

## **BAB 2**

### **TINJAUAN PUSTAKA**

Bagian ini berisi tentang beberapa teori-teori yang mendukung dalam proses analisis dan implementasi berdasarkan masalah yang diangkat dalam pembangunan Aplikasi Identifikasi Jalan Rusak Memanfaatkan Sensor Accelerometer Dan GPS Pada Smartphone Android.

#### **2.1 Definisi jalan**

Jalan adalah prasarana transportasi darat yang meliputi segala bagian jalan, termasuk bangunan pelengkap dan perlengkapannya yang diperuntukkan bagi lalu lintas, yang berada pada permukaan tanah, di atas permukaan tanah, di bawah permukaan tanah dan/atau air, serta di atas permukaan air, kecuali jalan kereta api, jalan lori, dan jalan kabel. Jalan juga merupakan prasarana transportasi darat yang mempunyai peranan sangat penting dalam mendukung bidang ekonomi, sosial dan budaya serta berbagai aspek kehidupan masyarakat lainnya. Berbagai aktivitas masyarakat dipengaruhi oleh kondisi jalan. Kondisi jalan yang baik akan memberikan kenyamanan dan memperlancar aktifitas suatu daerah [1].

##### **2.1.1 Faktor kecelakaan jalan**

Ancaman dari lubang bisa lebih parah daripada sekadar risiko finansial. Terkadang seorang pengendara dapat berbelok berlebihan ketika menghindari lubang sehingga kehilangan kendali atas kendaraannya yang akan menyebabkan pengendara tersebut mengalami kecelakaan. Adapun faktor-faktor yang menyebabkan kecelakaan akibat lubang sebagai berikut.

1. Di Jalan Basah: Saat mengemudi di jalan basah dan di mana Anda melihat genangan air. Lubang mungkin terlihat seperti genangan air saat jalan basah.
2. Mengemudi di Malam Hari : Visibilitas di malam hari berkurang dan Anda mungkin dibutakan oleh lampu depan kendaraan yang melaju, mengurangi kemampuan Anda untuk mengidentifikasi lubang dengan jelas.
3. Di Area dengan Penerangan Jalan yang Buruk: Penerangan dapat dikurangi dengan pemadaman listrik, pelepasan muatan, atau bahkan kejahatan.

Berhati-hatilah saat Anda mungkin tidak melihat dengan jelas kondisi permukaan jalan yang jauh dari kendaraan.

4. Di mana Anda melihat Rambu Peringatan: Rambu peringatan mungkin dengan rambu lalu lintas yang menunjukkan pemeliharaan jalan. Ini juga bisa berupa serpihan kerikil di jalan sebagai solusi perbaikan cepat yang menutupi lubang [2].

## 2.2 Aplikasi

Definisi aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi suatu pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas tertentu.

Aplikasi software yang dirancang untuk suatu tugas khusus dapat dibedakan menjadi dua jenis, yaitu:

1. Aplikasi software spesialis, program dengan dokumentasi terdapat yang dirancang untuk menjalankan tugas tertentu.
2. Aplikasi software paket, suatu program dengan dokumentasi terdapat yang dirancang untuk jenis masalah tertentu [5].

## 2.3 Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat seluler layar sentuh seperti komputer pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google yang kemudian membelinya pada tahun 2005. Sistem operasi ini kemudian di rilis pada tahun 2007.

Beberapa fitur utama dari Android antara lain WiFi hotspot, Multi-touch, Multitasking, GPS, accelerometers, support java, mendukung banyak jaringan (*GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE & WiMAX*) serta juga kemampuan dasar handphone pada umumnya [6].

Banyak *smartphone* dan PC Tablet menggunakan sistem operasi dengan versi yang berbeda. Semakin tinggi versi, fiturnya semakin canggih dan banyak. Telepon pertama yang memakai sistem operasi *Android* adalah HTC Dream yang

dirilis pada tanggal 22 Oktober 2008. Beberapa uraian versi *android* seperti dibawah ini :

#### 1. Android Versi 1.1

Pada 9 Maret 2009, *Google* merilis *Android* versi 1.1. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan Email.

#### 2. Android Versi 1.5 (*Cupcake*)

Pada pertengahan Mei 2009, *Google* kembali merilis telepon seluler dengan menggunakan *Android* dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth* A2DP, kemampuan terhubung secara otomatis ke *headset bluetooth*, animasi layar dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

#### 3. Android Versi 1.6 (*Donut*)

*Donut* (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, *camcorder* dan galeri yang diintegrasikan pada CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel).

#### 4. Android Versi 2.0/2.1 (*Éclair*)

Pada 3 Desember 2009 kembali diluncurkan ponsel *Android* dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah mengoptimalkan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML 5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, *digital zoom*, dan *Bluetooth* 2.1. Untuk bergerak cepat dalam persaingan perangkat

generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *mobile* terbaik (*killer apps*-aplikasi unggulan). Kompetisi ini berhadiah \$25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah *handset Android*, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi *Android*. Aplikasi terkenal yang diubah ke dalam sistem operasi *Android* adalah Shazam, *Backgrounds* dan *WeatherBug*. Sistem operasi *Android* dalam situs internet juga dianggap penting untuk menciptakan aplikasi *Android* asli, contohnya oleh *MySpace* dan *Facebook*.

#### 5. Android Versi 2.2 (*Froyo : Frozen Yoghurt*)

Pada 20 Mei 2010, *Android* versi 2.2 (*Froyo*) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi dengan V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel dan kemampuan *auto update* dalam aplikasi *Android Market*.

#### 6. Android Versi 2.3 (*Gingerbread*)

Pada 6 Desember 2010, *Android* versi 2.3 (*Gingerbread*) diluncurkan. Perubahan-perubahan umum yang didapat dari versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antarmuka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM. efek audio baru (*reverb, equalization, headphone virtualization, dan bass boost*), dukungan kemampuan *Near Field Communication* (NFC) dan dukungan jumlah kamera yang lebih dari satu.

#### 7. Android Versi 3.1 (*Honeycomb*)

*Android Honeycomb* dirancang khusus untuk tablet. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk tablet. *Honeycomb* juga

mendukung multi prosesor dan juga akselerasi perangkat keras (*hardware*) untuk grafis. Tablet peratam yang dibuat dengan menjalankan *Honeycomb* adalah Motorola Xoom.

#### 8. Android Versi 4.0 (*Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara *offline* dan berbagi informasi dengan menggunakan NFC.

