



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Menurut Sahyar (2016:1) “Komputer adalah alat pengolah data yang bekerja secara elektronik dan dikontrol oleh program, dengan keterlibatan pemakai atau pengguna seminimal mungkin”.

Wahyudin dan Munir (2018:1) Mengatakan bahwa “komputer adalah suatu peralatan 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 Matondang, dkk (2019:1) mengatakan bahwa komputer (*computer*) diambil dari bahasa latin “*computare*” yang berarti menghitung (*to compute* atau *to reckon*) yaitu sebuah perangkat yang dapat digunakan untuk mengolah suatu data dan kemudian dapat disimpan ke dalam perangkat penyimpanan yang terdapat di dalam komputer.

Dari beberapa definisi diatas penulis menyimpulkan bahwa komputer adalah sebuah alat elektronik yang dapat mengelola data menjadi informasi yang berguna dalam mempermudah pekerjaan manusia.

2.1.2 Pengertian Perangkat Lunak (*Software*)

Menurut Sahyar (2016:1) “*Software* adalah perintah-perintah dalam bentuk rekaman untuk mengendalikan kerja perangkat keras komputer”.

Menurut Rosa A.S dan M. Salahuddin (2018:2) “Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”.

Menurut Kadir (2021:2) mengatakan bahwa Perangkat Lunak adalah instruksi-instruksi yang ditujukan kepada komputer sehingga dapat melaksanakan tugas sesuai dengan kehendak pemakai.



Berdasarkan pengertian di atas, penulis menyimpulkan bahwa perangkat lunak (*software*) adalah sebuah program komputer yang dibuat untuk membantu mempermudah para pengguna atau *user* dalam mengerjakan tugas-tugas tertentu.

2.1.3 Pengertian Basis Data (*Database*)

Menurut Kadir (2008:3) “Basis data dianggap sebagai suatu penyusunan data yang terstruktur yang disimpan dalam media pengingat (*hard disk*) yang tujuannya adalah agar data tersebut dapat diakses dengan mudah dan cepat”.

Menurut Priyanto Hidayatullah dan Jauhari Khairul Kawistara (2017:142) “Basis data dapat didefinisikan sebagai himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Menurut Rosa A. S dan M. Shalahuddin (2018:43) “Basis data (*database*) adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Dari beberapa pendapat mengenai definisi basis data yang telah dikemukakan oleh beberapa para ahli diatas, maka dapat disimpulkan bahwa basis data atau *database* merupakan suatu kumpulan data yang dapat berupa berbagai macam file yang tersimpan di dalam *storage* khusus yang digunakan untuk kepentingan suatu organisasi dan dapat diakses ataupun digunakan oleh siapapun yang berhak dan juga membutuhkannya.

2.1.4 Metode Pengembangan Sistem

Menurut Leod R.M (1996:184), Metode Pengembangan Sistem atau SDLC (*System Development Life Cycle*) merupakan gambaran dari suatu usaha dalam merancang sistem yang akan selalu bergerak seperti roda, yang melewati beberapa langkah atau tahapan, antara lain tahap *investigate*, *analyze*, desain, implementasi dan perawatan. Dan langkah selanjutnya akan kembali pada tahap *investigate*, jika dirasakan bahwa sistem yang ada sudah tidak efisien lagi untuk diterapkan.



Menurut Jabu Baso, dkk (2021:90), Metode Pengembangan Sistem atau SDLC (*System Development Life Cycle*) adalah siklus yang digunakan dalam pembuatan atau pengembangan sistem informasi yang bertujuan untuk menyelesaikan masalah secara efektif.

Menurut Rosa A.S dan M. Salahuddin (2018:26), Metode Pengembangan Sistem atau SDLC (*System Development Life Cycle*) adalah proses mengembangkan atau mengubah suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan orang untuk mengembangkan sistem-sistem perangkat lunak sebelumnya (berdasarkan *best practice* atau cara-cara yang sudah teruji baik).

Rosa A.S dan M. Salahuddin (2018:133), Metode 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, desai, pengodean, pengujian, dan tahap pendukung (*support*).

