Aplikasi ini dibangun di atas sebuah perangkat lunak yang dinamakan IntelliJ IDEA milik JetBrains. Android Studio menawarkan berbagai fitur canggih yang akan meningkatkan kemampuan produktivitas dalam proses pembuatan sebuah aplikasi [6].

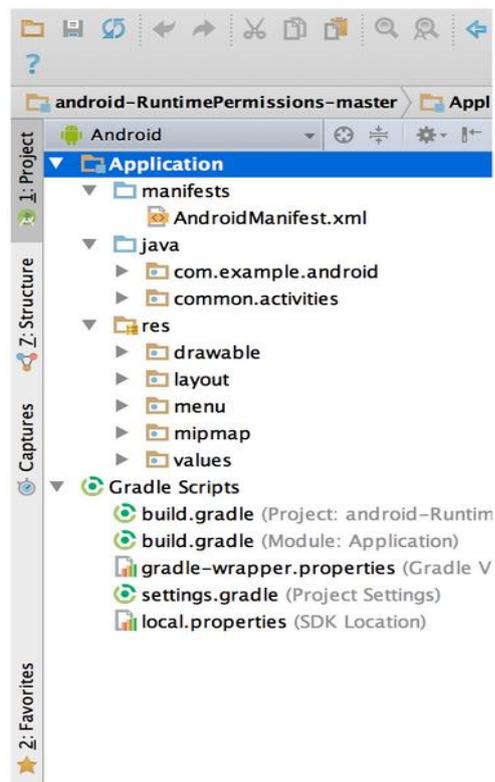
Berikut ini adalah beberapa fitur yang ditawarkan Android Studio adalah sebagai berikut :

1. Sistem versi berbasis Gradle yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
4. Instant Run untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
5. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
6. Alat pengujian dan kerangka kerja yang ekstensif.

7. Alat Lint untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk Google Cloud Platform, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.

### 2.8.1 Struktur Project

Pada dasarnya, dalam membuat sebuah projek di Android Studio terdapat 3 komponen yang saling terhubung. Komponen tersebut adalah Manifests yang mengatur tampilan antar muka dari sebuah aplikasi, Res yang menampung berbagai komponen penunjang dari manifests dan Java yang berperan dalam mengatur proses pemrograman dibalik sebuah aplikasi [6].



Sumber Gambar : <https://developer.android.com/studio/intro/?hl=id>

**Gambar 2. 4 Struktur Project Android Studio**

Pada setiap project yang ada di dalamnya berisikan satu hingga lebih modul – modul dengan berbagai file kode dan sumber daya yang beragam. Berikut adalah tiga file dasar yang membangun Modul Aplikasi :

1. Manifests

Berisikan file Android / Manifests.xml sebagai lembar kerja bawaan yang mengurus dasar tampilan project.

2. Java

Berisikan file – file kode sumber Java, termasuk Junit sebagai kode pengujian. File java berfungsi untuk menjalankan setiap fungsi yang ada pada tampilan dan display pada layer aplikasi Android.

3. Res

Berisikan file – file kode sumber daya seperti layout xml, penyimpanan gambar bit, string UI, warna dan lain sebagainya.

Di dalam file res terdapat folder resource sebagai berikut :

- a. Drawable : file bitmap (PNG, JPEG, atau GIF) dan file XML yang menggambarkan bentuk Drawable.
- b. Layout : file XML yang dikompilasi kedalam layout layar.
- c. Menu : file XML yang mendefinisikan menu aplikasi.
- d. Mipmap : ikon peluncur aplikasi
- e. Values : file XML yang mendefinisikan sumber daya dengan jenis elemen XML :

1. String.xml : berisi data text yang akan digunakan
2. Mipmap : resource tentang tema seperti toolbar untuk komponen pada aplikasi
3. Color.xml : berisi resource warna – warna dari file style.xml

## **2.9 GPS (*Global Positioning System*)**

*Global Positioning System* atau sering disingkat dengan GPS adalah sistem navigasi yang menggunakan satelit yang didesain agar dapat menyediakan posisi secara instan, kecepatan dan informasi waktu di hampir semua tempat di muka bumi, setiap saat dan dalam kondisi cuaca apapun. GPS pertama kali digunakan untuk kepentingan militer.