#### 9. Android Versi 4.1 (*Jelly Bean*)

Pada tanggal 27 Juni 2012 Google mengumumkan *Android* versi 4.1 (*Jelly Bean*) dalam konferensi Google I/O berdasarkan kernel Linux 3.0.31. *Jelly Bean* adalah pembaruan penting yang bertujuan untuk meningkatkan fungsi dan kinerja antarmuak pengguna (*User Interface*). Pembaruan ini diwujudkan dalam proyek *Butter*, perbaikan ini termasukantisipasi sentuh, *triple buffering*, perpanjangan waktu vsync dan peningkatan *frame rate* hingga 60fps untuk menciptakan UI yang lebih halus. Perangkat pertama yang menggunakan sistem operasi ini adalah tablet Nexus 7, yang dirilis pada 13 Juli 2012.

#### 10. Android Versi 4.4 (*KitKat*)

*Android* versi 4.4 (*KitKat*) direncanakan akan dirilis pada bulan Oktober 2013. Setelah sebelumnya beredar rumor bahwa *Android* versi berikutnya setelah *Jelly Bean* diperkirakan akan diberi nomor 5.0 dan dinamai *Key Lime Pe* [6].

## 2.4 Android Studio

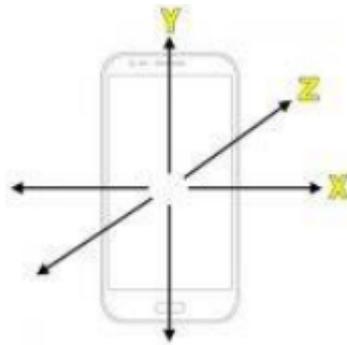
Android Studio adalah Lingkungan Pengembangan Terpadu - *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan *IntelliJ IDEA*. Selain merupakan editor kode *IntelliJ* dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas Anda saat membuat aplikasi Android, misalnya:

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 [7].

## 2.5 Accelerometer

Sensor adalah sebuah piranti yang dapat mengubah suatu besaran fisis, misal : jarak, cahaya, panas, magnet, listrik, dan lain sebagainya, ke bentuk besaran fisis lainnya

Sensor accelerometer merupakan perangkat elektronik yang digunakan untuk mengukur percepatan dari suatu objek [8]. *Accelerometer* ini mengukur percepatan, bahwa perangkat mengalami perubahan yang sesuai dengan nilai dari tiga sumbu X,Y,Z lalu data dari tiga sumbu tersebut akan dihitung untuk mengetahui percepatan atau perlambatan yang terjadi. Gambar sumbu accelerometer dapat dilihat pada gambar 2.1 Sumbu *Accelerometer*.



Sumber : <https://mainthebest.com>

**Gambar 2.1 Pergerakan pada sensor accelerometer**

Percepatan adalah suatu keadaan yang mengakibatkan perubahan bertambahnya kecepatan terhadap waktu, sedangkan apabila perubahan berkurangnya kecepatan terhadap waktu disebut perlambatan. Accelerometer yang diletakkan di permukaan bumi dapat mendeteksi percepatan 1g (ukuran gravitasi bumi) pada titik vertikalnya. Sedangkan percepatan yang arah pergerakannya secara horizontal, sensor accelerometer akan mengukur percepatannya secara langsung ketika bergerak secara horizontal [9].

Prinsip dari accelerometer adalah sebuah per dengan beban dan dilepaskan, beban bergerak dengan suatu percepatan sampai kondisi tertentu lalu berhenti. Bila ada sesuatu yang mengguncangkannya maka beban berayun kembali. Accelerometer dapat digunakan untuk mengukur getaran pada mobil, mesin, bangunan, dan instalasi pengaman. Sensor accelerometer juga dapat diaplikasikan pada pengukuran aktivitas gempa, pedometer atau penghitung langkah.

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

GPS adalah sistem navigasi yang berbasis satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (Departemen of Defense) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS receiver yang berfungsi untuk menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama Waypoint yang nantinya

akan berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik.

Sejak tahun 1980, layanan GPS yang dulunya hanya untuk keperluan militer mulai terbuka untuk publik. Uniknya, walau satelit-satelit tersebut berharga ratusan juta dolar, namun setiap orang dapat menggunakannya dengan gratis. Satelit-satelit ini mengorbit pada ketinggian sekitar 12.000 mil dari permukaan bumi. Posisi ini sangat ideal karena satelit dapat menjangkau area coverage yang lebih luas. Satelit-satelit ini akan selalu berada pada posisi yang bisa menjangkau semua area di atas permukaan bumi sehingga dapat meminimalkan terjadinya blank spot (area yang tidak terjangkau oleh satelit).

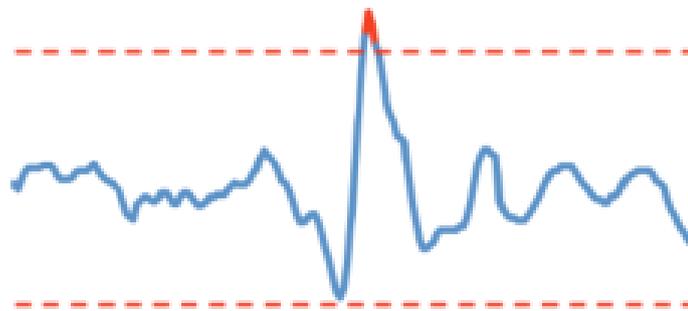
Setiap satelit mampu mengelilingi bumi hanya dalam waktu 12 jam. Sangat cepat, sehingga mereka selalu bisa menjangkau dimana pun posisi anda diatas permukaan bumi. GPS receiver sendiri berisi beberapa integrated circuit (IC) sehingga murah dan teknologinya mudah untuk di gunakan oleh semua orang. GPS dapat digunakan untuk berbagai kepentingan, misalnya mobil, kapal, pesawat terbang, pertanian dan di integrasikan dengan komputer maupun laptop [10].

## **2.7 Algoritma Deteksi**

Sudah ada beberapa sistem penginderaan kendaraan dalam rangka deteksi lubang. Beberapa sistem menggunakan accelerometers untuk mendapatkan data. Dalam sub bagian ini berisi ulasan tentang algoritma deteksi lubang yang diimplementasikan dalam sistem berbasis accelerometer. Kajian kelayakan implementasi sistem ini dilakukan pada platform dengan hardware dan software bersumber daya terbatas, yakni berbasis telepon pintar Android oleh Artis Mednis, Girts Starzdins, Reinholds Zviedris, Georgijs Kanonirs, Leo Selavo dalam tulisan ilmiah mereka "Real Time Pothole Detection ,Using Android Smartphones with Accelerometers" pada tahun 2011. Dalam jurnal ilmiah mereka disampaikan kajian terkait algoritma deteksi Z-THRESH, Dengan menggunakan ketersediaan sumber daya perangkat keras dan perangkat lunak pada platform ini mampu untuk diimplementasikan algoritma yang lebih kompleks dengan parameter deteksi lubang yang lebih baik serta melakukan deteksi secara realtime [11].

