



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Menurut Ansori dan Yulmaini (2019:56), “Komputer adalah alat elektronik yang dapat menerima input, mengolah input, memberikan informasi, menggunakan suatu program yang tersimpan di memori komputer, dapat menyimpan program dan hasil pengolahan, serta bekerja secara otomatis”.

Menurut Hanafri, dkk (2019:88), “Komputer adalah sebuah alat elektronik otomatis yang digunakan untuk mengolah data sesuai perintah secara cepat dan tepat, serta menerima dan menyimpan data berdasarkan perintah yang telah tersimpan di dalam memorinya”.

Menurut Yusuf, dkk (2022:28), “Komputer adalah alat elektronis otomatis yang dapat menghitung atau mengolah data secara cermat menurut yang diinstruksikan dan memberikan hasil pengolahan, biasanya terdiri atas unit pemasukan, unit pengeluaran, unit penyimpanan, serta unit pengontrolan”.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut Ali (dalam Sudarso, 2022:5), “Perangkat lunak aplikasi merupakan program yang ditujukan untuk menyelesaikan suatu permasalahan dalam aplikasi yang tertentu yang sudah dibuat oleh pabrik pembuat perangkat lunak aplikasi. Program aplikasi dibuat dengan menggunakan perangkat lunak bahasa (*language software*)”.

Menurut Indra (dalam Sudarso, 2022:5), “*Software* adalah sekumpulan data elektronik yang disimpan dan diatur oleh komputer. Data elektronik yang disimpan dapat berupa program atau instruksi yang akan menjalankan perintah. *Software* dapat diartikan juga sebagai segala jenis program yang digunakan untuk pengoperasian komputer dan peralatannya”.



Menurut Utami (dalam Sudarso, 2022:5), “Perangkat lunak (software) adalah program yang berisi kumpulan instruksi untuk melakukan proses pengolahan data. Software sebagai penghubung antara manusia sebagai pengguna dengan perangkat keras komputer, berfungsi menerjemahkan Bahasa manusia ke dalam Bahasa mesin sehingga perangkat keras komputer memahami keinginan pengguna dan menjalankan intruksi yang diberikan dan selanjutnya memberikan hasil yang diinginkan oleh manusia tersebut”.

2.1.3 Pengertian Data

Menurut Gunadi dan Widiyanto (dikutip Wahono dan Ali, 2021:227), “Data adalah bahan mentah yang perlu dilakukan pengolahan sehingga menghasilkan informasi atau keterangan, baik kualitatif maupun kuantitatif yang menunjukkan fakta sehingga dapat memberi manfaat bagi peneliti atau memberi gambaran kepada peneliti tentang kondisi atau suatu keadaan. Sedangkan informasi adalah sekumpulan data yang sudah diolah sehingga menghasilkan suatu analisa untuk digunakan oleh pihak yang membutuhkan”.

Menurut Dewi (2020:226), “Data adalah deskripsi dari suatu benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan dalam jumlah yang besar dan belum diolah menjadi sebuah informasi”.

Menurut Turban (dikutip Dewi, 2020:226), “Data adalah deskripsi dasar dari benda, peristiwa, aktivitas dan transaksi yang direkam, dikelompokkan, dan disimpan tetapi belum terorganisir untuk menyampaikan arti tertentu”.

2.1.4 Pengertian Internet

Menurut Ahdan dan Susanto (dalam Priyanga dan Febriyo, 2022:4), “Internet adalah organisasi dari berbagai jaringan komputer yang menghubungkan semua komputer, mulai dari super komputer berukuran besar di berbagai badan pemerintahan, server perusahaan, hingga computer personal di rumah”.



Menurut Putri (2020:93), “Internet merupakan jaringan komputer yang terdiri dari berbagai perangkat komputer yang terhubung melalui protokol untuk memberikan informasi dan internet menyediakan sumber-sumber yang dapat di akses oleh manusia di seluruh dunia”.

Menurut Susana, dkk (2022:2), “Internet adalah sebuah jaringan komunikasi public dan global yang menyediakan koneksi langsung kepada siapa saja melalui Local area network (LAN) dan Internet Service Provider (ISP) pengertian internet dalam arti sederhana adalah komunikasi antara konsumen, marketer, dan jutaan organisasi lainnya”.

2.1.5 Pengertian Perangkat Keras (*Hardware*)

Menurut Ansori (2019:56), “Perangkat keras komputer (*hardware*) adalah semua bagian fisik komputer, dan dibedakan dengan data yang berada di dalamnya atau yang beroperasi di dalamnya, seperti: *Motherboard, Power Supply, Prosesor, RAM, Hardisk, CD Drive, Battery CMOS, VGA Card, Sound Card, dll*”.

