



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Wahyudin (2018:1), “Komputer adalah suatu peralatan elektronik yang dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) menggunakan suatu program yang tersimpan di memori komputer, dapat menyimpan program dan hasil pengolahan serta bekerja secara otomatis”.

Menurut Watrianthos (2020:2), “Komputer adalah barang teknologi terpenting yang diperlukan. Sebagian kalangan masyarakat, secara langsung atau tidak langsung ikut menggunakan komputer. Hampir kalangan masyarakat tampaknya melihat komputer dari wujud yang nyata yaitu persegi dan mempunyai monitor, tetapi pada dasarnya komputer memiliki dua buah perangkat utama meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*). Komponen dari hardware yaitu monitor, *keyboard*, *mouse* dan *Central Processing Unit* (CPU) yang terdiri dari komponen *motherboard*, *VGA card*, *processor*, *harddisk*, *RAM* serta *power supply*”.

2.1.1.1 Fungsi Komputer

Menurut Prawiro (2019), Komputer memiliki 4 fungsi utama, yaitu :

1. *Data Input*

Komputer dapat menerima informasi atau data dari sumber lain. Data tersebut diterima melalui aktivitas di *keyboard*, *mouse*, dari komputer lain atau peralatan lainnya.

2. *Data Processing*

Salah satu fungsi utama komputer adalah untuk melakukan pengolahan data sehingga menghasilkan *output*, yaitu berupa informasi. Umumnya, data yang diolah di dalam komputer adalah berbentuk teks, gambar, audio, video, grafik, dan lainnya.



3. Data Output

Fungsi komputer berikutnya adalah untuk menghasilkan *output*/ informasi setelah melalui proses pengolahan data. Informasi tersebut dapat disajikan melalui monitor, alat *printer*, dan *speaker*.

4. Data Storage

Komputer juga dapat berfungsi sebagai tempat untuk menyimpan data sehingga dapat ditemukan dengan mudah dan digunakan kembali. Data tersebut dapat disimpan di dalam memori internal komputer maupun memori eksternal.

5. Data Movement

Fungsi lain dari komputer adalah untuk memindahkan data dari satu komputer ke komputer lainnya atau berbagai alat *output* lainnya.

2.1.2 Pengertian Perangkat Lunak

Menurut Pahul dalam Sindu dan Paramartha (2018:192), “*Software* atau perangkat lunak adalah kumpulan dari data-data elektronik yang diformat, disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer”.

Menurut Prehanto (2020:55), “*Software* atau perangkat lunak adalah kerangka berpikir dan logika, tidak seperti elemen yang dapat dilihat secara fisik”

2.1.2.1 Jenis-Jenis Perangkat Lunak

Menurut Biztech Proxis Group (2020), ada dua klasifikasi utama perangkat lunak yaitu :

1. Perangkat Lunak Sistem

Perangkat lunak sistem membantu pengguna dan perangkat keras untuk berfungsi dan berinteraksi satu sama lain. Pada dasarnya, ini adalah perangkat lunak untuk mengelola perilaku perangkat keras komputer sehingga dapat menyediakan fungsionalitas dasar yang dibutuhkan oleh pengguna. Dengan kata sederhana, kita dapat mengatakan bahwa perangkat lunak sistem adalah perantara atau lapisan tengah antara pengguna dan perangkat keras.



2. Perangkat Lunak Aplikasi

Perangkat Lunak Aplikasi, juga dikenal sebagai program pengguna akhir atau program produktivitas adalah perangkat lunak yang membantu pengguna dalam menyelesaikan tugas-tugas seperti melakukan penelitian *online*, mencatat catatan, menyetel alarm, merancang grafik, membuat log akun, melakukan perhitungan, atau bahkan bermain game.

2.1.3 Pengertian Data

Menurut Dantes et.al. (2018:3), “Data adalah *value* yang dapat merepresentasikan deskripsi dari suatu objek atau kejadian (*event*). Objek yang dimaksud dapat berupa manusia (pegawai, siswa, pembeli, pelanggan), barang dan hewan”.

Menurut Nawassyarif et al (2020:34), “Data adalah kenyataan yang menggambarkan suatu kejadian serta merupakan suatu bentuk yang masih mentah yang belum dapat bercerita banyak sehingga perlu diolah lebih lanjut melalui suatu model untuk menghasilkan informasi”.

