

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Refrensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal lokal.

Berdasarkan penelitian dari Abadi Nugroho (2020) dengan judul “Penerapan Metode Haversine Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana” Kota Bontang terletak di Provinsi Kalimantan Timur Indonesia merupakan salah satu kota yang paling rawan terjadinya bencana. Saat ini belum ada media khusus dalam proses penyampaian informasi mengenai kejadian bencana dan informasi titik kumpul, banyaknya berita atau informasi yang tidak dapat dipercaya sumbernya dan lambatnya penanganan kebencanaan karena harus mengecek keakuratan informasi kejadian bencana terlebih dahulu. Tujuan dari penelitian ini adalah membuat aplikasi yang dapat memberikan informasi bencana dan mencari titik kumpul dengan jarak terdekat menggunakan metode Haversine Formula serta teknologi Google Maps. Haversine Formula merupakan sebuah persamaan dalam navigasi dengan cara memberikan jarak radius (lingkaran besar) antara dua titik pada permukaan bola (bumi) berdasarkan lintang dan bujur.

Berdasarkan penelitian dari Muh. Iqbal (2021) dengan judul “Penerapan Metode Haversine Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar” Tujuan Penelitian ini dapat memberikan rekomendasi

jasa lokasi laundry terdekat dengan posisi user dan memberi petunjuk arah menuju lokasi tempat laundry sehingga meminimalisir tenaga waktu dan biaya. Metode yang digunakan dalam penelitian ini adalah Haversine Formula menghitung jarak antara dua titik di bumi berdasarkan panjang garis lurus antara dua titik tanpa mengabaikan kelengkungan yang dimiliki bumi, dengan menerapkan teknologi tersebut dalam sebuah SIG pengguna sistem dapat mengetahui posisinya terhadap lokasi tertentu walaupun masih mengabaikan relief bumi yang sebenarnya.

Berdasarkan penelitian dari Ramadiani (2018) dengan judul “Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal” Kota Samarinda terdapat lebih 20 lapangan futsal, pengguna membutuhkan informasi atau kriteria untuk memilih lapangan futsal terbaik sesuai kebutuhan sedangkan informasi tentang lokasi, harga dan kondisi lapangan masih terbatas. Hal ini dapat membuat sebagian pengguna memilih lapangan futsal yang letaknya jauh karena tidak memiliki informasi yang cukup tentang lapangan futsal terdekat. Berdasarkan permasalahan tersebut diperlukan sistem yang dapat menyajikan informasi dan difasilitasi dengan pencarian jarak terdekat.

Berdasarkan penelitian dari Chandra Husada (2019) dengan judul “Implementasi Haversine Formula untuk Pembuatan SIG Jarak Terdekat ke RS Rujukan COVID-19” Perbandingan hasil pengukuran jarak lokasi terdekat antara SIG hasil rancangan berbasis Haversine formula dan Find Nearest Tool diperoleh nilai selisih rata-rata sebesar 13 meter, dengan nilai selisih terkecil 3 meter dan

nilai selisih terbesar 40 meter. Selain itu terdapat juga selisih antara hasil perhitungan berbasis Haversine formula terhadap hasil pengukuran jarak garis lurus dengan aplikasi Google Maps, dengan nilai selisih terkecil 0 meter dan nilai selisih terbesar 5 meter, sehingga nilai selisih rata-rata sebesar 3 meter. Pembuatan SIG juga telah diperoleh melalui rancang bangun use case, activity, dan class diagram yang dilengkapi dengan user interface.

Berdasarkan penelitian dari Sugeng Purwantoro (2021) dengan judul “Perancangan Sistem Informasi Geografis Pariwisata Pasaman Barat Berbasis Android Menggunakan Metode Haversine” Perancangan sistem ini memberikan perhitungan estimasi jarak dan rekomendasi lokasi jarak terdekat menggunakan metode Haversine yang berfungsi untuk menghitung jarak antara dua titik. Sistem dirancang pada platform android, database MySQL dan bahasa pemograman PHP. Pengujian menggunakan pengujian White Box, Black Box, dan pengujian perbandingan dengan Google Maps Distance. Pada pengujian White Box pada kode program Haversine Formula dan kode program tampilan wisata didapatkan hasil nilai Cyclomatic Complexity 1 dan 2, disimpulkan kode program memiliki level resiko yang sederhana dan tanpa banyak resiko sehingga mudah dilakukan maintenance dan diuji alur programnya.

