



BAB II

TINJAUAN PUSTAKA

2.1 Referensi Penelitian Sebelumnya

1. Penelitian yang dilakukan oleh Erviana Devie dan Edy Winarno pada tahun 2019 yang berjudul “Aplikasi *Location Based Service* Untuk Informasi Kuliner Di Yogyakarta”. Pada jurnal ini menjelaskan mengenai aplikasi pencarian rute berbasis android menggunakan *Location Based Service* serta memberikan informasi dan pemahaman mengenai *Location Based Service*.
2. Penelitian yang dilakukan oleh Swono Sibagariang pada tahun 2016 yang berjudul “Penerapan *Web Service* Pada Perpustakaan Berbasis *Android*”. Tujuan dari penelitian ini adalah membahas tentang pengembangan dan penggunaan *webservice* pada perpustakaan.
3. Penelitian yang dilakukan oleh Rinaldy Maulidiansyah, Deny Fauzy Rakhman, dan Muhammad Ali Ramadhani pada tahun 2017 yang berjudul “Aplikasi Kerusakan Jalan Tol Menggunakan Layanan *Webservice* Berbasis *Android*”. Pada penelitian ini menjelaskan mengenai bagaimana mengimplementasikan layanan *webservice* dengan metode REST (*Representational State Transfer*) pada perangkat mobile berbasis *android*.
4. Penelitian yang dilakukan oleh Gusmira dan Heri Sutanta pada tahun 2019 yang berjudul “Pengembangan Sisem Pelaporan dan Pemetaan Kerusakan Infrastruktur Berbasis Android Menggunakan Metode *Volunteered Geographic Information* (VGI)”. Pada jurnal ini menjelaskan mengenai merancang sisitem yang dapat melaaporkan serta memetakan kerusakan infrastruktur sesuai kondisi nyata.
5. Penelitian yang dilakukan oleh Iin Parlina, Agus Perdana Windarto, Anjar Wanto dan M.Ridwan Lubis pada tahun 2018 yang berjudul “Memanfaatkan Algoritma K-Means Dalam Menentukan Pegawai Yang Layak Mengikuti Assesment Canter Untuk Clustering Program SDP”.



Pada jurnal ini menjelaskan mengenai algoritma K-Means dan mengimplementasikan didalam sebuah aplikasi.

2.2 Teori Judul

2.2.1 *Location Based Service (LBS)*

Menurut Devie dan Winarno (2018:16) *Location Based Service (LBS)* merupakan salah satu metode layanan berbasis lokasi untuk mengetahui rute perjalanan, yang memberikan hasil dengan tingkat keakuratan yang cukup tinggi serta dapat memberikan informasi lokasi. Layanan ini menggunakan teknologi *global positioning service (GPS)* dan *cell-based location* dari Google.

Jadi dapat disimpulkan bahwa *Location Based Service (LBS)* adalah sebuah layanan informasi yang dapat diakses menggunakan perangkat *mobile* dengan menggunakan jaringan internet dan *smartphone* yang memanfaatkan kemampuan maps pada perangkat *smartphone*.

Budiman (2016:138) mengemukakan bahwa terdapat dua unsur utama pada *Location Based Service (LBS)* yaitu:

1. *Location Manager (API Maps)*

Menyediakan tool untuk LBS, *Application Programming Interface (API)* Maps menyediakan fasilitas untuk menampilkan, memanipulasi maps atau peta beserta fitur-fitur lainnya seperti tampilan satelit, jalan, maupun gabungannya.

2. *Location Provider (API Location)*

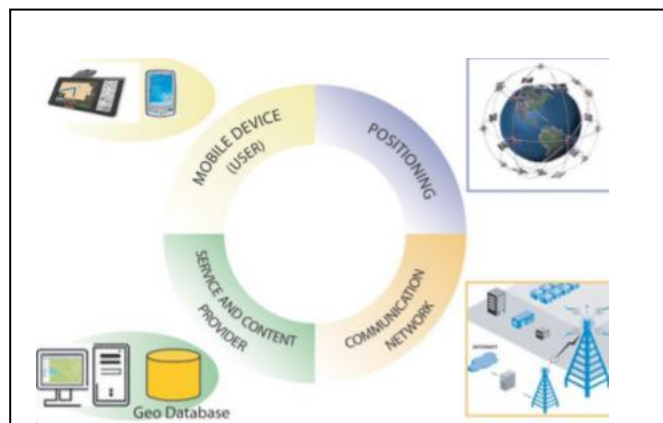
Menyediakan teknologi pencarian lokasi yang digunakan oleh device/perangkat. API Location berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi real time.

