

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terkait

Penelitian terkait bertujuan untuk mengambil beberapa referensi jurnal terkait yang digunakan dalam mendukung penelitian publikasi ilmiah dalam jurnal local yaitu:

Berdasarkan penelitian dari Muhammad Yanis (2020) dengan judul “Rancang Bangun Aplikasi Pencarian Lokasi Penjualan Tanah Menggunakan Metode Haversine” Proses jual beli tanah yang lazim dilakukan saat ini di kota Lhokseumawe dan Aceh utara yaitu dengan memanfaatkan media massa, media cetak maupun situs dagang online untuk memasarkan tanah . proses ini terbilang tidak efisien dan efektif karena bercampur dengan iklan item yang tidak sejenis. Selain itu, Banyaknya lokasi tanah yang dijual di Kota Lhokseumawe dan Aceh Utara juga tidak diikuti dengan informasi yang cukup bagi masyarakat yang ingin mencari informasi tanah yang dijual di wilayah tersebut.Maka dibangunlah sebuah sistem informasi pencarian lokasi penjualan tanah secara online yang areanya khusus Lhokseumawe dan Aceh Utara.Sistem yang dibangun dapat memberikan informasi lokasi penjualan tanah terdekat berdasarkan posisi pembeli,informasi lokasinya ditampilkan melalui google maps.Metode pengembangan sistem yang digunakan yaitu Metode Haversine (menghitung jarak terdekat).Analisis perancangan meliputi use case diagram.

Berdasarkan penelitian dari Algi Muhkamaat Abdillah (2019) dengan judul “Penerapan Metode Haversine pada Aplikasi Layanan Perbaikan Kendaraan Berbasis Location Based Service” Tujuan dari penelitian ini adalah membangun sebuah aplikasi dengan menerapkan metode haversine untuk menemukan bengkel terdekat pada aplikasi layanan perbaikan kendaraan. Formula Haversine membutuhkan masukan latitude dan longitude lokasi pengguna dan lokasi bengkel sebagai masukan untuk menemukan bengkel terdekat. Penelitian ini didukung dengan metodologi pengembangan perangkat lunak yaitu metode Extreme Programming dalam melakukan analisis dan desain perancangan terhadap aplikasi yang dibangun.

Berdasarkan penelitian dari Anita Bawaiqi Wandanaya (2021) dengan judul “Aplikasi Pencarian Lokasi Dan Informasi Alamat Rumah Dengan Menggunakan LBS Berbasis Web” Dan terkadang ada orang tak dikenal masuk kekomplek perumahan tersebut dengan tujuan untuk mencuri/merampok rumah yang kosong. Berdasarkan permasalahan yang ada dibutuhkan sistem pencarian lokasi berbasis android yang dapat memudahkan kurir, tamu dan warga untuk mencari lokasi di perumahan permata tangerang. Aplikasi ini dibuat menggunakan Bahasa pemrograman android dan database Mysql, metode analisis menggunakan PIECES, metode perancangan menggunakan Unified Modelling Language (UML). Hasil dari penelitian menunjukkan bahwa aplikasi berbasis Android dapat memudahkan kurir, tamu dan warga untuk mencari lokasi di perumahan permata tangerang.

Berdasarkan penelitian dari Denka Amalia Putri (2021) dengan judul “Perancangan Aplikasi Pencarian Lokasi Penitipan Hewan Peliharaan Menggunakan Metode Haversine Formula (Studi Kasus: Wilayah Jakarta Barat)” Jakarta merupakan ibu kota dari Indonesia, Jakarta merupakan kota yang banyak dikunjungi oleh pendatang baru dari berbagai daerah. Dan memiliki jumlah penduduk yang terus meningkat tiap tahunnya. kebutuhan akan informasi adalah hal mendasar bagi kebutuhan masyarakat, salah satu yang di butuhkan masyarakat adalah kebutuhan akan perawatan hewan peliharaan sehingga pemilik harus meluangkan waktu untuk mengurus hewan pemeliharaan mereka akan tetapi pemilik hewan terkadang memiliki banyak pekerjaan harus diselesaikan. Hal ini menyebabkan hewan peliharaan tidak terawat dengan baik.

