

BAB II

LANDASAN TEORI

2.1. Algoritma Haversine Formula

Algoritma Haversine Formula adalah sebuah persamaan yang penting dalam bidang navigasi, untuk mencari jarak busur antara dua titik pada bola dari longitude dan latitude. Ini merupakan bentuk persamaan khusus dari trigonometri bola, law of haversine, mencari hubungan sisi dan sudut pada segitiga dalam bidang bola. Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza y Rios di tahun 1801. Istilah haversine ini sendiri diciptakan pada tahun 1833 oleh Prof.James Inman. Josef de Mendoza y Rios menggunakan haversine pertama kali dalam penelitiannya tentang “Masalah Utama Astronomi Nautical”, Proc.Royal Soc, Dec 22. 1796. Haversine digunakan untuk menemukan jarak antara bintang. Haversine Formula ialah persamaan penting dalam sistem navigasi, nantinya formula haversine ini akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (longitude) dan garis lintang (latitude). haversine formula merupakan penerapan dari konsep trigonometri yang merupakan bagian dari geometri. Rumus Haversine ialah persamaan yang penting dalam navigasi, yang menghasilkan jarak lingkaran besar antara dua titik (latitude dan longitude) pada permukaan bola (bumi) berdasarkan bujur dan lintang. Penggunaan rumus ini cukup akurat untuk sebagian besar perhitungan, juga mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi.

Berikut bentuk Rumus Haversine Formula :

$$\begin{aligned}
 x &= (\text{lon2} - \text{lon1}) \times \cos\left(\frac{\text{lat1} + \text{lat2}}{2}\right) \\
 y &= (\text{lat2} - \text{lat1}) \\
 d &= \sqrt{((x \times x) + (y \times y))} \times R
 \end{aligned}$$

Keterangan :

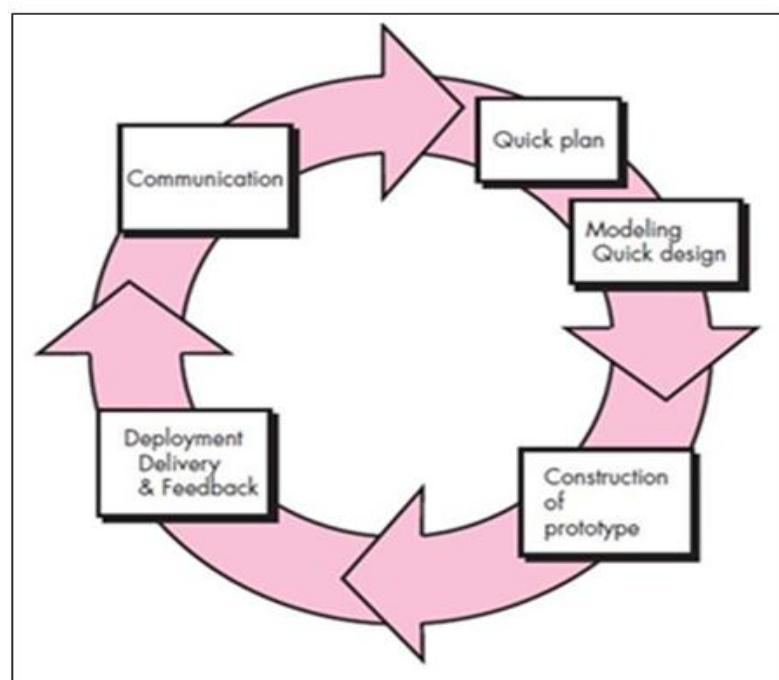
- lat1 = latitude awal
- lon1 = longitude awal
- lat2 = latitude tujuan
- lon2 = longitude tujuan
- x = longitude (Lintang)
- y = latitude (Bujur)
- d = jarak
- R = radius bumi (6371 km)
- 1 derajat = 0,0174532925

2.2. Metode Pegembangan Perangkat Lunak yang digunakan

2.2.1. *Prototype*

Pressman (2017) menguraikan bahwa Prototype adalah proses pembuatan model sederhana software yang mengijinkan pengguna memiliki gambaran dasar tentang program serta melakukaan pengujian awal. Prototype memberikan fasilitas bagi pengembang dan pengguna untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan dibuat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali. Metode ini dimulai dengan pengumpulan kebutuhan pengguna. Kemudian membuat sebuah rancangan kilat yang

selanjutnya akan dievaluasi kembali sebelum di produksi secara benar. Prototype bukanlah merupakan sesuatu yang lengkap, tetapi sesuatu yang harus dievaluasi dan dimodifikasi kembali. Segala perubahan dapat terjadi pada saat prototype dibuat untuk memenuhi kebutuhan pengguna dan saat yang sama memungkinkan pengembangan untuk lebih memahami kebutuhan pengguna secara baik. Gambar 2.1 berikut merupakan Diagram Prototype.