Selain itu *Location Based Service (LBS)* terdiri dari beberapa komponen di antaranya:

1. *Mobile Device* yaitu sebuah alat yang digunakan untuk meminta informasi yang dibutuhkan. Biasanya perangkat yang memungkinkan yaitu *Mobile Phone*, Laptop, dan perangkat lainnya yang mempunyai fasilitas navigasi.
2. *Communication Network* adalah jaringan seluler yang mengirimkan data pengguna dan permintaan layanan.



3. *Positioning Component* untuk pengolahan layanan biasanya posisi pengguna harus ditentukan. Posisi pengguna dapat diperoleh menggunakan jaringan komunikasi atau dengan menggunakan *Global Positioning System (GPS)*.
4. *Service and Application Provider* yaitu penyedia layanan pengguna seluler yang bertanggung jawab untuk memproses layanan. Provider melakukan proses komputasi seperti menemukan rute perjalanan, informasi tempat terdekat, mencari ke google API untuk menghasilkan informasi yang dibutuhkan pengguna.
5. *Data and Content Provider* yaitu penyedia layanan informasi data yang dapat diminta oleh pengguna. Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang atau pihak ketiga yang memang memiliki otoritas untuk menyimpannya.



Gambar 2.1. Komponen *Location Based Service*

Sederhananya, dengan layanan *Location Based Service (LBS)* kita dapat mengetahui posisi dimana kita berada. Langkah awal yang dilakukan oleh metode *Location Based Service* dalam pencarian lokasi yaitu mengaktifkan fungsi pencarian dimana posisi pengguna yang ada di ponsel diperoleh dari *positioning service* pada GPS. Setelah itu perangkat ponsel akan mengirim informasi yang berisi tujuan untuk mencari dan mengirimkan posisi melalui jaringan komunikasi ke gateway. Kemudian, service menganalisis pesan dan memutuskan informasi



tambahan pencarian dan posisi pengguna dan selanjutnya service akan menemukan bahwa informasi mengenai jalan, jarak dan memeriksa apakah tujuan tersebut dapat dicapai dan akan ditampilkan kepada pengguna dalam bentuk peta.

2.2.2 Pelaporan

Pelaporan merupakan suatu kegiatan atau aktivitas yang dilakukan untuk menyampaikan hal-hal yang berhubungan dengan hasil pekerjaan yang dilakukan dalam satu periode tertentu, pelaporan dapat berupa lisan maupun tulisan.

Menurut Gullick di dalam buku *Papers on the Science of Administration* (Kusriyanti, Saragih., *et.al* 2019) mengemukakan bahwa pelaporan merupakan salah satu fungsi manajemen berupa penyampaian perkembangan atau hasil kegiatan atau pemberian keterangan mengenai segala hal yang bertalian dengan tugas dan fungsi-fungsi kepada pejabat yang lebih tinggi.

Mukhsinuddin (2017:50) mengatakan bahwa reporting atau pelaporan adalah salah satu fungsi penyampaian manajemen berupa perkembangan atau hasil kegiatan atau pemberian keterangan mengenai segala hal yang bertalian dengan tugas dan fungsi kepada pejabat yang lebih tinggi baik secara lisan maupun tulisan.