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 di miliki bumi, dengan menerapkan teknologi tersebut dalam sebuah SIG pengguna sistem dapat mengetahui posisinya terhadap lokasi tertentu walupun masih mengabaikan relief bumi yang sebenarnya.

Berdasarkan penelitian dari Farhan Ramadhan (2018) dengan judul “Sistem Informasi Penjualan Berbasis Web Pada PT. Mustika Jati” Hasil dari penelitian ini berupa Sistem Informasi Penjualan PT. Mustika Jati berbasis web yang mampu melakukan pencatatan, pengolahan, dan mengintegrasikan data serta menjadi tempat penyimpanan data yang terstruktur sehingga dapat meningkatkan efektifitas dan efisiensi kinerja maupun waktu. Berdasarkan hasil pengujian dan evaluasi dari penelitian, sistem yang telah dibangun mampu memberikan informasi lengkap mengenai penjualan, stok barang, dan pelanggan di PT. Mustika Jati.

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 dari 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.2.1.1. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem lainnya yaitu:

1. Batasan (*Boundary*) adalah Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*Environment*) adalah Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*Input*) adalah Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*Output*) adalah Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar Komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*Component*) adalah Kegiatan-kegiatan atau proses dalam sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

6. Penghubung (*Interface*) adalah Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*Storage*) adalah Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama (Aris, 2017)

II.2.2. 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.2.3. JQuery

jQuery Mobile adalah framework yang berbasis *HTML/CSS* dan *javascript* untuk aplikasi *web* yang dijalankan di perangkat *mobile*. Dalam membangun web biasa menggunakan *framework jquery*, begitu juga apabila ingin membuat aplikasi web untuk perangkat *mobile (smartphone/tablet pc)* bisa digunakan *jquery mobile*. Penggunaan *jquery mobile* akan menghasilkan tampilan yang menyesuaikan dengan lebar layar. *Script jquery mobile* merupakan bawaan dari *framework jquery mobile*, seperti *jquery.mobile-1.0a1.min.css* dan *styles.css*. Untuk menggunakan *jQuery Mobile*, pertama perlu memasukkan tiga *file* yaitu.

1. *jQuery CSS Mobile file (jquery.mobile-1.0a1.min.css)*
2. *Perpustakaan jQuery (jquery1.4.3.min.js)*
3. *Mobile perpustakaan jQuery (jquery.mobile-1.0a1.min.js)*. (Koko Mukti Wibowo : 2018)

II.2.4. Metode Haversine

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)

Formula *Haversine* adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bola (bumi) berdasarkan bujur dan lintang. *Formula Haversine* merupakan suatu metode untuk mengetahui jarak antar dua titik dengan memperhitungkan bahwa bumi bukanlah sebuah bidang datar namun adalah sebuah bidang yang memiliki derajat

kelengkungan. Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus *haversine*:

$$\Delta\text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta\text{long} = \text{long2} - \text{long1}$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta\text{long}/2)$$

$$c = 2 \cdot \arctan2(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Dimana :

R = jari-jari bumi sebesar 6371(km)

Δlat = besaran perubahan latitude

Δlong = besaran perubahan longitude

C = kalkulasi perpotongan sumbu

d = jarak (km) 1 derajat = 0.0174532925 radian.