Menurut Rizky Dhanta dalam Putri (dikutip Jatmiko, 2022:187), “Hardware adalah perangkat komputer yang terdiri atas susunan komponen-komponen elektronik berbentuk fisik (berupa benda)”.

Menurut Putri, dkk (2022:69), “Perangkat keras komputer yaitu suatu benda atau perangkat yang memiliki fungsi buat melakukan proses tuang data (input) proses pengolahan data (process) dan keluaran (hasil)”.

2.1.6 Pengertian Basis Data (*Database*)

Menurut Sudarso (2022:3), “Basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (perusahaan/organisasi) bertujuan agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”.



Menurut Hardiansyah dan Dewi (2020:223), “Basis data merupakan data yang dapat didesain dan berintegrasi sehingga dapat memenuhi kebutuhan user dalam perusahaan atau organisasi”.

Menurut Ali (dalam Sudarso, 2022:3), “Basis data (database) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan disimpan luar komputer (external memory) dan digunakan perangkat lunak tertentu untuk memanipulasinya. Database merupakan salah satu komponen yang penting di SI, karena berfungsi sebagai basis penyedia informasi bagi para pemakainya. Sistem basis data (database system) ini adalah suatu SI yang mengintegrasikan kumpulan data yang saling berhubungan satu dengan lainnya dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam dalam suatu organisasi”.

2.1.7 Metode Pengembangan Sistem *Waterfall*

Penelitian Laporan Akhir ini menggunakan metode pengembangan sistem *Waterfall*. Menurut Novitasari (dalam Kurniawan., dkk, 2020:161), “Metode *waterfall* adalah hal yang menggambarkan pendekatan secara sistematis dan juga berurutan (*step by step*) pada sebuah pengembangan perangkat lunak. Tahapan dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan yaitu planning, permodelan, konstruksi, sebuah sistem dan penyerahan sistem kepada pengguna, dukungan pada perangkat lunak lengkap yang dihasilkan”.

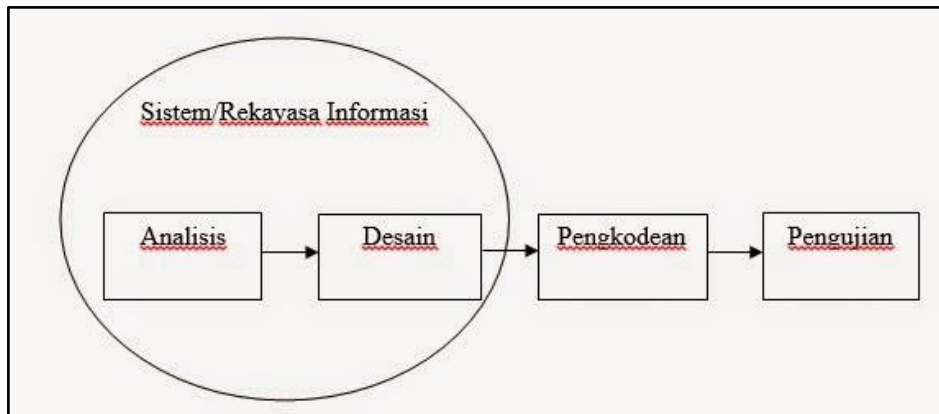
Menurut Ibrahim, dkk (dikutip Perdana dan Rohman, 2022:195), “Metode pengembangan sistem *waterfall* merupakan metode pengembangan perangkat lunak yang sistematis dan sekuensial, mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan”.

Menurut Sukamto dan Shalahuddin (2016:28), mengungkapkan bahwa “Model SDLC air terjun (*waterfall*) sering juga disebut 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*)”.

Berikut adalah gambar model air terjun:



Gambar 2.1 Ilustrasi Model *Waterfall*

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspezifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk didokumentasikan.

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 diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.



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 segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

e. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

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 yang sudah ada, tapi tidak untuk membuat perangkat lunak baru.

2.1.8 Studi Kelayakan

Menurut Subagyo (dikutip Anugrahani dan Mutia, 2023:2), “Studi kelayakan merupakan sebuah riset secara mendalam terhadap suatu ide bisnis tentang kelayakan pada ide tersebut”.