2.1.4 Pengertian Basis Data (*Database*)

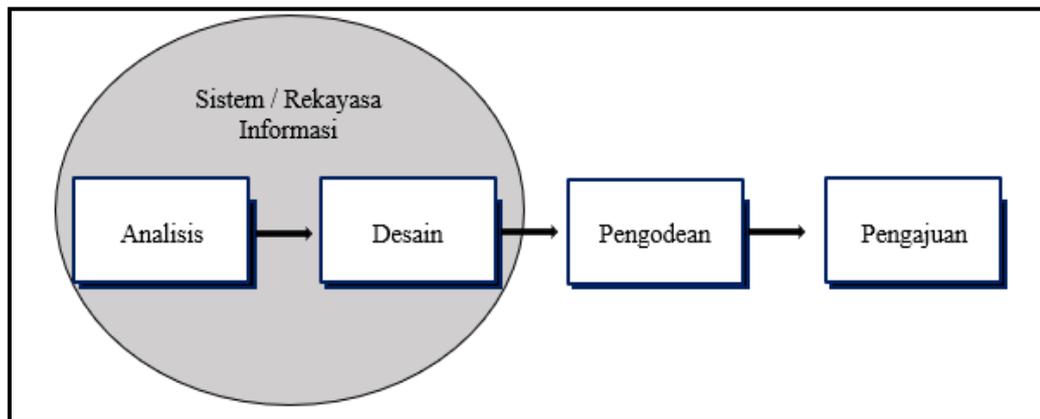
Menurut Yanti et al (2018), “*Database* adalah suatu susunan atau kumpulan catatan data yang tersimpan di dalam komputer. Hubungan antar entri dalam database dapat digunakan sebagai sumber informasi bagi pengguna. Sampai saat ini, masih banyak *record database* yang ditampilkan dalam bentuk teks sebagai informasi kepada pengguna. Ini adalah salah satu kerentanan yang dimiliki analisis kriptografi dalam mengakses, memanipulasi atau membocorkan dan mendistribusikan catatan basis data”.

Menurut Andaru (2018), “*Database* adalah kumpulan informasi yang disimpan secara sistematis di dalam komputer sehingga dapat dikendalikan oleh program komputer untuk mengambil informasi dari database. Istilah “basis data” berasal dari ilmu komputer”.



2.1.5 Metode Pengembangan Sistem *Waterfall*

Menurut Sukamto & Shalahuddin (2018:28), “Model SDLC air terjun (*waterfall*) sering juga disebut juga model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*)”. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).



Gambar 2.1 Model *Waterfall* menurut Sukamto dan Shalahuddin (2018:29)

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat di pahami perangkat lunak seperti apa yang di butuhkan oleh *user*.

b. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur, data, arsitektur perangkat lunak representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya, desain perangkat lunak yang di dihasilkan pada tahap ini juga perlu di dokumentasikan.

c. Pembuatan Kode Program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat



pada tahap desain.

d. Pengujian

Pengujian fokus pada perangkat lunak secara dari *logic* dan fungsional dan memastikan bahwa semua bagian sudah di uji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau pemeliharaan (*maintance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung atau pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak baru

2.2 Teori Judul

2.2.1 Pengertian Sistem

Menurut Kristanto (2018), “Sistem adalah kumpulan elemen – elemen yang saling terkait dan bekerja sama untuk memproses masukan (*input*) yang ditunjukkan kepada sistem tersebut dan mengolah masukan tersebut sampai menghasilkan keluaran (*output*) yang diinginkan”.

Menurut Erawati (2019), “Sistem adalah jaringan proses kerja yang saling terkait dan berkumpul guna untuk mencapai sebuah tujuan serta melakukan suatu kegiatan”.

2.2.2 Pengertian Informasi

Menurut Tukino (2020), ” Informasi adalah sebuah data yang dikelola menjadi sesuatu yang lebih bernilai tinggi bagi penerima guna untuk membantu membuat sebuah pengambilan keputusan”.

Menurut Sudjiman (2020), “ Informasi itu sendiri di definisikan sebagai data yang telah diubah menjadi sesuatu yang memiliki arti dan pernyataan bisa digunakan bagi pengguna akhir”.