### 2.7.1 Z-TRESH

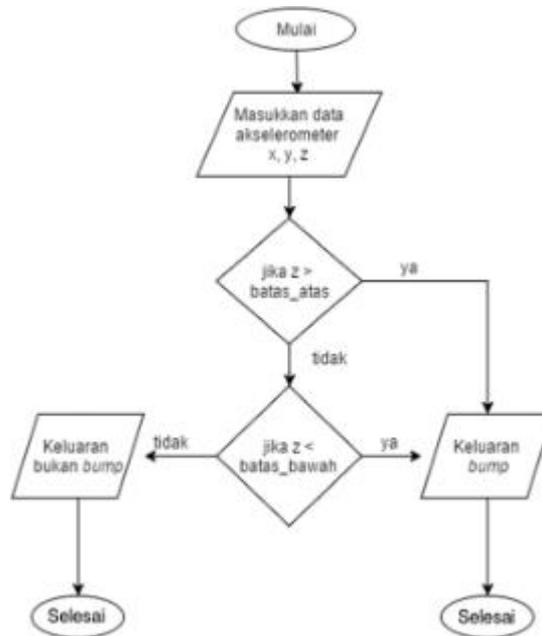
Algoritma digunakan yaitu algoritma deteksi metode Z-THRESH yang diuji pada suatu kumpulan data. Metode ini menggunakan amplitudo batasan yang telah ditentukan pada sumbu Z. Metode ini mengklasifikasikan pengukuran berdasarkan nilai yang melebihi batas yang telah ditetapkan dalam mengidentifikasi kondisi jalan raya. Algoritma ini menggunakan informasi tentang posisi sumbu Z di accelerometer. Reorientasi virtual tambahan dari accelerometer dapat diterapkan. Reorientasi ini terkait dengan posisi meletakkan telepon pintar dalam melakukan uji coba [11].



**Gambar 2.2 Algoritma Z-THRESH**

### 2.7.2 Alur Deteksi

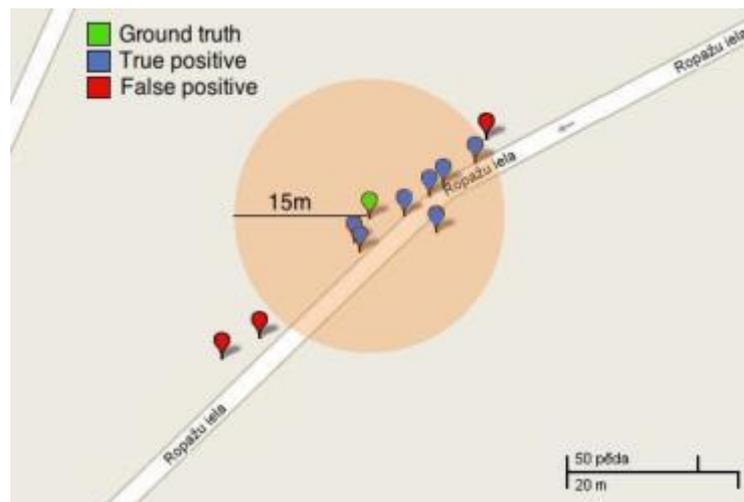
Alur yang digunakan dalam pendeteksian lubang ini menggunakan alur Z-Tresh yang sudah dibuat oleh Otniel Yehezkiel Bornok Hutabarat, Fajar Baskoro, dan Rizky Januar Akbar dalam tulisan ilmiah “Rancang Bangun Sistem Pendeteksi Bump Menggunakan Android Smartphone dengan Sensor Akselerometer” pada tahun 2016 [12]. Dimana diagram alurnya dapat dilihat sebagai berikut.



**Gambar 2.3 Diagram Alur Z-TRESH**

### 2.7.3 Validasi Lubang

Validasi lubang adalah pengecekan lubang jalan apakah sudah diperbaiki atau tidak dimana menggunakan pengecekan yang sudah dibuat oleh Artis Mednis, Girts Strazdins, Martins Liepins, Andris Gordjusins, and Leo Selavo dalam tulisan ilmiah “RoadMic: Road Surface Monitoring Using Vehicular Sensor Networks with Microphones” pada tahun 2010.



**Gambar 2.4 Roadmic Validasi Lubang**

Dalam penelitian itu, mereka mengasumsikan periode posisi GPS setiap 1 detik akurasi estimasi posisi-nya adalah  $\pm 3-30$  meter. Dalam percobaan-nya, median akurasi GPS terakurat adalah  $\pm 10-15$ m. dimana pada gambar diatas 7 dari 10 deteksi dianggap True Positive dan 3 dari 10 di luar jarak 15 meter dari lubang False Postive. Mereka juga mendefinisikan kriteria pendeteksian: lubang yang terdeteksi akan dianggap telah diakui oleh Algoritma jika memiliki setidaknya dari 10 kali pengujian dalam melewati lubang setidaknya ada 4 kali terjadi deteksi lubang [21].

## **2.8 Google Maps API**

Google Maps API adalah salah satu Application Programming Interface (API) yang dimiliki Google. API ini mempunyai fitur untuk melakukan aktivitas-aktivitas yang berkaitan dengan Google Maps, antara lain menampilkan peta, mencari rute terdekat antara dua tempat, dan lain sebagainya. Google Maps API tersedia untuk platform android, iOS, web, dan juga web service. Google maps API juga menyediakan layanan seperti direction, geocoding, distance matrix API, dan elevation API.

Google maps merupakan sebuah layanan gratis yang diberikan oleh google dan sangat populer. Google maps adalah suatu peta dunia yang dapat kita gunakan untuk dapat untuk dapat melihat suatu daerah. Dengan kata lain, google maps merupakan suatu peta yang dapat dilihat dengan menggunakan suatu web browser. Kita dapat menambahkan fitur google maps dalam sistem atau aplikasi yang kita buat. Google maps API adalah suatu library yang berbentuk javascript dimana kita dapat mengubah dan menambahkan variabel-variabel tertentu sehingga bisa dibuat sesuai dengan keinginan kita. Berkas yang mengandung bytecode kemudian akan dikonversikan oleh java interpreter menjadi bahasa mesin sesuai dengan jenis dan platform yang digunakan [13].

Dalam perkembangannya google maps API diberikan kemampuan untuk mengambil peta statis. Melakukan geocoding dan memberikan penuntun arah. Kekurangan pada google maps API yaitu jika ingin melakukan akses harus terdapat layanan internet pada perangkat yang digunakan, sedangkan kelebihan yang dimiliki yaitu :

1. Dukungan penuh yang dilakukan google sehingga terjamin dan fitur yang bervariasi pada google maps API.
2. Banyak pengembang yang menggunakan google maps API sehingga mudah dalam mencari referensi dalam pengembangan aplikasi.