1) Analisis kebutuhan perangkat lunak

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

2) 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.



3) 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.

4) 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 (*output*) yang dihasilkan sesuai dengan yang diinginkan.

5) 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 Judul

2.2.1 Pengertian Aplikasi

Menurut Kadir (2021:4) mengatakan bahwa, Aplikasi adalah suatu perangkat lunak atau program yang biasanya dipasang di komputer untuk melakukan instruksi-instruksi dari pemakai.

Menurut Habibi, dkk (2020:4), Aplikasi adalah suatu program komputer yang bertujuan untuk mengerjakan tugas dari *user*.

Menurut Sutabri (2012:147), Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya.

Jadi, dari definisi di atas dapat disimpulkan bahwa Aplikasi adalah suatu program yang tidak terpisahkan pada sistem komputer dengan tujuan untuk melaksanakan tugas dari *user* atau pengguna sesuai kemampuan yang dimilikinya.



2.2.2 Pengertian Invoice

Menurut Siswosoediro H.S (2008:91), Invoice/faktur adalah dokumen yang diterbitkan oleh penjual (eksportir) kepada pembeli (importir) yang mencantumkan tanggal pengeluaran invoice, tanggal pengiriman barang, uraian barang (berat dan ukuran), harga, biaya-biaya lain, jumlah total yang harus dibayar pembeli, syarat penyerahan barang dan syarat pembayaran, nama kapal laut/kapal udara/kereta api dan nama pelabuhan/bandara/stasiun muat, serta pelabuhan/bandara/stasiun bongkar.

Menurut Mahmudi A. (2009:249), Invoice adalah proses pengiriman tagihan (faktur) kepada pelanggan (*customer*).

Menurut Hamdani dan Bushindo (2017:270), Invoice adalah sangat esensial dalam perdagangan, sebab didalam invoice itu tercantum data-data tentang nota perincian barang-barang yang dijual, harga barang, nama dan alamat pembeli, cara pengapalan, Nomor Post Tarif dan sebagainya.

Dari beberapa pendapat mengenai definisi Invoice yang telah dikemukakan oleh beberapa para ahli diatas, maka penulis menyimpulkan bahwa Invoice adalah dokumen tagihan yang diterbitkan oleh penjual kepada pelanggan yang mencantumkan perincian transaksi-transaksi yang dilakukan oleh kedua belah pihak.

2.2.3 Pengertian Transaksi Kontrak

Menurut Mahtumah B. (2021:257), Transaksi adalah suatu kegiatan yang menyebabkan adanya suatu perubahan yang berdampak pada keuangan si pelaku transaksi.

Menurut Kartomo dan Sudarman L. (2019:15), Transaksi adalah suatu aktivitas perusahaan yang menimbulkan perubahan terhadap posisi harta keuangan perusahaan, misalnya seperti menjual, membeli, membayar gaji, serta membayar berbagai macam biaya yang lainnya.

Menurut Jumardi (2020:35), Transaksi adalah kesepakatan dalam kegiatan jual-beli.



Menurut Naja D. (2006:249), Kontrak merupakan sesuatu yang sakral, yaitu merupakan wujud dari kebebasan (*freedom of contract*) dan kehendak bebas untuk memilih (*freedom of choise*).

Menurut Emirzon J. dan Sadi M. (2021:1), Kontrak merupakan dasar bagi para pihak untuk melakukan tuntutan seandainya ada salah satu pihak tidak melaksanakan apa yang diperjanjikan.

Menurut Dirdjosisworo S. (2002:1), Kontrak adalah perjanjian atau kesepakatan antara dua pihak yang menimbulkan pengikatan antara keduanya untuk melaksanakan apa yang telah diperjanjikan.

Dari beberapa pendapat mengenai definisi mengenai Transaksi dan Kontrak yang telah dikemukakan oleh beberapa para ahli diatas, maka penulis menyimpulkan bahwa Transaksi Kontrak adalah suatu aktivitas yang menyebabkan adanya kesepakatan dalam kegiatan keuangan perusahaan dan menggunakan perjanjian atau kesepakatan antara dua pihak.