2.2.3 Pengertian Layanan

Menurut Zeithaml et al (2019), “Layanan adalah tindakan, proses atau kinerja yang ditawarkan satu pihak kepada pihak lain. Hal ini melibatkan interaksi antara penyedia layanan dan penerima layanan, dan dapat mencakup produk fisik, pengetahuan, pengalaman, atau ide”.

Menurut Tjiptono (2020:6), “Layanan adalah setiap tindakan yang dapat ditawarkan oleh suatu pihak kepada pihak lain, yang pada dasarnya bersifat *intangible* (tidak berwujud fisik) dan tidak menghasilkan kepemilikan sesuatu. Produk jasa bisa berhubungan dengan produk fisik maupun tidak”.

2.2.4 Pengertian Kesehatan

Menurut Menurut Greenberg et al (2021), “Kesehatan adalah keadaan yang optimal dari tubuh, pikiran, dan jiwa yang memungkinkan seseorang untuk mencapai tujuan hidupnya dan berpartisipasi secara efektif dalam kehidupan sehari-hari. Kesehatan mencakup berbagai faktor, seperti lingkungan, perilaku, dan akses ke layanan kesehatan yang tepat”.

Kesehatan berdasarkan Undang-undang Nomor 36 Tahun 2009 bahwa kesehatan adalah “keadaan sehat baik secara fisik, mental spiritual maupun sosial yang memungkinkan setiap orang untuk hidup produktif secara sosial dan ekonomi. Sedangkan upaya kesehatan adalah setiap kegiatan untuk memelihara dan meningkatkan kesehatan yang dilakukan oleh pemerintah dan atau masyarakat”.

2.2.5 Pengertian Pasien

Berdasarkan Pasal 1 angka 10 Undang-Undang Republik Indonesia Nomor 29 Tahun 2004, “Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatannya untuk memperoleh pelayanan kesehatan yang diperlukan baik secara langsung maupun tidak langsung kepada dokter”.

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 069/MENKES/PER/I/2008, "Pasien adalah setiap orang yang melakukan konsultasi masalah kesehatan untuk memperoleh pelayanan kesehatan yang



diperlukan baik secara langsung maupun tidak langsung kepada dokter atau dokter gigi”.

2.2.6 Pengertian Klinik

Berdasarkan peraturan Menteri Kesehatan Republik Indonesia Nomor 9 Tahun 2014, “Klinik merupakan fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan medis dasar dan spesialistik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis”.

Menurut Peraturan Menteri Kesehatan Republik Indonesia Nomor 028/MENKES/PER/I/2011, “Klinik adalah fasilitas pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan yang menyediakan pelayanan medis dasar dan spesialistik, diselenggarakan oleh lebih dari satu jenis tenaga kesehatan dan dipimpin oleh seorang tenaga medis”.

2.2.7 Pengertian Website

Menurut Sa'ad (2020:3), “*Website* merupakan halaman-halaman yang berisi informasi yang dapat diakses oleh browser dan mampu memberikan informasi yang berguna bagi para pemakainya.

Menurut Maulia Usnaini, dkk (2021:37), “*Website* atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara atau gabungan dari semuanya , baik yang bersifat statis maupun dinamis yang berbentuk satu rangkaian bangunan yang saling terkait, yang masing – masing dihubungkan dari jaringan – jaringan halaman.