2.2.3 Kerusakan Jalan

Kerusakan pada jalan akan mempengaruhi laju kendaraan yang melintasi jalan tersebut, dikarenakan bagian jalan yang rusak akan cenderung dihindari oleh pengendara kendaraan, dan pengendara kendaraan harus mengurangi kecepatan kendaraannya untuk menghindari kerusakan jalan tersebut. Sehingga menyebabkan kemacetan serta menjadi salah satu faktor penyebab terjadinya kecelakaan lalu lintas (Wicaksono, Pradana., *et.al*, 2019:3825). Kerusakan jalan menunjukkan suatu kondisi dimana struktural dan fungsional jalan sudah tidak mampu memberikan pelayanan yang optimal terhadap pengendara lalu lintas yang melintasi jalan tersebut.

Juwita dan Ariadi (2018:67) berpendapat kerusakan pada konstruksi perkerasan jalan dapat disebabkan oleh:



1. Lalu lintas, yang dapat berupa peningkatan beban dan repitisi beban.
2. Air hujan, sistem drainase jalan yang tidak baik, naiknya air akibat sifat kapilaritas.
3. Material kontruksi perkerasan.
4. Iklim
5. Kondisi tanah dasar yang tidak stabil.
6. Proses pemadatan lapisan di atas tanah dasar yang kurang baik.

Menurut Juwita dan Ariadi (2018:69) jenis-jenis kerusakan jalan umumnya dapat diklasifikasikan sebagai berikut:

1. Deformasi berupa bergelombang, alur, amblas, sungkur, mengembang, benjol dan turun.
2. Retak berupa retak diagonal, retak memanjang, retak melintang, retak reflektif.
3. Kerusakan tekstur permukaan berupa pelepasan butiran, pengausan agrerat, dan pengelupasan.
4. Kerusakan lubang berupa tambalan dan persilangan rel.
5. Kerusakan dipinggir perkerasan berupa retak pinggir dan penurunan bahu jalan.

2.2.4 Android

Menurut Yudhanto dan Wijayanto (2017:1), menyatakan android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Selain itu android didefinisikan sebagai sebuah sistem operasi perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi (Supardi, 2017:1). Dengan demikian penulis dapat menarik kesimpulan dari definisi android di atas, bahwa android merupakan sistem operasi yang dirancang berbasis Linux untuk mendukung kinerja sebuah smartphone yang mencakup middleware dan aplikasi.



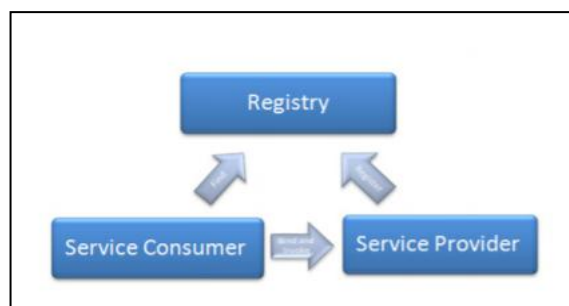
2.2.5 Webservice

Sibagariang (2016:29) berpendapat bahwa *webservice* adalah suatu bentuk sistem perangkat lunak yang didesain untuk mendukung interaksi mesin-ke-mesin melalui jaringan.

Menurut Maulidiansyah, Rakhman., *et.al* (2016:118) *webservice* adalah sebuah entitas komputasi yang dapat diakses melalui jaringan internet maupun intranet dengan standar protokol tertentu dalam platform dan antarmuka bahasa pemrograman yang independen.

Christianto dan Santosa (2018:7) mengemukakan terdapat beberapa arsitektur webservice yaitu:

- Find*, dilakukan oleh service consumer terhadap server yang memiliki daftar layanan yang dapat digunakan serta cara untuk menggunakannya.
- Bind and invoke*, dilakukan apabila aggreemant antara service consumer dan service provider telah didaftarkan pada registry server.
- Register*, dilakukan oleh services provider untuk mendaftarkan service yang dapat digunakan oleh services consumer pada registry server.
- Service contract*, bagian ini adalah antarmuka service yang mendefinisikan komponen-komponen yang terdapat pada sebuah services.



Gambar 2.2. Arsitektur Webservices

Webservice memiliki interface yang dideskripsikan dalam format yang dapat dibaca oleh mesin. *Webservice* digunakan sebagai suatu fasilitas yang disediakan oleh suatu website untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut. Umumnya sistem-sistem tersebut berinteraksi dengan *webservice*



menggunakan pesan SOAP yang umumnya dikirim melalui HTTP dalam bentuk XML.

