

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Untuk mendukung keberhasilan penelitian ini, penyusun melakukan pendekatan teoritis melalui beberapa literatur yang berhubungan dengan penelitian yang dilakukan. Beberapa uraian penelitian terdahulu yang menjadi acuan dalam penelitian ini yaitu:

1. Berdasarkan penelitian Septya Maharani, Awang Harsa K dan Atik Tia Nalarwati (2017) dengan judul “Sistem Informasi Geografi (SIG) Pencarian ATM Bank Kaltim Terdekat dengan Geolocation dan Haversine Formula Berbasis Web” SIG akan sangat tepat digunakan sebagai sarana untuk mengolah informasi lokasi ATM Bank Kaltim di Kota Samarinda. Salah satu metodenya yaitu Haversine Formula, yang merupakan metode yang diterapkan di dalam aplikasi SIG dalam menentukan lokasi ATM yang terdekat dengan user. Haversine Formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan garis bujur dan lintang. SIG yang dibuat dapat memberikan informasi mengenai lokasi ATM terdekat dan petunjuk arahnya. Hasil dari penelitian ini, Haversine Formula untuk proses perhitungan menentukan lokasi ATM terdekat dapat diterapkan dan memberikan informasi ATM untuk pengguna atau nasabah Bank Kaltim.

2. Berdasarkan penelitian Muhammad Julda dan Elida Tuti Siregar (2020) dengan judul “Sistem Informasi Geografi Menghitung Jarak Rute Angkot di Kota Medan Menggunakan Metode Haversine Formula” SIG akan sangat tepat digunakan sebagai sarana untuk mengolah informasi rute angkot di Kota Medan. Salah satu metodenya yaitu Haversine Formula, yang merupakan metode yang diterapkan di dalam aplikasi SIG dalam menghitung jarak dari lokasi awal ke lokasi tujuan. Haversine Formula adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan garis bujur dan lintang. SIG yang dibuat dapat memberikan informasi rute angkot di kota medan.
3. Berdasarkan Penelitian Irwan dan Dedy Atmajaya (2017) yang berjudul “Sistem Informasi Pencarian Lokasi Perguruan Tinggi di Makassar Dengan Metode Haversine Formula” Metode yang digunakan dalam penelitian ini adalah metode Haversine Formula dan metode Euclidean. Dari hasil penelitian menggunakan metode Haversine Formula, hasil pengukuran jarak lintasan terdekat ke perguruan tinggi lebih kecil/dekat dibandingkan dengan menggunakan metode Euclidean. Hal ini dikarenakan metode Haversine Formula mengabaikan ketinggian bukit dan kedalaman lembah di permukaan bumi. Tujuan dari 2 penelitian ini adalah untuk mempermudah dalam menginformasikan kepada masyarakat umum yang sedang membutuhkan informasi perguruan tinggi di Makassar.
4. Berdasarkan Penelitian Lubis, Jahria dan Jannati (2016) yang berjudul “Sistem Informasi Geografis Jasa Ekspedisi Angkutan Darat Di Kota Medan Dengan

Menggunakan Metode Haversine Formula” Adapun manfaat yang diperoleh dari penelitian ini adalah Sistem Informasi Geografis Letak Lokasi Jasa ekspedisi angkutan darat akan meningkatkan keuntungan untuk perusahaan Jasa ekspedisi angkutan darat, kemudahan yang dirasakan konsumen dalam mengakses informasi lokasi Jasa ekspedisi angkutan darat dapat digunakan sebagai upaya pengenalan teknologi berbasis sistem informasi geografis kepada konsumen dan melakukan perkembangan metode haversine formula dapat dijadikan sebagai referensi untuk melakukan perhitungan jarak.