Menurut Anugrahani dan Mutia Edwy (2023:2), “Studi kelayakan merupakan bahan pertimbangan bagi pengusaha dalam pengambilan keputusan. Keputusan ini akan digunakan untuk menerima atau menolak suatu gagasan atau proyek yang telah direncanakan atau sudah dilaksanakan”.

Menurut Ibrahim (dikutip Nurlita Merin., dkk, 2022:221), “Studi kelayakan (*feasibility study*) merupakan bahan pertimbangan untuk mengambil suatu



keputusan, dari suatu gagasan/proyek yang sedang atau akan direncanakan apakah bisa diterima atau ditolak”.

2.1.9 Kebutuhan Fungsional

Menurut Apriyogi dan Hamidy (2022:20), menjelaskan bahwa “Analisis kebutuhan fungsional dilakukan untuk mengetahui proses-proses apa saja yang dilakukan oleh sistem”.

Menurut Nurkholis, dkk (2022:24), “Kebutuhan fungsional adalah kebutuhan yang berisi proses-proses apa saja yang nantinya dapat dilakukan dan informasi apa yang dihasilkan oleh sistem, mencakup bagaimana sistem harus bereaksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu”.

Menurut Harahap, dkk (2022:196), “Kebutuhan fungsional merupakan suatu kebutuhan yang berisi mengenai proses yang disediakan oleh sistem”.

2.1.10 Kebutuhan Non Fungsional

Menurut Apriyogi dan Hamidy (2022:2), menjelaskan bahwa “Analisis kebutuhan non fungsional dilakukan untuk mengetahui perilaku yang dimiliki oleh sistem. Spesifikasi kebutuhan non fungsional melibatkan analisis perangkat keras/*hardware*, analisis perangkat lunak/*software* dan analisis keamanan/*security*”.

Menurut Nurkholis, dkk (2022:24), “Kebutuhan non-fungsional merupakan dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Kebutuhan ini juga menentukan spesifikasi masukan yang dibutuhkan sistem. Pada analisis kebutuhan sistem non-fungsional ini dijelaskan mengenai analisis perangkat keras (*hardware*), perangkat lunak (*software*), dan pengguna (*user*)”.

Menurut Harahap, dkk (2022:196), “Kebutuhan non-fungsional adalah suatu kebutuhan yang tidak berpengaruh terhadap sistem sehingga sistem dapat beroperasi dan digunakan tanpa kebutuhan tersebut”.



2.2 Teori Judul

2.2.1 Pengertian Aplikasi

Menurut Harip Santoso (dikutip Syabania dan Rosmawarni, 2021:45), “Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya”.

Menurut Huda dan Priyatna (2019:82), “Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia”.

Menurut Andi (dikutip Danny., dkk, 2022:2), “Aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya aplikasi merupakan suatu perangkat komputer yang siap pakai bagi user”.

2.2.2 Pengertian Surat Perintah Perjalanan Dinas

Menurut Putri, dkk (2021:31), “Surat perintah perjalanan dinas adalah surat ini berfungsi untuk menerangkan perihal penugasan seseorang pada sebuah instansi maupun kantor untuk melaksanakan tugas di suatu tempat tertentu dengan biaya kantor dan dilakukan dalam waktu yang sudah ditentukan”.

Menurut Novianti dan Amin (2021:2720), “Surat perintah perjalanan dinas (SPPD) merupakan surat perintah yang dibutuhkan oleh seorang pegawai dalam melaksanakan tugas kerja dan harus pergi ke daerah lain (luar kota)”.

Menurut Sapitri, dkk (2022:97), “Surat Perintah Perjalanan Dinas (SPPD) merupakan dokumen yang dibuat ketika seorang pimpinan dan anggota yang bekerja pada sebuah instansi atau perusahaan tersebut akan melakukan pekerjaan diluar kantor dengan jarak yang telah ditentukan”.

2.2.3 Pengertian Web View

Menurut Labrinidis dan Roussopoulos (dikutip Putra., dkk, 2021:374), “Dalam teknologi pengembangan aplikasi android terdapat fitur bernama WebView



yang merupakan teknologi halaman web yang dibuat secara otomatis dari data dasar yang biasanya disimpan dalam DBMS ke dalam aplikasi android”.

Satriadi, dkk (2021:8), mengungkapkan bahwa android webview merupakan aplikasi android yang dapat menampilkan halaman website tanpa harus membuka web browser.

Menurut Yunita, dkk (2023:173), “Webview dapat didefinisikan sebagai komponen aplikasi android studio yang memungkinkan aplikasi android menampilkan suatu informasi dari web secara langsung”.