2.2.6 Definisi Judul Secara Keseluruhan

Penerapan Metode *Location Based Service* (LBS) Untuk Pelaporan Kerusakan Jalan Berbasis *Android Webservice* adalah sebuah aplikasi yang dirancang untuk menyediakan informasi mengenai daerah yang jalannya mengalami kerusakan agar proses dalam menangani kerusakan jalan dapat menjadi lebih efektif dan efisien dengan pemanfaatan teknologi informasi khususnya *smartphone*. Aplikasi yang dibangun menggunakan metode *Location Based Service* (LBS) untuk memberikan informasi mengenai posisi lokasi saat ini dan rute lokasi yang akan dituju. Serta aplikasi yang akan dibangun berbasis *Android Webservice*.

2.3 Teori Khusus

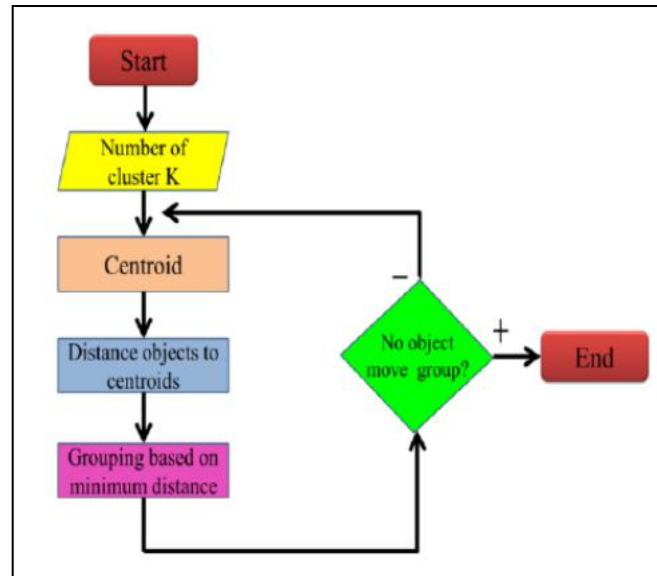
2.3.1 *K-Means*

Menurut Wu dan Kumar dalam Sibuea dan Sapta (2017:87) *K-Means* merupakan suatu algoritma pengklasteran yang cukup sederhana yang mempartisi data ke dalam beberapa kluster. *K-Means* algoritma cukup mudah diimplementasikan dan dijalankan, relatif cepat, mudah disesuaikan dan banyak digunakan.

Darmi dan Setiawan (2016:149) berpendapat bahwa *K-Means* merupakan algoritma *clustering* yang berulang-ulang. Algoritma *K-Means* dimulai dengan pemilihan secara acak K , K disini merupakan banyaknya *cluster* yang ingin dibentuk. Kemudian tetapkan nilai K secara random, untuk sementara nilai tersebut menjadi pusat dari *cluster* atau biasa disebut dengan *centroid*, mean atau means.

K-Means akan memilih pola k sebagai titik awal centroid secara acak. Jumlah iterasi untuk mencapai cluster centroid akan dipengaruhi oleh kandidat cluster centroid awal yang ditentukan secara random dimana jika posisi *centroid* baru tidak berubah.

Proses dasar algoritma *K-Means* menurut Parlina, Windarto, *et.al*, (2018:88) dapat dilihat dibawah ini:

Gambar 2.3. Flowchart *K-Means*

Algoritma *K-means* yang digambarkan dalam gambar 2.3 dijelaskan sebagai berikut:

1. Inisialisasi: menentukan nilai *K centroid* yang diinginkan dan metrik ketidakmiripan (jarak) yang diinginkan.
2. Memilih *K* data dari set *X* sebagai *centroid*. Untuk menentukan *centroid* dapat menggunakan persamaan.
3. Mengalokasikan semua data ke *centroid* terdekat dengan metrik jarak yang telah ditetapkan
4. Menghitung kembali *centroid* *C* berdasarkan data yang mengikuti *cluster* masing-masing.
5. Mengulangi langkah 3 dan 4 hingga kondisi konvergen tercapai.

