



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”.

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”.

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”.

2.1.5 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 dirumah”.

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

2.1.6 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, Prosessor, 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)”.

2.1.7 Pengertian Basis Data (Database)

Menurut Sudarso (2022:3), “Basis data adalah tempat berkumpulnya data yang saling berhubungan dalam suatu wadah (perusahaan/organisasi) bertujuan



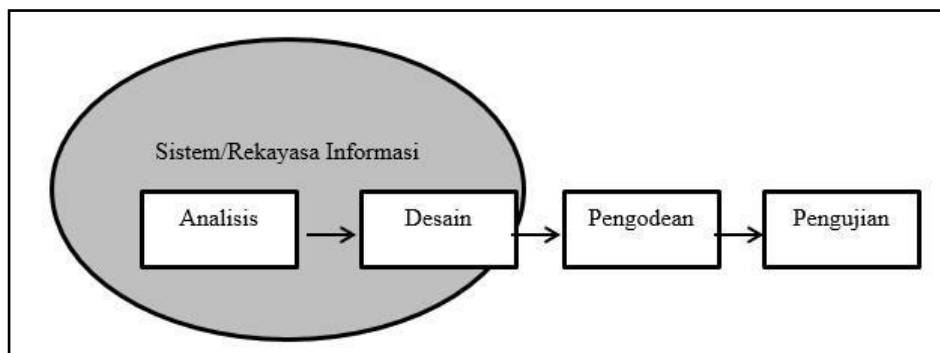
agar dapat mempermudah dan mempercepat untuk pemanggilan atau pemanfaatan kembali data tersebut”.

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

2.1.8 Metode Pengembangan Sistem Waterfall

Penelitian Laporan Akhir ini menggunakan metode pengembangan sistem Waterfall. Menurut Rosa A.S dan M.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

(Sumber: Rosa A S dan M Shalahuddin, 2014)

a. Analisis Kebutuhan Perangkat Lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk mespesifikasikan 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.



2.2 Teori Khusus

2.2.1 Kamus Data

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”.

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 | n kali diulang/ bernilai banyak |
| 5 | () | data opsional |

(sumber: (Sukamto dan Shalahuddin)