5. Berdasarkan Penelitian Indra Permana, Deni Ahmad dan Zakaria (2018) yang berjudul “Aplikasi Dan Panduan Angkutan Kota Tasik Malaya Berbasis Android” Adapun manfaat yang diperoleh dari penelitian ini adalah Sistem Informasi Geografis Letak Lokasi Jasa ekspedisi angkutan darat akan meningkatkan keuntungan untuk perusahaan Jasa ekspedisi angkutan darat, kemudahan yang dirasakan konsumen dalam mengakses informasi lokasi Jasa ekspedisi angkutan darat dapat digunakan sebagai upaya pengenalan teknologi berbasis sistem informasi geografis kepada konsumen dan melakukan perkembangan metode haversine formula yang dapat dijadikan sebagai referensi untuk melakukan perhitungan jarak dari suatu tempat ke tempat lainnya.

Sedangkan hasil penelitian yang dilakukan oleh penulis, dibangun sebuah sistem informasi geografis berbasis Android yang memudahkan pengguna dalam mencari informasi mengenai jarak dan rute yang dilalui angkutan umum PT Mitra Swadaya Mandiri di Kota Medan.

II.2. Landasan Teori

II.2.1. Sistem

Sistem berasal dari bahasa latin (*systēma*) dan bahasa Yunani (*sustēma*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat.

Sistem juga merupakan kesatuan bagian-bagian yang saling berhubungan yang berada dalam suatu wilayah serta memiliki item-item penggerak, contoh umum misalnya seperti negara. Negara merupakan suatu kumpulan dari beberapa elemen kesatuan lain seperti provinsi yang saling berhubungan sehingga membentuk suatu negara di mana yang berperan sebagai penggerakya yaitu rakyat yang berada dinegara tersebut.

Kata "sistem" banyak sekali digunakan dalam percakapan sehari-hari, dalam forum diskusi maupun dokumen ilmiah. Kata ini digunakan untuk banyak hal, dan pada banyak bidang pula, sehingga maknanya menjadi beragam. Dalam pengertian yang paling umum, sebuah sistem adalah sekumpulan benda yang memiliki hubungan di antara mereka (Monica Kumaat, dkk ; 2016 : 81).

II.2.2. Informasi

Informasi adalah sekumpulan fakta-fakta yang telah diolah menjadi bentuk data, sehingga dapat menjadi lebih berguna dan dapat digunakan oleh siapa saja

yang membutuhkan data-data tersebut sebagai pengetahuan ataupun dapat digunakan dalam pengambilan keputusan.

Informasi bisa dikatakan sebagai pengetahuan yang didapatkan dari belajar, pengalaman atau instruksi. Namun, istilah ini masih memiliki banyak arti tergantung pada konteksnya. Dalam beberapa pengetahuan tentang suatu peristiwa tertentu yang telah dikumpulkan ataupun dari sebuah berita dapat juga dikatakan sebagai informasi. Lain halnya dalam ilmu komputer, informasi adalah data yang disimpan, diproses atau ditransmisikan (Monica Kumaat, dkk ; 2016 : 81).

II.2.3. Sistem Informasi

Sistem Informasi adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algorithmic, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis (Monica Kumaat, Steven R. Sentinuwo, Agustinus Jacobus; 2016: 81).

II.2.4. Konsep Dasar Sistem Informasi

Sistem informasi memberikan nilai tambah terhadap proses, produksi, kualitas, manajemen, pengambilan keputusan, dan pemecahan masalah serta

keunggulan kompetitif yang tentu saja sangat berguna bagi kegiatan bisnis.

Kegiatan yang terdapat pada sistem informasi antara lain:

- a. Input, menggambarkan suatu kegiatan untuk menyediakan data yang akan diproses.
- b. Proses, menggambarkan bagaimana suatu data diproses untuk menghasilkan suatu informasi yang bernilai tambah.
- c. Output, suatu kegiatan untuk menghasilkan laporan dari proses diatas.
- d. Penyimpanan, suatu kegiatan untuk memelihara dan menyimpan data.
- e. Kontrol, suatu aktifitas untuk menjamin bahwa sistem informasi tersebut berjalan sesuai dengan yang diharapkan (Rice Novita, dkk; 2015: 2).