Berikut adalah rumus untuk menentukan jumlah *cluster*:

$$K = \sqrt{\frac{N}{2}}$$

Keterangan:

K= Klaster

N= Jumlah Data

Menghitung jarak pada ruang jarak *Euclidean* menggunakan formula:



$$d_{ik} = \sqrt{\sum_j^m (C_{ij} - C_{kj})^2}$$

Keterangan:

d_{ik} = jarak antara data ke *centroid* dengan *centroid* ke- k

m = jumlah atribut

C_{ij} = data ke- i

C_k = data pusat klaster ke- k

2.3.2 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan pemodelan yang menggunakan konsep orientasi object, untuk menspesifikasikan dan membangun sistem perangkat lunak. Dharwiyanti dalam Rusmawan (2019:71) berpendapat bahwa *Unified Modeling Language* (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

Menurut Isa dan Hartawan (2017:141) *Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek.

Beberapa diagram *Unified Modeling Language* (UML) yang akan digunakan dalam pembuatan tugas akhir ini antara lain sebagai berikut:

2.3.2.1 Usecase Diagram

Rusmawan (2019:72) berpendapat *Use case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

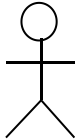
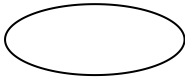
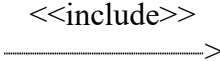
Menurut Isa dan Hartawan (2017:141) *Use case* diagram digunakan untuk menggambarkan sistem dari sudut pandang pengguna sistem tersebut (*user*).

Use case diagram menjelaskan interaksi yang terjadi antara aktor dan inisiator dari interaksi sistem tersebut dengan sistem yang ada, sebuah *use case* dipresentasikan dengan urutan langkah yang sederhana. *Usecase* diagram dapat digunakan untuk menentukan kebutuhan apa saja yang diperlukan suatu sistem,



jadi dapat digambarkan dengan detail bagaimana suatu sistem memproses atau melakukan sesuatu. Berikut merupakan simbol-simbol diagram *usecase*:

Tabel 2.1 Simbol-simbol Diagram *Usecase*

No.	Simbol	Nama	Keterangan
1.		Actor	Seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun
2.		Use case	Menggambarkan bagaimana seseorang menggunakan sistem
3.		Relasi asosiasi	Relasi yang dipakai untuk menunjukkan hubungan antara aktor dan use case
4.		Relasi include	Memungkinkan suatu use case menggunakan fungsionalitas yang disediakan oleh use case lainnya.
5.		Relasi extend	Memungkinkan suatu use case secara optional menggunakan fungsionalitas yang disediakan oleh use case lainnya.

(Sumber : Rusmawan 2019:72)

2.3.2.2 Activity Diagram

Nugroho dalam Rusmawan (2019:79) berpendapat bahwa *Activity diagram* digunakan untuk menampilkan rangkaian kegiatan, menunjukkan alur kerja dari suatu titik awal ke titik akhir keputusan, merinci banyak jalur yang ada dalam perkembangan peristiwa yang terkandung dalam kegiatan.

Isa dan Hartawan (2017:141) berpendapat bahwa *Activity Diagram* menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya.



Dari kedua pengertian diatas dapat disimpulkan bahwa activity diagram adalah diagram yang menggambarkan aliran kerja dari sebuah sistem yang digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi.

Berikut merupakan simbol-simbol yang digunakan dalam *activity diagram* yang dapat dilihat pada tabel berikut:

Tabel 2.2 Simbol-simbol *Activity Diagram*

No.	Simbol	Nama	Keterangan
1.		Start state	Titik awal atau permulaan
2.		End state	Titik akhir atau akhir dari aktivitas
3.		Activity	Activity atau aktivitas yang dilakukan oleh aktor
4.		Decision	Pilihan untuk mengambil keputusan
5.		Interaction	Alur

(Sumber : Rusmawan 2019:80)

2.3.2.3 Sequence Diagram

Menurut Nugroho dalam Rusmawan (2019:84) *sequence diagram* adalah diagram interaksi yang disusun berdasarkan urutan waktu. Setiap diagram sekuensial mempresentasikan satu flow dari beberapa flow di dalam use case.