Penggunaan GPS untuk penentuan posisi saat ini di antaranya adalah Department of Defense (DoD) Amerika serikat yang dapat dikatakan merupakan pembuat sistem ini. Manajemennya sendiri dilaksanakan oleh US Air Force dengan panduan dari komite eksekutif DoD Positioning/Navigation. Komite ini menerima masukan dari komite yang sama dari Department of Transportation (DoT) yang bertindak sebagai suara sipil untuk urusan aturan GPS (NAPA, 1995). GPS asli hasil desain oleh US Department of Defense (DoD) terdiri atas tiga komponen utama, yaitu control segment, space segment, dan user segment.

### **2.9.1 GPS Control Segment**

Control segment GPS terdiri dari lima stasiun yang berada di pangkalan Falcon Air Force, Colorado Springs, Ascension Island, Hawaii, Diego Garcia dan Kwajalein. Kelima stasiun ini adalah mata dan telinga bagi GPS. Sinyal-sinyal dari satelit diterima oleh bagian kontrol, kemudian dikoreksi, dan dikirimkan kembali ke satelit. Data koreksi lokasi yang tepat dari satelit ini disebut data ephemeris, yang kemudian nantinya dikirimkan ke alat navigasi yang kita miliki.

### 2.9.2 GPS Space Segment

Space segment terdiri atas sebuah jaringan satelit yang terdiri dari beberapa satelit yang berada dalam orbit lingkaran yang terdekat dengan tinggi nominal sekitar 20,183 km di atas permukaan bumi. Sinyal yang dipancarkan oleh seluruh satelit tersebut dapat menembus awan, plastik dan kaca, namun tidak bisa menembus benda padat seperti tembok dan rapatnya pepohonan.

### 2.9.3 GPS User Segment

User segment merupakan bagian penerima dari sistem GPS. Bagian ini menerima data dari satelit-satelit melalui sinyal radio yang dikirimkan setelah mengalami koreksi oleh stasiun pengendali (GPS Control Segment). Segmen pengguna ini terdiri dari beberapa bagian utama yaitu : antena dan prosesor receiver yang kemudian untuk menentukan dan menampilkan posisi, kecepatan, waktu dan sebagainya.

## 2.10 Geofencing

*Geofencing* adalah teknologi yang digunakan untuk memantau objek bergerak (kendaraan, orang), dengan menggunakan GPS. Koordinat geografis dari objek secara otomatis dilacak dan secara berkala dikirim ke pusat pengawasan, melalui jaringan perangkat seluler.

Fungsi utama *geofencing* yaitu untuk melakukan pemantauan jarak jauh (monitoring) suatu perangkat mobile dari peta virtual, *geofence* juga dapat mengirim alert berupa notifikasi apabila perangkat mobile keluar atau memasuki daerah yang dibatasi *geofence* (pagar virtual). Berbagai macam teknik *geofencing* telah dikembangkan untuk berbagai kebutuhan yang berbeda [8].

## 2.11 Google Maps

*Google Maps* adalah layanan pemetaan web yang dikembangkan oleh Google. *Google Maps* dapat diakses melalui browser web atau melalui perangkat mobile. Layanan ini memberikan citra satelit, peta jalan, panorama 360°, kondisi

lalu lintas, dan perencanaan rute untuk bepergian dengan berjalan kaki, mobil, sepeda (versi beta), atau angkutan umum.

Google Maps dimulai sebagai program desktop C++, dirancang oleh Lars dan Jens Eilstrup Rasmussen pada Where 2 Technologies. Pada Oktober 2004, perusahaan ini diakuisisi oleh Google, yang diubah menjadi sebuah aplikasi web. Setelah akuisisi tambahan dari perusahaan visualisasi data geospasial dan analisis lalu lintas, Google Maps diluncurkan pada Februari 2005. Layanan ini menggunakan Javascript, XML, dan AJAX. Google Maps menawarkan API yang memungkinkan peta untuk dimasukkan pada situs web pihak ketiga, dan menawarkan penunjuk lokasi untuk bisnis perkotaan dan organisasi lainnya di berbagai negara di seluruh dunia. Google Map Maker memungkinkan pengguna untuk bersama-sama mengembangkan dan memperbarui pemetaan layanan di seluruh dunia.

