



BAB II TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Widharma (2022:2), “Komputer adalah sebuah mesin hitung elektronik yang secara cepat menerima informasi masukan digital dan mengolah informasi tersebut menurut seperangkat instruksi yang tersimpan dalam komputer dan menghasilkan keluaran informasi setelah diolah”.

Tuhuteru (2021:19) mengatakan, “komputer merupakan suatu peralatan elektronik yang dapat menerima *input*, mengolah *input*, menyimpan, serta memberikan informasi menggunakan suatu program yang tersimpan di memori *computer* (standar program), dapat menyimpan program dan hasil pengolahan, serta bekerja secara otomatis”.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa komputer adalah suatu peralatan elektronik yang dapat mengolah data atau informasi secara digital melalui suatu program yang digunakan untuk mempermudah pekerjaan manusia.

2.1.2 Pengertian Perangkat Lunak (*Software*)

“Perangkat lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*). Sebuah program komputer tanpa terasosiasi dengan dokurnentasinya maka belum dapat disebut perangkat lunak (*software*)” (Rosa & Salahudin, 2019:2).

Utami dalam Sudarso (2022:5) mengatakan, “Perangkat lunak (*software*) adalah program yang berisi kumpulan instruksi untuk melakukan proses pengolah data. *Software* adalah penghubung antara manusia sebagai pengguna dengan perangkat keras komputer yang berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga perangkat keras komputer dapat



memahami keinginan pengguna dan menjalankan intruksi yang diberikan dan selanjutnya memberikan hasil yang diinginkan oleh manusia tersebut”.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa perangkat lunak adalah program komputer yang ditulis dengan bahasa pemrograman yang dapat melaksanakan tugas sesuai kehendak pengguna yang dijalankan oleh perangkat elektronik dan perangkat pintar lainnya.

2.1.3 Pengertian Basis Data (*Database*)

Pamungkas dan Putranto (2021:46), “*Database* adalah kumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. *Database* adalah representasi kumpulan fakta yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (redundansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Menurut Ali dalam Sudarso (2022:3) basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di luar komputer (*external memory*) dan digunakan perangkat lunak tertentu untuk memanipulasinya.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa basis data atau *database* adalah kumpulan data yang tersimpan secara terstruktur dalam satu tempat atau lokasi tertentu, yang dapat diakses dan dimanipulasi oleh pengguna atau program komputer.

2.1.4 Pengertian Titik Koordinat

Titik koordinat adalah sistem koordinat geografi yang digunakan untuk menunjukkan suatu titik di bumi berdasarkan garis lintang dan garis bujur (Samsudin *et al.*, 2019:24).

“Kordinat merupakan suatu sistem yang menunjukkan suatu titik di bumi berdasarkan garis lintang dan garis bujur dimana garis lintang menunjukkan garis



horizontal dan garis bujur menunjukkan garis *vertical*” (Duskarnaen & Nurfalaha, 2018:136).

Dari definisi diatas penulis menyimpulkan bahwa Titik Koordinat adalah titik pertemuan antara kedua garis yang berfungsi untuk menentukan suatu lokasi.

2.1.5 Pengertian *Latitude*

“*Latitude* adalah garis lintang yang mengarah dari khatulistiwa (0) ke kutub selatan, atau khatulistiwa ke kutub utara (sudut 0-90 dan 0-90)” (Samsudin *et al.*, 2019:24).

Awangga (2019:77) mengatakan bahwa *latitude* atau garis lintang adalah garis yang menentukan lokasi berada di sebelah utara atau selatan ekuator.

Dari beberapa definisi diatas penulis menyimpulkan bahwa *latitude* adalah garis lintang, garis yang dari atas ke bawah sejajar dengan bagian bumi (khatulistiwa)

2.1.6 Pengertian *Longitude*

“*Longitude* adalah garis yang menghubungkan antara sisi utara dan sisi selatan bumi (kutub)”. *Longitude* adalah garis bujur atau garis horizontal seperti garis khatulistiwa. Sudut O (*Greenwich*) ke arah Hawaii adalah 0-180, sedangkan kebalikannya dari O ke -180” (Samsudin *et al.*, 2019:24).

Awangga (2019:77), “*Longitude* adalah garis yang menggambarkan lokasi sebuah tempat di timur atau barat bumi dari sebuah garis utara-selatan yang disebut meridian utama. *Longitude* diberikan berdasarkan pengukuran sudut yang berkisar dari 0 derajat meridian utama ke +180 derajat arah timur dan -180 derajat arah barat”.

Dari beberapa definisi diatas penulis menyimpulkan bahwa *longitude* adalah garis mendatar yang menghubungkan antara sisi utara dan sisi selatan bumi.



2.2 Teori Judul

2.2.1 Pengertian Aplikasi

“Aplikasi adalah suatu perangkat lunak (*software*) atau program komputer yang beroperasi pada sistem tertentu yang diciptakan dan dikembangkan untuk melakukan perintah tertentu” (Pane *et al.*, 2020:53).