2.2.4 Pengertian *Website*

Menurut Yuhefizar (2009:2), *Website* adalah keseluruhan halaman-halaman web yang terdapat dalam sebuah domain yang mengandung informasi.

Menurut Widia D.M dan Asriningtias S.R (2021:3), *Website* adalah kumpulan dokumen berupa halaman web yang berisi teks dalam format *Hyper Text Markup Language* (HTML).

Menurut Elgamar (2020:3), *Website* merupakan sebuah media yang memiliki banyak halaman yang saling terhubung (*hyperlink*), dimana *website* memiliki fungsi dalam memberikan informasi berupa teks, gambar, video, suara dan animasi atau penggabungan dari semuanya.

Dari beberapa pendapat mengenai definisi mengenai *Website* yang telah dikemukakan oleh beberapa para ahli diatas, maka penulis menyimpulkan bahwa *Website* adalah sebuah media yang memiliki halaman-halaman web yang saling terhubung dan berisi informasi, serta terdapat dalam sebuah domain.



2.2.5 Pengertian Aplikasi Invoice Transaksi Kontrak pada PT Bhandha Ghara Reksa (Persero) Berbasis Website

Aplikasi Invoice Transaksi Kontrak merupakan aplikasi yang membantu karyawan PT Bhandha Ghara Reksa (Persero) dalam memproses pengajuan dan persetujuan invoice berbasis *website*.

2.3 Teori Khusus

2.3.1 *Unified Model Language (UML)*

Menurut Azis F. (2005:116), *Unified Modeling Language (UML)* adalah sekumpulan simbol dan diagram untuk memodelkan software.

Menurut Sari I.P (2021:156), *Unified Modeling Language (UML)* adalah bahasa grafis yang menggunakan sekelompok simbol grafis.

Menurut Rosa A.S dan M. Salahuddin (2018:133), “*Unified Modeling Language (UML)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain”.

UML muncul karena adanya kebutuhan pemodelan visual untuk menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. *UML* hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan *UML* tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya *UML* paling banyak digunakan pada metodologi berorientasi objek.

Dalam membangun perancangan sistem dengan alat bantu perancangan *Unified Modeling Language (UML)* ada beberapa tahapan yang akan dilakukan, yaitu sebagai berikut:

2.3.1.1 *Use Case Diagram*

Menurut Swain G. (2010:110), *Use Case Diagram* merupakan diagram yang menunjukkan satu set *use case*, aktor dan hubungan mereka.

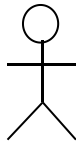
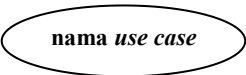


Menurut Kulak D. dan Guiney E. (2004:29), *Use Case Diagram* adalah penggambaran grafis dari hubungan antara aktor dan *use case* dan antara *use case* dan *use case* lainnya.

Menurut Rosa A.S dan M. Shalahuddin (2018:155), “*Use Case Diagram* 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. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu”.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel 2.1 Simbol-Simbol *Use Case Diagram*

No.	Gambar	Nama	Keterangan
1.		Aktor/ <i>Actor</i>	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 aktor.
2.		<i>Use Case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .

Lanjutan Tabel 2.1 Simbol-Symbol *Use Case Diagram*

No.	Gambar	Nama	Keterangan
3.		Asosiasi/ <i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4.		Ekstensi / <i>extend</i>	Relasi <i>use case</i> tambahan 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 pemrograman berorientasi objek, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan. 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
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. Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).

Lanjutan Tabel 2.1 Simbol-Simbol *Use Case Diagram*

No.	Gambar	Nama	Keterangan
6.		<i>Include</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.

(Sumber : Rosa A.S dan M. Shalahuddin, 2018:156)

2.3.1.2 *Activity Diagram* (Diagram Aktivitas)

John S., 2010, dalam buku *System Analysis and Design in a Changing World* menyatakan bahwa “*Activity Diagram* adalah sebuah diagram alur kerja yang menjelaskan berbagai kegiatan pengguna (sistem), orang yang melakukan masing-masing aktivitas, dan aliran sekuensial dari aktivitas-aktivitas tersebut”.