II.2.5. *Global Positioning System (GPS)*

Google Maps adalah layanan pemetaan berbasis *Android* service yang disediakan oleh Google dan bersifat gratis, yang memiliki kemampuan terhadap banyak layanan pemetaan berbasis *Android*. *Google Maps* juga memiliki sifat server side, yaitu peta yang tersimpan pada server Google dapat dimanfaatkan oleh pengguna. *Google Maps* API adalah suatu library yang berbentuk javascript yang berguna untuk memodifikasi peta yang ada di *Google Maps* sesuai

kebutuhan. Untuk membangun aplikasi yang memanfaatkan *Google Maps* di desktop dan mobile device maka akan digunakan *Google Maps Javascript API v3* yang memiliki keunggulan lebih cepat dari versi sebelumnya. (Koko Mukti Wibowo : 2018)

II.2.6. Website

Website merupakan sebuah media informasi yang ada di internet. Website tidak hanya dapat digunakan untuk penyebaran informasi saja melainkan bisa digunakan untuk membuat toko online. Website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada di dalam World Wide Web (WWW) di Internet. Sebuah halaman web adalah dokumen yang ditulis dalam format HTML (Hyper Text Markup Language), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser. Semua publikasi dari website-website tersebut dapat membentuk sebuah jaringan informasi yang sangat besar (Yunita Trimarsiah : 2017).

World wide web atau sering dikenal 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 mengizinkan 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.2.7. PHP

PHP adalah bahasa pelengkap HTML yang memungkinkan dibuatnya aplikasi dinamis yang memungkinkan adanya pengolahan data dan pemrosesan data. Semua syntax yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. Kemudian merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirimkan ke client, tempat pemakai menggunakan browser. PHP dikenal sebagai sebuah bahasa scripting, yang menyatu dengan tag-tag HTML, dieksekusi di server, dan digunakan untuk membuat halaman web

yang dinamis seperti halnya Active Server Pages (ASP) atau Java Server Pages (JSP). PHP merupakan sebuah software Open Source (Reza Hermiati : 2021).

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.2.8 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.2.9. Normalisasi

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik tanpa redundansi. Tujuan normalisasi adalah mengorganisasikan data kedalam tabel-tabel untuk memenuhi kebutuhan pemakai, menghilangkan kerangkapan data, mengurangi kompleksitas, mempermudah modifikasi data. (Mukhlisulfatih Latief : 2018)

1. Proses Normalisasi

- a. Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu kebeberapa tingkat.
- b. Apabila tabel yang diuji belum memenuhi persyaratan tertentu maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2. Tahapan Normalisasi :

- 1) Bentuk tidak normal : Menghilangkan perulangan grup.
- 2) Bentuk Normal pertama (1NF) : Menghilangkan ketergantungan sebagian.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kesatu bila setiap data bersifat atomik yaitu setiap irisan baris dan kolom hanya mempunyai satu nilai data.
- 3) Bentuk Normal kedua (2NF) : Menghilangkan ketergantungan transitif.

Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal kedua bila relasi tersebut sudah memenuhi bentuk normal kesatu dan atribut yang bukan key sudah tergantung penuh terhadap key-nya.

4) Bentuk Normal ketiga (3NF) : Menghilangkan anomali-anomali hasil dari ketergantungan fungsional. Yaitu : suatu relasi dikatakan sudah memenuhi bentuk normal ketiga bila relasi tersebut sudah memenuhi bentuk normal kedua dan atribut yang bukan key tidak tergantung transitif terhadap key-nya.

II.2.10. UML (*Unified Modelling Language*)

Menurut Alfina & Harahap (2019) 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.

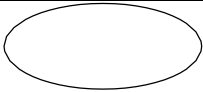
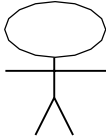
- b. Menggambarkan kegiatan atau proses bisnis yang dilaksanakan secara umum, dibuat dengan interaction diagrams.
- c. Menggambarkan representasi struktur static sebuah sistem dalam bentuk class diagrams.
- d. Membuat model behavior “yang menggambarkan kebiasaan atau sifat sebuah sistem” dengan state transition diagrams.
- e. Menyatakan arsitektur implementasi fisik menggunakan component and development.
- f. Menyampaikan atau memperluas fungsionalitas dengan stereotypes.

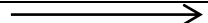
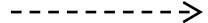
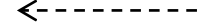
Diagram-diagram yang terdapat dalam UML sangat banyak, berikut ini beberapa diagram yang sering digunakan dalam pengembangan sistem yaitu :

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat.