Wijaya dalam (Parina *et al.*, 2022), “Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut”.

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa aplikasi adalah program komputer yang dikembangkan untuk melakukan suatu fungsi atau perintah tertentu bagi pengguna nya melalui komponen atau *hardware* komputer.

2.2.2 Pengertian Pelaporan

“Pelaporan adalah proses menghasilkan informasi yang bermanfaat dan relevan bagi pihak yang membutuhkan. Proses pelaporan ini melibatkan pengumpulan data, pengolahan data, dan penyajian informasi dalam bentuk laporan atau dokumen tertentu. Laporan ini kemudian dapat digunakan sebagai dasar pengambilan keputusan atau evaluasi kinerja suatu organisasi atau perusahaan” (Aprilianti *et al.*, 2020:151).

“Pelaporan adalah kegiatan untuk menyampaikan data dan informasi yang dilakukan oleh unit pelaksana terkait dengan tujuan agar semua data dan informasi dapat disajikan dan disampaikan kepada pihak yang berkepentingan dengan akurat. Hal ini bertujuan untuk mendukung pengambilan keputusan dalam rangka pengelolaan data” (Amaliah *et al.*, 2019:120).

Berdasarkan beberapa definisi diatas, dapat disimpulkan bahwa pelaporan adalah proses penyampaian suatu informasi baik secara lisan maupun tulisan yang dapat dilakukan secara manual atau menggunakan aplikasi perangkat lunak.



2.2.3 Pengertian Sampah

Rohim (2023:1) mengatakan bahwa, “Sampah adalah bahan sisa yang tidak diinginkan dan tidak terpakai yang dihasilkan oleh aktivitas manusia, baik itu dalam bentuk padat, cair, gas, maupun bahan berbahaya. Sampah terdiri dari berbagai macam bahan seperti sisa makanan, kertas, plastik, logam, kaca, tekstil, bahan kimia, dan limbah medis. Sampah dapat berasal dari rumah tangga, institusi, industri, dan sektor publik lainnya. Sampah yang tidak dikelola dengan baik dapat menimbulkan masalah lingkungan dan kesehatan, seperti polusi udara, pencemaran air, penyebaran penyakit, dan sebagainya. Oleh karena itu, pengelolaan sampah yang baik sangat penting untuk menjaga kebersihan dan kesehatan lingkungan”.

Ps (2022:6), “Sampah adalah suatu bahan yang terbuang atau dibuang yang belum memiliki nilai ekonomis. Bentuk sampah bisa berada dalam setiap *fase* materi yaitu padat, cair, dan gas”.

Berdasarkan definisi diatas, dapat disimpulkan bahwa sampah adalah benda yang dihasilkan dari aktivitas manusia yang sudah tidak mempunyai nilai lagi dan umumnya akan dibuang karena dianggap sudah tidak berguna dan tidak dibutuhkan lagi.

2.2.4 Pengertian Metode Algoritma *Haversine*

Menurut Longitudestore dalam (Purnawan *et al.*, 2018) mengatakan bahwa *haversine* formula atau algoritma *haversine* ialah persamaan penting dalam sistem navigasi, dimana formula *haversine* ini akan menghasilkan jarak terpendek antara dua titik, misalnya pada bola yang diambil dari garis bujur (*longtitude*) 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*:

$$x = (\text{lon}2 - \text{lon}1) \times \cos\left(\frac{\text{lat}1 + \text{lat}2}{2}\right)$$

$$y = (\text{lat}2 - \text{lat}1)$$

$$d = \sqrt{((x \times x) + (y \times y))} \times R$$

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

2.2.5 Pengertian Aplikasi *Mobile*

Mubarak (2022:187), “Aplikasi *mobile* adalah sebuah aplikasi sistem perangkat lunak yang memungkinkan setiap pemakai melakukan mobilitas dengan perlengkapan asisten digital perusahaan pada telepon genggam atau seluler. Aplikasi *mobile* juga dikenal sebagai *web app*, *online app*, *iphone app* atau *smartphone app*”.

Pane (2022:11), “Aplikasi *mobile* atau aplikasi seluler adalah segmen teknologi informasi dan komunikasi global yang baru dan berkembang pesat. Aplikasi seluler mudah, ramah pengguna, murah, dapat diunduh, dan dapat dijalankan di sebagian besar ponsel termasuk ponsel murah dan *entry level*. Aplikasi seluler memiliki kegunaan yang luas seperti menelepon, mengirim pesan, menjelajah, mengobrol, komunikasi”.



Berdasarkan definisi diatas, dapat disimpulkan bahwa aplikasi *mobile* adalah program komputer yang dirancang khusus untuk dijalankan pada perangkat *mobile*, seperti *smartphone* atau tablet.