Gambar 2.1 Diagram *Prototype*

Penjelasan dari gambar 2.1 berikut adalah :

- 1) Komunikasi (*Communication*) : pengumpulan data awal, yaitu komunikasi dengan klien dan user untuk menentukan kebutuhan.
- 2) Perencanaan Cepat (*Quick Plan*) : pembuatan perencanaan analisis terhadap kebutuhan pengguna.
- 3) Pemodelan Perancangan Cepat (*Modeling Quick Design*) : membuat rancangan desain program.

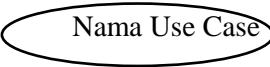
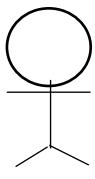
- 4) Pembentukkan *Prototype* (*Construction of prototype*) : pembuatan aplikasi berdasarkan dari pemodelan desain yang telah dibuat.
- 5) Penyerahan Sistem dan Umpan Balik (*Development Delevery and Feedback*) : memproduksi perangkat secara benar sehingga dapat digunakan oleh pengguna.

2.3. Komponen *Unified Modelling language* (UML)

2.3.1. *Use Case Diagram*

Rosa & Shalahuddin (2018) menguraikan bahwa *Use Case Diagram* merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya *use case* tidak hanya penting pada tahap analisis, tetapi juga sangat penting untuk perancangan, untuk mencari kelas-kelas yang terlibat dalam aplikasi, serta untuk melakukan pengujian. *Diagram use case* bersifat statis, diagram ini memperlihatkan himpunan use-case dan aktor-aktor (suatu jenis khusus dari kelas). Simbol – simbol Use Case dapat dilihat pada tabel 2.1.

Tabel 2.1 Simbol Use Case Diagram

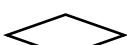
Simbol	Keterangan
Use Case 	Fungsionalitas yang disediakan sistem sebagai unit – unit yang saling bertukar pesan antar unit dan aktor.
Aktor / Actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi.
Asosiasi / Association 	Komunikasi antar aktor dan Use Case yang berpartisipasi.

Ekstensi / extend 	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambah dapat berdiri sendiri walau tanpa Use Case tambahan.
Generalisasi / generalization 	Hubungan generalisasi dan spesialisasi antara dua buah Use Case yang mana fungsi yang satu lebih umum dari yang lainnya.
Include / Use Case  	Relasi Use Case tambahan ke sebuah Use Case dimana Use Case yang ditambahkan memerlukan Use Case ini untuk menjalankan fungsinya.

2.3.2. *Activity Diagram*

Rosa & Shalahuddin (2018) menguraikan bahwa *Activity diagram* menggambarkan *workflow* (alir kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Tahap perancangan *activity diagram* menjabarkan masing – masing *activity* pada perancangan *use case*. Simbol – simbol *Activity Diagram* dapat dilihat pada tabel 2.2

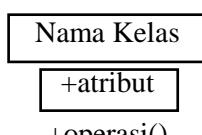
Tabel 2.2. Simbol Activity Diagram

Simbol	Keterangan
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Status awal 	Bagaimana objek dibentuk atau diawali.
Status akhir 	Bagaimana objek dibentuk dan diakhiri.
Percabangan 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu

2.3.3. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi (A.S Rosa, & Shalahuddin, M., 2018). simbol-simbol yang ada pada diagram kelas pada tabel *class diagram* 2.3 di bawah ini:

Tabel 2.3. Simbol Class Diagram

Simbol	Keterangan
Kelas 	Kelas pada struktur system.
Antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity.
Asosiasi berarah / directed association 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity.
Generalisasi 	Relasi antarkelas dengan makna Generalisasi – spesialisasi (umum khusus).
Kebergantungan / dependency 	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi / aggregation 	Relasi antar kelas dengan makna semua bagian