Google Maps lebih dari sekedar petunjuk jalan. Ada banyak informasi tentang tempat-tempat yang berbeda yang dapat dilihat langsung di Google Maps. Misalnya, anda bisa mencari alamat dari sebuah kedai kopi lokal hanya dengan mencari namanya. Anda juga dapat melihat jam operasional kedai tersebut dan mengakses informasinya pada website. Google Maps dapat membantu Anda memberitahukan petunjuk arah kemana Anda harus pergi dan apa saja yang perlu Anda ketahui sebelum Anda sampai di sana. Google Maps berbeda dengan peta berbentuk kertas yaitu tidak terbatas pada satu negara atau daerah. Kenyataannya, Google Maps menyediakan peta hampir setiap negara di dunia. Google Maps menggunakan kombinasi dari citra satelit, foto-foto yang dikirimkan pengguna, dan fitur streetview sehingga menjadikan Google Maps lebih dari sekedar peta biasa. Google Maps adalah atlas interaktif atau sebuah cara untuk mengetahui tentang berbagai tempat dan orang-orang di seluruh dunia.

## **2.12 API (*Application Programming Interface*)**

API adalah sekumpulan instruksi program dan protokol yang digunakan untuk membangun aplikasi perangkat lunak dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara

bersamaan. API berperan sebagai pembawa pesan yang menerima permintaan pengguna dan memberitahu sistem apa yang harus dilakukan, lalu memberikan respons yang sesuai untuk permintaan tersebut. Sebagai analogi, dalam sebuah rumah makan anggaplah pelanggan adalah pengguna aplikasi dan dapur adalah sistem. Dapur akan menyiapkan pesanan atau permintaan pelanggan, dalam analogi ini, API adalah pelayan yang mengkomunikasikan pesanan pelanggan ke dapur atau sistem dan membawa kembali makanan dan minuman yang sesuai dengan permintaan. Dengan adanya API, maka terdapat aturan bagaimana software dapat berinteraksi dengan software lain untuk mengakses resources melalui interface yang telah tersedia.

### **2.13 Java**

Java adalah sebuah bahasa pemrograman dasar dalam sebuah pembuatan aplikasi. Java juga merupakan bahasa pemrograman yang multi platform dan multi device. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, “Tulis sekali, jalankan di mana pun”. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

Sekarang ini Java menjadi sebuah bahasa pemrograman yang populer dan dimanfaatkan secara luas untuk pengembangan berbagai jenis perangkat lunak

ataupun aplikasi. Kebanyakan perangkat lunak yang menggunakan Java adalah ponsel feature dan ponsel pintar atau smartphone.

Beberapa fitur yang ditawarkan Java API antara lain sebagai berikut :

1. Applet

Program Java yang dapat berjalan di atas browser, yang dapat membuat halaman HTML lebih dinamis dan menarik.

2. Java Networking

Sekumpulan API (*Application Programming Interface*) yang menyediakan fungsi-fungsi untuk aplikasi-aplikasi jaringan, seperti penyediaan akses untuk TCP, UDP, IP address dan URL. Namun pada Java Networking tidak menyediakan ICMP karena *security* dan hanya administrator (root) yang bias memanfaatkan protocol ICMP.

3. Java Database Connectivity (JDBC)

JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses database seperti Oracle, MySQL, PostgreSQL, Microsoft SQL Server.

4. Java *Security*

Java *Security* menyediakan sekumpulan API untuk mengatur security dari aplikasi Java baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates*.

5. Java Swing

Java Swing menyediakan sekumpulan API untuk membangun aplikasi – aplikasi GUI (*Graphical User Interface*) dan model GUI yang diinginkan bisa bermacam-macam, bisa model Java, model Motif/CDE atau model yang dependent terhadap platform yang digunakan.

#### 6. Java RMI

Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (Remote Procedure Call) jadi objek-objek Java bisa di panggil secara remote pada jaringan komputer.

#### 7. Java 2D / 3D

Java 2D/3D menyediakan sekumpulan API untuk membangun grafik-grafik 2D/3D yang menarik dan juga akses ke printer.