2.2.6 Pengertian Aplikasi Pelaporan Sampah Dengan Menggunakan Metode Algoritma *Haversine* di Kota Palembang Berbasis *Mobile*

Aplikasi pelaporan sampah dengan menggunakan metode *haversine* di kota Palembang berbasis *mobile* adalah aplikasi berbasis android yang dapat mempermudah masyarakat dalam melaporkan sampah agar dapat segera diangkut oleh petugas pengangkut sampah terdekat. Selain itu, aplikasi ini juga membantu petugas pengangkut sampah dan pegawai Dinas Lingkungan Hidup dan Kebersihan Kota Palembang dalam menentukan rute menuju lokasi dan memantau hasil laporan pengaduan sampah.

2.3 Teori Khusus

2.3.1 Pengertian UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen-komponen yang diperlukan dalam sistem *software* (Jauhari *et al.*, 2022:37).

UML (*Unified Modeling Language*) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi, penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML



paling banyak digunakan pada metodologi berorientasi objek (Rosa & Salahudin, 2019).

2.3.2 Pengertian *Use Case Diagram*

“Diagram *use case* merupakan salah satu tahapan dalam berbagai macam diagram yang dipergunakan dalam UML, diharapkan dengan mempergunakan diagram *use case*, pembangunan atau rancangan sistem akan menjadi lebih mudah sesuai fungsionalitas dari masing-masing fungsi pada sistem. Dalam Diagram *Use case* ada berbagai unsur atau komponen yang terlibat yaitu *actor*, *system* dan *use case* serta hubungan asosiasi, generalisasi serta dependensi di dalam diagram *use case*. Model kasus yang digunakan dapat dikembangkan dari sudut pandang pemangku kepentingan pada proyek sistem yang dibangun tidak hanya dari sudut pandangan pengembang (*Desainer*)” (Ahmad *et al.*, 2022:71).

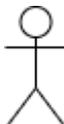
Pressman dalam Sa`ad (2020:46), “*Use Case Diagram* membantu menentukan fungsi dan fitur dari perangkat lunak. Dalam diagram ini, gambar yang menyerupai boneka kayu mewakili aktor yang berhubungan dengan kategori dari pengguna. Dalam diagram *Use Case*, para aktor terhubung oleh garis ke *Use Case* yang mereka kerjakan”.

Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

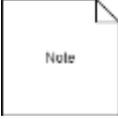
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2. 1 Simbol-Simbol *Use Case* Diagram

No	Gambar	Nama	Keterangan
1.		<i>Actor</i>	Spesifikasi himpunan peran pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2.		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i> akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri).
3.		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
4.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case target</i> memperluas perilaku dan <i>use case</i> sumber pada suatu titik yang diberikan.
6.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.

Lanjutan **Tabel 2. 1** Simbol-Simbol *Use Case Diagram*

9.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
----	---	-------------	--

Sumber: Sa`ad, 2020 (*Software Engineering : A Practitioner's Approach, 7th Edition*)

2.3.3 Pengertian *Activity Diagram* (Diagram Aktivitas)

Pressman dalam Sa`ad (2020:51), “Sebuah diagram *activity* menggambarkan perilaku dinamis dari sistem atau bagian dari sistem melalui aliran kontrol antara tindakan yang sistem lakukan. Hal ini mirip dengan sebuah *flowchart* bahwa sebuah diagram *activity* dapat menunjukkan arus bersamaan”.

Diagram *activity* juga banyak digunakan untuk mendefinisikan hal-hal berikut:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dan sistem/*user interface*, di mana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:



Tabel 2. 2 Simbol-Simbol Activity Diagram

No	Gambar	Nama	Keterangan
1.		<i>Activity</i>	Memperlihatkan bagaimana kelas antar muka saling berinteraksi satu sama lain.
2.		<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu.
3.		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4.		<i>Activity Final Node</i>	Bagaimana objek dibentuk atau diakhiri.
5.		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

Sumber: Sa`ad, 2020 (*Software Engineering : A Practitioner's Approach, 7 th Edition*)

2.3.4 Pengertian Class Diagram (Diagram Kelas)

Pressman dalam Sa`ad (2020:49), “Unsur-unsur utama dari diagram kelas adalah kotak, yang merupakan ikon yang digunakan untuk mewakili kelas dan *interface*. Setiap kotak dibagi menjadi bagian-bagian horizontal. Bagian atas berisi nama kelas. Bagian tengah berisi daftar atribut kelas dan bagian tengah bawah merupakan operation dari kelas tersebut”.