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 mengidentifikasi aktor, harus ditentukan pembagian

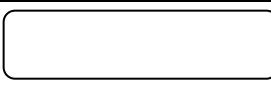
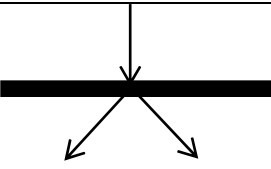
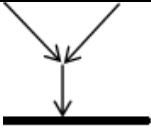
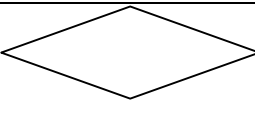
	tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan use case, tetapi tidak memiliki control terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan 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: Rosa dan Shalahudin (2018))

b. Activity Diagram

Activity adalah model yang menggambarkan aktivitas yang tersusun dari sistem yang akan dibuat.

Tabel II.2. Simbol Activity Diagram

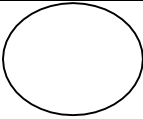
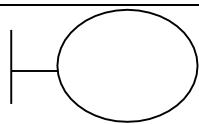
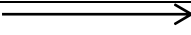
Gambar	Keterangan
	<i>Start point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	<i>End point</i> , akhir aktifitas.
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
	<i>Join</i> (penggabungan) atau <i>merge</i> , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, true, false.
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

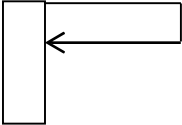
(Sumber: Rosa dan Shalahudin (2018))

c. *Sequence Diagram*

Sequence Diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu.

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.
	Proses atau sistem lain yang berinteraksi dengan sistem informasi.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan <i>form entry</i> dan form 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 class.

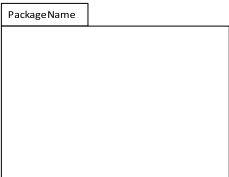
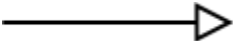
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

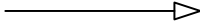
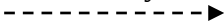
(Sumber: Rosa dan Shalahudin (2018))

d. Class Diagram

Class Diagram adalah adalah model yang menggambarkan hubungan antarkelas untuk membangun sebuah sistem.

Tabel II.4. Simbol *Class Diagram*

Gambar	Keterangan
	Package merupakan sebuah bungkusan dari satu atau lebih kelas.
	<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.
Nama_kelas	<i>Class</i> , untuk sebuah kelas pada struktur sistem.

+atribut +operasi	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 interface 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>Asosiasi Berarah / Directed Association</i> , adalah relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai dengan multiplicity (contohnya : one-to-one, one-to-many, many-to-many)
	<i>Agregasi / Aggregation</i> , adalah relasi antar kelas dengan makna semua bagian.
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
Kebergantungan / defedency 	Relasi antar kelas dengan makna kebergantungan antar kelas.

Tabel II.5. Simbol *Multiplicity*

<i>Multiplicity</i>	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih

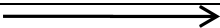
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber: Rosa dan Shalahudin (2018))

e. *Statemachine Diagram*

Statemachine diagram yaitu salah satu jenis diagram pada UML yang menggambarkan transisi maupun perubahan keadaan suatu objek pada sistem.

Tabel II.6. Simbol *Statemachine Diagram*

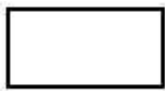
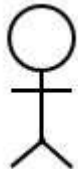
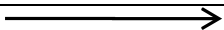
Gambar	Keterangan
	<i>State</i> , digambarkan berbentuk segi empat dengan sudut membulat dan memiliki nama sesuai kondisi saat itu.
	Titik awal (<i>start</i>) Digunakan untuk menggambarkan awal dari kejadian dalam suatu diagram flowchart.
	Titik Akhir(<i>end</i>) Digunakan untuk menjelaskan/ menggambarkan akhir (<i>end</i>) dari kejadian dalam suatu diagram <i>state chart</i> .
[Guard]	<i>Guard</i> ,Merupakan syarat transisi yang bersangkutan.
	<i>Point</i> , digunakan untuk menggambarkan /menjelaskan apakah akan masuk (<i>entry point</i>) ke dalam status atau keluar (<i>exit point</i>).
<i>event</i>	<i>Event</i> , digunakan untuk menjelaskan kondisi yang menyebabkan sesuatu pada status.