2.4. Perangkat Lunak Pengembangan Sistem

2.4.1. Android studio

Firly (2018) menjelaskan bahwa Android studio merupakan integrated development environment (IDE) atau dalam artian lain adalah sebuah lingkungan pengembangan terintegrasi resmi yang memang merancang khusus untuk pengembangan sistem operasi Google Android. Aplikasi ini dibangun di atas sebuah perangkat lunak yang dinamakan IntelliJ IDEA milik JetBrains. Bisa juga dibilang bahwa android studio merupakan pengganti dari Eclipse android development tool atau ADT sebagai IDE utama dalam pengembangan aplikasi android yang asli. Android studio diluncurkan pada tanggal 16 mei 2013 dalam konferensi google I/O yang pada saat itu masih dalam tahap pratinjau akses versi 0.1 sebagai perintis. Hingga pada akhirnya versi stabil 3.0 yang liris pada pertengahan bulan oktober 2017 dan menjadi software terlaris dikalangan developer muda. Aplikasi ini dapat digunakan diberbagai sistem operasi yaitu windows, linux dan macOS. Aplikasi ini menawarkan berbagai fitur canggih yang akan meningkatkan kemampuan produktivitas dalam proses pengembangan aplikasi. Berikut ini adalah beberapa hal yang akhirnya banyak mengundang developer untuk melirik android studio sebagai software pengembang :

- a. Dukungan dari C++, NDK dan sekarang kotlin
- b. Perkembangan yang up to date
- c. Sistem berbasis Gradle yang dinilai fleksibel
- d. Lingkungan yang mencakup seluruh perangkat android
- e. Emulator yang cepat dan kaya akan fitur
- f. Alat pengujian dan kerangka yang juga ekstensif
- g. Instant Run
- h. Dukungan google cloud platfor

2.4.2. Java

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari sun microsystem yang memiliki kode green dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitektur tertentu, mulanya disebut oak, tetapi karena oak sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada, maka sun mengubahnya menjadi java.sun kemudian meluncurkan browser dari java yang disebut hot java yang mampu menjalankan applet. Setelah itu teknologi java diadopsi oleh Netscape yang memungkinkan program java dijalankan di browser netscape yang kemudian diikuti Internet Explore. Karena keunikan dan kelebihannya, teknologi java mulai menarik banyak vendor seperti IBM, Symantec, Inprise, dll. Sun merilis versi awal java secara resmi pada awal tahun (Suyanto, 2015), 1996 yang kemudian terus berkembang hingga muncul JDK 1.1 kemudian JDK 1.2 yang mulai disebut sebagai versi java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah swing yang merupakan teknologi GUI (Graphical UserInterface) yang mampu menghasilkan windows yang portable. Dan pada tahun 1998-1999 lahirlah teknologi J2EE (Java 2 Enterprise Edition).

1. JDK (Java Development Kit)

Java Development Kit atau JDK merupakan sebuah perangkat lunak yang berfungsi dalam proses memanajemenasi aplikasi java. Hal tersebut dikarenakan Anda akan menggunakan bahasa pemrograman Java dalam membuat aplikasi di Android Studio (Nadia Firly, 2019).

2. SDK (Software Development Kit)

Software Development Kit atau SDK, merupakan sebuah kit

yang berfungsi untuk mengembangkan berbagai aplikasi berbasis Android oleh para developer. Di dalam SDK telah terdapat berbagai tools yang bertujuan untuk proses pengembangan aplikasi seperti proses debugger, software libraries, emulator, dokumentasi, sample cod (Nadia Firly, 2019).

2.4.3. *Firebase*

Firebase adalah Cloud Service Provider dan Backend as a Service yang dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah dalam pengembangan aplikasi mobile maupun web dan bersifat Realtime Database. Firebase memiliki banyak SDK yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, IOS, Javascript, C++ hingga Unity. Pada penggunaan Firebase diperlukan akses internet dalam menjalankan aplikasi tersebut. Dikarenakan data yang tersimpan pada tempat penyimpanan berbasis Online (Nadia Firly, 2019).

a. Fitur Firebase

Google Firebase (2016), setelah API Firebase dimasukkan ke Dalam aplikasi Android atau iOS, pengembang bisa menggunakan fitur Firebase dengan coding yang simpel. Beberapa fitur yang disediakan Firebase, yaitu :

1. Analytics

Fitur ini membuat para pengembang aplikasi dapat mengerti bagaimana para pengguna menggunakan aplikasi mereka. SDK analytics menangkap sendiri data yang dibutuhkan oleh para developer. Dashboard juga menyediakan detail seperti pengguna paling aktif atau fitur apa saja yang paling digunakan dalam aplikasi tersebut. Analytics juga menyediakan data yang telah dirangkum.