II.3. Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau Geographic Information System (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis. Akronim GIS terkadang dipakai sebagai istilah untuk geographical information science atau geospatial information studies yang merupakan ilmu studi atau pekerjaan yang berhubungan dengan Geographic Information System. Dalam artian sederhana sistem informasi geografis dapat kita simpulkan sebagai gabungan kartografi, analisis statistik dan teknologi sistem basis data (database) (Achmad Maulana, dkk; 2018: 4)

Konsep SIG telah diperkenalkan di Indonesia sejak pertengahan tahun 1980-an, dan kini telah dimanfaatkan di berbagai bidang baik negeri maupun

swasta. Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti *query*, menganalisisnya, dan menyimpan serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain. Komponen SIG terdiri atas *hardware*, *software*, data, dan *user*. Dengan adanya SIG diharapkan tersedia informasi yang cepat, benar dan akurat tentang keadaan di lingkungannya.

II.3.1. Komponen Sistem Informasi Geografis

Sistem Informasi Geografis terdiri dari empat komponen, yaitu:

1. Perangkat keras (Hardware)

Pada saat ini SIG tersedia untuk berbagai platform perangkat keras mulai dari PC desktop, workstations, hingga multiuser host yang dapat digunakan oleh banyak orang secara bersamaan dalam jaringan komputer yang luas, berkemampuan tinggi, memiliki ruang penyimpanan (hard disk) yang besar, dan mempunyai kapasitas memori (RAM) yang besar. Walaupun demikian, fungsionalitas SIG tidak terikat secara ketat terhadap karakteristik-karakteristik fisik perangkat keras ini sehingga keterbatasan memori pada PC30 pun dapat diatasi. Adapun perangkat keras yang sering digunakan untuk SIG adalah komputer (PC), mouse, digitizer, printer, plotter, dan scanner.

2. Perangkat lunak (Software)

Bila dipandang dari sisi lain, SIG juga merupakan sistem perangkat lunak yang tersusun secara modular dimana basis data memegang peranan kunci. Setiap subsistem diimplementasikan dengan menggunakan perangkat lunak yang

terdiri dari beberapa modul, hingga tidak mengherankan jika ada perangkat SIG yang terdiri dari ratusan modul program yang masing-masing dapat dieksekusi sendiri.

3. Data dan Informasi Geografi SIG

Dapat mengumpulkan dan menyimpan data dan informasi yang diperlukan baik secara tidak langsung dengan cara mengimpor-nya dari perangkat-perangkat lunak SIG yang lain maupun secara langsung dengan cara mendigitasi data spasialnya dari peta dan memasukkan data atributnya dari tabel-tabel dan laporan dengan menggunakan keyboard.

4. Manajemen Suatu proyek SIG

Akan berhasil jika di manage dengan baik dan dikerjakan oleh orang-orang memiliki keahlian yang tepat pada semua tingkatan. (Koko Mukti Wibowo, dkk; 2015: 11)

II.3.2. Sub Sistem Informasi Geografis

Sistem informasi geografis dapat diuraikan menjadi beberapa subsistem sebagai berikut (Eko Budi Setiawan,2016:2).

1. Data Masukan (Input Data)

Subsistem ini bertugas untuk mengumpulkan dan mempersiapkan data spasial dan atribut dari berbagai sumber. Subsistem ini bertanggungjawab dalam mengkonversi atau mentransformasikan format-format data aslinya ke dalam format yang dapat digunakan dalam SIG.

2. Data Keluaran (Output Data)

Subsistem ini menampilkan atau menghasilkan keluaran seluruh atau sebagian basis data baik dalam bentuk softcopy maupun hardcopy seperti tabel, grafik, peta dan lain-lain.

3. Data Manajemen

Subsistem ini mengorganisasikan data spasial maupun atribut ke dalam sebuah basis data sedemikian rupa sehingga mudah dipanggil, diupgrade, dan di-edit.

4. Manipulasi Data Analisis Data

Subsistem ini menentukan informasi-informasi yang dapat dihasilkan oleh sistem informasi geografis. Selain itu, subsistem ini juga melakukan manipulasi dan Pemodelan data untuk menghasilkan informasi yang diharapkan.

II.3.3. Kemampuan Sistem Informasi Geografis

Sistem informasi geografis mempunyai kemampuan untuk menghubungkan berbagai data pada suatu titik tertentu di bumi, menggabungkan, menganalisis dan akhirnya memetakan hasilnya (Eddy Prahasta, 2014 :115).