#### 8. Java Server Pages

Berkembang dari Java Servlet yang digunakan untuk menggantikan aplikasi-aplikasi CGI, JSP (Java Server Pages) yang mirip ASP dan PHP merupakan alternatif terbaik untuk solusi aplikasi Internet.

#### 9. Java Native Interface (JNI)

JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi-fungsi pada library (\*.dll atau \*.so) yang dibuat dengan bahasa pemrograman yang lain seperti C,C++, dan Basic.

#### 10. Java Sound

Java Sound menyediakan sekumpulan API untuk manipulasi suara.

#### 11. Java IDL + CORBA

Java IDL (*Interface Definition Language*) menyediakan dukungan Java untuk implementasi CORBA (*Common Object Request Broker*) yang merupakan model distributed-Object untuk solusi aplikasi besar di dunia *networking*.

#### 12. Java Card

Java Card utamanya digunakan untuk aplikasi-aplikasi pada smart card, yang sederhana wujudnya seperti SIM Card pada handphone.

### 13. Java Telephony API (JTAPI)

Java Telephony API menyediakan sekumpulan API untuk memanfaatkan devices-devices telephony, sehingga akan cocok untuk aplikasi-aplikasi CTI (*Computer Telephony Integration*) yang dibutuhkan seperti ACD (*Automatic Call Distribution*), PC-PBX dan lainnya.

Adapun hubungan Java pada penelitian ini karena aplikasi yang dibuat menggunakan tools Android Studio dengan bahasa pemrograman yang digunakan yaitu bahasa pemrograman Java. Android Studio dipilih karena memiliki banyak fitur yang memudahkan saat pembangunan aplikasi android.

### 2.14 OOP (*Object Oriented Programming*)

Pemrograman berorientasi objek atau OOP (*Object Oriented Programming*) adalah suatu pendekatan pemrograman untuk membuat kode program yang lebih terkelompok berdasarkan objek - objek yang terlibat. Tujuan dari OOP adalah untuk mempermudah pembuatan program dengan cara membagi aplikasi tersebut menjadi bagian - bagian kecil, lalu semua data dan fungsi di dalam aplikasi dibungkus dalam kelas - kelas atau objek - objek. Dimana setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Sebuah bahasa pemrograman dinyatakan bisa menggunakan OOP jika sudah memenuhi syarat-syarat berikut, yaitu :

#### 1. *Encapsulation* (Pembungkusan)

Yaitu pembungkusan variabel dan method dalam sebuah objek dalam bagian yang terlindungi disebut dengan *enkapsulasi*. Jadi, *enkapsulasi* dapat diartikan sebagai mekanisme pemrograman yang membungkus kode dan data yang sedang diolah. Pembungkus ini mendefinisikan untuk melindungi program dan data yang sedang diolah agar terhindar dari interferensi dan penggunaan yang tidak perlu dan tidak diakses sembarangan oleh program lain.

## 2. *Inheritance* (Pewarisan)

Yaitu merupakan pewarisan dari suatu *Class* untuk dipakai *Class - class* yang lain yang memiliki beberapa sifat atau perilaku dari *Class* induknya. *Inheritance* merupakan suatu hubungan dari dua buah class atau lebih dimana dari class yang berhubungan tersebut memiliki atribut dan method pada sebuah class yang diperoleh dari class yang sama. Class yang mengandung member yang sama dari beberapa class lain dinamakan superclass atau parent class. Class yang mewarisi dinamakan subclass atau child class.

## 3. *Polymorphism* (Banyak Bentuk)

Yaitu artinya mempunyai banyak bentuk. Dua objek atau lebih dikatakan sebagai *polymorphisme*, bila objek - objek itu mempunyai antar muka yang sama namun mempunyai perilaku perilaku yang berbeda.

Hubungan OOP (*Object Oriented Programming*) pada penelitian ini adalah pada penelitian ini menggunakan konsep OOP yaitu membagi-bagi kode program menjadi kelas-kelas yang di buat berdasarkan objek-objek yang terlibat saat pembangunan program.