## 2.9 Google Direction API

Google Direction API adalah layanan yang menghitung arah antar lokasi menggunakan permintaan HTTP. Google Direction API memiliki beberapa parameter yang di butuhkan untuk meminta lama perjalanan. Daftar dari parameter yang ada dapat dilihat pada tabel 2.1 Daftar parameter yang dibutuhkan Google Direction API [20].

**Table 2.1 Daftar parameter yang dibutuhkan Google Direction API**

<b>Nama Parameter</b>	<b>Masukan</b>
Origin	Koordinator atau Nama Lokasi
Destination	Koordinator atau Nama Lokasi
Key	Kunci API

Parameter yang Terdapat pada Tabel 2.1 merupakan parameter yang dibutuhkan untuk Google Maps Google Maps Direction API untuk dapat digunakan parameter Origin merupakan parameter untuk Titik Awal perhitungan, parameter Destination Merupakan parameter untuk titik akhir perhitungan, parameter Key merupakan kunci API yang digunakan pada aplikasi, Google Maps Direction API sendiri memiliki beberapa parameter Opsional yang diberikan daftar dari parameter opsional yang ada dapat dilihat pada tabel 2.2 Daftar parameter opsional Google Maps Direction API

**Table 2.2 Daftar parameter opsional Google Maps Direction API**

<b>Nama Parameter</b>	<b>Masukan</b>
mode	Driving,walking,bicycling,transit
waypoints	Koordinat Lokasi atau Nama Lokasi
alternatives	True atau flase
language	Bahasa yang digunakan

region	country code top-level domain
Avoid	Tolls,highway,ferries,indoor
Units	Satuan yang digunakan
Arrival_time	Unix Time
Departure_time	Unix Time,now
Traffic_model	Best_guess,Pessimistic,optimistic
Transit_mode	Bus,subway,train,tram,rail
Transit_routing_preference	Less_walking,fewer_transfer

Penjelasan mengenai parameter parameter pada Tabel 3.2 adalah sebagai berikut:

1. Mode

Merupakan parameter untuk menentukan mode transportasi yang akan digunakan saat menghitung arah

2. Waypoints

Merupakan parameter untuk menentukan susunan lokasi perantara untuk disertakan di sepanjang rute antara titik asal dan tujuan sebagai melewati atau menghentikan lokasi. Titik jalan mengubah rute dengan mengarahkannya melalui lokasi yang ditentukan

3. Alternatives

menetapkan bahwa layanan Arah dapat memberikan lebih dari satu alternatif rute dalam respons

4. Language

Merupakan parameter bahasa yang akan di gunakan untuk hasil dari Google Maps Direction API

5. Region

Merupakan Parameter area untuk membatasi hasil dari geocoder

6. Avoid

Merupakan Parameter yang membatasi jalur yang digunakan seperti tidak melewati toll

7. Units

merupakan Parameter yang menentukan satuan jarak yang akan di gunakan pada hasil

#### 8. Arrival Time

Merupakan parameter yang digunakan untuk menspesifikasi waktu Sampai yang di inginkan namun parameter ini hanya dapat digunakan bila berkendara menggunakan transit

#### 9. Departure Time

Merupakan parameter yang digunakan untuk menspesifikasi waktu keberangkatan yang di inginkan parameter ini membutuhkan masukan Unix Time yaitu detik semenjak 1970 januari 1 UTC(Coordinated universal Time)

#### 10. Traffic model

Merupakan Parameter yang dipakai untuk asumsi perhitungan prediksi lama Perjalanan

#### 11. Transit model

Merupakan Parameter yang dipakai untuk asumsi perhitungan prediksi lama Perjalananan transit

#### 12. Transit routing preference

Merupakan parameter untuk memilih jalur transit apakah lebih banyak jalan atau berkendara

### **2.10 Smartphone**

Smartphone atau ponsel cerdas merupakan salah satu wujud realisasi ubiquitous computing (ubicomp) di mana teknologi tersebut memungkinkan proses komputasi dapat terintegrasi dengan berbagai aktivitas keseharian manusia dengan jangkauannya yang tidak dibatasi dalam satu wilayah atau suatu scope area. [14]

Disebut smartphone karena ponsel ini dapat bekerja dengan kemampuan yang mirip dengan komputer. Ponsel cerdas ini biasanya dilengkapi dengan kamera, layanan internet, pemutar musik, layanan untuk membaca dan mengedit sebuah dokumen dan lain-lain yang pada umumnya didukung oleh berbagai macam aplikasi yang dapat diunduh oleh pengguna ponsel. Terdapat berbagai macam sistem operasi yang dapat ditemukan pada ponsel cerdas. Contoh sistem operasi

yang biasa ditemukan pada ponsel cerdas atau smartphone adalah Windows Mobile, Android, serta iOS.

Smartphone menjadi dasar perangkat yang digunakan dalam penelitian ini dikarenakan smartphone memiliki banyak keunggulan yang dapat dimanfaatkan bagi pengembang aplikasi.

## **2.11 Firebase**

Firebase merupakan sebuah layanan infrastruktur Backend-as-a-service (BaaS) yang diakuisisi oleh Google pada oktober 2014. Firebase menawarkan kemudahan kepada para pengembang perangkat lunak dalam membangun aplikasi yang lebih baik serta bisnis yang sukses melalui seluruh fitur komplementernya. Firebase dibangun diatas tiga pilar yang meliputi Develop, Grow, dan Earn, dapat dilihat pada gambar 2.3 Pilar Firebase [15].



**Gambar 2.4 Logo Firebase**

Firebase memiliki fitur yang dirancang untuk meningkatkan pengalaman pengembang perangkat lunak dalam mengembangkan aplikasinya. Berikut adalah beberapa teknologi tersebut :

### **2.11.1 Firebase Authentication**

Firebase Authentication merupakan layanan siap pakai yang dimiliki oleh Firebase SDK. Firebase Authentaction memungkinkan aplikasi untuk melakukan autentikasi yang aman, sekaligus meningkatkan pengalaman login dan pengalaman aktivasi bagi end-user. Metode autentikasi yang digunakan meliputi email and password based authentication, federated identity provider identeegration (authentication menggunakan akun Google, Facebook, Twitter atau Github), custom authentication system integration hingga anonymous authentication. Firebase User Authentication ini bekerja dengan cara mengirimkan server response

dari Firebase Server berdasarkan credential yang dikirimkan oleh client ke Firebase Server. Credential tersebut dapat berupa alamat email dan password ataupun sebuah token OAuth dari sebuah federated identity provider. Melalui server response yang diterima dari Firebase Server, aplikasi dapat mengakses informasi dasar profil pengguna dan mengontrol akses pengguna terhadap produk atau layanan Firebase yang terdapat pada aplikasi. Firebase Authentication dibangun untuk memberikan API yang mudah kepada pengembang yang dapat digunakan untuk proses sign-in dari federated providers dengan skema email dan password, atau yang sudah terintegrasi dengan autentikasi yang sudah ada. Firebase Authentication telah terintegrasi dengan Firebase Realtime Database sehingga administrator dapat mengontrol siapa yang dapat mengakses data.