Adapun kemampuan sistem informasi geografis sebagai berikut:

1. Memasukkan dan mengumpulkan data geografis (spasial dan atribut).
2. Memeriksa, meng-update (mengedit) data geografis.
3. Menyimpan atau memanggil kembali data geografis.
4. Mempresentasikan atau menampilkan data geografis.
5. Mengelola, memanipulasi dan menganalisis data geografis.

6. Menghasilkan output data geografis dalam bentuk peta tematik (view dan layout), tabel, grafik (chart) laporan, dan lainnya baik dalam bentuk hardcopy maupun softcopy.

II.3.4. Definisi Sistem Informasi Geografis

Beberapa definisi sistem informasi geografis yang telah beredar diberbagai sumber pustaka, diantaranya yaitu (Eddy Prahasta, 2014:100):

1. SIG adalah sistem komputer untuk memasukkan, menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisis, dan menampilkan data yang berhubungan dengan posisinya dipermukaan bumi.
2. SIG adalah kombinasi perangkat keras & perangkat lunak sistem komputer yang memungkinkan penggunaanya untuk mengelola, menganalisa, dan memetakan informasi spasial berikut atributnya dengan akurasi kartografi.
3. SIG adalah sistem yang terdiri dari perangkat keras, perangkat lunak, data manusia, organisasi & lembaga yang digunakan untuk mengumpulkan, menyimpan, menganalisis, dan menyebarkan informasi - informasi mengenai daerah - daerah di permukaan bumi.

II.4. Metode Haversine Formula

Metode Haversine digunakan untuk menghitung jarak antara titik di permukaan bumi menggunakan garis lintang (longitude) dan garis bujur (latitude) sebagai variabel inputan. Haversine formula adalah persamaan penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola

(bumi) berdasarkan bujur dan lintang.

Dengan mengasumsikan bahwa bumi berbentuk bulat sempurna dengan jari-jari R 6.371 km, dan lokasi dari 2 titik di koordinat bola (lintang dan bujur) masing-masing adalah long1, lat1, dan long2, lat2, maka rumus Haversine dapat ditulis dengan persamaan sebagai berikut:

$$d = 2r \arcsin^1 \left(\sqrt{\sin^2 \left(\frac{\varphi 1 - \varphi 2}{2} \right) + \cos(\varphi 1) \cos(\varphi 2) \sin^2 \left(\frac{\lambda 1 - \lambda 2}{2} \right)} \right)$$

(Sumber: I Gede Oka Gartria Atitama; 2017)

Keterangan:

Ket: Δ =delta ini seperti konstanta dari variable

d = jarak

r = radius

$\varphi 1$ = latitude awal

$\varphi 2$ = latitude akhir

$\lambda 2$ =longitude awal

$\lambda 1$ =longitude akhir

Dimana, d merupakan jarak antara dua titik (sepanjang lingkaran besar dari bola, melihat jarak bola), r merupakan jari - jari bola, $\varphi 1$ merupakan nilai latitude asal, $\varphi 2$ merupakan nilai latitude tujuan, $\lambda 1$ merupakan nilai longitudinal asal, dan $\lambda 2$ adalah nilai longitudinal tujuan.

Menghitung Jarak Antara Dua Titik Koordinat Formula Haversine dipublikasikan oleh R.W. Sinnott pada tahun 1984. Haversine merupakan fungsi

trigonometri yang digunakan untuk melakukan perhitungan jarak antara dua titik.