2.3 Teori Khusus

2.3.1 *Unified Modeling Language (UML)*



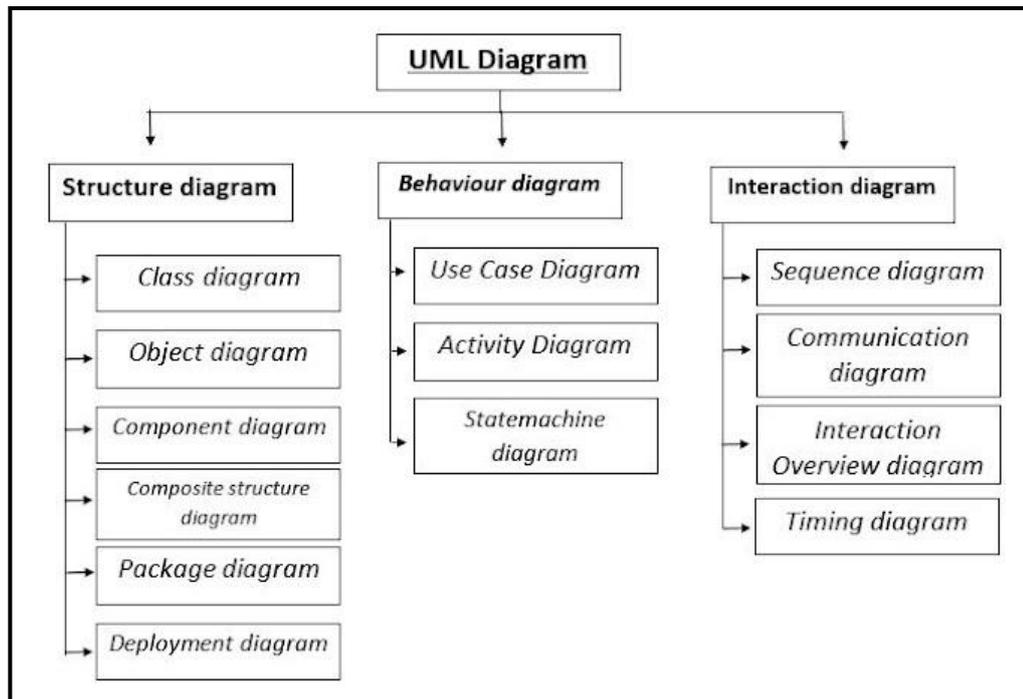
Gambar 2.2 Logo UML

Sumber : www.wikipedia.org

Menurut Rosa A.S dan M. Shalahuddin (2018:137), “UML adalah bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung. UML hanya sebagai pemodelan”.

Menurut Huda (2019:110), ”UML adalah bahasa pemodelan standar yang terdiri dari seperangkat diagram ter-integrasi untuk membantu pengembangan sistem dan perangkat lunak menentukan, membuat gambaran, mengembangkan, dan mendokumentasikan artefak sistem perangkat lunak, serta untuk pemodelan bisnis dan sistem non-perangkat lunak lainnya”.

Unified Modeling Language (UML) terdiri dari 13 macam diagram dan dikelompokkan menjadi 3 kategori sebagai berikut:



Gambar 2.3 Kategori dan Diagram UML

Sumber : www.ansoriweb.com

Berikut penjelasan singkat dari pembagian kategori pada diagram UML menurut Rosa A.S dan M. Shalahudin (2018:137) :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

2.3.2 Jenis-Jenis Diagram *Unified Modeling Language (UML)*

2.3.2.1 Pengertian *Use Case Diagram*

Menurut Tohari dalam Tabrani dan Aghniya (2019:46), “*Use case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”.



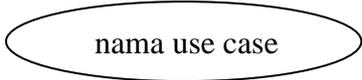
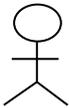
Menurut Pratama (2019), “*Use case diagram* adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detail tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem. Di dalam *use case* ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat”.

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

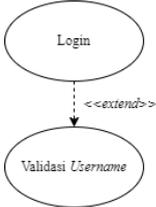
- a. **Aktor** merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas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

Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

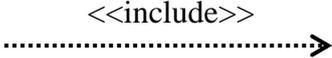
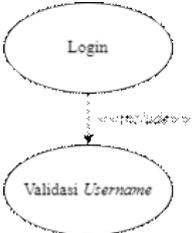
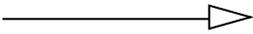
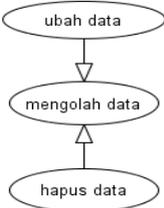
Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
1.	<p><i>Use Case</i></p>  <p>nama use case</p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i> .
2.	<p>Aktor / <i>actor</i></p>  <p>nama aktor</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi

Lanjutan Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
		walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i> .
3.	<p>Aasosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan <i>actor</i> .
4.	<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p>  <p>arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>
5.	<p><i>Include / uses</i></p>	Relasi tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan

Lanjutan Tabel 2.1 Simbol-simbol *Use Case Diagram*

No.	Simbol	Deskripsi
		<p>memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <p><i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, <i>include</i> juga berarti relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan tugasnya, misalnya pada kasus berikut:</p> 
6.		<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p> <p>misalnya:</p> <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p> 

Sumber: Rosa dan Shalahuddin, 2018.