Untuk membuat pengguna login ke aplikasi, dapatkan kredensial autentikasi dari pengguna terlebih dahulu. Kredensial ini dapat berupa alamat email dan sandi pengguna atau token OAuth dari penyedia identitas tergabung. Kemudian, teruskan kredensial ini ke Firebase Authentication SDK. Layanan backend kami selanjutnya akan memverifikasi kredensial tersebut dan menampilkan respons ke klien. Setelah berhasil login, Anda dapat mengakses informasi profil dasar pengguna dan Anda dapat mengontrol akses pengguna ke data yang disimpan di produk Firebase lainnya. Anda juga dapat menggunakan token autentikasi yang disediakan untuk memverifikasi identitas pengguna di layanan backend Anda sendiri [15].

### **2.11.2 Firebase Realtime Database**

Firebase Realtime Database merupakan sebuah layanan NoSQL cloudhosted database yang dimiliki oleh Firebase SDK. Firebase Realtime Database adalah database event-driver yang cara kerjanya berbeda dari database SQL. Tidak ada kode sisi server dan tingkat akses database semua pengkodean dilakukan di klien. Layanan ini menawarkan layanan penyimpanan data yang dapat disinkronisasikan secara realtime terhadap seluruh klien yang terhubung. Maksud dari realtime adalah jika terdapat perubahan pada data pada database, maka seluruh client yang terhubung akan secara otomatis mendapatkan perubahannya dalam hitungan milidetik. Kemudian offline, yaitu aplikasi yang menggunakan Firebase Realtime Database akan tetap responsif bahkan saat offline. Hal ini disebabkan

karena Firebase SDK dapat mempertahankan data dan perubahannya pada media penyimpanan klien. Pada saat klien terhubung ke jaringan internet, maka Firebase SDK akan melakukan penyesuaian otomatis atas catatan perubahan data yang disimpan pada media penyimpanan klien dengan kondisi terkini dari Firebase server. Kemampuan inti yang terakhir adalah *accessible from client devices*. Layanan ini menawarkan kemudahan untuk mengakses Firebase Realtime Database secara langsung dari sebuah perangkat mobile atau sebuah peramban web tanpa membutuhkan server application.

Firebase Realtime Database memungkinkan Anda 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 Firebase Realtime Database, untuk menentukan metode strukturisasi data dan kapan data dapat dibaca atau ditulis. Ketika diintegrasikan dengan Firebase 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 [15].

### 2.11.3 Firebase Cloud Messaging

FCM adalah sebuah layanan yang digunakan untuk melakukan pemberitahuan (notifications) pada aplikasi berbasis Android, iOS maupun aplikasi web. Sebelumnya Firebase Cloud Messaging ini bernama Google Cloud Messaging atau GCM, namun sekarang sudah berubah dan menjadi lebih besar di Firebase. Dengan FCM, pengembang aplikasi dapat mengirim pemberitahuan ke aplikasi yang didukung FCM dengan mudah. Dalam hal responsif, GCM / FCM memiliki hasil yang lebih baik (lebih cepat) dari pada SMS [22]. Dengan menggunakan FCM memungkinkan terjadinya pengiriman dua jenis pesan ke klien :

1. Notification messages, terkadang diangg Display messages ditangani oleh FCM SDK secara otomatis. Digunakan FCM pengembang ingin menangani pemberitahuan pada aplikasi klien.
2. Data messages, yang ditangani oleh aplikasi klien. Digunakan jika pengembang ingin memproses pesan di aplikasi klien.

Pesan pemberitahuan berisi serangkaian kunci yang dapat dilihat pengguna yang telah ditentukan sebelumnya. Pesan data, sebaliknya, hanya berisi pasangan nilai kunci khusus yang ditetapkan pengguna. Pesan pemberitahuan dapat berisi muatan data opsional. Payload maksimum untuk kedua jenis pesan adalah 4KB, kecuali saat mengirim pesan dari Firebase console, yang memberlakukan batas 1024 karakter.

FCM dapat mengirim pesan pemberitahuan termasuk muatan data opsional. Dalam kasus semacam itu, FCM menangani tampilan payload notifikasi, dan aplikasi klien menangani payload data. Implementasi FCM mencakup dua komponen utama untuk mengirim dan menerima:

1. Lingkungan terpercaya seperti Cloud Functions untuk Firebase atau server aplikasi yang digunakan untuk membuat, menargetkan, dan mengirim pesan. Aplikasi klien iOS, Android, atau web (JavaScript) yang menerima pesan.
2. Anda dapat mengirim pesan melalui SDK Admin atau API HTTP dan XMPP. Untuk menguji atau mengirim pesan pemasaran atau keterlibatan dengan penargetan dan analitik internal yang kuat, Anda juga dapat menggunakan komposer Notifications [15].

#### 2.11.4 Geofire

Geo-Fire adalah sebuah library open source yang diperuntukan untuk Java/Android. Library ini memungkinkan dan mengizinkan pengguna untuk menyimpan dan meng-query set kunci-kunci berdasarkan dari lokasi geografik.

Pada dasarnya, library Geo-fire memudahkan untuk menyimpan lokasi-lokasi (latitude, longitude) menggunakan String key atau kunci. Keuntungan utama penggunaan library ini adalah memungkinkan mengolah key pada lokasi-lokasi spesifik secara real time [16]. Geo-Fire menyimpan data pengguna dalam formatnya sendiri dan lokasinya sendiri di dalam basis data Firebase. Ini memungkinkan format data dan aturan keamanan pengguna aplikasi tidak berubah serta mempermudah permintaan geografik tanpa mengubah data yang ada.

Objek Geo-Fire digunakan untuk membaca dan menulis data lokasi geo ke basis data dan untuk membentuk queri. Untuk membuat instance GeoFire baru, pengguna harus memasukkannya ke dalam referensi basis data Firebase.