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

2.2.2 Pengertian Unified Modeling Language (UML)



Gambar 2. 2 Logo UML

Sumber: Google

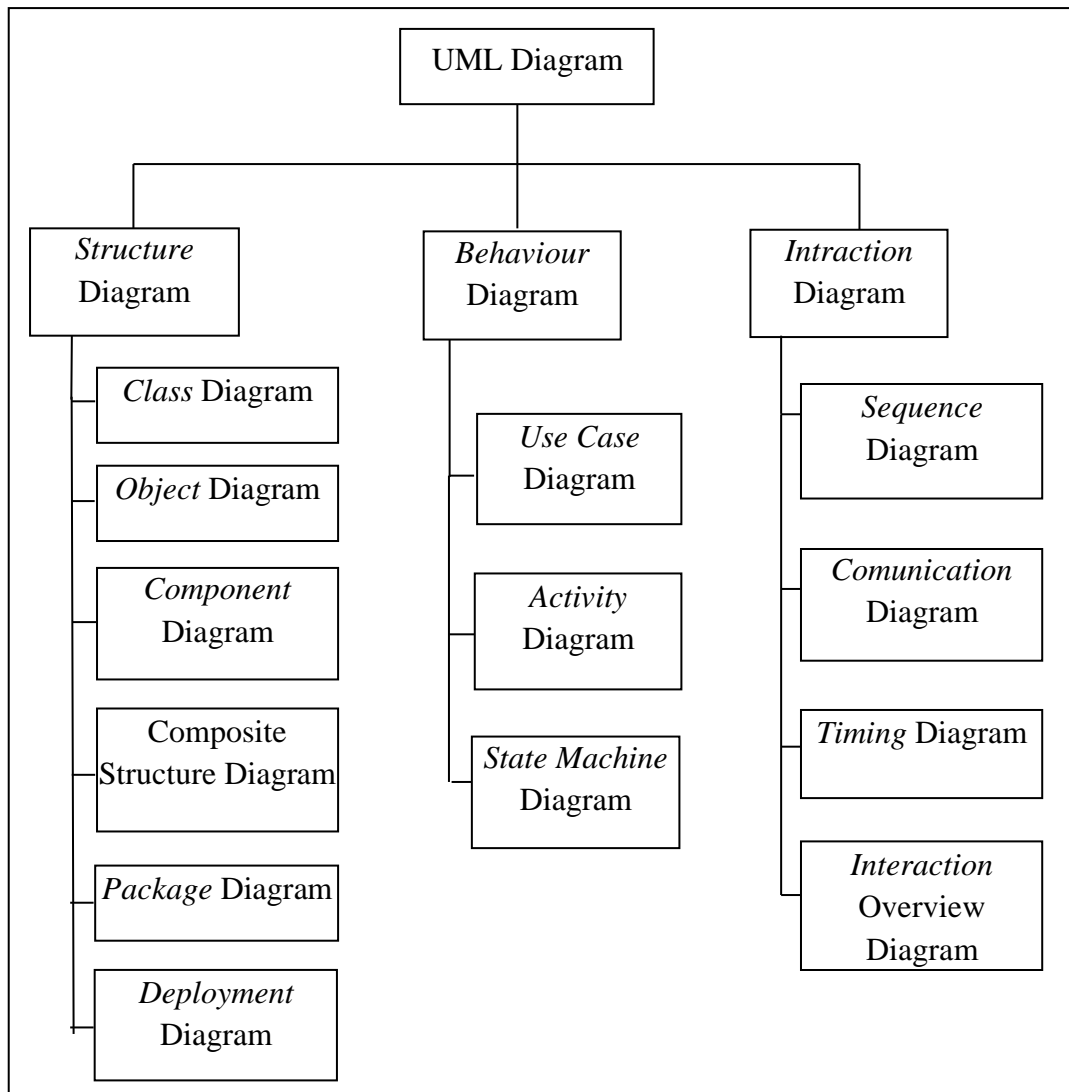


Menurut Rosa A.S dan M.Shalahuddin (2016:133), “UML (*Unified Modelling Language*) adalah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

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”.

2.2.3 Klasifikasi Diagram UML (Undefined Modelling Language)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun dan dokumentasi dari system perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah system dengan menggunakan diagram dan teks-teks pendukung. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar dibawah ini:



(Sumber: Rosa A.S dan M.Shalahuddin (2016:140))

Gambar 2. 3 Klasifikasi Diagram UML (Undefined Modelling Language)

2.2.4 Jenis-Jenis Diagram Unified Modeling Language (UML)

2.2.4.1 Pengertian Use Case Diagram

Menurut Tohari dalam Tabrani dan Aghniya (dikutip Apriliah et al, 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”.

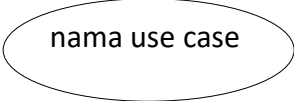
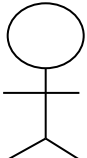


Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *use case*.

- 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.

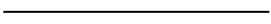
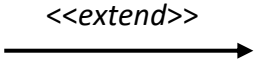
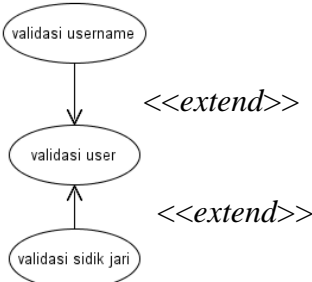
Simbol-simbol pada *Use Case Diagram* adalah sebagai berikut:

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 - |

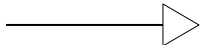
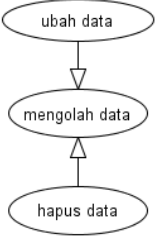
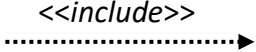
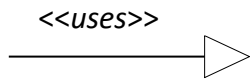
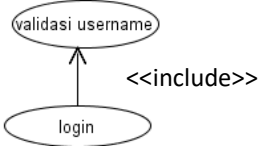


Tabel 2. 3 Lanjutan Simbol-Symbol Use Case Diagram

| No | Simbol | Deskripsi |
|----|--|---|
| | | merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor. |
| 3. | Asosiasi / <i>association</i>  | 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. | Ekstensi / <i>extend</i>  | <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> |



Tabel 2. 4 Lanjutan Simbol-Simbol Use Case Diagram

| No | Simbol | Deskripsi |
|----|---|--|
| 5. | Generalisasi/ <i>generalization</i>  | 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:  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum). |
| 6. | Menggunakan <i>/include /uses</i>   | Relasi 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. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:  b. <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan - |



Tabel 2. 5 Lanjutan Simbol-Simbol Use Case Diagram

| No | Simbol | Deskripsi |
|----|--------|---|
| | | <p>apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p> <pre> graph TD UC1(ubah data) -- <<include>> UC2(validasi user) </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p> |

(Sumber: Rosa A.S dan M.Shalahuddin (2016:156-158))

2.2.4.2 Pengertian Class Diagram

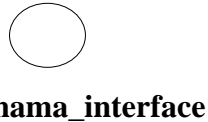


Menurut Tohari (dikutip Tabrani dan Aghniya, 2019:43), mendefinisikan bahwa “Kelas (*class*) adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan perancangan berorientasi objek”.

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

Tabel 2. 6 Simbol-Simbol Class Diagram

| No | Simbol | Deskripsi |
|----|--|-----------------------------|
| 1. | <p>Kelas</p> <pre> classDiagram class nama_kelas { + atribut + operasi () } </pre> | Kelas pada struktur sistem. |



| | | |
|----|---|---|
| 2. | Antarmuka / <i>interface</i>  | 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: Rosa A.S dan M.Shalahuddin (2016:146-147))



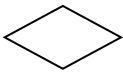


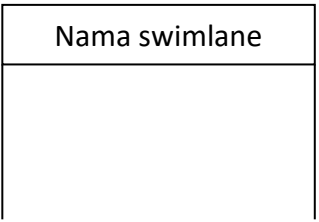
2.2.4.3 Pengertian Activity Diagram

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”.

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




Tabel 2. 7 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>  | <i>Swimlane</i> memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |



Tabel 2. 8 Lanjutan Simbol-simbol Activity Diagram

| No | Simbol | Deskripsi |
|----|---|-----------|
| 6. | atau  | |

(Sumber: Rosa A.S dan M.Shalahuddin (2016:162-163))

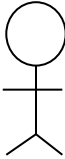
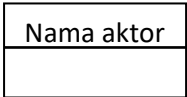

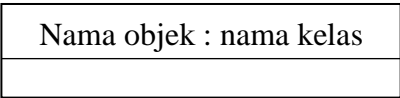
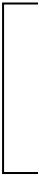
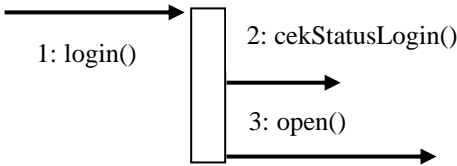
2.2.4.4 Pengertian Sequence Diagram

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

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

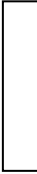
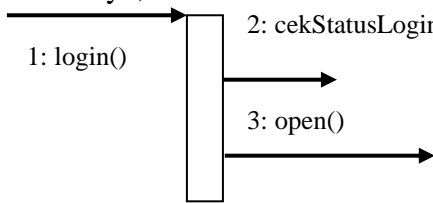
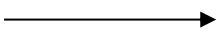
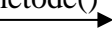
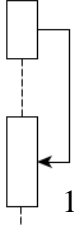


Tabel 2. 9 Simbol-Simbol Sequence Diagram

| No | Simbol | Deskripsi |
|----|---|---|
| 1. | <p>Aktor</p>  <p>Atau</p>  <p>tanpa waktu aktif</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> . |
| 2. | <p>Garis hidup/ <i>lifeline</i></p>  | Menyatakan kehidupan suatu objek. |
| 3. | <p>Objek</p>  | Menyatakan objek yang berinteraksi pesan. |
| 4. | <p>Waktu aktif</p>  | <p>, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login()</p> <p>Aktor tidak memiliki waktu aktif.</p> |

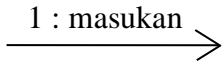
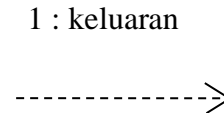
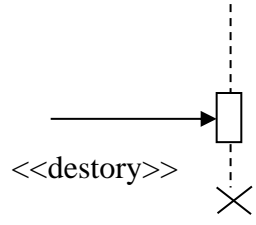


Tabel 2. 10 Lanjutan Simbol-Symbol Sequence Diagram

| No | Simbol | Deskripsi |
|----|--|---|
| 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, misalInva</p>  <p>maka cek Status Login () dan open() dilakukan di dalam metode login() Aktor tidak memiliki waktu aktif.</p> |
| 5. | <p>pesan tipe <i>create</i></p> <p><code><<create>></code></p>  | <p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p> |
| 6. | <p>Pesan tipe <i>call</i></p> <p><u>1 : nama_metode()</u></p>  | <p>Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,</p>  <p>arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus -</p> |



Tabel 2. 11 Lanjutan Simbol-Symbol Sequence Diagram

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

(Sumber: Rosa A.S dan M.Shalahuddin (2016:165-167))

2.3 Teori Judul

2.3.1 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”.