Berikut ini rumus Haversine: $R = 6,371\text{km}$ $\Delta\text{lat} = \text{lat}_2 - \text{lat}_1$ $\Delta\text{long} = \text{long}_2 - \text{long}_1$

$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat}_1) * \cos(\text{lat}_2) * \sin^2(\Delta\text{long}/2)$ $c = 2 * \text{atan2}(\sqrt{a}, \sqrt{1-a})$ $d =$

$R * c$

II.5. Peta

Peta merupakan gambaran wilayah geografis, bagian permukaan bumi yang disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Peta dapat digambarkan dengan berbagai gaya, masing-masing menunjukkan permukaan yang berbeda untuk subjek yang sama untuk memvisualisasikan dunia dengan mudah, informatif dan fungsional. Peta berbasis komputer (digital) lebih serba guna dan dinamis karena bisa menunjukkan banyak *view* yang berbeda dengan subjek yang sama. Peta ini juga memungkinkan perubahan skala, animasi gabungan, gambar, suara, dan bisa terhubung ke sumber informasi tambahan melalui internet. Peta digital dapat diupdate ke peta tematik baru dan bisa menambahkan detail informasi geografi lainnya (Monica kumaat, dkk; 2016: 82).

II.6. Assisted – Global Positioning System (A-GPS)

Assisted Global Positioning System (A-GPS) merupakan pengembangan dari sistem GPS biasa sebagai penentu posisi. A-GPS memperbaiki performansi GPS biasa dengan menyediakan informasi lewat kanal komunikasi alternatif yang terhubung ke suatu server pembantu, dimana A-GPS receiver akan mendapatkan

beberapa informasi yang biasanya diterima dari satelit melalui server tersebut. Inilah yang membedakan antara A-GPS dengan GPS, yaitu penambahan elemen assistance server atau juga disebut LBS (*location Base Service*). Dengan demikian, membantu proses penerimaan data untuk menentukan posisi user menjadi benar-benar lebih mudah dan dapat mengurangi waktu serta jumlah informasi yang dibutuhkan dari satelit. Lebih lanjut, karena A-GPS receiver didesain untuk bisa mengetahui terlebih dalam penentuan posisinya, maka A-GPS receiver dapat meningkatkan jumlah energy yang diterima dari frekuensi-frekuensi tertentu saja. Hal ini meningkatkan sensitivitas dari A-GPS *receiver* dan memungkinkan untuk melakukan penentuan posisi *user* walaupun memperoleh sinyal GPS pada kekuatan sinyal yang lebih rendah.



Gambar II.1. Contoh (*Assisted – Global Positioning System*) A-GPS

(Sumber: Andi Juansyah ; 2015 : 2)

II.7. Pengenalan Sistem Operasi Android

Android merupakan sistem operasi yang sekarang sedang terkenal di pasaran *smartphone* saat ini. Berikut adalah pengenalan android:

II.7.1. Sejarah Android

Android adalah sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel atau *smartphone*. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia (Andi Juansyah: 2015: 2).

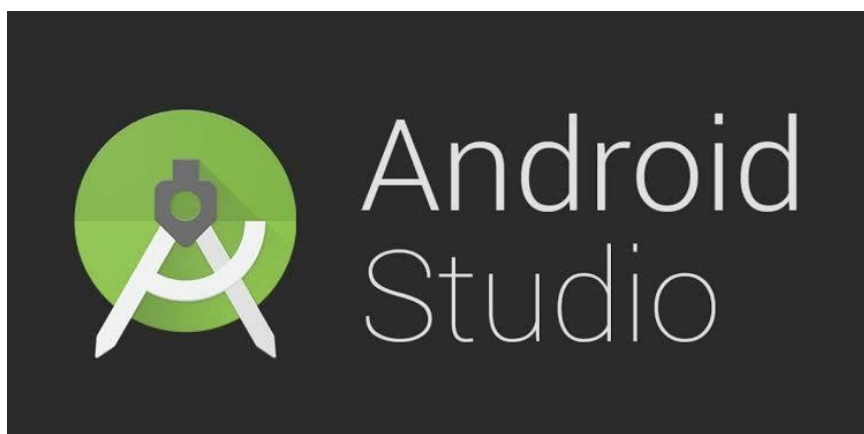
Pada saat perilis perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan open source pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

II.8. Perangkat Lunak Pendukung

Berikut adalah perangkat lunak pendukung dalam penunjang pembangunan aplikasi yang akan di bangun.