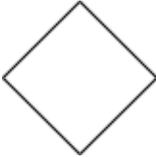
Setiap atribut bisa memiliki nama, tipe dan visibilitas yang berbeda-beda. Tipe mengikuti nama dan dipisahkan oleh titik dua, sedangkan visibilitas berada pada sebelah kiri nama -, #, ~, atau + masing-masing bernilai, *private*, *protected*, *package* atau *public*. Class Diagram juga dapat menunjukkan hubungan antar kelas-kelas. Sebuah kelas yang merupakan anak kelas dari kelas lain akan dihubungkan dengan garis lurus dengan tanda panah di ujungnya, sedangkan yang dalam UML disebut sebagai *generalization*. Garis putus-putus dengan tanda panah pada



ujungnya digunakan untuk menspesifikasikan hubungan antara sebuah *interface* dengan kelas yang megimplementasikan kelas tersebut dan yang dalam UML disebut sebagai *realization*. *Association* di antara dua kelas berarti ada hubungan struktural di antara kelas-kelas tersebut. *Association* diwakili oleh garis yang solid dan lurus.

Berikut adalah simbol-simbol yang ada pada *class diagram*:

Tabel 2. 3 Simbol-Simbol *Class Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Generalization</i>	Hubungan di mana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk.
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi- aksi yang ditampilkan sistem menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.

Lanjutan **Tabel 2. 3** Simbol-Simbol *Class Diagram*

6.		<i>Dependency</i>	Hubungan di mana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>), akan mempengaruhi elemen yang bergantung padanya (elemen yang tidak mandiri).
7.		<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.

Sumber: Sa`ad, 2020 (*Software Engineering : A Practitioner's Approach, 7th Edition*)

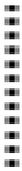
2.3.5 Pengertian *Sequence Diagram* (Diagram Sekuen)

Pressman dalam Sa`ad (2020:52), “*Sequence diagram* adalah suatu diagram yang menggambarkan interaksi antar objek dan mengindikasikan komunikasi di antara objek-objek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang diperutukan oleh objek-objek yang melakukan suatu tugas atau aksi tertentu. Objek-objek tersebut kemudian diurutkan dari kiri ke kanan, *actor* yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram”.

Sequence diagram berisi informasi yang sama dengan *collaboration diagram*, tetapi menekankan alur pada sekuensial sebuah pesan dari pada hubungan antar objek-objek, *sequence diagram* menggambarkan alur dari logika di dalam sistem secara visual sehingga memungkinkan kita untuk menyimpan dan memvalidasi logika kita. *Sequence diagram* juga digunakan secara umum untuk keperluan analisis dan desain. Diagram ini lebih memfokuskan diri dan eksekusi pada alur sistem dari pada bagaimana sistem itu dirakit. Diagram aktivitas menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi dengan kata lain *sequence diagram* menggambarkan alur dari logika di dalam sistem secara visual.

Berikut adalah simbol-simbol yang ada pada diagram sekuen: *object*, *actor*, *Lifeline*. *Activation*, *Message*.

Tabel 2. 4 Simbol-Simbol *Sequence Diagram*

No	Gambar	Nama	Keterangan
1.		<i>Object</i>	Menyatakan objek yang berinteraksi, pesan atau sebuah <i>instance</i> dari sebuah <i>class</i> dengan nama objek di dalamnya yang diawali titik koma.
2.		<i>Actor</i>	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang dibuat sendiri, jadi walaupun <i>actor</i> adalah gambar orang, tapi belum tentu merupakan orang.
3.		<i>Lifeline</i>	Menyatakan kehidupan suatu <i>object</i> atau menyatakan keberadaan sebuah <i>object</i> , <i>lifeline</i> biasanya garis putus vertikal yang di tarik dari sebuah objek.
4.		<i>Activation</i>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan, digambarkan segi empat yang disimpan di <i>lifeline</i> yang menggambarkan suatu objek yang akan melakukan aksi.
5.		<i>Message</i>	Digambarkan berbentuk anak panah, <i>message</i> menyatakan komunikasi antar objek.

Sumber: Sa''ad, 2020 (*Software Engineering : A Practitioner's Approach, 7th Edition*).



2.4 Teori Program

2.4.1 Android Studio



Gambar 2. 1 Logo Android Studio

Android studio adalah salah satu *software* untuk membuat aplikasi android. Android studio ini banyak digunakan oleh para *start up* atau *developer* android untuk membuat sebuah aplikasi berbasis android, saat ini sudah banyak kampus menggunakan mata kuliah pemrograman *mobile* yang dijadikan mata kuliah wajib bagi mahasiswa fakultas ilmu komputer. Android studio juga sangat memudahkan, karena banyak *developer* yang memakai *software* ini karena lebih mudah saat *coding*, *debug*, dan *share* pada GitHub (Purnomo *et al.*, 2020:6).

Herlinah & Musliadi (2019:4-5), “Android studio merupakan lingkungan pengembangan perangkat *Integrated Development Environment (IDE)* lunak terpadu untuk pengembangan aplikasi android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, android studio juga menawarkan banyak fitur untuk meningkatkan produktivitas saat membuat aplikasi android, misalnya:

1. *Build system* berbasis *gradle* yang fleksibel.
2. Emulator yang kaya fitur.
3. Mampu mengembangkan semua perangkat android.
4. *Instant run* untuk memberikan perubahan ke aplikasi yang sedang berjalan tanpa membuat APK baru.
5. *Template* kode dan integrasi GitHub yang membantu dalam membuat aplikasi dengan fitur biasa dan mengimpor sampel kode.
6. Alat pengujian dan *frameworks* yang ekstensif.