2.2.4 Pengertian Aplikasi Surat Perintah Perjalanan Dinas pada Badan Kepegawaian dan Pengembangan Sumber Daya Manusia (BKPSDM) Kota Palembang Berbasis *Web View*

Aplikasi Surat Perintah Perjalanan Dinas pada Badan Kepegawaian dan Pengembangan Sumber Daya Manusia (BKPSDM) Kota Palembang Berbasis *Web View* adalah sebuah aplikasi yang terdiri dari beberapa aplikasi yaitu aplikasi pembuatan NPPD, SPPD, SPT, kwitansi, serta pengelolaan laporan perjalanan dinas, yang mana aplikasi ini dapat membantu pihak Badan Kepegawaian dan Pengembangan Sumber Daya Manusia (BKPSDM) Kota Palembang dalam pengelolaan data perjalanan dinas agar lebih efektif dan efisien dalam pelaksanaannya.

2.3 Teori Khusus

2.3.1 Kamus Data

Menurut Sukamto dan Shalahuddin (2016:73), mengungkapkan bahwa “Kamus Data (*data dictionary*) dipergunakan untuk memperjelas aliran data yang digambarkan pada DFD yang berupa kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan)”.



Menurut Puspita dan Anggadini (dikutip Prastianto dan Rostiani, 2020:26), “Kamus data (*Data Dictionary*) adalah suatu penjelasan tertulis mengenai data yang ada di dalam *database*. Kamus data pertama berbasis dokumen, kamus data itu tersimpan berbentuk *hard copy* dengan mencatat semua penjelasan data dalam bentuk tercetak”.

Menurut Anderha dan Maskar (dikutip Pratiwi, 2022:6), “Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi”.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut:

Tabel 2.1 Simbol-Simbol pada Kamus Data

No.	Simbol	Keterangan
1	=	disusun atau terdiri dari
2	+	dan
3	[]	baik ...atau...
4	{ }”	n kali diulang/ bernilai banyak
5	()	data opsional
6	*...*	batas komentar

(Sumber: Sukamto dan Shalahuddin (2016:74))

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur.



2.3.2 *Unified Modeling Language (UML)*

2.3.2.1 *Pengertian Unified Modeling Language (UML)*



Gambar 2.2 Logo UML

Sumber: Google

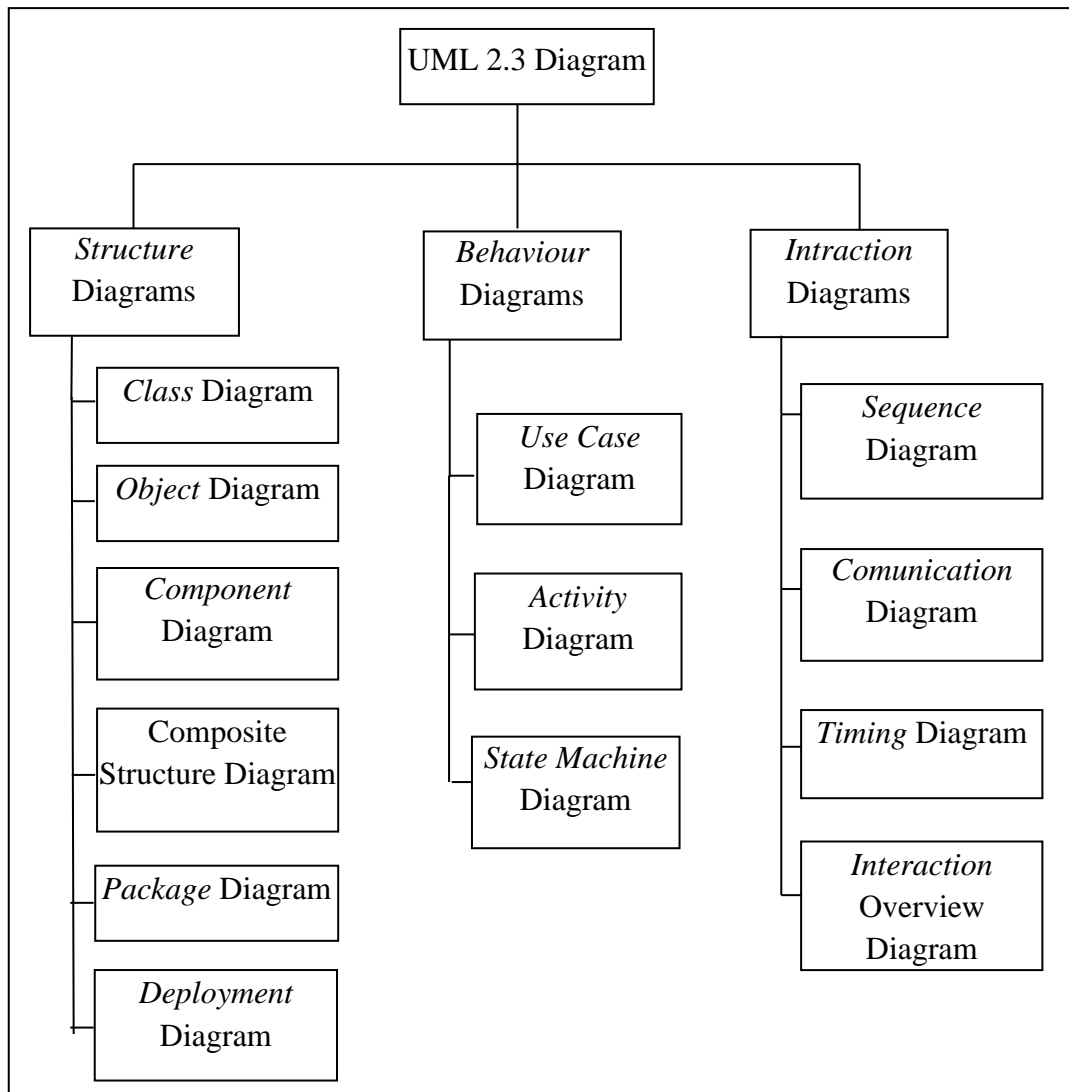
Menurut Sukamto dan Shalahuddin (2016:133), “*UML (Unified Modelling Language)* adalah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Menurut Feby, dkk (2022:15), “UML adalah bahasa untuk menspesifikasi, memvisualisasi, membangun dan mendokumentasikan artifacts (bagian dari informasi yang digunakan untuk dihasilkan oleh proses pembuatan perangkat lunak, artifact tersebut dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya”.