### 2.15 UML (Unified Modelling Language)

*Unified Modeling Language* (UML) adalah sebuah bahasa pemodelan yang digunakan untuk menentukan atau mendeskripsikan sebuah sistem *software* berdasarkan objek-objek yang ada di sistem tersebut. UML tidak menentukan metode apa yang harus digunakan dalam mengembangkan suatu sistem, namun hanya menentukan notasi-notasi standar yang biasa digunakan untuk *object modeling*.

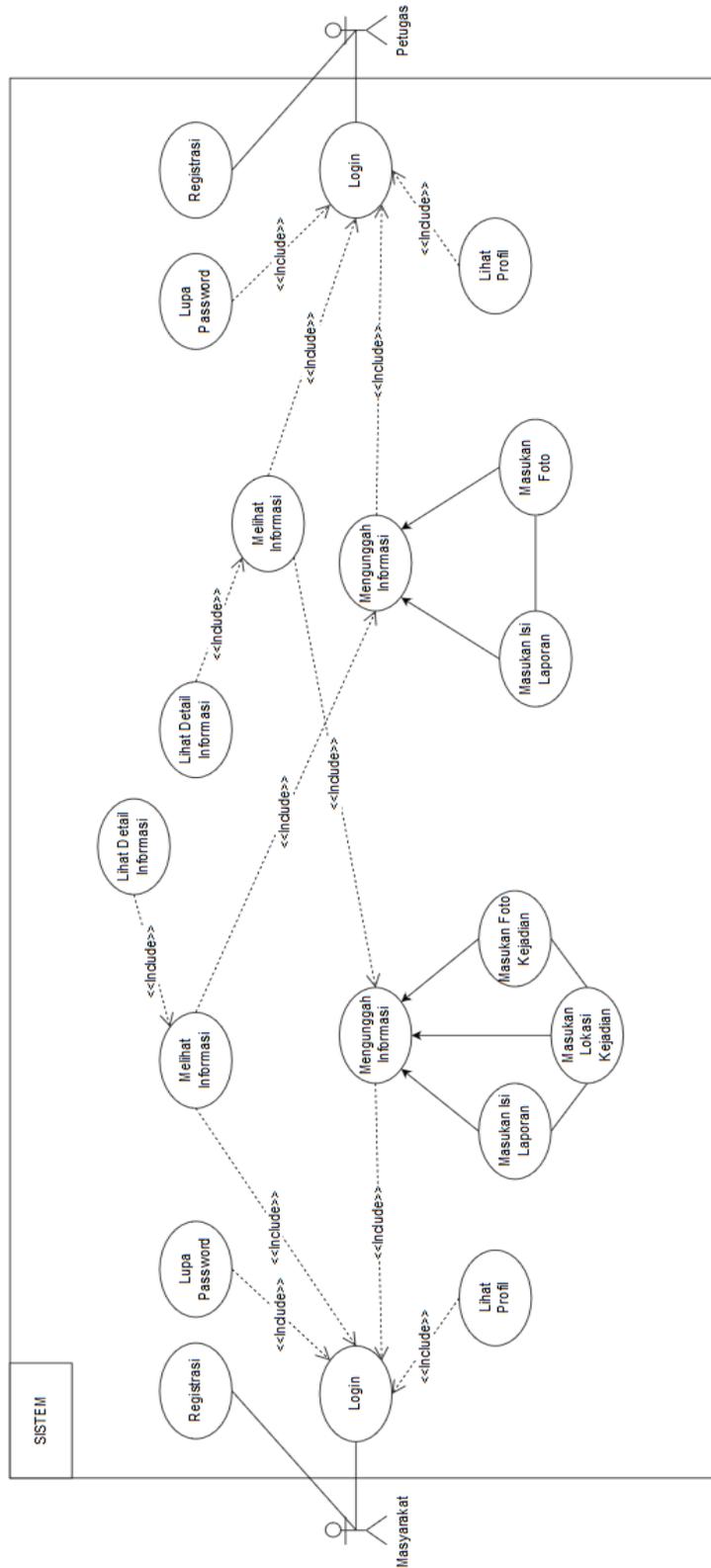
Berikut ini diagram-diagram UML yang paling sering digunakan dalam memodelkan sebuah sistem berorientasi objek diantaranya :