7. *Lint tools* untuk menganalisis kinerja, kegunaan, kompatibilitas versi dan masalah-masalah lain.
8. Dukungan C++ dan *Native Development Kit* (NDK).
9. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian *Google Cloud Messaging* dan *App Engine*.

Berdasarkan pengertian di atas, dapat disimpulkan bahwa pengertian Android studio adalah sebuah IDE (*Integrated Development Environment*) untuk pengembangan aplikasi android resmi yang didasarkan pada IntelliJ IDEA.

2.4.2 Visual Studio



Gambar 2. 2 Logo Visual Studio

Visual studio code (VS Code) ini adalah sebuah teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *linux*, *mac*, dan *windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *javascript*, *typescript*, dan *node.js*, serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang via *marketplace* visual studio code (seperti *php*, *c++*, *c#*, *pyhton*, *go*, *java*, dst) (Setyawan & Munari, 2020).

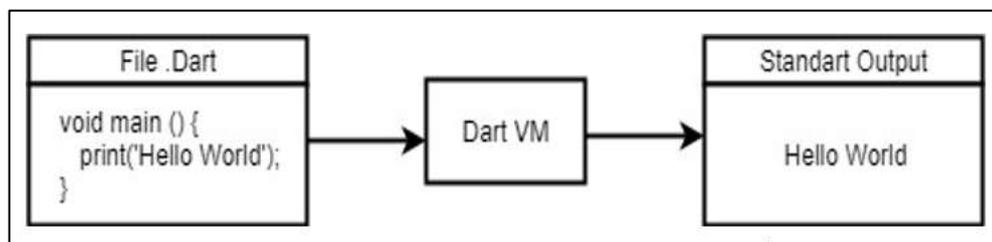
2.4.3 Dark



Gambar 2. 3 Logo Dart



“*Dart* adalah bahasa pemrograman yang dikembangkan oleh Google sejak tahun 2012 yang dipimpin oleh Lars Bak dan Kasper Lund, dan mulai tersedia untuk umum pada tahun 2013. Pada tahun 2018 bulan Agustus *Dart* 2.0 dirilis dengan banyak perubahan termasuk *sound-null-safety*. Bahasa *Dart* adalah dasar pemrograman untuk proyek *Flutter Framework*. Bahasa *Dart* menggunakan prinsip pemrograman berorientasi objek (OOP) yang berfokus untuk optimalisasi *client*. *Dart* termasuk bahasa tingkat tinggi dengan manajemen memori otomatis (*garbage-collection*) yang membuat aplikasi berjalan lebih cepat pada banyak *platform*. *Dart* menggunakan sintak kode bergaya bahasa C” (Wali *et al.*, 2023:178).



Gambar 2. 4 Proses Eksekusi Aplikasi *Dart*

2.4.4 Flutter



Gambar 2. 5 Logo *Flutter*

Dian dalam Krisnada & Tanone (2019:121), “*Flutter* adalah sebuah SDK atau *framework open source* yang dikembangkan oleh Google untuk membuat atau mengembangkan aplikasi yang dapat berjalan dalam sistem operasi Android dan iOS. *Flutter* menggunakan bahasa pemrograman *dart* dalam pengkodean. Perbedaan *framework flutter* dengan yang lainnya yaitu dalam *build* aplikasi, pada *framework* ini semua kodenya di *compile* dalam kode *native*



(Android NDK, LLVM, AOT-compiled) tanpa ada *interpreter* pada prosesnya sehingga proses *compile*-nya menjadi lebih cepat”.

2.4.5 Firebase Realtime Database



Gambar 2. 6 Logo *Firebase*

“*Firebase* adalah suatu layanan dari google untuk memberikan kemudahan bahkan mempermudah para *developer* aplikasi dalam mengembangkan aplikasinya. *Firebase* atau *Backend as a Service* (BaaS) merupakan solusi yang ditawarkan oleh google untuk mempercepat pekerjaan *developer*. Dengan menggunakan *firebase developer* bisa fokus dalam mengembangkan aplikasi tanpa memberikan *effort* yang besar untuk urusan *backend*. *Firebase* menyediakan *library* untuk berbagai *client platform* yang memungkinkan *integrasi* dengan android, iOS, JavaScript, Java, *Objective-C* dan *Node* aplikasi Js. *Firebase* digunakan untuk mempermudah dalam penambahan fitur-fitur yang akan dibangun oleh *developer*” (Fiorenza & Tolle, 2023).