Menurut Wulandari dan Nurmiati (2022:81), “UML (Unified Modeling Language) ialah bahasa pemodelan buat sistem atau software yang berparadigma berorientasi objek”.

2.3.2.2 *Kategori Unified Modeling Language (UML)*

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



Gambar 2.3 Kategori dan Macam-Macam Diagram

(Sumber: Sukamto dan Shalahuddin (2016:140))

Berikut penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141):

- 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.

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

2.3.3.1 Pengertian *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2016:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

Menurut Tohari dalam Tabrani dan Aghniya (dikutip Apriliah., dkk, 2019:83), menyimpulkan bahwa “*Use case* adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor”.

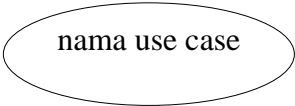
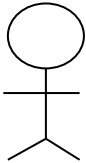

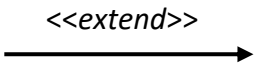
Menurut Wulandari dan Nurmiati (2022:81), “*Use Case Diagram* ialah diagram yang wajib dirancang pertama kali saat pemodelan software berorientasi di objek yang dilakukan”.

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

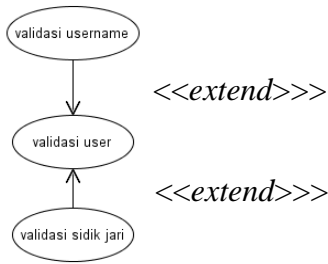
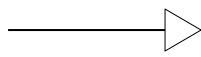
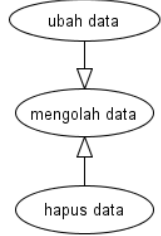
- a. **Aktor** merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. *Use case* merupakan fungsionalitas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

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

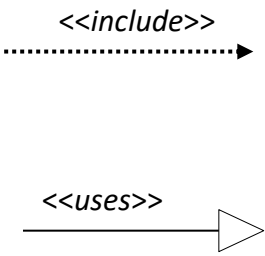
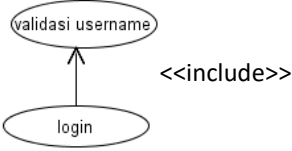
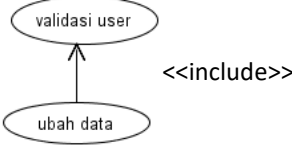
Tabel 2.2 Simbol-simbol *Use Case Diagram*