Menurut Destriana R., dkk (2021:10), *Diagram Aktivitas* adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan *concurrency*.

Menurut Rosa A.S dan M. Shalahuddin (2018:161), “*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel 2.2 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
1.		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

Lanjutan Tabel 2.2 Simbol-Simbol *Activity Diagram*

No	Simbol	Nama	Deskripsi
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.

(Sumber : Rosa A.S dan M. Shalahuddin, 2018:162)

2.3.1.3 Class Diagram (Diagram Kelas)

Menurut Eder J. dan Bajec M. (2012:429), Diagram Kelas adalah diagram yang hanya menggambarkan dimensi struktur sistem. Semua elemen menggambarkan struktur sistem tetapi beberapa properti perilaku dihasilkan dari *Operation*.



Menurut Aldo D., dkk (2020:98), *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* dapat membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

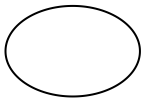

Menurut Rosa A.S dan M. Shalahuddin (2018:141), “Diagram Kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan operasi:

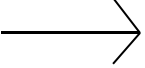
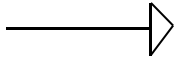
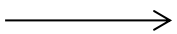
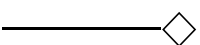
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1.	<div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid black; padding: 5px; width: 150px;"> <p style="text-align: center;">Class</p> <hr/> <p>Attribute</p> <hr/> <p>Operation</p> </div> <div style="border: 1px solid black; width: 50px; height: 50px; margin-left: 10px;"></div> </div>	Kelas	Kelas pada struktur sistem
2.		Antarmuka/ <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi obyek
3.		Asosiasi/ <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>

Lanjutan Tabel 2.3 Simbol-Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
4.		Asosiasi berarah/ <i>directed</i>	Asosiasi antarkelas dengan makna kelas yang satu digunakan oleh kelas lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.		Generalisasi	Asosiasi antar kelas dengan makna generalisasi spesialisasi (umum – khusus)
6.		Kebergantungan/ <i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antarkelas
7.		Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>)

(Sumber : Rosa A.S dan M. Shalahuddin, 2018:146)

2.3.1.4 *Sequence Diagram* (Diagram Sekuen)

Menurut Destriana R., dkk (2021:14), *Sequence Diagram* adalah diagram yang menggambarkan interaksi dengan berfokus pada urutan pesan yang dipertukarkan, bersama dengan spesifikasi kejadian yang sesuai di *Lifelines*.

Menurut Aldo D., dkk (2020:98), *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek.


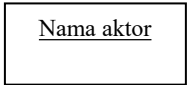


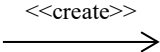
Menurut Rosa A.S dan M. Shalahuddin (2018:165), “*Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek”. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang



diinstansiasi menjadi objek tersebut. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.

Sequence diagram digunakan untuk memperlihatkan interaksi antar obek dalam perintah yang berurut. Tujuan utama *sequence diagram* adalah mendefinisikan urutan kejadian yang dapat menghasilkan *output* yang diinginkan. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1.	 Atau 	Aktor	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 aktor
2.		Garis hidup <i>/ lifeline</i>	Menyatakan kehidupan suatu objek
3.		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
4.		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat

Lanjutan Tabel 2.4 Simbol-Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
5.		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/metode
6.		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
7.		Pesan tipe keluaran	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

(Sumber : Rosa A.S dan M. Shalahuddin, 2018:165)

2.4 Teori Program

2.4.1 *HTML (HyperText Markup Language)*

Sulhan (2006:23) “*HTML* adalah suatu sistem untuk menambahkan dokumen dengan tabel yang menandakan bagaimana teks di dokumen harus disajikan dan bagaimana dokumen dihubungkan bersama-sama”.

Budiharto (2013:27) Mengatakan bahwa “*HTML* merupakan dokumen web yang statis, hanya mampu menampilkan teks dan gambar yang statis sehingga untuk membuatnya lebih dinamis (misal : teks dan gambar bergerak, menampilkan database suatu perusahaan, dan lainnya) harus menggunakan bahasa lainnya seperti JavaScript, Java, PHP, dll”.