2.4.6 JSON

“JSON adalah singkatan dari *JavaScript Object Notation*, yaitu format data yang digunakan untuk keperluan penyimpanan dan pertukaran data. Dalam bidang *web*, JSON biasanya digunakan untuk mengambil data dari *server* atau mengirim data ke *server*. Secara fungsinya JSON sama dengan XML, tapi JSON lebih cepat dan lebih mudah dibaca. Karena kelebihan ini, JSON lebih sering digunakan daripada XML sebagai pendahulunya” (Abdulloh, 2023:299).



2.4.7 Google Maps Api

“*Google Maps API (Application Programming Interface)* adalah aplikasi *interface* yang dapat diakses lewat javascript agar *google maps* dapat ditampilkan pada halaman web yang akan dibangun. Pengetahuan yang diperlukan untuk mengembangkan sebuah *google maps api* adalah tentang HTML, javascript, sedangkan peta sudah disediakan oleh *google*. *Google maps API* merupakan fasilitas dari *google* yang dapat digunakan untuk menambahkan peta ke dalam *website* dengan menggunakan JavaScript. *Google maps API* menyediakan banyak fasilitas dan utilitas untuk memanipulasi peta dan menambahkan konten ke peta melalui berbagai layanan” (Utomo & Hamdani, 2021).



2.5 Referensi Jurnal

Beberapa jurnal yang digunakan penulis sebagai referensi, antara lain :

Tabel 2. 5 Referensi Jurnal Terkait

No	Resume Jurnal	
1.	Judul	Aplikasi Penghitung Jarak Koordinat Berdasarkan <i>Latitude</i> dan <i>Longitude</i> dengan Metode <i>Euclidean Distance</i> dan Metode <i>Haversine</i>
	Volume, Nomor ISSN	Vol. 5, Nomor 2, ISSN : 2442-7942
	Tahun	2019
	Penulis	Canggih Ajika Pamungkas
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan dampak ketergantungan manusia terhadap teknologi informasi yaitu <i>google maps</i> yang mampu meningkatkan produktivitas kerja
	Metode Penelitian	Metode <i>Euclidean Distance</i> dan Metode <i>Haversine</i>
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan aplikasi yang <i>sederhana</i> dan mudah dipahami saat dijalankan - Kekurangan penelitian ini adalah kurang menampilkan desain perancangan UML secara detail mengenai alur atau gambaran aplikasi
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan sebuah aplikasi yang mampu menyajikan informasi mengenai kondisi geografis dengan <i>google map</i>	

Lanjutan **Tabel 2. 5 Referensi Jurnal** Terkait

No	Resume Jurnal	
2.	Judul	Penerapan <i>Haversine</i> Formula Dalam Pencarian Lokasi Bank Sampah di Kabupaten Bantul
	Volume, Nomor ISSN	Volume 19, Edisi 2, ISSN: 2686-0546 E-ISSN: 08521468
	Tahun	2022
	Penulis	Ika Arfiani, Probo Prasetyo, Muhammad Dzikrullah Suratin
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan belum adanya sebuah sistem manajemen yang mampu menentukan lokasi bank sampah terdekat dengan posisi pengguna
	Metode Penelitian	Metode <i>Haversine</i> Formula
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah memiliki tampilan menarik dan mudah dipahami dan proses yang cepat dalam pencarian rute untuk membantu pengguna mendapatkan rute terdekat menuju tempat bank sampah - Kekurangan penelitian ini adalah belum menerapkan <i>multipoint</i> terhadap data pengguna
	Hasil Penelitian	Hasil dari penelitian ini berupa aplikasi yang mampu membantu masyarakat dalam menemukan jarak terdekat lokasi bank sampah dengan lokasi penggunanya disertai dengan deskripsi dari bank sampah dan navigasi rute dengan <i>google maps</i> menuju lokasi bank sampah yang dituju, agar bank sampah ini dapat dimanfaatkan dengan baik oleh Masyarakat

Lanjutan **Tabel 2. 5** Referensi Jurnal Terkait

No	Resume Jurnal	
3.	Judul	Penerapan Metode <i>Haversine</i> Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana
	Volume, Nomor ISSN	Volume 4, Nomor 2, ISSN-P 2442-9562, ISSN-E 2580-1503
	Tahun	2020
	Penulis	Abadi Nugroho, Rio Jumardi , Nur Fajariah Ramadhania
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan belum ada media khusus dalam proses penyampaian informasi mengenai kejadian bencana dan informasi titik kumpul, dan lambatnya penanganan kebencanaan karena harus mengecek keakuratan informasi kejadian bencana terlebih dahulu.
	Metode Penelitian	Metode <i>Haversine</i> Formula
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan yang terdapat pada aplikasi sudah cukup lengkap dan menampilkan titik kumpul beserta jarak - Kekurangan penelitian ini adalah belum adanya fitur menambahkan foto untuk menambah detail pada lokasi dan juga fitur lain yang dapat memudahkan pengguna
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan sebuah aplikasi berbasis android berdasarkan jarak tempuh terdekat menggunakan metode <i>haversine</i> Formula ini diimplementasikan pada <i>smartphone</i> berbasis Android.	