### 1. *Use Case Diagram*

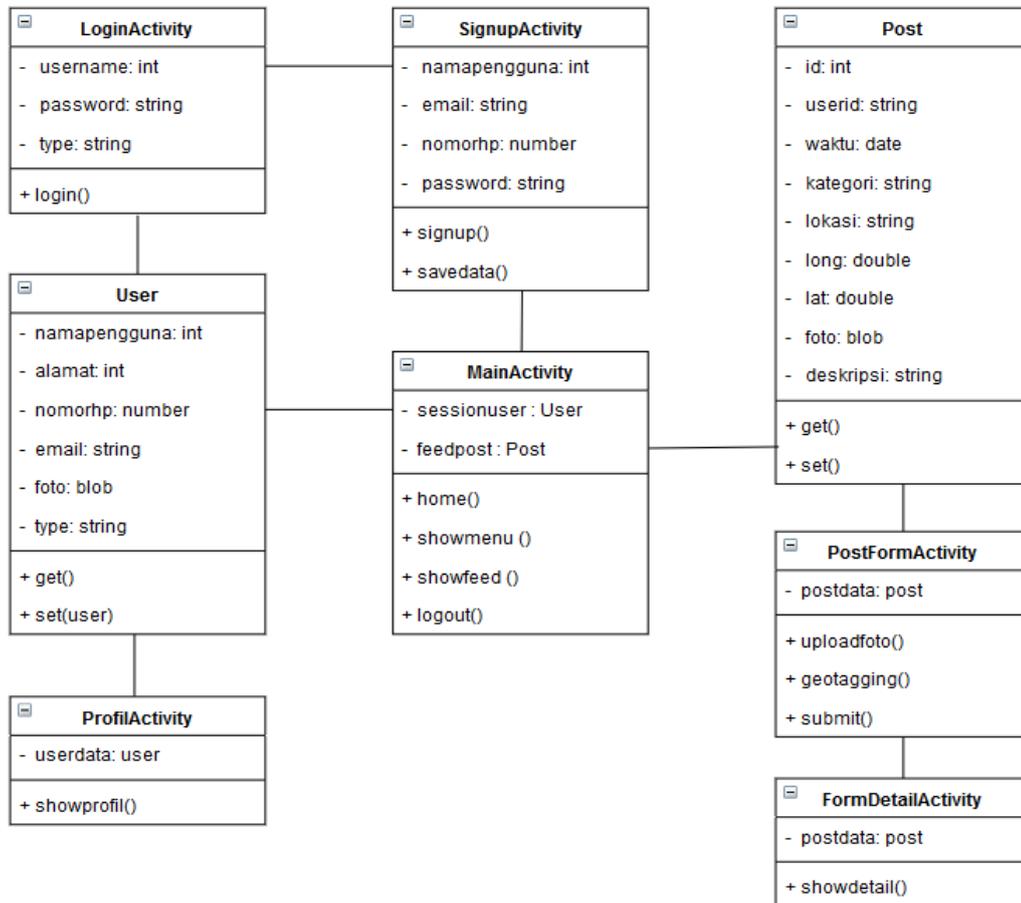
*Use case diagram* adalah sebuah diagram yang mendeskripsikan interaksi antara sistem dengan bagian eksternal dari sistem serta dengan *user*. Secara grafis, *Use-case diagram* ini mendeskripsikan siapa yang akan menggunakan sistem yang ada dan bagaimana ekspektasi *user* saat berinteraksi dengan sistem tersebut.

*Use case diagram* memiliki unsur yang harus dipenuhi yaitu :

- a. *Use case*, yaitu sekumpulan fungsi yang terdapat dalam sistem dimana fungsi – fungsi tersebut dapat dilakukan oleh *actor (user)* untuk melakukan pekerjaannya dengan sistem yang ada.
- b. *Actor*, yaitu segala sesuat yang berinteraksi dengan sistem untuk bertukar informasi, baik *user* maupun sistem. *Relationship*, yaitu garis yang menghubungkan antara *actor* dengan *use case* yang dapat menggambarkan hubungan antara *actor* dengan *use case* itu sendiri.