Untuk pengaturan lokasi (set location) pada GeoFire, cukup dengan memanggil method 'setLocation()'. Method ini meneruskan kunci sebagai String dan lokasi sebagai objeknya yang berisi koordinat lintang (longitude) dan bujur (latitude). Adapun untuk menghapus lokasi dan menghapusnya dari basis data, pengguna cukup menggunakan method 'removeLocation()' kunci lokasi ke remove location.

GeoFire memungkinkan pengguna untuk melakukan queri semua kunci dalam area geografis menggunakan objek GeoQuery. Saat lokasi untuk kunci berubah, kueri diperbarui dalam waktu real-time dan memunculkan event yang memberi tahu pengguna jika ada kunci yang relevan telah dipindahkan. Parameter GeoQuery dapat diperbarui untuk mengubah ukuran dan pusat area yang diminta.

Berikut ini adalah 5 Key Events untuk menerima event untuk Geo Query.

1. Key Entered: Lokasi kunci saat cocok dengan kriteria query.
2. Key Exited: Lokasi kunci saat sudah tidak cocok dengan kriteria query (keluar dari kriteria query).
3. Key Moved: Lokasi kunci berubah, namun lokasi masih di dalam atau masih cocok dengan kriteria query.

4. Query Ready: Semua data saat ini telah diambil dari server.
5. Query Error: Adanya suatu kesalahan dalam melakukan query lokasi.

Selain hal-hal diatas, pada GeoFire pun menyediakan event data. Maksudnya, jika pengguna menyimpan data model dan data geo di lokasi basis data yang sama, Pengguna mungkin ingin mengakses DataSnapshot sebagai bagian dari event geo. Dalam hal ini, bisa menggunakan GeoQueryDataEventListener daripada listener kunci.

Listener "data event" ini memiliki semua event yang sama dengan listener kunci dengan satu jenis acara tambahan. DataSnapshot mendasarinya data telah berubah. Setiap event "data dipindahkan" diikuti oleh event yang diubah data tetapi pengguna juga bisa mendapatkan perubahan event tanpa bergerak jika data yang diubah tidak memengaruhi lokasi. Menambahkan event listener data mirip dengan menambahkan listener event utama.

## **2.12 Java**

Java adalah bahasa pemrograman yang berorientasi objek (OOP) dan dapat dijalankan pada berbagai platform sistem operasi. Perkembangan java tidak hanya terfokus pada satu sistem operasi, tetapi juga dikembangkan untuk berbagai sistem operasi dan bersifat open source [17]. 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. 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 berbasis web.

Java menggunakan model pengamanan tiga lapis (three-layer security model) untuk melindungi sistem dari untrusted Java code. Pertama, bytecode verifier membaca bytecode sebelum dijalankan dan menjamin bytecode memenuhi aturan-aturan dasar bahasa Java. Kedua, class loader menangani pemuatan kelas Java ke runtime interpreter. Ketiga, manajer keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, port jaringan, proses eksternal dan sistem window.

Java termasuk bahasa Multithreading. Thread adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. Java menyediakan kelas untuk menulis program multithreaded, program mempunyai lebih dari satu thread eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren.

Program Java melakukan garbage collection yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat di bahasa yang memungkinkan alokasi dinamis.

Java mempunyai mekanisme exception-handling yang ampuh. Exceptionhandling menyediakan cara untuk memisahkan antara bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih dan menjadikan aplikasi lebih tegar. Ketika kesalahan yang serius ditemukan, program Java menciptakan exception. Exception dapat ditangkap dan dikelola program tanpa resiko membuat sistem menjadi turun.

Program Java mendukung native method yaitu fungsi ditulis di bahasa lain, biasanya C/C++. Dukungan native method memungkinkan pemrogram menulis fungsi yang dapat dieksekusi lebih cepat dibanding fungsi ekivalen di java. Native method secara dinamis akan di-link ke program java, yaitu diasosiasikan dengan program saat berjalan.

Adapun kelebihan dari Java yaitu :

Java mempunyai beberapa keunggulan dibandingkan dengan Bahasa pemrograman lainnya. Keunggulan bahasa pemrograman Java antara lain:

1. Berorientasi objek

Java adalah bahasa pemrograman yang berorientasi pada objek. Java membagi program menjadi objek - objek serta memodelkan sifat dan tingkah laku masing-masing dalam menyelesaikan suatu masalah.

2. Java bersifat multiplatform

Java dirancang untuk mendukung aplikasi yang dapat beroperasi di lingkungan jaringan berbeda. Untuk mengakomodasi hal tersebut, Java compiler membangkitkan bytecodes (sebuah format yang tidak tergantung pada arsitektur tertentu yang didesain untuk mengirimkan kode ke banyak platform perangkat keras dan perangkat lunak secara efisien). Java dapat dijalankan oleh banyak platform seperti Linux, Unix, Windows, Solaris, maupun Mac.

3. Java bersifat multithread

Multithreading adalah kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. Thread dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosesor apabila sistem operasi yang digunakan mendukung multi prosesor.

4. Dapat didistribusi dengan mudah

Java memiliki library rutin yang lengkap untuk dirangkai pada protocol TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan networking Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah socket koneksi. Java juga memudahkan pembuatan CGI (Common Gateway Interface).

5. Bersifat dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang. Java bersifat dinamis dalam tahap linking. Class yang ada dapat di-link sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di-link dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

Kekurangan Java yaitu:

1. Tulis sekali

Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.

2. Mudah didekompilasi

Dekompilasi adalah proses membalikkan dari kode jadi menjadi kod sumber. Ini dimungkinkan karena kode jadi Java merupakan bytecode yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Microsoft .NET Platform. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/di-reverse-engineer.

3. Penggunaan memori yang banyak

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal). Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berlutut dengan mesin komputer berumur lebih dari 4 tahun.

### **2.13 Extensible Markup Language (XML)**

XML adalah bahasa markup yang menggunakan tag untuk memberi label, mengategorikan, dan mengatur informasi dengan cara spesifik. Markup menggambarkan struktur dokumen atau data dan organisasi. XML tidak terbatas pada kumpulan markup tertentu yang dapat digunakan, pengguna dapat membuat markup sendiri agar sesuai dengan kebutuhan data dan dokumen pengguna. Fleksibilitas XML telah menyebabkan penggunaan luas untuk pertukaran data dalam banyak bentuk. [18]

Di dalam XML pengguna dapat menambahkan format instruksi untuk tampilan konten dalam dokumen terpisah yang biasa disebut stylesheet. Pengguna dapat mengubah tampilan pada stylesheet tanpa mengubah format XML yang telah dibuat. Pengguna dapat mengubah seluruh tampilan XML yang telah dibuat hanya dengan mengedit stylesheet. XML dapat menerima kombinasi dua tipe stylesheet

yang berbeda yakni CSS (Cascading Style Sheets) serta XSLT (Extensible Stylesheet Language Transformations).