Lanjutan **Tabel 2. 5** Referensi Jurnal Terkait

No	Resume Jurnal	
4.	Judul	Sistem Pencarian Lokasi dan Rute Terdekat Menggunakan Metode <i>Haversine</i> Formula pada Aplikasi Donatur Pakaian Berbasis Android
	Volume, Nomor ISSN	Vol. 7 No. 1
	Tahun	2021
	Penulis	Silvia Kartika, Suendri, dan Raissa Amanda Putri
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan proses penggalangan donasi di Kota Medan yang masih dilakukan secara manual, dengan membuat dan menyebarkan <i>flyer</i> atau brosur
	Metode Penelitian	Metode <i>Haversine</i> Formula
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan aplikasinya yang sudah cukup lengkap dan mudah dimengerti oleh pengguna - Kekurangan penelitian ini adalah keterbatasan spesifikasi android dalam mengakses aplikasi ini yaitu hanya dapat digunakan android versi 4 ke atas
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan sebuah aplikasi yang dapat menampilkan lokasi <i>user</i> saat ini dan dengan mengimplementasikan metode <i>haversine formula</i> pada aplikasi sehingga dapat menampilkan lokasi donatur terdekat yang sedang berdonasi dalam jangkauan jarak ≤ 5 Km dari posisi penggalang donasi	

Lanjutan **Tabel 2. 5** Referensi Jurnal Terkait

No	Resume Jurnal	
5.	Judul	Aplikasi Pencarian Pariwisata dan Tempat Oleh-Oleh Terdekat Menggunakan <i>Metode Haversine</i> Berbasis Android
	Volume, Nomor ISSN	Vol 3, No 2, ISSN 2503-1945
	Tahun	2018
	Penulis	Sarif Ifan Purnawan, Fitri Marisa, Indra Dharma Wijaya
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan banyaknya masyarakat yang belum mengetahui informasi setiap lokasi mengenai pariwisata dan tempat oleh-oleh di Malang
	Metode Penelitian	Metode <i>Haversine</i> Formula
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan dari aplikasi yang sangat menarik dan mudah digunakan oleh pengguna - Kekurangan penelitian ini adalah fitur yang disajikan pada aplikasi belum lengkap dan perlu adanya penambahan fitur lainnya
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan aplikasi berbasis android untuk pencarian pariwisata dan tempat oleh-oleh berdasarkan jarak tempuh terdekat menggunakan metode <i>Haversine Formula</i> guna menghemat waktu, tenaga dan biaya	

Lanjutan **Tabel 2. 5** Referensi Jurnal Terkait

No	Resume Jurnal	
6.	Judul	Aplikasi Android untuk Pencarian Rute Terdekat Mini Market Menggunakan Metode <i>Floyd-Warshall</i> Berbasis Gis (Studi Kasus di Kecamatan Taman)
	Volume, Nomor ISSN	Volume 2, Nomor 1, ISSN 2622-7983 <i>online</i> , ISSN 2622-7746 <i>print</i>
	Tahun	2019
	Penulis	Mochamad Afandi
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan banyaknya mini market yang didirikan, namun masih banyak orang yang kesulitan untuk mencari mini market terdekat karena tidak semua orang mengetahui jalan tersebut
	Metode Penelitian	Metode <i>Floyd-Warshall</i>
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah mempunyai banyak alternatif untuk menuju lokasi, dapat memanfaatkan aplikasi untuk mencari rute lain di saat rute tersebut macet, dan mengetahui mini market terdekat - Kekurangan penelitian ini adalah proses untuk menampilkan pencarian setiap rute relatif lebih lama
	Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan sebuah aplikasi yang menampilkan jalur terpendek untuk menentukan rute terpendek dari lokasi pengguna menuju mini market dengan Metode <i>Floyd Warshall</i> sehingga membantu pengguna untuk merealisasikan kebutuhan informasi mini market

Lanjutan **Tabel 2. 5 Referensi Jurnal** Terkait

No	Resume Jurnal	
7.	Judul	Aplikasi Pencarian Jalur Terdekat Panggilan Darurat Supir Ambulans Kota Medan Menggunakan Algoritma Boruvka Berbasis Android
	Volume, Nomor ISSN	Volume 8, No 1, ISSN 2301-9425 (Media Cetak)
	Tahun	2020
	Penulis	Veronica Suci
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan sering terjadi permasalahan ambulans yang datang terlambat saat ada panggilan darurat sehingga memberikan efek kerugian yang ditimbulkan, antara lain menimbulkan korban jiwa
	Metode Penelitian	Metode Algoritma Boruvka
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan program aplikasi yang sudah cukup lengkap sehingga memudahkan supir ambulans dalam mengoperasikan aplikasi tersebut serta banyaknya alternatif jalan yang disediakan - Kekurangan penelitian ini adalah tampilan aplikasinya yang kurang menarik dan terkesan seperti putih polos saja di beberapa bagian tampilan halaman
	Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan sebuah perancangan aplikasi pencarian jalur terdekat supir ambulans berhasil dengan menerapkan algoritma Boruvka untuk mencari jalur terdekat dengan mendapatkan jarak terdekat serta waktu yang dibutuhkan