II.8.1. Android Studio

Android studio adalah IDE (Integrated Development Environment) resmi untuk pengembangan aplikasi Android dan bersifat open source atau gratis. Peluncuran Android Studio ini diumumkan oleh Google pada 16 mei 2013 pada event Google I/O Conference untuk tahun 2013. Sejak saat itu, Android Studio menggantikan Eclipse sebagai IDE resmi untuk mengembangkan aplikasi Android (Andi Juansyah: 2015: 2)



Gambar II.2. Android Studio
(Sumber: Andi Juansyah ; 2015: 3)

Android studio sendiri dikembangkan berdasarkan IntelliJ IDEA yang mirip dengan Eclipse disertai dengan ADT *plugin* (Android Development *Tools*). Android studio memiliki fitur:

- a. Projek berbasis pada Gradle Build
- b. *Refactory* dan pembenahan *bug* yang cepat
- c. *Tools* baru yang bernama “Lint” dikalim dapat memonitor kecepatan, kegunaan, serta kompatibilitas aplikasi dengan cepat.
- d. Mendukung *Proguard* And *App-signing* untuk keamanan.
- e. Memiliki GUI aplikasi android lebih mudah

- f. Didukung oleh Google *Cloud Platfrom* untuk setiap aplikasi yang dikembangkan.

II.8.2. Java Development Kit (JDK)

Java Development Kit (JDK) adalah sekumpulan perangkat lunak yang dapat kamu gunakan untuk mengembangkan perangkat lunak yang berbasis Java, sedangkan JRE adalah sebuah implementasi dari Java *Virtual Machine* yang benar-benar digunakan untuk menjalankan program java. Biasanya, setiap JDK berisi satu atau lebih JRE dan berbagai alat pengembangan lain seperti sumber compiler java, bundling, debuggers, development libraries dan lain sebagainya (Andi Juansyah: 2015: 3)

II.8.3. Web Service

Web service adalah sekumpulan application logic beserta objek-objek dan metode-metode yang dimilikinya, yang terletak di suatu server yang terhubung ke internet sehingga dapat diakses menggunakan protocol HTTP dan SOAP (Simple Object Access Proctocl). Dalam penggunaannya, web service dapat digunakan hanya untuk memeriksa data *user* yang *login* ke sebuah *website* ataupun untuk digunakan pada transaksi perbankan *online* yang rumit. Tujuan dari teknologi ini adalah untuk memudahkan beberapa aplikasi komponennya untuk saling terhubung dengan aplikasi lain dalam sebuah organisasi maupun dilaur organisasi menggunakan standar yang tidak terikat platform (*platform-neutral*)

dan tidak terikat akan bahasa pemrograman yang digunakan. Hal tersebut dapat terjadi karena XML standar yang didukung oleh banyak perusahaan besar didunia, yang digunakan untuk bertukar data. Selain daripada itu, penggunaan SOAP menjadi metode dari objek-objek yang ada dalam sebuah web service dapat di akses dari aplikasi lain seperti halnya aplikasi tersebut mengakses metode local (Andi Juansyah: 2015: 3)

II.8.4. Java Script Object Nonation (JSON)

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data [JSON]. JSON terbuat dari dua struktur:

1. Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (object), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
2. Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

II.8.5. Google Cloud Messaging API

Google *Cloud Messaging* (GCM) untuk android adalah layanan yang membantu pengembang mengirim data dari server untuk aplikasi android mereka ke perangkat android lainnya. Ini bisa menjadi pesan singkat untuk memberitahu aplikasi android bahwa data baru yang akan diambil dari server (misalnya, film diunggah oleh seorang teman), atau bisa juga pesan yang berisi sampai dengan 4KB data *payload* (sehingga aplikasi seperti *instant messaging* dapat menerima pesan langsung) (Andi Juansyah: 2015: 3)

Layanan GCM menangani semua aspek antrian pesan dan pengiriman ke aplikasi target, dan berjalan pada perangkat target. GCM memungkinkan aplikasi android untuk mengirim pesan kepada server untuk membroadcast sebuah notifikasi kepada seluruh *client* yang ada. Hanya membutuhkan sebuah *account* Gmail maka akan langsung dapat menggunakan fasilitas GCM ini.