Gambar 2. 5 Contoh Use Case Diagram



**Gambar 2. 6 Contoh Class Diagram**

## 2. Class Diagram

*Class diagram* adalah sebuah diagram menggambarkan struktur objek dari sistem yang ada, dimana *class diagram* ini memperlihatkan *object class* yang menyusun diagram ini beserta hubungan antara *object class* tersebut.

*Class Diagram* memiliki tiga area pokok :

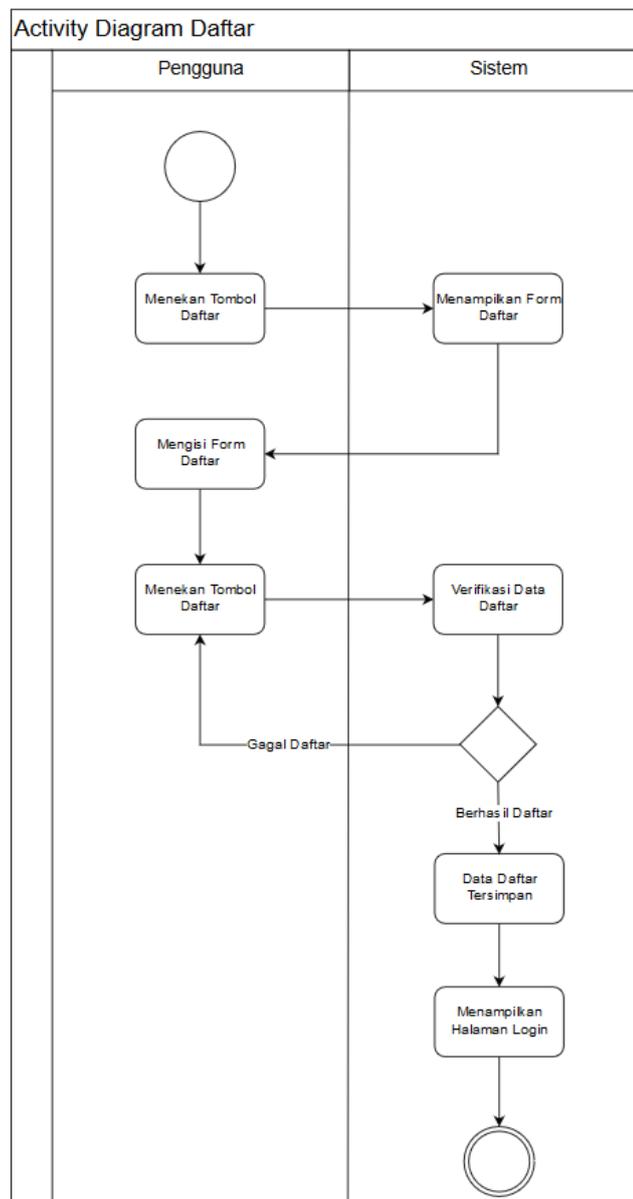
- a. Nama, memiliki fungsi merepresentasikan identitas dari sebuah class.
- b. Atribut, memiliki fungsi merepresentasikan atribut – atribut yang ada di dalam suatu class.

- c. Metode, memiliki fungsi merepresentasikan metode-metode yang ada pada suatu class.

Pada atribut dan metode bisa memiliki sifat sifat berikut :

- a. *Private*, hanya bisa dipanggil oleh *Class* yang bersangkutan, tidak bisa dipanggil oleh *Class* lain yang tidak bersangkutan.
- b. *Public*, bisa di panggil oleh seluruh *Class*.
- c. *Protected*, hampir sama sifatnya dengan *Private* dan *Class* turunannya.