Lanjutan **Tabel 2. 5 Referensi Jurnal** Terkait

No	Resume Jurnal	
8.	Judul	Aplikasi Berbasis Android untuk Mencari Lokasi Puskesmas Terdekat dengan Algoritma <i>A-Star</i> di Provinsi Dki Jakarta
	Volume, Nomor ISSN	Volume 9, Nomor 1, p-ISSN 2089-0265, e-ISSN 2598-3016
	Tahun	2018
	Penulis	Vicky Budiman, Yogi Swara Hendro Leksmo, Halim Agung
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan keterbatasan informasi mengenai puskesmas yang menjadi kendala masyarakat, seperti ketidaktahuan lokasi puskesmas terdekat dan jalan yang harus ditempuh menuju puskesmas
	Metode Penelitian	Metode Algoritma <i>A-Star</i>
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah tampilan aplikasi yang sangat menarik dan mudah dioperasikan oleh pengguna - Kekurangan penelitian ini adalah belum dilengkapi dengan sistem antrian <i>online</i> pada puskesmas sehingga fitur yang dibutuhkan oleh pasien puskesmas belum lengkap sepenuhnya
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu bahwa dari 50 kali pengujian mencari jarak terpendek dari lokasi pengguna menuju puskesmas yang diinginkan menunjukkan 44 kali algoritma <i>A-Star</i> berhasil menghasilkan jarak terpendek atau sama dengan <i>Google Maps</i>	



Lanjutan Tabel 2. 5 Referensi Jurnal Terkait

No	Resume Jurnal	
9.	Judul	Aplikasi Pencari Tambal Ban Area Magelang Berbasis Android dengan Menggunakan Metode <i>Haversine</i>
	Volume, Nomor ISSN	Vol. 2 No. 1, <i>Online</i> ISSN 2580-734X
	Tahun	2018
	Penulis	Helmy Dewantara, Mukhtar Hanafi, Setiya Nugraha
	Latar Belakang	Jurnal ini memiliki latar belakang berdasarkan banyaknya pengguna kendaraan bermotor di Indonesia, serta tingkat keseringan pengendara yang mengalami ban bocor di lokasi yang belum dikenal, sehingga kesulitan untuk menemukan bengkel tambal ban terdekat
	Metode Penelitian	Metode <i>Haversine</i>
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini perhitungan jarak dengan menggunakan metode <i>haversine</i> untuk menentukan lokasi bengkel tambal ban terdekat memiliki tingkat akurasi perhitungan jarak sebesar 90% - Kekurangan penelitian ini adalah hanya menampilkan bengkel yang memiliki radius maksimal 2 km saja
Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan informasi data tambal ban yang dinamis sesuai dengan perkembangan bengkel dan <i>output</i> yang dihasilkan lebih efisien dengan radius pencarian disesuaikan dengan jarak tempuh maksimal 2 km	

Lanjutan **Tabel 2. 5 Referensi Jurnal** Terkait

No	Resume Jurnal	
10.	Judul	Sistem Pemesanan Makanan Tradisional Berbasis Android Menggunakan Metode <i>Haversine</i> Formula
	Volume, Nomor ISSN	Vol. 6, No.1, p-ISSN: 2406-7768, e-ISSN : 2581-2181
	Tahun	2019
	Penulis	Dzakaul Malik , Vidila Rosalina
	Latar Belakang	Jurnal ini memiliki latar belakang untuk membantu penjual makanan tradisional di Kabupaten Pandeglang mempromosikan makanan tradisional yang mereka jual serta meningkatkan retensi pelanggannya dengan memberi kemudahan para pembeli untuk mendapatkan informasi dan memesan makanan tradisional di Kabupaten Pandeglang
	Metode Penelitian	Metode <i>Haversine Formula</i>
	Kelebihan dan Kekurangan	<ul style="list-style-type: none"> - Kelebihan penelitian ini adalah mampu menghitung jarak pemesanan beserta tibanya makanan ketika pemesanan sedang berlangsung - Kekurangan penelitian ini adalah tampilan dari aplikasi yang kurang menarik dan masih perlu dikembangkan lagi agar menjadi <i>user friendly</i>
	Hasil Penelitian	Hasil dari pengujian yang telah dilakukan yaitu menghasilkan aplikasi yang dapat mempermudah calon pembeli dengan memberikan informasi penjual makanan tradisional di Kabupaten Pandeglang