Penggunaan XML pada penelitian ini dikarenakan XML dapat digunakan untuk membuat format tampilan antarmuka aplikasi dalam pemrograman android.

#### **2.14 Unified Modelling Language (UML)**

Unified Modelling Language (UML) merupakan sebuah bahasa yang telah menjadi standar dalam untuk merancang dan mengembangkan sebuah sistem perangkat lunak [19]s. UML memiliki enam keunggulan yang utama yakni:

1. Bahasa Formal

Setiap elemen yang terdapat pada uml memiliki makna yang jelas, sehingga tidak akan terjadi perbedaan pendapat saat memodelkan sistem dengan elemen-elemen uml.

2. Singkat

Seluruh bahasa yang digunakan dalam uml terdiri dari notasi yang sederhana dan mudah dipahami.

3. Komprehensif

Dapat menggambarkan semua aspek penting dari sebuah sistem yang akan dibangun.

4. Dapat diminimalisir

Dapat digunakan untuk proyek besar maupun proyek kecil.

5. Dibangun dari pelajaran yang terjadi

UML merupakan puncak dari penggambaran terbaik untuk kasus sistem berorientasi objek selama 15 tahun terakhir.

6. Standar

UML dikendalikan oleh kelompok standar terbuka yang merupakan kelompok vendor dan akademisi dari seluruh dunia.

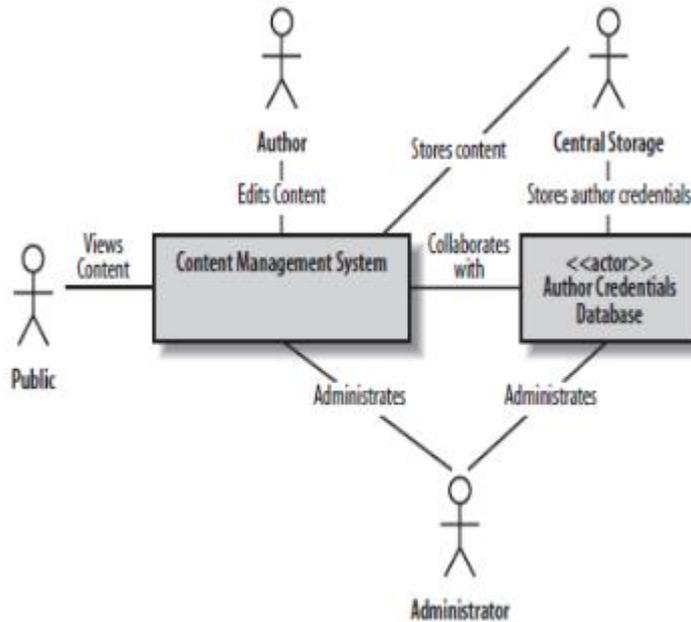
Dalam UML terdapat diagram-diagram yang dapat mewakili gambaran dari sistem yang akan dibangun, seperti: Use Case Diagram, Activity Diagram, Sequence Diagram, dan Class Diagram. Versi pertama dari UML memungkinkan pengguna untuk menjelaskan desain sistem dengan jelas.

Penggunaan UML pada penelitian ini bermaksud untuk mempermudah dalam penggambaran perancangan sistem yang akan dibangun pada tahap analisis. Program yang dibangun akan menggunakan bahasa pemrograman android yakni Java, yang sangat cocok untuk digambarkan dengan bahasa UML [19].

### **2.15 Use Case Diagram**

Use case diagram merupakan interaksi antara pengguna dengan sistem atau sistem eksternal lainnya yang dapat membantu untuk pemetaan sistem yang akan dibangun. Use case merupakan titik awal yang dilakukan untuk setiap aspek pengembangan, desain serta pengujian dan dokumentasi sistem berorientasi objek. Use case menggambarkan persyaratan yang ada dalam sebuah sistem. Use case juga membantu dalam pembangunan tes untuk sebuah sistem. [19]

Sebuah use case dapat melakukan include dan extend pada use case lain. Include menyatakan bahwa sebuah use case harus melewati semua langkah dari use case yang melakukan include. Sedangkan extend merupakan deklarasi warisan dari use case lain, dimana sebuah use case meluas ke use case lainnya. Use case menggambarkan cara bagaimana sebuah sistem berperilaku untuk memenuhi syarat. Dalam sebuah diagram use case tidak memiliki batasan maksimal untuk menggunakan use case. Jumlah kasus penggunaan tergantung pada pekerjaan yang dilakukan oleh sistem serta harus sesuai dengan persyaratan sistem yang akan dibangun.



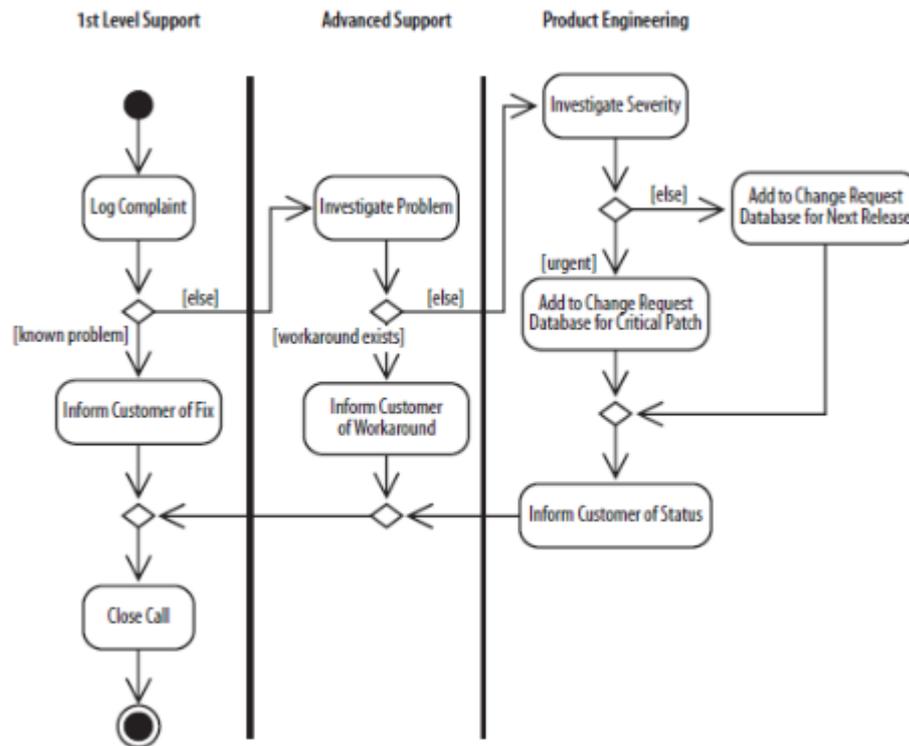
**Gambar 2.5 Contoh Usecase Diagram**

Penggunaan use case diagram dapat membantu penggambaran mengenai pekerjaan yang dapat dilakukan oleh aktor di dalam sebuah sistem yang akan dibangun. Oleh karena itu use case diagram sangat dibutuhkan dalam penelitian ini.