II.8.6. Google Geocoding API

Geocoding adalah proses konversi alamat ke dalam koordinat geografis seperti didapatkan sebuah alamat (1600 Amphiteatre Mountain View, CA) dan di konversikan menjadi koordinat *latitude* 37,423021 *longitude* – 122,083739, yang dapat digunakan untuk menempatkan penanda pada peta atau posisi peta. Adapun *reverse geocoding* adalah kebalikan dari *geocoding* yaitu konversi koordinat menjadi sebuah alamat yang memungkinkan dapat menemukan sebuah alamat (Andi Juansyah: 2015: 4)

II.8.7. Google Maps API

Google *Maps* merupakan layanan dari Google yang mempermudah penggunaannya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan Google *Maps* API memungkinkan pengembangan untuk mengintegrasikan Google *Maps* ke dalam situs web. Dengan menggunakan Google *Maps* Api memungkinkan untuk menanamkan situs Google *Maps* ke dalam situs eksternal, di mana situs data tertentu dapat dilakukan overlay (Andi Juansyah: 2015: 4)

Meskipun pada awalnya hanya javascript API, API *Maps* sejak diperluas untuk menyertakan sebuah API untuk Adobe Flash aplikasi, layanan untuk mengambil gambar peta status, dan layanan web untuk melakukan *geocoding*, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil elevasi.

II.8.8. Extensible Markup Language (XML)

Extensible Markup Language, disingkat XML, mendefinisikan sebuah kelas objek-objek data yang disebut dokumen XML dan secara parsial mendefinisikan perilaku program komputer yang memprosesnya. XML merupakan sebuah profil aplikasi atau bentuk terbatas dari SGML, *Standard Generalized Markup Language*. Berdasarkan konstruksinya, dokumen XML selaras dengan dokumen SGML. (Handayani, Herlina Fitri, 2019).

II.9. Unified Modelling Language (UML)

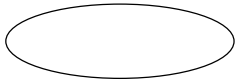
UML (Unified Modeling Language) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. UML juga salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena UML menyediakan bahasa pemodelan Visual yang memungkinkan pengembangan sistem membuat blue print atas visinya dalam bentuk yang baku. (Ommi Alfina, Fitriana Harahap, 2019).

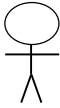


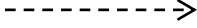
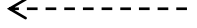
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut:

1. *Use case* Diagram

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol *Use Case*

| Gambar | Keterangan |
|---|---|
|  | <p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama <i>use case</i>.</p> |


| | |
|---|---|
|  | <p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki control terhadap <i>use case</i>.</p> |
|  | <p>Asosiasi antara aktor dan <i>use case</i>, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p> |
|  | <p>Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.</p> |
|  | <p><i>Include</i>, merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program.</p> |
|  | <p><i>Extend</i>, merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.</p> |


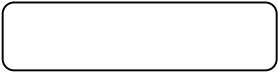
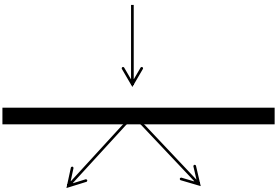
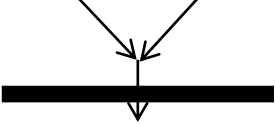
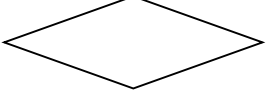

(Sumber: Gellysa Urva dan Helmi Fauzi Siregar: 2015: 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2:

Tabel II.2. Simbol *Activity Diagram*

| Gambar | Keterangan |
|---|--|
|  | <p><i>Start point</i>, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.</p> |

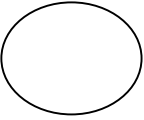
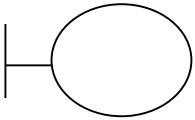
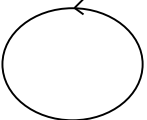

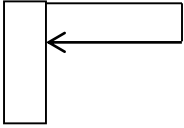


| | |
|---|--|
|  | <i>End point</i> , akhir aktifitas. |
|  | <i>Activites</i> , menggambarkan suatu proses/kegiatan bisnis. |
|  | <i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu. |
|  | <i>Join</i> (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi. |
|  | <i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> , <i>false</i> . |
|  | <i>Swimlane</i> , pembagian <i>activity</i> diagram untuk menunjukkan siapa melakukan apa. |