2. Authentication

Fitur ini membuat para pengembang dapat mengijinkan pengguna untuk mengakses aplikasi. Firebase menyediakan fitur login melalui Gmail, Github, Twitter, Facebook dan juga autentikasi buatan sendiri.

3. Messaging

FCM (Firebase Cloud Messaging) membuat para pengguna dapat mengirimkan pesan ke berbagai platform tanpa biaya tambahan. Messaging juga dapat digunakan untuk kebutuhan notifikasi.

4. Real-time Database

Database di Firebase adalah database berbasis cloud dan tidak membutuhkan SQL untuk mengambil dan menyimpan data atau bisa disebut juga NoSQL. Database ini sangatlah cepat dan dapat diandalkan yang artinya data dapat dibaharui dan disinkronisasikan dengan cepat. Data juga dijaga meskipun pengguna kehilangan koneksi internetnya.

5. Storage

Firebase juga menyediakan fasilitas penyimpanan. Firebase dapat menyimpan dan mengambil konten seperti gambar, video, dan audio langsung dari SDK. Meng-upload dan men-download juga dilakukan di background. Data yang disimpan akan aman dan hanya pengguna yang diijinkan yang dapat mengaksesnya.

6. Hosting

Firebase juga menyediakan fitur hosting. Firebase mengirimkan konten web secara cepat dan konten selalu dikirim dengan aman.

7. Crash reporting

Fitur crash reporting di Firebase membuat pengembang dapat mengetahui kesalahan ketika terjadi crash

2.5. Smartphone

Gadget adalah perangkat alat elektronik kecil yang memiliki banyak fungsi. Gadget (smartphone) memiliki banyak fungsi bagi penggunanya sehingga dinilai lebih memudahkan. Gadget (smartphone) atau dengan kata sederhana telephone gengam yang saat ini telah memiliki beragam fiture dan fungsi yang semakin kompleks guna memudahkan pemakainya merupakan trobosan baru dari telephone gengam sebelumnya. (Garini dalam Rohman, 2017).

2.6. Android

Pemakaian android di Indonesia semakin pesat, terlihat dari jumlah penggunaan android yang terus meningkat. Menurut data Badan Pusat Statistik (BPS) pemakaian android di Indonesia kini telah mencapai angka 171 juta pada tahun 2019. Jika disesuaikan dengan jumlah populasi penduduk Indonesia mencapai 264 juta jiwa, maka pengguna internet di Indonesia mengalami pengingkatan sebanyak 54,86% dari tahun 2017 yang lalu. Android adalah sistem operasi yang 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 macam aplikasi *mobile* (Firly, 2018).

Berikut ini adalah kelebihan dalam menggunakan Android, yaitu :

1. *User interface* yang mudah digunakan
2. Memiliki sifat *open source*
3. Memiliki tampilan yang mudah diubah.
4. Variasi harga produk yang beragam
5. Cepat dan *Responsive*
6. *User Friendly*

Berikut ini adalah kekurangan dalam menggunakan Android, yaitu :

1. Kerja sistemnya cukup berat
2. Hasil modifikasi sering menyebabkan sistem bekerja tidak stabil dan kurang optimal
3. Kurang responsive jika disandungkan dengan spesifikasi hardware yang tidak baik.

