



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Aplikasi

Menurut Reksoatmodjo (2018:24), “Aplikasi adalah program komputer yang digunakan untuk membuat maupun memelihara basis data, kemudian melakukan diseminasi data dan informasi kepada semua pengguna akhir”.

Menurut Chan (2017:4), “Aplikasi adalah koleksi window dan objek-objek yang menyediakan fungsi untuk aktivitas user, seperti pemasukan data, proses, dan pelaporan”.

Dapat disimpulkan bahwa, Aplikasi adalah suatu program komputer yang dibuat untuk membantu penggunaannya dalam berbagai kegunaan.

##### 2.1.2 Pengertian Perangkat Lunak

Menurut Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai”.

Menurut Sukamto dan Shalahuddin (2018:2), “Perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (user manual)”.

Dapat disimpulkan bahwa, Perangkat lunak adalah suatu bagian dari sistem komputer dalam memproses beberapa perintah dari penggunaannya.

##### 2.1.3 Pengertian Komputer

Wahyudin, Munir (2018:1), 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 berkerja secara otomatis.



Menurut Kadir (2019:2), Komputer merupakan peralatan elektronik yang umum di jumpai di mana saja dan bahkan dalam bentuk apa saja. Komputer tidak selalu berbentuk *PC* maupun *notebook*, tetapi bisa melekat dalam perangkat mikro gelombang, pembuat kopi, maupun mesin cuci.

## **2.2 Teori Judul**

### **2.2.1 Pengertian Prediksi**

Menurut Sujana dan Jayadinata (2018:68), Prediksi adalah suatu perkiraan yang spesifik pada bentuk observasi yang akan datang. Prediksi harus didasarkan hasil observasi dan pengukuran yang teliti bukan sekedar menebak tanpa ada dasarnya.

Menurut Minarni dan Aldyanto (2016:59-65), Prediksi adalah suatu proses memperkirakan secara sistematis tentang suatu yang paling mungkin terjadi dimasa depan berdasarkan informasi masa lalu dan sekarang yang dimiliki, agar kesalahannya (selisih antara suatu yang terjadi dengan hasil perkiraan) dapat diperkecil. Prediksi tidak harus memberikan jawaban sedekat mungkin yang akan terjadi.

Dapat disimpulkan bahwa, Prediksi adalah suatu proses untuk memperkirakan tentang sesuatu yang mungkin bisa terjadi di masa depan.

### **2.2.2 Pengertian Kebutuhan**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Onilne, “Butuh adalah yang dibutuhkan”.

Dapat disimpulkan bahwa, Kebutuhan adalah segala sesuatu yang dibutuhkan untuk keperluan mempertahankan hidup.

### **2.2.3 Pengertian Perawatan**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Onilne, “Perawatan adalah proses cara perbuatan merawat dan pemeliharaan”.



Dapat disimpulkan bahwa, Perawatan adalah sebuah proses pemeliharaan tanaman agar tanaman tetap segar, indah dan tidak rusak.

#### **2.2.4 Pengertian Kebun**

Menurut Kamus Besar Bahasa Indonesia (KBBI) Onilne, “Kebun adalah sebidang tanah yang ditanami pohon musiman (buah-buahan dan sebagainya); tanah luas yang ditanami kopi, karet dan sebagainya”.

#### **2.2.5 Pengertian Kelapa Sawit**

Menurut Gunawan (2017:2), Kelapa sawit merupakan salah satu komoditi potensial yang dikembangkan saat ini dengan alasan mempunyai peranan sangat strategis sebagai sumber pendapatan masyarakat, mempunyai prospek pasar yang sangat baik di dalam negeri maupun luar negeri (ekspor), mampu menyerap tenaga kerja baru dan mempunyai peranan dalam pelestarian fungsi lingkungan hidup.

#### **2.2.6 Pengertian Metode *Moving Average***

Menurut Eunike, dkk (2018:32), *Moving Average* menghasilkan peramalan untuk periode berikutnya dengan merata-rata permintaan aktual sejumlah  $n$  periode terakhir. Metode ini merupakan kompromi antara metode naif dan *simple average* untuk masing-masing kelebihan dan kelemahannya. Penentuan jumlah  $n$  didasarkan pada percobaan atau simulasi dengan mempertimbangkan situasi riil di lapangan. Penentuan  $n$  yang terlalu kecil menghasilkan peramalan yang mendekati metode naif, dan menyebabkan peramalan yang fluktuatif, dipengaruhi oleh variasi acak dari permintaan. Penentuan  $n$  yang terlalu besar memberikan hasil peramalan yang mendekati *simple average*, dan hasil peramalan menjadi terlalu stabil dan tidak dapat mendektisi adanya tren.

Menurut Agustian dan Wibowo (2019:156-162), *Moving average* (rerata bergerak) adalah metode menghitung rerata dari pergerakan sejumlah data berurutan dalam jangka waktu tertentu, pada sekelompok data berkala/berurut



waktu (*time series*).