## 2.16 Activity Diagram

Activity diagram menggambarkan bagaimana sistem yang dibangun dapat mencapai tujuannya. Activity diagram menunjukkan tingkat dari tindakan yang dihubungkan bersama untuk mewakili proses yang terjadi pada sebuah sistem. Activity diagram merupakan satu-satunya diagram di dalam UML yang menampilkan model proses sebuah sistem. [19]

Activity Diagram sangat baik dalam pemodelan proses bisnis. Proses bisnis merupakan serangkaian tugas yang terkoordinasi untuk mencapai tujuan bisnis seperti pengiriman pesan pelanggan. Activity diagram memungkinkan pengguna untuk menentukan bagaimana sebuah sistem akan mencapai tujuannya.



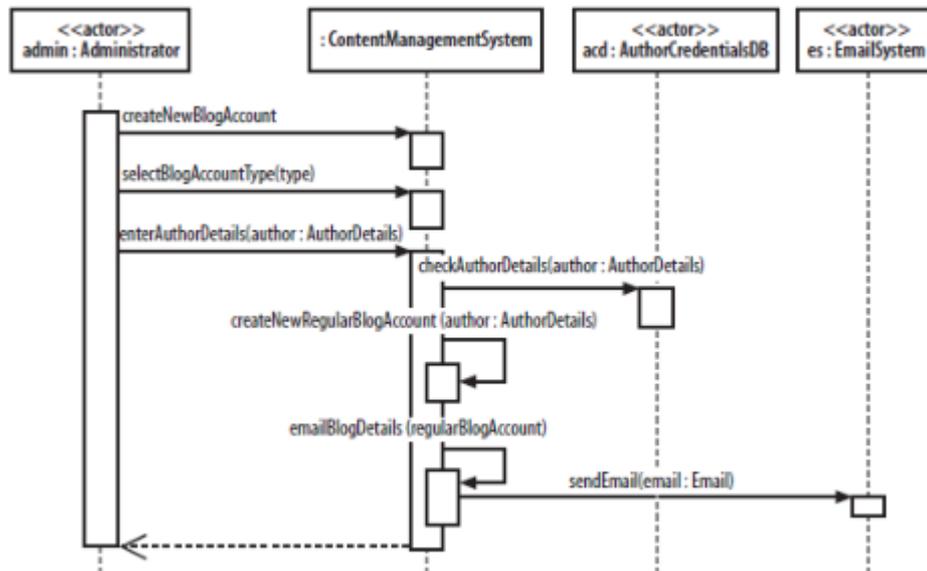
**Gambar 2.6 Contoh Activity Diagram**

Penggunaan activity diagram sangat dibutuhkan dalam penelitian ini karena dalam tahap analisis dibutuhkan gambaran proses aktivitas yang dapat dilakukan oleh sebuah sistem.

### 2.17 Sequence Diagram

Sequence diagram merupakan diagram penting dalam UML yang dikenal sebagai diagram interaksi. Sequence diagram adalah diagram interaksi yang paling terkenal. Hal ini dikarenakan sequence diagram dapat menunjukkan jenis informasi secara tepat. [20]

Sequence diagram menggambarkan urutan interaksi antar bagian pada sebuah sistem. Dengan menggunakan sequence diagram, pengguna dapat menggambarkan interaksi mana yang akan dipicu saat kasus penggunaan tertentu dieksekusi. Sequence diagram terdiri dari kumpulan bagian sistem yang berinteraksi satu sama lain secara berurutan, di mana bagian sistem yang terdapat dalam diagram ini merupakan bagian yang penting dalam sistem.



**Gambar 2.7 Contoh Sequence Diagram**

Penggunaan sequence diagram sangat dibutuhkan dalam penelitian ini karena dalam tahap analisis dibutuhkan gambaran mengenai masukan serta keluaran yang dihasilkan dalam sebuah aktivitas yang dapat memperjelas proses yang dapat dilakukan oleh sistem yang akan dibangun.

## 2.18 Class Diagram

Class diagram merupakan kumpulan class dari sebuah sistem. Sebagai permulaan, class bisa memiliki hubungan satu sama lainnya. Sebuah class bisa menjadi generalisasi dari class lain atau bisa juga berisi objek-objek dari kelas lain tergantung pada seberapa kuat hubungan antara sebuah class dengan class lainnya. [19]

Kekuatan hubungan antar class didasarkan pada seberapa tergantungnya class yang terlibat dalam hubungan itu satu sama lain. Dua class yang saling bergantung satu sama lainnya dapat dikatakan digabungkan. Perubahan pada suatu class kemungkinan besar akan berdampak pada class lainnya. [19]

Dalam sebuah class terdapat beberapa hubungan di antaranya:

1. Dependency

Ketika objek dari suatu class bekerja secara singkat dengan objek dari class lainnya.

2. Association

Ketika objek dari satu class bekerja dengan objek class lain untuk waktu yang lama.

3. Aggregation

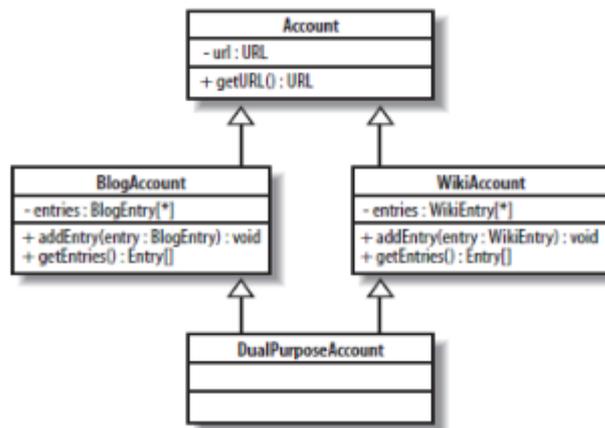
Ketika satu class utama berbagi referensi objek ke class lainnya.

4. Composition

Ketika satu class berisikan objek dari class lainnya.

5. Inheritance

Ketika class tersebut merupakan jenis dari class lainnya. [19]



**Gambar 2.8 Contoh Class Diagram**

Penggunaan Class Diagram sangat dibutuhkan pada penelitian ini karena sangat dibutuhkan gambaran untuk memperjelas hubungan antar class pada sistem nantinya akan berjalan.