Menurut Priyanto H. dan Jauhari K.K (2017:15) “*Hypertext Markup Language (HTML)* adalah bahasa standard yang digunakan untuk menampilkan halaman web”.



2.4.2 *PHP (Hypertext Preprocessor)*

Sulhan (2006:118) “*PHP* merupakan bahasa pemrograman yang digunakan untuk membangun aplikasi-aplikasi berbasis web khususnya aplikasi web yang bersifat dinamis”.

Kadir (2008:358) *PHP* merupakan bahasa pemrograman skrip yang diletakkan dalam server yang biasa digunakan untuk membuat aplikasi Web yang bersifat dinamis”.

Menurut Sidik (2017:4) “*PHP* merupakan secara umum dikenal sebagai bahasa pemrograman 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*”.

2.4.3 *CSS (Cascading Style Sheets)*

Sakur (2005:31) “*CSS* merupakan salah satu bagian dari design web yang berfungsi untuk mengatur tampilan setiap unsur yang ada di dalam website”.

Menurut Azis, dkk (2019:36), *CSS* adalah sebuah dokumen yang terdiri dari kode program yang digunakan untuk membuat elegan tampilan dari tampilan halaman website yang dibuat.

Menurut Priyanto H. dan Jauhari K.K (2017:56) mengatakan bahwa ada 4 cara memasang kode *CSS* ke dalam kode *HTML* atau halaman *web*, yaitu :

- 1) *Inline Style Sheet* (Memasukkan kode *CSS* langsung pada tag *HTML*)
 - 2) *Internal Style Sheet (Embed* atau memasang kode *CSS* ke dalam bagian *<head>*)
 - 3) Me-link ke external *CSS*
 - 4) Import *CSS* file
-



2.4.4 MySQL

Nugroho (2004:29) “MySQL (*My Structure Query Language*) atau yang biasa dibaca “*mai-se-kuel*” adalah sebuah program pembuat database yang bersifat *open source*, artinya siapa saja boleh menggunakannya dan tidak dicekal. MySQL juga merupakan program pengakses database yang bersifat jaringan sehingga dapat digunakan untuk aplikasi *Multi User* (Banyak Pengguna)”.

Kadir (2008:2) “MySQL merupakan software yang tergolong sebagai DBMS (Database Management System) yang bersifat *Open Source*”.

Menurut Rahmadya T.H dan Herlawati (2018:43) “MySQL merupakan salah satu *software database* terkenal yang banyak digunakan oleh pengembang perangkat lunak karena sifatnya yang *open source*”.

2.4.5 XAMPP



Gambar 2.1 XAMPP

Riyanto (2011:1) “XAMPP merupakan paket *PHP* dan *MySQL* berbasis open source yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis *PHP*. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda ke dalam satu paket”.

Menurut Habibi, dkk (2020:5), XAMPP merupakan perangkat lunak bebas, yang mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa program. XAMPP adalah perangkat yang menggabungkan tiga aplikasi ke dalam satu paket yaitu *Apache*, *MySQL*, dan *PHP my Admin*.

Menurut Enterprise (2018:3), XAMPP merupakan server yang paling banyak digunakan karena fiturnya tergolong lengkap dan mudah digunakan.



2.4.6 *Sublime Text*



Gambar 2.2 *Sublime Text*

Menurut Nugroho, dkk (2021:39), *Sublime Text* adalah aplikasi editor kode dan teks yang dapat berjalan di berbagai *platform system* dengan menggunakan teknologi *Phyton API*.

Menurut Habibi, dkk (Faridl, 2015:3), “*Sublime text* adalah teks editor berbasis *Phyton*, sebuah teks editor yang elegan, kaya fitur, *cross platform*, mudal dan simpel yang cukup terkenal di kalangan *developer* (pengembang), penulis dan desainer”.

Menurut Habibi, dkk (Supono dan Putratama, 2018:14), “*Sublime Text* adalah perangkat lunak *text editor* yang digunakan untuk membuat atau mengubah suatu aplikasi”.