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 frase nama <i>use case</i> .
2.	<p>Aktor / actor</p>  <p>nama aktor nama_interface</p>	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; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.
3.	<p>Asosiasi / association</p> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> yang memiliki interaksi dengan actor.
4.	<p>Ekstensi / extend</p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
		<p>pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</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>Generalisasi/<i>generalization</i></p> 	<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, misalnya:</p> 

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case* Diagram

No	Simbol	Deskripsi
		arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).
6.	Menggunakan <i>/include /uses</i> 	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 fungsinya atau sebagai syarat dijalankan <i>use case</i> ini ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : a. Include berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  b. Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut: 



Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*

No	Simbol	Deskripsi
		Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

(Sumber: Sukamto dan Shalahuddin (2016:156-158))

2.3.3.2 Pengertian *Class Diagram*

Menurut Sukamto dan Shalahuddin (2013:141), “*Class Diagram* atau diagram kelas 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 Tohari (dikutip Tabrani dan Aghniya, 2019:43), mendefinisikan bahwa “Kelas (*class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek”.

Menurut Wulandari dan Nurmiati (2022:81), “*Diagram Class* ialah mendeskripsikan struktur sistem asal segi pendefinisian kelas-kelas yang akan dirancang untuk menciptakan sistem”.

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

Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
1.	<div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;"> <div style="border-bottom: 1px solid black; padding: 2px 5px;">nama_kelas</div> <div style="border-bottom: 1px solid black; padding: 2px 5px;">+ atribut</div> <div style="padding: 2px 5px;">+ operasi ()</div> </div>	Kelas pada struktur sistem.

Lanjutan Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Deskripsi
2.	Antarmuka / <i>interface</i>  nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	Asosiasi / <i>association</i> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antarkelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua- bagian (<i>whole-part</i>).

(Sumber: Sukamto dan Shalahuddin (2016:146-147))

2.3.3.3 Pengertian *Activity Diagram*

Menurut Sukamto dan Shalahuddin (2016:161), “Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem”.



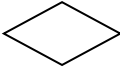




Menurut Tohari (dikutip Tabrani dan Aghniya, 2019:42), mendefinisikan bahwa “*Activity diagram* memodelkan *workflow* proses bisnis dan urutan aktifitas dalam sebuah proses. Diagram ini sangat mirip dengan *flowchart* karena memodelkan *workflow* dari suatu aktifitas lainnya atau dari aktifitas ke status”.

Menurut Wulandari dan Nurmiati (2022:81), “*Activity Diagram* ialah diagram yang dapat menampilkan prosedur logika dan proses bisnis dalam sebuah sistem informasi”.

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

Tabel 2.4 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.
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.
6.	<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Lanjutan Tabel 2.4 Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 2px; text-align: center;">Nama swimlane</div> </div> <p style="text-align: center;">atau</p> <div style="border: 1px solid black; padding: 5px;"> <div style="border: 1px solid black; padding: 2px; text-align: center; writing-mode: vertical-rl; transform: rotate(180deg);">Nama swimlane</div> </div>	

(Sumber: Sukamto dan Shalahuddin (2016:162-163))

2.3.3.4 Pengertian *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2016:165), "Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat *scenario* yang ada pada *use case*".

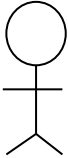
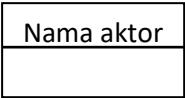

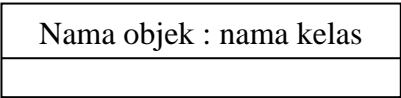
Menurut Tohari (dikutip Tabrani dan Aghniya, 2019:43), menyimpulkan bahwa "*Sequence diagram* menggambarkan interaksi antara sejumlah objek dalam urutan waktu".

Menurut Wulandari dan Nurmiati (2022:81), "*Diagram Sequence* ialah diagram yang dirancang buat mengetahui alur asal hubungan antar objek".


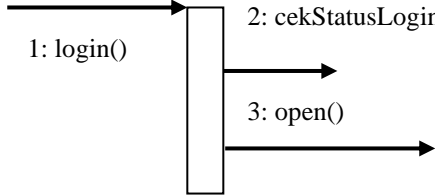
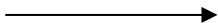
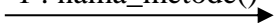
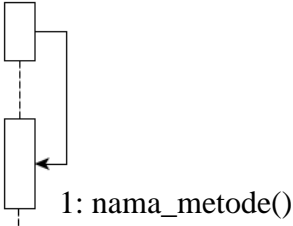


Simbol-simbol pada *Sequence Diagram* adalah sebagai berikut:

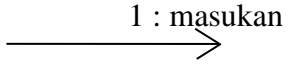
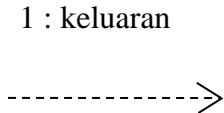
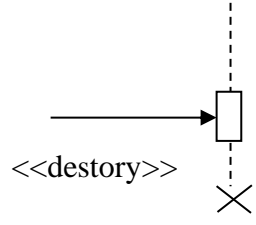
Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>tanpa waktu aktif</p>	<p>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; biasanya dinyatakan menggunakan kata benda di awal frase nama <i>actor</i>.</p>
2.	<p>Garis hidup/ <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek.</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan.</p>

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
4.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnva  maka cek Status Login () dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif
5.	pesan tipe <i>create</i> <<create>> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6.	Pesan tipe <i>call</i> 1 : nama_metode() 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No	Simbol	Deskripsi
		operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.
7.	<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8.	<p>Pesan tipe <i>return</i></p> 	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.
9.	<p>Pesan tipe <i>destrory</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destrory</i> .

(Sumber: Sukamto dan Shalahuddin (2016:165-167))



2.4 Teori Program

2.4.1 Pengertian HTML



Gambar 2.4 Logo HTML

Sumber: Google

Menurut Yuda dan Faris (2023:43), “HTML (*Hypertext Markup Language*) yaitu sebuah intruksi pemrograman yang dituliskan dalam bentuk baris-baris kode, yang dikenal dengan istilah Tag. *Script* HTML dikenal hanya mampu menghasilkan sebuah *website* dengan *output* yang sederhana, baik dari segi tampilan maupun kapabilitas yang dimilikinya. Maintenance terhadap halaman web (seperti menambah dan meremajakan isi *website*), hanya dapat dilakukan dengan mengubah *script* HTML yang ada dalam program”.

Menurut Zainy, dkk (2022:336), “HTML adalah salah satu Bahasa pemrograman yang digunakan dalam pembuatan sebuah *website*”.

Menurut Rio dan Marsehan (2023:45), “*Hypertext Mark Up Language* (HTML) bukan merupakan bahasa pemrograman melainkan markup yang menjadi kerangka utama dari tampilan sebuah *website*”.



2.4.2 Pengertian JavaScript



Gambar 2.5 Logo JavaScript

Sumber: Google

Menurut Sahi (2020:122), “*Javascript* adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk *web*”.

Menurut Sibero (dikutip Noviantoro., dkk, 2022:91), “*JavaScript* adalah suatu bahasa pemrograman yang dikembangkan untuk dapat berjalan pada *web browser*. Pada awalnya *JavaScript* dikembangkan pada *web browser Netscape* oleh Brenden Eich dengan nama Mocha, kemudian berubah menjadi *Live-Script* dan yang akhirnya menjadi *JavaScript*”.

Menurut Von, dkk (2021:48), “*Javascript* merupakan bahasa pemrograman tingkat tinggi dan dinamis, javascript bekerja disebagian besar *server web* seperti *google chrome*, dan lain-lain”.

2.4.3 Pengertian CSS



Gambar 2.6 Logo CSS

Sumber: Google



Menurut Saputra (dikutip Handayani., dkk, 2019:34), “Kepanjangan dari CSS adalah *Cascading Style Sheet* yang merupakan suatu bahasa pemrograman suatu bahasa pemrograman *web* yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam *web* sehingga tampilan *web* akan lebih rapi, terstruktur, dan seragam”.

Menurut Kurniawan (dikutip Arizha., dkk, 2019:52), menjelaskan bahwa “CSS atau *Cascading Style Sheets* adalah sebuah dokumen yang berisi aturan yang digunakan untuk memisahkan isi dengan layout dalam halaman-halaman web yang dibuat”.

Menurut Andre (dikutip Sari dan Hidayat, 2022:24), “CSS atau singkatan dari *Cascading Style Sheet* adalah suatu aturan untuk mengatur tampilan dari website sehingga tampilan dalam web lebih terstruktur. CSS sendiri bukanlah bahasa pemrograman, CSS lebih seperti konfigurasi tampilan dari suatu tag pada website. CSS dapat merubah text, warna, background dan posisi dari suatu tag”.

2.4.4 Pengertian PHP



Gambar 2.7 Logo PHP

Sumber: Google

Menurut Betha Sidik (dikutip Noviyanti., dkk, 2021:70), “*PHP* merupakan secara umum dikenal sebagai bahasa pemrograman *script-script* yang membuat dokumen *HTML* secara *on the fly* yang dieksekusi di *server web*, dokumen *HTML* yang dihasilkan dari suatu aplikasi bukan dokumen *HTML* yang dibuat dengan menggunakan editor teks atau editor *HTML*”.