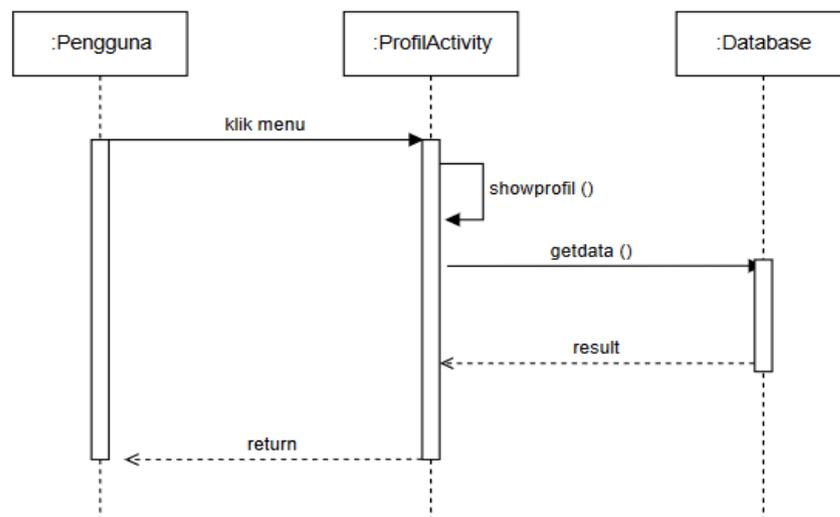
3. *Activity Diagram* *Activity diagram* adalah sebuah diagram yang bisa digunakan untuk menggambarkan secara grafis alur dari sebuah proses bisnis, langkah-langkah dari sebuah *use-case*, atau logika dari sebuah objek. *Activity diagram* sangat berguna untuk model *action* yang akan dikerjakan ketika sebuah operasi dieksekusi serta hasil dari *action* tersebut.



**Gambar 2. 7 Contoh Activity Diagram**

#### 4. Sequence Diagram

*Sequence diagram* adalah sebuah diagram yang menggambarkan interaksi antara *actor* dan *system* untuk sebuah skenario *use-case*. Pada tahap pembuatan *sequence diagram*, kita belum menganalisa lebih lanjut *individual object class*, namun hanya memikirkan keseluruhan sistem yang ada. *Sequence diagram* membantu kita untuk mengidentifikasi setiap data yang masuk dan keluar dari sebuah sistem. Pada *sequence diagram* hanya sebuah skenario dari sebuah *use-case*, sehingga sebuah *use-case* dapat memiliki beberapa *sequence diagram* untuk menggambarkan keseluruhan *use-case* tersebut



**Gambar 2. 8 Contoh Sequence Diagram**

#### 2.16 Kuesioner

Kuesioner merupakan daftar pertanyaan yang akan digunakan oleh periset untuk memperoleh data dari sumbernya secara langsung melalui proses komunikasi atau dengan mengajukan pertanyaan. Metode yang digunakan dalam kuisionare ini menggunakan Skala Likert yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan

paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif.

Ada dua macam bentuk pernyataan dalam skala likert, pertama adalah pernyataan yang diharapkan untuk disetujui oleh responden, disebut bentuk positif. Yang kedua adalah pernyataan yang diharapkan untuk tidak disetujui oleh responden, bentuk negatif. Biasanya kuesioner berisikan pernyataan positif dan negatif agar responden berpikir dulu sebelum memberi jawaban.

**Tabel 2. 1 Skala Likert**

Jenis Pernyataan	Pernyataan	
	Positif	Negatif
Sangat Setuju	5	1
Setuju	4	2
Ragu	3	3
Tidak Setuju	2	4
Sangat Tidak Setuju	1	5

Hubungan pengujian kuisionar dengan penelitian ini adalah pada panelitian ini menggunakan kuisionare untuk mendapatkan jawaban dari responden berupa pernyataan positif dan negatif terhadap penelitian yang dilakukan. Dengan mengukur pendapat, dan persepsi seseorang mengenai suatu fenomena untuk mengoptimalkan penggunaan media sosial dan mengefektifkan cara berkomunikasi dalam menyebarkan dan mendapatkan informasi berupa kemacetan jalan, kecelakaan, perbaikan jalan, dan bencana yang terjadi seperti kebakaran, banjir, dan tanah longsor di keadaan sekitar.

### **2.17 Pengujian *Black Box***

Pengujian *Black Box* merupakan pengujian fungsional dari perangkat lunak, Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan kesalahan antara lain :

1. Fungsi yang tidak benar atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data
4. Kesalahan kinerja

Alasan menggunakan Pengujian *Black Box* pada penelitian ini karena pengujian pada penelitian ini berfokus pada spesifikasi fungsional dari perangkat lunak. Pada pengujian *Black Box* dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak yang dibangun.