(Sumber: Rosa dan Shalahudin (2018))

f. *Communication Diagram*

Communication diagram yaitu salah satu jenis diagram pada UML yang dapat menggambarkan tahapan terjadinya suatu aktivitas dan diagram ini juga menggambarkan interaksi antara objek yang ada pada sistem. Hampir sama seperti *sequence* diagram akan tetapi *communication* diagram lebih menekankan kepada peranan masing-masing objek pada sistem.

Tabel II.7. Simbol *Communication Diagram*

Gambar	Keterangan
	<i>Objek</i> , adalah instance dari sebuah kelas yang tersusun secara horizontal. Digambarkan sebagai sebuah kelas (kotak) dengan nama objek didalamnya yang diawali dengan sebuah titik koma (;).
	<i>Actor</i> , juga dapat berhubungan dengan objek, maka actor juga dapat diurutkan sebagai kolom. Simbolnya sama pada <i>Actor Use Case Diagram</i> .
	<i>Message</i> , digambarkan dengan anak panah yang mengarah antar obyek dan diberi label urutan nomor yang mengindikasikan urutan komunikasi yang terjadi

	antar obyek.
--	--------------

(Sumber: Rosa dan Shalahudin (2018))

g. *Deployment Diagram*

Deployment diagram yaitu salah satu diagram pada UML yang menunjukkan tata letak suatu sistem secara fisik, dapat juga dikatakan untuk menampilkan bagian-bagian *software* yang terdapat pada *hardware* dan digunakan untuk menerapkan suatu sistem dan hubungan antara komponen *hardware*. Jadi *Deployment* diagram intinya untuk menunjukan letak *software* pada *hardware* yang digunakan sistem.

Tabel II.8. Simbol *Deployment Diagram*

Gambar	Keterangan
	<i>Package</i> , adalah sebuah bungkusan dari satu atau lebih <i>node</i> .
	<i>Node</i> , biasanya mengacu pada hardware, perangkat lunak yang tidak dibuat sendiri (<i>software</i>). Jika didalam node disertai komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
	<i>Dependency</i> , adalah kebergantungan antar node, arah

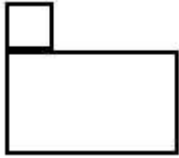
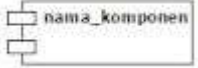
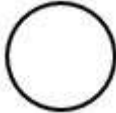
	panah mengarah pada node yang dipakai.
	<i>Link</i> , merupakan relasi antar node.

(Sumber: Rosa dan Shalahudin (2018))

h. *Component Diagram*

Component diagram yaitu salah satu jenis diagram pada UML yang menggambarkan *software* pada suatu sistem. *Component* diagram merupakan penerapan *software* dari satu ataupun lebih *class*, dan biasanya berupa file data atau .exe, source kode, table, dokumen dsb.

Tabel II.9. Simbol *Component Diagram*

Gambar	Keterangan
	<i>Package</i> , merupakan sebuah simbol yang dipakai untuk tempat komponen.
	<i>Component system</i> . adalah simbol yang menjelaskan perangkat keras atau objek dalam sistem tersebut.
	<i>Dependency</i> , simbol yang menjelaskan sebuah keterkaitan antara komponen, satu komponen dengan yang lain. Arah panah dalam simbol tersebut diarahkan pada komponen yang dipakai.
	<i>Interface</i> . hal ini mirip dengan bahasa pemrograman berorientasi objek (PBO), dimana simbol ini dipakai untuk antar muka dengan fungsi supaya tidak langsung mengakses objek.
	<i>Link</i> , simbol link ini dipakai untuk mengarahkan relasi antar komponen, jika suatu komponen memiliki relasi

	atau keterkaitan dengan komponen lainnya maka dipakailah simbol link ini.
--	---

(Sumber: Rosa dan Shalahudin (2018))