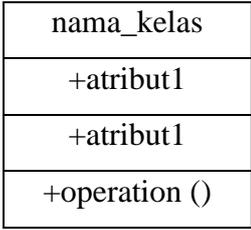
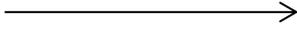


2.3.2.2 Pengertian *Class Diagram*

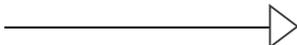
Menurut Sukamto dan Shalahuddin (2018:141), menjelaskan bahwa “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

Menurut Wira et al (2019), “*Class diagram* ialah menjelaskan secara garis besar mengenai kelas-kelas perancangan sistem dari sudut pandang struktur sistem yang dapat memperjelas fungsi-fungsinya. Atribut dan operasi merupakan bagian dari *class diagram* yang dapat memberi gambaran hubungan antara perancangan dan perangkat lunaknya sehingga sesuai dengan pembuatan programnya”.

Tabel 2.2 Simbol-simbol *Class Diagram*.

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem.
2.	<p>Antarmuka / <i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi <i>antarclass</i> dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

Lanjutan Tabel 2.2 Simbol-simbol *Class Diagram*.

No.	Simbol	Deskripsi
5.	Generalisasi / <i>generalization</i> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
6.	Ketergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>).

Sumber : Rosa dan Shalahuddin, 2018.

2.3.2.3 Pengertian *Activity Diagram*

Menurut Sukanto dan Shalahuddin (2018:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

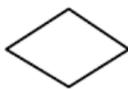
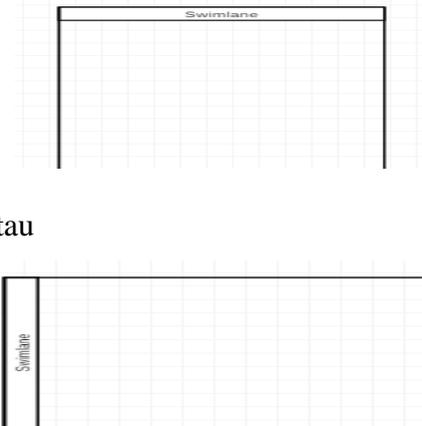
Menurut Tohari dalam Tabrani dan Aghniya (2019:45), “*Activity diagram* memodelkan *workflow* proses bisnis dan urutan aktifitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas lainnya atau dari aktifitas ke status”.

Simbol-simbol pada *Activity Diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.

Lanjutan Tabel 2.3 Simbol-simbol *Activity Diagram*.

No.	Simbol	Deskripsi
3.	Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir, dan bagaimana objek dibentuk dan dihancurkan.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Rosa dan Shalahuddin (2018:162).

2.3.2.4 Pengertian *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165), menjelaskan bahwa “Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendekati waktu hidup objek dan message yang dikirimkan dan diterima antar objek”.

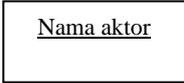
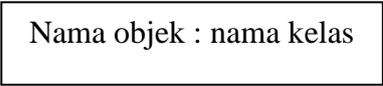
Menurut Tohari dalam Tabrani dan Aghniya (2019:46), menyimpulkan bahwa, “*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam



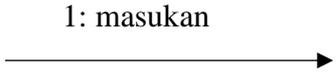
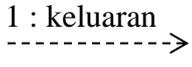
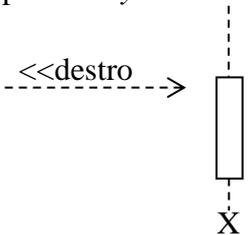
urutan waktu”.

Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Keterangan
1.	<p>Aktor</p>  <p>Atau</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi dibuat sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi <i>actor</i> adalah gambar orang, tapi belum tentu merupakan orang, tapi <i>actor</i> adalah gambar orang, tapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu <i>object</i> atau menyatakan keberadaan sebuah <i>object</i>, <i>lifeline</i> biasanya garis putus <i>vertical</i> yang ditarik dari sebuah objek.</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>
4.	<p>Waktu Aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.</p>

Lanjutan Tabel 2.4 Simbol-simbol *Sequence Diagram*

No.	Simbol	Keterangan
5.	Pesan tipe <i>send</i> 	Menyatakan bahwa suatu objek mengirimkan data /masukan /informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
6.	Pesan tipe <i>return</i> 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
7.	Pesan tipe <i>destroy</i> 	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah yang mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber : Rosa dan Shalahuddin, 2018.