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015: 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* dapat dilihat pada tabel II.3:

Tabel II.3. Simbol *Sequence Diagram*

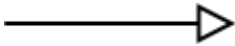
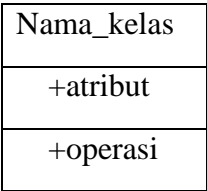


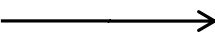
| Gambar | Keterangan |
|---|--|
|  | <i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |
|  | <i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan <i>form</i> cetak. |
|  | <i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek. |
|  | <i>Message</i> , simbol mengirim pesan antar <i>class</i> . |
|  | <i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri. |
|  | <i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi. |
|  | <i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i> . |

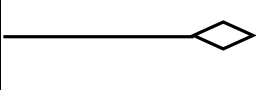
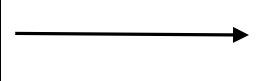
(Sumber : Gellysa Urva dan Helmi Fauzi Siregar:2015: 95)

4. Diagram Kelas (*Class Diagram*)

Class diagram juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi: Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti (Ade Hendini, 2016 : 111). Simbol *class diagram* dan *multiplicity class diagram* dapat dilihat pada Tabel II.4 dan Tabel II.5. di bawah ini :

Tabel II.4. Simbol *Class Diagram*

| Gambar | Keterangan |
|---|---|
|  | <i>Generalization</i> , untuk menghubungkan antar kelas dengan arti umum-khusus. Jadi jika ada kelas dengan arti umum-khusus. Jadi jika ada kelas bermakna umum dan kelas bermakna khusus dapat menggunakan simbol ini. |
|  | <i>Class</i> , untuk sebuah kelas pada struktur sistem. Penulisan tidak boleh menggunakan spasi. Simbol ini memiliki 3 susunan, yaitu kotak pertama adalah nama kelas, kedua atribut dan ketiga operasi. |
|  | <i>Interface</i> , untuk simbol <i>interface</i> atau dalam bahasa indonesianya antar muka. Konsep yang digunakan pun sama dengan pemrograman berorientasi object (OOP). |
|  | <i>Association</i> , digunakan untuk menghubungkan atau merelasikan kelas satu dengan kelas yang lainnya dengan makna umum. |
|  | <i>Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain. |

| | |
|---|--|
|  | <i>Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian. |
|  | <i>Dependency</i> , adalah relasi antar kelas dengan makna kebergantungan antar kelas. |

(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

Tabel II.5. Multiplicity Class Diagram

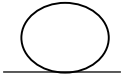
| Multiplicity | Penjelasan |
|---------------------|---|
| 1 | Satu dan hanya satu |
| 0..* | Boleh tidak ada atau 1 atau lebih |
| 1..* | 1 atau lebih |
| 0..1 | Boleh tidak ada, maksimal 1 |
| n..n | Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4 |

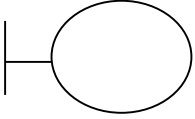
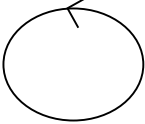

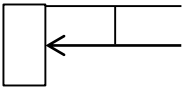


(Sumber : Gellysa Urva dan Helmi Fauzi Siregar : 2015 : 95)

II.10. Sequence diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence* diagram dapat dilihat pada tabel II.5. dibawah ini:

Tabel II.6. Simbol Sequence Diagram

| Gambar | Keterangan |
|---|--|
|  | <i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data. |

| | |
|---|--|
|  | <p><i>Boundary Class</i>, berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan furmenty dan <i>form</i> cetak.</p> |
|  | <p><i>Control class</i>, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.³</p> |
|  | <p><i>Message</i>, simbol mengirim pesan antar <i>class</i>.</p> |
|  | <p><i>Recursive</i>, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.</p> |
|  | <p><i>Activation</i>, <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang Kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.</p> |
|  | <p><i>Lifeline</i>, garis titik-titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>.</p> |

(Sumber: Ade Hendini; 2016: 4)