II.2. Uraian Teoritis

II.2.1. Sistem

Definisi sistem adalah “kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.” Definisi sistem adalah “sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek

bisa dilihat sebagai suatu kesatuan yang dirancang untuk mencapai suatu tujuan. Sistem adalah penggabungan dari bagian-bagian atau komponen-komponen yang terpisah-pisah dan disatukan menjadi satu rangkaian dan menjadi suatu fungsi yang baru (Aris, 2017)

Kata sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*sustema*) yang artinya adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi. Secara umum sistem adalah kumpulan dari beberapa bagian tertentu yang saling berhubungan secara harmonis untuk mencapai suatu tujuan tertentu. Elemen-elemen yang mewakili suatu sistem secara umum adalah masukan (*Input*) pengolahan (*Processing*) dan keluaran (*Output*). (Sutopo et al., 2018)

II.2.2. Informasi

Pengertian informasi masih bersifat kontradiktif dan belum mempunyai ciri-ciri khusus. Maka para pakar teori atau ilmu pengetahuan informasi belum dapat membuat satu definisi yang jelas dan lengkap mengenai apa yang disebut informasi.

Menurut para pakar informasi, informasi meliputi aspek abstrak dan khusus. Informasi dipandang sebagai ilmu pengetahuan tertulis atau yang disampaikan secara lisan dan sebagai hasil dari data yang diolah, biasanya diolah secara formal. Ada pula yang menyebut informasi sebagai sarana pengurang ketidakpastian. Pengertian ini sama dengan arti informasi dalam istilah ekonomi,

yakni informasi adalah penghalau ketidakpastian, misalnya di dalam pasar, *consumer preference*, dan harga (Tyoso ; 2018 : 31)

II.2.3 Sistem Informasi Geografis

SIG adalah sistem komputer yang digunakan untuk mengumpulkan, memeriksa, mengintegrasikan, dan menganalisa informasi-informasi yang berhubungan dengan permukaan bumi. Pada dasarnya, istilah sistem informasi geografi merupakan gabungan dari tiga unsur pokok yaitu sistem, informasi, dan geografi. Dengan demikian, pengertian terhadap ketiga unsur-unsur pokok ini akan sangat membantu dalam memahami SIG. Dengan melihat unsur-unsur pokoknya, maka jelas SIG merupakan salah satu sistem informasi. SIG merupakan suatu sistem yang menekankan pada unsur informasi geografi. Istilah “geografis” merupakan bagian dari spasial (keruangan). Kedua istilah ini sering digunakan secara bergantian atau tertukar hingga timbul istilah yang ketiga, geospasial. Ketiga istilah ini mengandung pengertian yang sama di dalam konteks SIG. Penggunaan kata “geografis” mengandung pengertian suatu persoalan mengenai bumi: permukaan dua atau tiga dimensi. Istilah “informasi geografis” mengandung pengertian informasi mengenai tempat-tempat yang terletak di permukaan bumi, pengetahuan mengenai posisi dimana suatu objek terletak di permukaan bumi, dan informasi mengenai keterangan-keterangan (atribut) yang terdapat di permukaan bumi yang posisinya diberikan atau diketahui. (Koko Mukti Wibowo : 2019)

II.2.4. Metode

Teorema Haversine merupakan metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat kelengkungan. Teorema haversine digunakan untuk menghitung jarak antara 2 titik dengan berdasarkan panjang garis lurus antara 2 titik pada garis bujur (lattitude) dan garis lintang (longitude). Dalam menentukan jarak garis lurus dari dua lokasi, Formula Haversine adalah algoritma yang bagus. Haversine melakukan perhitungan berdasarkan garis lintang dan garis bujur. Haversine hanya bisa digunakan di peta dunia, jadi Haversine tidak cocok untuk mengukur jarak dengan satuan selain lintang dan bujur. Algoritma Haversine memiliki akurasi yang sangat baik. Dengan menerapkan algoritma Haversine, jarak dua titik koordinat pada peta dunia dapat dihitung. Haversine Formula pada sistem ini digunakan untuk menghitung jarak antara dua titik koordinat yaitu koordinat pengguna dengan GPS dan koordinat lokasi objek pariwisata yang telah disimpan pada database. Haversine Formula dipilih karena berdasarkan penelitian yang dilakukan oleh Miftahuddin. Metode perhitungan Haversine menghasilkan akurasi perhitungan jarak tertinggi yaitu 98,66%. Dan metode perhitungan Haversine menghasilkan akurasi keputusan tertinggi dalam menentukan keputusan lokasi keberadaan karyawan yaitu 90%. Haversine Formula menjadi kunci untuk menghitung jarak antara pengguna dan objek pariwisata terdekat. (Sugeng Purwantoro : 2021)