2.3.2.5 Pengertian *Entity Relationship Diagram (ERD)*

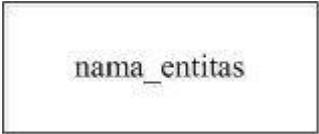
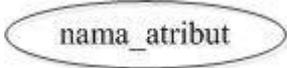
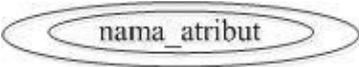
Menurut Sukamto dan Shalahudin dalam Apriliah, dkk (2018:30), menyimpulkan bahwa “*Entity Relationship Diagram (ERD)* adalah sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data”.

Menurut Dawan (2019), “*ERD* atau *Entity Relationship Diagram* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh sistem analis dalam tahap analisis persyaratan proyek pengembangan sistem”.



Simbol- simbol *Entity Relationship Diagram* adalah sebagai berikut:

Tabel 2.5 Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Deskripsi
1.	Entitas/ Entity 	Entitas merupakan data inti yang akan disimpan, bakal tabel pada basis data, benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer, penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel.
2.	Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas.
3.	Atribut Kunci Primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan, biasanya berupa id, kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama).
4.	Atribut Multinilai/ <i>Multi Value</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu.

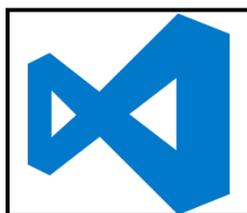
Lanjutan Tabel 2.5 Simbol-simbol *Entity Relationship Diagram*

No	Simbol	Deskripsi
5.	Relasi 	Relasi yang menghubungkan antar entitas, biasanya diawali dengan kata kerja.
6.	Assosiasi/ <i>association</i> 	Penghubung antara relasi dan entitas dimana di kedua ujungnya memiliki <i>multiplicity</i> kemungkinan jumlah pemakaian kemungkinan jumlah maksimum keterhubungan antara entitas satu dengan entitas yang lain disebut dengan kardinalitas.

Sumber : Sukamto dan Shalahuddin (2018:50-51).

2.4 Teori Program

2.4.1 Pengertian *Visual Studio Code (VS Code)*



Gambar 2.4 Logo *Visual Studio Code (VS Code)*

Sumber : www.wikipedia.org

Menurut Salamah Ummy Gusti (2021:1), “*Visual Studio Code (VS Code)* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi multiplatform, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan



bantuan plugin yang dapat dipasang via *marketplace Visual Studio Code* (seperti C++, C#, Python, Go, Java, dst)”.

Menurut Nurhayati (2022:34), “*Visual Studio Code (VS Code)* adalah sebuah teks editor ringan dan handal yang dibuat oleh Microsoft untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux, Mac, dan Windows*”.

2.4.2 Pengertian *Hypertext Markup Language (HTML)*



Gambar 2.5 Logo *Hyper Text Mark Up Language*

Sumber : www.wikipedia.org

Menurut Enterprise (2018:21), “HTML adalah Bahasa *markup (markup language)* seperti yang ada di dalam singkatan HTML itu sendiri. Itu artinya, HTML adalah bahasa struktur untuk menandai bagian-bagian dari sebuah halaman”.

Menurut Andri (2018), “HTML (*Hypertext Mark up Langunge*) adalah semacam bahasa pengkodean bukan sebagai bahasa pemrograman dan juga sekumpulan perintah yang terformat yang digunakan untuk membuat halaman dokumen web *Hypertext*. HTML berperan sebagai penyusun struktur halaman website yang menempatkan setiap elemen website sesuai layout yang diinginkan”.



2.4.3 Pengertian *Cascading Style Sheets (CSS)*



Gambar 2.6 Logo *Cascading Style Sheets*

Sumber : www.wikipedia.org

Menurut Abdulloh (2018:45), “CSS adalah singkatan dari *Cascading Style Sheet* yaitu dokumen web yang berfungsi mengatur elemen HTML dengan berbagai property yang tersedia sehingga dapat tampil dengan berbagai gaya yang diinginkan”.

Menurut Lewenusa (2019:1), “CSS (*Cascading Style Sheet*) secara sederhana adalah sebuah metode yang digunakan untuk mempersingkat penulisan tag HTML, seperti *font, color, text* dan tabel menjadi lebih ringkas sehingga tidak terjadi pengulangan penulisan. CSS digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format berbeda”.