2.3.2 Pengertian Aduan

Menurut Ramita et al (2020:79), “Pengaduan adalah pemberitahuan atau penyampaian keluhan yang disampaikan kepada pihak yang berwenang atas ketidaksesuaian standar pelayanan yang bertujuan untuk ditindak lanjuti pihak yang berwenang”.

Menurut Tolba et al (dikutip Amilia & Rahayu, 2020), “pengaduan adalah semua komunikasi tertulis, lisan atau elektronik yang digunakan pelanggan untuk menyatakan ketidaksenangan mereka mengenai kekurangan atau kegagalan suatu produk atau jasa”.

2.3.3 Pengertian Layanan

Menurut Bashori (2016), “Layanan adalah suatu kegiatan atau urutan kegiatan yang terjadi dalam interaksi langsung antara seseorang dengan orang lain atau mesin secara fisik, dan menyediakan kepuasan pelanggan”.

Menurut Kai et al (2018), “Layanan adalah menyediakan atau memenuhi segala apa yang menjadi kebutuhan bagi orang lain sebagai penerima layanan tersebut”.

2.3.4 Pengertian Website

Menurut Hasan dan Muhammad (2020:47), “Pengertian web yaitu sebuah dokumen yang ditulis dalam *hype text markup language* (HTML) yang dapat diakses melalui *protocol hype text tranfer protocol* (HTTP) yang merupakan protokol untuk menyampaikan informasi dari sebuah pusat situs *web* untuk ditampilkan dihadapan pengguna program pembaca informasi yang ada pada situs *web*”.

Menurut Beki (dikutip Riko, 2020:110), “*Website* merupakan kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang



saling terkait, yang masing-masing dihubungkan dengan jaringan-jaringan halaman”.

2.3.5 Pengertian Aplikasi Pelayanan Pengaduan Masyarakat di RSUD Prabumulih Berbasis Website

Aplikasi Pelayanan Pengaduan Masyarakat di RSUD Prabumulih Berbasis *Website* adalah sebuah aplikasi yang terdiri dari beberapa aplikasi yaitu aplikasi tempat masyarakat bisa melakukan pengaduan terhadap rumah sakit, yang mana aplikasi ini dapat membantu pihak Rumah Sakit Umum Daerah (RSUD) Prabumulih dalam pengelolaan data pengaduan layanan agar lebih efektif dan efisien dalam pelaksanaannya.

2.4 Teori Program

2.4.1 Pengertian HTML



Gambar 2. 4 Logo HTML

Sumber: Google

Menurut Pahlevi et al (2018), “HTML (Hyper Text Mark Up Language) merupakan bahasa yang digunakan untuk mendeskripsikan struktur sebuah halaman web”.

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

2.4.3 Pengertian CSS



Gambar 2. 6 Logo CSS

Sumber: Google

Menurut Saputra (dikutip Handayani et al, 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 et al, 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”.

2.4.4 Pengertian PHP



Gambar 2. 7 Logo PHP

Sumber: Google

Menurut Betha Sidik (dikutip Noviyanti et al, 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 Henderson (dikutip Pahlevi et al, 2018), “CSS kepanjangan dari *Cascading Style Sheet* adalah bahasa-bahasa yang merepresentasikan halaman web. Seperti warna, layout, dan font”.

2.4.5 Pengertian Bootstrap



Gambar 2. 8 Logo Bootstrap

Sumber: Google



Menurut Mediana dan Nurhidayat (dikutip Purwanto dan Mubarok, 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 et al (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*”.

2.4.6 Pengertian XAMPP



Gambar 2. 9 Logo XAMPP

Sumber: Google

Menurut Imam Mulhim (dikutip Tumini dan Fitria, 2021:14), “XAMPP adalah paket 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”.



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”.

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*”.