II.2.5. Peta

Banyak sekali definisi tentang peta tetapi pada dasarnya hakekat peta adalah sebagai berikut :

1. Peta adalah gambaran seluruh atau sebagian permukaan bumi yang diproyeksikan dalam dua dimensi pada bidang datar dengan metode dan perbandingan tertentu.
2. Peta adalah suatu presentasi di atas bidang datar baik seluruh atau sebagian permukaan bumi yang dilihat dari atas dan diperkecil dengan perbandingan tertentu.
3. Peta adalah sebuah gambar suatu daerah yang dapat dibayangkan seolah olah kita melihat daerah itu dari udara.
4. Peta adalah gambar seluruh atau dari permukaan bumi yang dilukiskan ke suatu bidang datar dengan perbandingan tertentu yang dinamai kedar / skala.

Berdasarkan beberapa definisi tersebut, paling tidak kita dapat menyimpulkan satu pengertian tentang peta. Defenisi di atas dapat saling melengkapi dan memperjelas kesimpulan kita tentang pengetahuan peta (Samadi ; 2018 : 152).

II.2.6. Spasial

Menurut G. Manjela Eko Hartoyo (2010 : 2) Sebagian besar data yang akan ditangani dalam SIG merupakan data spasial yaitu sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari

data lain, yaitu informasi lokasi (spasial) dan informasi *deskriptif (attribute)* yang dijelaskan berikut ini :

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi *deskriptif (atribut)* atau informasi non spasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan dengannya, contohnya: jenis vegetasi, populasi, luasan, kode pos, dan sebagainya.

II.2.7. Digital

Peta yang merupakan konversi dalam bentuk *digital* (angka) yang tersimpan dalam komputer. Peta *digital* mempunyai beberapa kelebihan dibandingkan dengan peta *analog*, antara lain adalah: Peta ini dibuat dengan menggunakan komputer sehingga proses pembuatannya lebih cepat dan hasilnya lebih bagus. Untuk pembaruan (revisi) peta pun dapat dilakukan lebih cepat. Sedangkan untuk penyimpanannya, tidak memerlukan ruang sebanyak yang diperlukan untuk menyimpan peta *analog*. Peta *digital* juga dapat ditampilkan dalam bentuk *planimetri* maupun *stereometri*. Peta *digital* ini datanya dapat diperoleh dari beberapa cara, antara lain adalah: dari citra satelit, dari dijitasi foto udara dan atau peta garis, serta dari hasil pengukuran dengan alat ukur Total Station (TS). Penggambaran peta *digital* dapat dilakukan dengan menggunakan program-program Arc Info, Map Info, Auto Cad , Arc View dan lain sebagainya (IG Indradi ; 2018 : 6)

II.2.8. Web

Web adalah salah satu aplikasi yang berisikan dokumen-dokumen multimedia (teks, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol HTTP (*Hypertext transfer protocol*) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser. Beberapa jenis browser yang populer saat ini di antaranya : *internet Explorer* yang diproduksi oleh *Microsoft*, *Mozilla Firefox*, *Opera* dan *Safari* yang diproduksi oleh *Apple*. *Browser* (perambah) adalah aplikasi yang mampu menjalankan dokumen-dokumen web dengan cara diterjemahkan. Prosesnya dilakukan oleh komponen yang terdapat di dalam aplikasi *browser* yang biasa disebut *web engine*. Semua dokumen *web* ditampilkan oleh *browser* dengan cara diterjemahkan (M. Rudyanto Arief ; 2019 : 7-8).

World wide web atau sering di kenal sebagai web adalah suatu layanan sajian informasi yang menggunakan konsep *hyperlink* (tautan), yang memudahkan surfer (sebutan para pemakai komputer yang melakukan browsing atau penelusuran informasi melalui internet). Keistimewaan inilah yang telah menjadikan web sebagai service yang paling cepat pertumbuhannya. Web mengijinkan pemberian highlight (penyorotan atau penggaris bawahan) pada kata-kata atau gambar dalam sebuah dokumen untuk menghubungkan atau menunjuk ke media lain seperti dokumen, *frase*, *movie clip*, atau *file* suara. Web dapat menghubungkan dari sembarang tempat dalam sebuah dokumen atau gambar ke

sembarang tempat di dokumen lain. Dengan sebuah *browser* yang memiliki *Graphical User Interface* (GUI), link-link dapat di hubungkan ke tujuannya dengan menunjuk link tersebut dengan *mouse* dan menekannya (Muhammad Susilo, dkk : 2018)