2.4.4 Pengertian *Hypertext Preprocessor (PHP)*



Gambar 2.7 Logo *Hypertext Preprocessor*

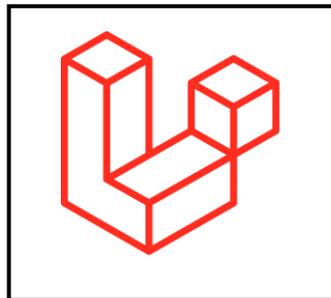
Sumber : www.wikipedia.org



Menurut Jubilee Enterprise (2018:1), “PHP adalah pemrograman yang digunakan untuk membuat website dinamis dan Interaktif. Dinamis artinya *website* tersebut bisa berubah – ubah tampilan dan konten sesuai kondisi tertentu. Sebagai contoh , PHP bisa menampilkan tanggal dan hari saat ini secara berganti-ganti di dalam sebuah *website*. Interaktif artinya, PHP dapat memberi *feedback* bagi *user* (misalnya menampilkan hasil pencarian produk)”.

Menurut Supono (2018:3), “*PHP (HyperText PreProcessor)* merupakan suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat di mengerti oleh komputer yang bersifat *server- side* yang dapat di tambahkan ke dalam HTML”.

2.4.5 Pengertian *Laravel*



Gambar 2.8 Logo *Laravel*

Sumber : www.wikipedia.org

Menurut Abdulloh (2018), ” *Laravel* adalah *framework PHP* dengan kode terbuka (*open source*) dengan desain MVC (*Model-View-Controller*) yang digunakan untuk membangun aplikasi *website*”.

Menurut Supardi dan Sulaeman (2019:1), “*Laravel* adalah sebuah *framework PHP* yang dirilis di bawah lisensi MIT dan dibangun dengan konsep MVC (*Model View Controller*). *Laravel* merupakan pengembangan *website* berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi yang menyediakan sintaks yang espresif, jelas dan menghemat waktu”.



2.4.6 Pengertian Javascript



Gambar 2.9 Logo Javascript

Sumber : www.wikipedia.org

JavaScript merupakan bahasa pemrograman tingkat dinamis yang tinggi dan serta merupakan merupakan teknologi inti *World Wide Web* selain HTML dan CSS. JavaScript sangat membantu dalam pembuatan halaman web interaktif dan juga merupakan bagian aplikasi web yang esensial. JavaScript itu sendiri dikembangkan oleh Brendan Eich yang berasal dari Netscape dengan nama Mocha, yang kemudian dikembangkan Kembali menjadi *JavaScript* (Pahlevi et al., 2018).

Menurut Kadir dan Triwahyuni (2013:325) “*JavaScript* adalah bahasa pemrograman yang biasa diletakkan bersama kode HTML untuk menentukan suatu tindakan”.

2.4.7 Pengertian MySQL



Gambar 2.10 Logo MySQL

Sumber : www.wikipedia.org

Menurut Subagia (2018:67), “*MySQL* adalah *software database open source* yang sering digunakan untuk mengolah basis data yang menggunakan bahasa *SQL*”.

Menurut Jubilee Enterprise (2018:2), “*MySQL* adalah *server* yang melayani *database*. Untuk membuat dan mengolah *database*, kita dapat mempelajari



pemrograman khusus yang disebut *query* (perintah) *SQL*. *Database* itu sendiri dibutuhkan jika kita ingin menginput data user menggunakan form HTML untuk kemudian diolah *PHP* agar bisa disimpan ke dalam *database MySQL*”.

2.4.8 Pengertian *XAMPP*



Gambar 2.11 Logo *Xampp*

Sumber : www.wikipedia.org

Menurut Jubilee Enterprise (2018:27), “*XAMPP* adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *mysql* dan *phpmyadmin*. *XAMPP* adalah singkatan dari *X*, *Apache Server*, *MySQL*, dan *Phyton*. Huruf *X* di depan menandakan *XAMPP* bisa diinstal di berbagai *operating system*. *XAMPP* dapat diinstal pada *Windows*, *Linux*, *MacOS*, dan *Solaris*”.

Menurut Mawaddah dan Fauzi (2018), “*XAMPP* adalah *software* yang di dalamnya terdapat *server MySQL* dan didukung oleh *PHP* sebagai bahasa pemrograman untuk membuat *website* dinamis serta terdapat *web server apache* yang dapat dijalankan di beberapa platform seperti OS *X*, *Windows*, *Linux*, *Mac*, dan *Solaris*”.