Menurut Simare dan Yana (2022:72), “*PHP* adalah bahasa pemrograman *script server side* yang sengaja dirancang lebih cenderung untuk membuat dan



mengembangkan *web*. Bahasa pemrograman ini memang dirancang untuk para pengembang *web* agar dapat menciptakan suatu halaman *web* yang bersifat dinamis”.

Menurut Wulandari dan Nurmiati (2022:81), “*PHP* sendiri ialah singkatan berasal *Personal Home page Tools. Script* ini akan membuat suatu aplikasi dapat diintegrasikan ke dalam *HTML* sehingga suatu halaman web tidak lagi bersifat statis, namun menjadi dinamis”.

2.4.5 Pengertian *Bootstrap*



Gambar 2.8 Logo *Bootstrap*

Sumber: Google

Menurut Mediana dan Nurhidayat (dikutip Purwanto dan Mubarak, 2020:154), “*Bootstrap* merupakan sebuah kerangka *framework* atau wadah untuk pengembangan aplikasi *web* yang lebih cepat dan mudah untuk suatu pengembangan sebuah sistem yang akan dibangun”.

Menurut Sopian, dkk (2020:50), “*Bootstrap* juga merupakan salah satu *framework* HTML, CSS dan *javascript* yang paling populer di kalangan *web* developer yang digunakan untuk mengembangkan sebuah *website* yang responsive. Sehingga halaman *website* nantinya dapat menyesuaikan sesuai dengan ukuran *monitor device* (desktop, tablet, ponsel) yang digunakan pengguna disaat mengakses *website-website* dari *browser*”.

Menurut Nugroho dan Setyawati (dikutip Satria., dkk, 2022:115), “*Bootstrap* adalah *framework* *css* untuk membuat tampilan web yang menyediakan *class* dan komponen yang sudah siap dipakai dan digunakan”.



2.4.6 Pengertian XAMPP



Gambar 2.9 Logo XAMPP

Sumber: Google

Menurut Imam Mulhim (dikutip Tumini dan Fitria, 2021:14), “XAMPP adalah pake instalasi program yang terdiri atas program apache HTTP Server, MySQL, database dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan perl”.

Menurut Jubilee Enterprise (dikutip Tumini dan Fitria, 2021:14), “XAMPP adalah server yang paling banyak digunakan untuk keperluan belajar PHP secara mandiri, terutama bagi programmer pemula”.

Menurut Sari dan Hidayat (2022:26), “Xampp merupakan tool pembantu pengembangan paket perangkat lunak berbasis open source yang menggabungkan Apache web server, MySQL, PHP dan beberapa modul lainnya di dalam satu paket aplikasi”.

2.4.7 Pengertian MySQL



Gambar 2.10 Logo MySQL

Sumber: Google

Menurut Tebai dan Wijanarko (2022:119), “MySQL merupakan pemrograman / sistem manajemen database (kumpulan data yang terstruktur) yang menggunakan basis Bahasa SQL (*Structured Query Language*)”.



Menurut Adi Nugroho (dikutip Renaldy dan Rustam, 2022:29), “MySQL (*My Structured Query Language*) adalah suatu sistem basis data relation atau *Relational Database Management System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan oleh banyak user”.

Menurut Wulandari dan Nurmiati (2022:81), “MySQL ialah database yang menghubungkan script PHP memakai perintah query”.

2.4.8 Pengertian *Sublime Text*



Gambar 2.11 Logo *Sublime Text*

Sumber: Google

Menurut Rahman (2022:29), “Sublime Text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya akan fitur, cross-platform, mudah dan simpel yang cukup terkenal di kalangan pengembang, penulis, dan desainer”.

Menurut Haughee (dikutip Tebai dan Wijanarko, 2022:119), “Sublime text adalah aplikasi editor untuk kode dan teks yang dapat berjalan di berbagai platform operating system dengan menggunakan teknologi *Phyton APP*”.

Menurut Henry (dikutip Lengkong dan Reynaldo, 2023:47), “Aplikasi Sublime atau lebih dikenal dengan Sublime Text merupakan suatu text editor yang sering digunakan oleh programmer khususnya seorang web developer. Aplikasi ini menjadi sesuatu yang sangat penting bagi web developer sebagai senjata koding. Banyak web developer yang menggunakan aplikasi Sublime Text untuk mendukung website mereka. Warna dari tulisan koding yang sangat variasi dan interaktif tentu sangat menarik”.