Website diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara dan atau gabungan dari semuanya, baik bersifat statis maupun dinamis yang membentuk suatu rangkaian bangunan yang saling terkait, yang masing-masing dihubungkan dengan jaring-jaringan halaman (Sudaryono : 2020)

II.2.9. PHP

PHP adalah sebuah bahasa pemrograman *scripting* untuk membuat halaman *web* yang dinamis. Walaupun dikenal sebagai bahasa untuk membuat halaman *web*, tapi *php* sebenarnya juga dapat digunakan untuk membuat aplikasi *command line* dan juga GUI. *Website* yang dibuat menggunakan *PHP* memerlukan software bernama *webserver* tempat pemrosesan kode *PHP* dilakukan. *Server web* yang memiliki software *PHP* parser akan memproses input berupa kode *PHP* dan menghasilkan *output* berupa halaman *web*. *PHP* bersifat terbuka dan *multiplatform*, karenanya dapat dijalankan di banyak merek *web server* (seperti *apache* dan *IIS*) (Ali Zaki ; 2015 : 2).

PHP adalah bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah *web* dan bias digunakan pada *HTML*. *PHP* merupakan singkatan dari “*PHP : Hypertext Preprocessor*”, dan

merupakan bahasa yang disertakan dalam dokumen HTML, sekaligus bekerja di sisi server (*server-side HTML-embedded scripting*). Artinya sintaks dan perintah yang diberikan akan sepenuhnya dijalankan di *server* tetapi disertakan pada halaman HTML biasa, sehingga script-nya tak tampak disisi client. PHP dirancang untuk dapat bekerja sama dengan database *server* dan dibuat sedemikian rupa sehingga pembuatan dokumen HTML yang dapat mengakses database menjadi begitu mudah. Tujuan dari bahasa *scripting* ini adalah untuk membuat aplikasi di mana aplikasi tersebut yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server* (Muhammad Susilo, dkk : 2018)

II.2.10. Dreamweaver

Macromedia *dreamweaver* adalah sebuah perangkat lunak aplikasi untuk mendesain dan membuat halaman web. Dengan menggunakan *Dreamweaver 8*, ketika membuat sebuah halaman *web*, Anda tidak perlu lagi mengetik kode-kode *HTML* atau kode-kode lainnya secara manual. Anda cukup melakukan klik beberapa kali, maka simsalabin, halaman *web* yang Anda inginkan sudah jadi. Selain *HTML*, *Dreamweaver 8* juga mendukung *CSS*, *JavaScript*, *PHP*, *ASP*, dan bahasa pemrograman lainnya untuk membuat *web*. Hal ini akan sangat menguntungkan Anda. Sebagai contoh, jika dahulu Anda harus mengetikkan kode-kode *CSS* untuk membuat *Style* tertentu, maka dengan *Dreamweaver 8*, Anda cukup melakukan klik beberapa kali saja. *Dreamweaver 8* adalah versi terbaru dari keluarga *Dreamweaver*. Versi pertamanya sendiri diluncurkan sekitar

tahun 1994 oleh *Macromedia Inc.* Dalam versi terbaru ini, banyak sekali fasilitas baru yang ditambahkan. Contohnya, Anda akan dapat membuat dan menggunakan *Style* dalam *CSS* dengan mudah dan fleksibel. Panel untuk pengolahan *CSS* juga sudah diperbarui dan lebih mudah digunakan. *Dreamweaver 8* juga menyediakan beberapa template halaman *web* baru, termasuk *fasilitas Starter Pages*. Akhirnya, selamat memulai mempelajari *Dreamweaver 8* (Arief Ramadhan ; 2019 : 2).

II.2.11. Apache

Apache adalah paket aplikasi yang digunakan untuk *web server* yang handal dan stabil. Jika dibandingkan dengan *web server* lainnya, *Apache* masih menjadi andalan para webmaster. Perkembangan server ini sangat pesat sehingga hampir semua *server web* menggunakan *Apache*. Aplikasi *apache* dikenal dengan nama *httpd*. Jika menggunakan *linux redhat/Fedora* maka paket ini sudah ada dalam bentuk *.RPM* (Albertus Dwiyoga ; 2018 : 11).

II.2.12. CSS

(*CSS*) *Cascading Style Sheets* adalah suatu bahasa *stylesheet* yang digunakan untuk mengatur style suatu dokumen. Pada umumnya *CSS* dipakai untuk memformat tampilan halaman *web* yang dibuat dengan bahasa *HTML* dan *XHTML*. *CSS* memungkinkan *web developer* untuk memisahkan *HTML* dari aturan-aturan untuk membentuk tampilan sebuah *website*. *CSS* diperkenalkan untuk pengembangan website pada tahun 1996. Nama *CSS* didapat dari fakta bahwa setiap deklarasi *style* yang berbeda dapat diletakkan secara berurutan, yang

kemudian akan membentuk hubungan parent-child pada setiap *style* (Sulistiyawan ; 2018 : 32).