Menurut Isa dan Hartawan (2017:141) *sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem.

**Table 2.3** Simbol-simbol Diagram Sekuensial

No.	Simbol	Nama	Keterangan
1.		<i>Actor</i>	Orang ataupun pihak yang akan mengelola sistem.
2.		<i>Lifeline</i>	Menggambarkan sebuah objek dalam sebuah sistem atau salah satu komponennya.
3.		<i>Create Message</i>	Pembuatan sebuah message sederhana antar elemen dan juga mengindikasikan komunikasi antara objek.
4.		<i>Synchronous Message</i>	Message ini mengaktifkan sebuah proses dan sampai selesai, baru bisa mengirimkan sebuah message baru.
5.		<i>Message to self</i>	Suatu hasil kembalian sebuah operasi dan berjalan pada objek itu sendiri.

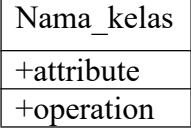
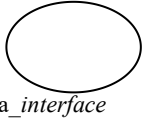
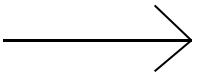
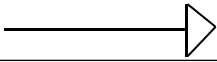
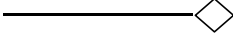
(Sumber : Rusmawan 2019:84)

2.3.2.4 Class Diagram

Isa dan Hartawan (2017:141) berpendapat bahwa *class diagram* adalah spesifikasi yang akan menghasilkan objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metode atau fungsi).

Menurut Hendini (2016:111) diagram class (*class diagram*) merupakan hubungan antar kelas dan penjelasan hasil detail tiap-tiap kelas di dalam model desain suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Berikut merupakan simbol-simbol class diagram.

**Tabel 2.4** Simbol-simbol Diagram Kelas

No	Simbol	Keterangan
1.	Kelas 	Kelas pada struktur sistem
2.	Antarmuka/Interface 	Sama dengan konsep interface dalam pemrograman berorientasi obyek.
3.	Asosiasi/association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity.
4.	Asosiasi berarah/directed association 	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan multiplicity.
5.	Generalisasi 	Asosiasi antar kelas dengan makna generalisasi spesialisasi (umum – khusus)
6.	Kebergantungan/dependency 	Ketergantungan antarkelas.
7.	Agregasi/ aggregation 	Relasi antar kelas dengan makna umum semua bagian (whole-part)

(Sumber: Aprianti dan Maliha, 2016:22)

2.4 Teori Program

2.4.1 Android Studio

Android Studio merupakan salah satu modern *code editor* yang sangat membantu dalam pembangunan suatu aplikasi android. Sebagai IDE (*Integrated Development Environment*) resmi yang disarankan oleh Google dalam pembangunan aplikasi android. Android studio menawarkan banyak fitur dan *tools* yang dapat digunakan oleh pengembang aplikasi Android (Hansun dan Kristanda, 2018:27).



Menurut Yudhanto dan Wijayanto (2017:17), Android Studio merupakan software tools *Integrated Development Environment* (IDE) untuk platform android.

Dari kedua pengertian diatas disimpulkan bahwa android studio merupakan salah satu modern code editor yang membantu dalam pembangunan suatu aplikasi android dan merupakan software tools *Integrated Development Environment* (IDE) resmi yang disarankan oleh Google untuk platform android.

2.4.2 Java

Java merupakan sekumpulan teknologi yang digunakan untuk membuat dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan (Haqi dan Setiawan, 2019:1).

Java merupakan bahasa pemrograman tingkat tinggi yang dipelopori oleh James Gosling yang merupakan engineer di Sun MicroSystem (Bachtiar dan Fakhrul, 2018:1).

Menurut kedua pengertian diatas dapat disimpulkan bahwa java merupakan bahasa pemrograman yang berorientasi objek dan program java terdiri dari bagian yang disebut kelas.