2.6.1. Versi Android

Android akan terus berusaha memperbarui sistem operasinya agar terus memuaskan kebutuhan pasar global. Kemajuan teknologi saat ini tentunya tidak lepas dari perkembangan teknologi yang semakin hari semakin terbaharu (Nadia Firly, 2019). Hal tersebut terlihat dari adanya versi demi versi yang terus diluncurkan oleh Android. Berikut adalah tabel yang menunjukkan berbagai versi Android yang telah dirilis :

Tabel 2.4. Versi Android

Versi	Nama	Tanggal Rilis
1.0 (API level 1)	-	23 September 2008
1.1 (API level 2)	-	9 Februari 2009
1.5 (API level 3)	<i>Cupcake</i>	27 April 2009
1.6 (API level 4)	<i>Donut</i>	15 September 2009
2.0 (API level 5)	<i>Eclair</i>	26 Oktober 2009
2.0.1 (API level 6)	<i>Eclair</i>	3 Desember 2009
2.1 (API level 7)	<i>Eclair</i>	12 Januari 2010
2.2-2.2.3 (API level 8)	<i>Froyo</i>	20 Mei 2010
2.3-2.3.2 (API level 9)	<i>Gingerbread</i>	6 Desember 2010
2.3-3.2.3.7 (API level 10)	<i>Gingerbread</i>	9 Februari 2011
3.0 (API level 11)	<i>Honeycomb</i>	22 Februari 2011
3.1 (API level 12)	<i>Honeycomb</i>	10 Mei 2011
3.2 (API level 13)	<i>Honeycomb</i>	15 Juli 2011
4.0-4.0.2 (API level 14)	<i>Ice Cream Sandwich</i>	19 Oktober 2011
4.0.3-4.0.4 (API level 15)	<i>Ice Cream Sandwich</i>	16 Desember 2011
4.1 (API level 16)	<i>Jelly Bean</i>	27 Juni 2012
4.2 (API level 17)	<i>Jelly Bean</i>	29 Oktober 2012
4.3 (API level 18)	<i>Jelly Bean</i>	24 Juli 2013
4.4 (API level 19)	<i>Kit-Kat</i>	31 Oktober 2013
5.0 (API level 21)	<i>Lollipop</i>	12 November 2014

6.0 (API level 22)	<i>MarshMallow</i>	5 Oktober 2015
7.0 (API level 23)	<i>Nougat</i>	9 Maret 2016
7.1 (API level 24)	<i>Nougat</i>	19 Oktober 2016
8.0 (API level 26)	<i>Oreo</i>	21 Maret 2017

2.7. Penelitian Terdahulu

Tabel 2.5. Berikut merupakan jurnal terkait dengan penelitian terdahulu :

Tabel 2.5. Penelitian Terdahulu

No	Nama	Judul	Terbit / Tahun	Keterangan
1	Elania Sukma Nabila	Konservasi dan Revitalisasi Cagar Budaya Pugung Raharjo Kabupaten Lampung Timur (Studi pada Objek Wisata Taman Purbakala Pugung Raharjo)	Fakultas ilmu sosial dan ilmu politik universitas Lampung 2019	<p>penelitian ini bertujuan untuk pengembangan dan pelestarian Taman Purbakala Pugung Raharjo, Kecamatan Sekampung Udk, Kabupaten Lampung Timur dengan menggunakan aspek-aspek pengembangan pariwisata.</p> <p>Keterbaruan : Cagar Budaya sudah dalam bentuk aplikasi Android</p>
2	Septilia Arfida1, Amnah, Hariyanto Wibowo	Pemetaan Lokasi Sekolah Dasar Negeri dengan Pencarian Jalur Terdekat pada Provinsi Lampung	Seminar Nasional Teknologi dan Bisnis 2018 IIB DARMAJAYA Bandar Lampung, 14 Agustus 2018	<p>Penelitian ini bertujuan untuk memberikan informasi terhadap pemetaan jalur terdekat lokasi Sekolah Dasar Negeri di Provinsi Lampung dengan guru bersertifikasi yang bisa dipantau serta diketahui oleh Dinas Pendidikan kota Bandar Lampung.</p> <p>Keterbaruan : Menggunakan algoritma Havarsine sebagai perbandingan metode Lain.</p>

3	M.Suriyanto, Basitha Febrinda ,Wahyu Dirgantara	Implementasi Haversine Formula pada Smart Garbage Can	Journal of Electrical Electronic Control and Automotiv e Engineering Vol.4, No.2, November 2019	Penelitian Ini Bertujuan untuk membuat Tempat Sampah Pintar dan aplikasi pencarinya dengan menggunakan metode Haversine Formula untuk menemukan posisi tempat sampah terdekat menggunakan fitur GPS dan location based tracker pada smartphone Android Keterbaruan : Ukuran jarak tiap Lokasi diketahui
---	---	---	---	--