II.2.13. *MySQL*

MySQL merupakan suatu jenis *database server* yang sangat terkenal. *MySQL* termasuk jenis *RDBMS (Relational Database Manajement System)*. *MySQL* mendukung bahasa pemrograman PH, bahasa permintaan yang terstruktur, karena pada penggunaannya *SQL* memiliki beberapa aturan yang telah distandarkan oleh asosiasi yang bernama *ANSI*. *MySQL* merupakan *RDBMS (Relational Database Management System) server*. *RDBMS* adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya. Beberapa keunggulan dari *MySQL* yaitu :

- a. Cepat, handal dan mudah dalam penggunaannya. *MySQL* lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan *MySQL*.
- b. Didukung oleh berbagai bahasa *Database Server MySQL* dapat memberikan pesan *Error* dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
- c. Mampu membuat tabel berukuran sangat besar. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan *MySQL* adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.

- d. Lebih murah *MySQL* bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk *UNIX* platform, *OS/2* dan *Windows Platform*. Melekatnya integrasi *PHP* dengan *MySQL*. Keterikatan antara *PHP* dengan *MySQL* yang sama-sama *Software Open-Source* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya. Modul *MySQL* di *PHP* telah dibuat *Built-in* sehingga tidak memerlukan konfigurasi tambahan pada *File* konfigurasi *Php* ini (Reza Hermiati : 2021).

II.2.14. Unified Modeling Language (UML)

Menurut Sri Dharwiyanti (2013) *Unified Modelling Language (UML)* adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.

Unified Modelling Language (UML) merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem software yang terkait dengan objek. *UML* merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena *UML* menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem membuat *blue print* atas visinya dalam bentuk yang baku. *UML* berfungsi sebagai jembatan dalam mengkomunikasikan beberapa aspek dalam sistem melalui jumlah elemen grafis yang bisa dikombinasikan menjadi. *Unified Modeling Language (UML)* biasa digunakan untuk :

- a. Menggambarkan batasan sistem dan fungsi - fungsi sistem secara umum, dibuat dengan *use case* dan *actor*.
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan *interaction diagrams*.
- c. Menggambarkan representasi struktur *static* sebuah sistem dalam bentuk *class diagrams*.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan *state transition diagrams*.
- e. Menyatakan arsitektur implementasi fisik menggunakan *component and development*.
- f. Menyampaikan atau memperluas *fungsi* dengan *stereotypes*. (Omni Alfina : 2019)

1. Use Case Diagram

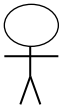
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk

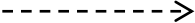
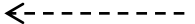
semua *feature* yang ada pada sistem.

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend* *use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

Tabel II.1. Simbol Use Case

Gambar	Keterangan
	<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> .
	Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan 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> .
	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.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
	<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.
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

(Sumber : Janiver W. Janis; 2020)

2. Class Diagram

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus

menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Tabel II.2. 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 : Janiver W. Janis; 2020)

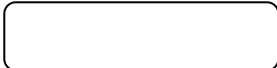
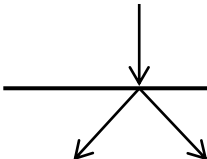
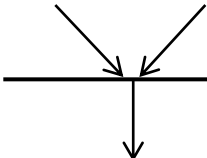
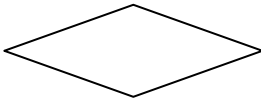
3. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem)

secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas

Tabel II.3. Simbol Activity Diagram

Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<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 diagram</i> untuk

	menunjukkan siapa melakukan apa.
--	----------------------------------

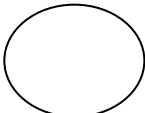
(Sumber : Janiver W. Janis; 2020)

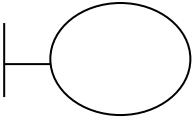
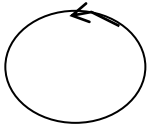
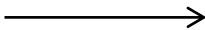
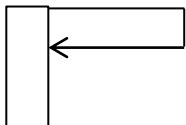
4. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*.

Tabel II.4. 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 objk, sepanjang <i>lifeline</i> terdapat <i>activation</i>.

(Sumber : Sri Dharwiyanti ; 2013 : 9)