2.4.3 Sublime Text Editor

Sublime text editor merupakan text editor yang di buat untuk mempermudah pekerjaan programmer dan digunakan untuk banyak bahasa pemrograman dan dibangun menggunakan python.

Putra, *et.al* dalam Pradiatiningtyas dan Suparwanto (2017:4) mendefinisikan sublime text adalah text editor berbasis Python, sebuah text editor yang elegan, kaya fitur, cross platform, mudah dan simple yang cukup terkenal dikalangan developer (pengembang) dan desainer.

Menurut Syifani dan Dores (2018:25) Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Python API. Terciptanya aplikasi ini terinspirasi



dari aplikasi Vim, Aplikasi ini sangatlah fleksibel dan powerfull. Fungsionalitas dari aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages.

2.4.4 PHP (*Hypertext PreProcessor*)

PHP (*Hypertext PreProcessor*) merupakan bahasa skrip pemrograman yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP (*Hypertext PreProcessor*) banyak dipakai untuk membangun sebuah CMS (Haqi dan Setiawan, 2019:9).

PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis website. Oleh karena itu, PHP dapat dijalankan menggunakan browser (Nugroho, 2019:1).

Dari kedua pengertian diatas dapat disimpulkan bahwa PHP adalah bahasa skrip pemograman berbasis *server side* yang dapat disisipkan ke dalam HTML untuk membuat aplikasi berbasis website dan dapat dijalankan menggunakan browser yang bersifat dinamis dan tampilan kontennya sesuai kondisi tertentu dan interaktif artinya dapat memberi *feedback* bagi *user*.

2.4.5 Pengertian Basis Data

C.J. Date dalam Rusmawan (2019:40), mengemukakan bahwa basis data adalah sekumpulan data persisten yang digunakan oleh sistem aplikasi dari suatu perusahaan. Sistem basis data pada dasarnya merupakan suatu sistem penyimpanan record atau data yang terkomputerisasi.

Menurut Kadir dalam Azis dan Sarmidi (2018:52), Basis data adalah suatu pengorganisasian sekumpulan data yang saling terkait sehingga memudahkan aktivitas untuk memperoleh informasi.

Menurut kedua pengertian diatas disimpulkan bahwa basis data merupakan sekumpulan data yang saling terkait dan disimpan secara bersama-sama dalam sebuah media tertentu yang digunakan oleh sistem aplikasi sehingga memudahkan aktivitas untuk memperoleh informasi.



2.4.6 MySQL

Menurut Rusmawan (2019:97) MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi diseluruh dunia.

Menurut Nugroho dalam Destiningrum dan Adrian (2017:33), MySQL (*My Structured Query Language*) adalah suatu sistem basis data relation atau *Relational Database Managemnt System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan. MySQL juga merupakan program pengakses database yang bersifat jaringan, sehingga sapat digunakan untuk aplikasi multi user (banyak pengguna). MySQL didistribusikan gratis dibawah lisensi GPL (General Public License). Dimana setiap program bebas menggunakan MySQL namun tidak bisa dijadikan produk turunan yang dijadikan closed source atau komersial.

Dapat disimpulkan bahwa MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang mampu bekerja secara cepat dan mudah untuk digunakan, serta merupakan sistem basis data yang bisa mengakses database yang bersifat jaringan.

2.4.7 Xampp

XAMPP merupakan perangkat lunak bebas (free software) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program (Haqi dan Setiawan, 2019:8).

Menurut Andi dalam Sandra dan Pratiwi (2017:60), XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP*, dan *Perl*. XAMPP adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Dalam paket XAMPP sudah terdapat *Apache* (*web server*), *MySQL* (*database*), *PHP* (*server side scripting*), *Perl*, *FTP server*, *PhpMyAdmin* dan berbagai pustaka bantu lainnya.

Dari kedua pengertian diatas dapat disimpulkan bahwa XAMPP perangkat lunak bebas (free software) yang merupakan sebuah paket kumpulan software



yang terdiri dari Apache, MySQL, PhpMyAdmin, PHP, Perl, Filezilla, dan lain-lain.