### **2.2.7 Pengertian PT. Andira Agro Tbk**

PT. Andira Agro adalah perseroan terbatas yang didirikan pada tahun 1995 berdasarkan hukum Republik Indonesia yang memiliki, mengoperasikan, dan mengelola perkebunan minyak dan Pabrik Minyak Mentah (“Perusahaan”).

### **2.2.8 Pengertian Website**

Menurut Abdulloh (2018:1), *website* dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang di sediakan melalui jalur koneksi internet sehingga dapat di akses dan dilihat oleh semua orang di seluruh dunia. Halaman *website* dibuat menggunakan bahasa standar yaitu *HTML*. Skript *HTML* ini akan diterjemahkan oleh *web browser* sehingga dapat ditampilkan dalam bentuk informasi yang dapat di baca oleh semua orang.

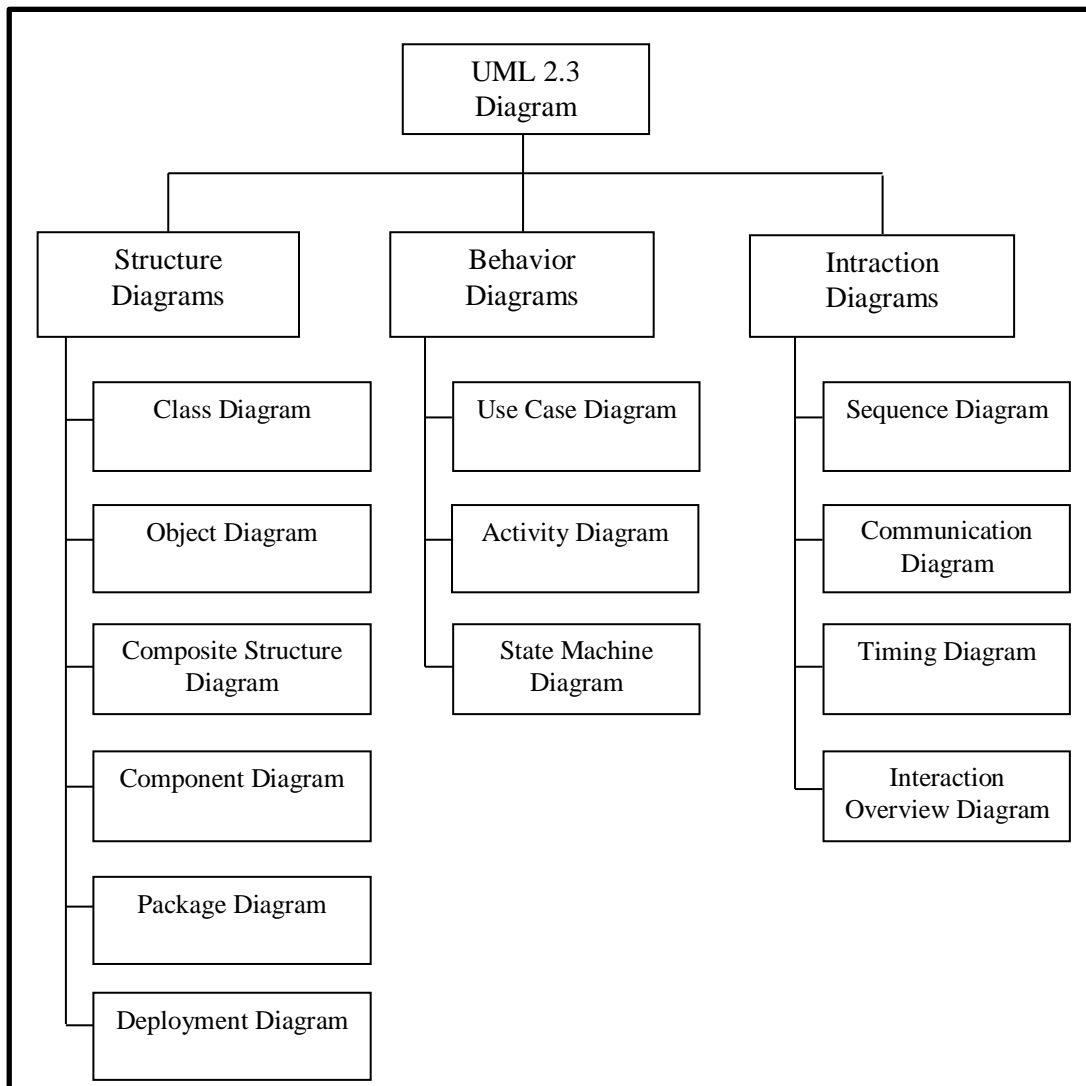
### **2.2.9 Pengertian Aplikasi Prediksi Kebutuhan Perawatan Kebun Kelapa Sawit dengan Menggunakan Metode *Moving Average* pada PT. Andira Agro Tbk Berbasis Website.**

Aplikasi Prediksi Kebutuhan Perawatan Kebun Kelapa Sawit dengan Menggunakan Metode *Moving Average* pada PT. Andira Agro Tbk Berbasis Website adalah aplikasi untuk membatu PT. Andira Agro Tbk melihat menlihat pola naik turunnya data dalam masa runtun waktu tertentu.

## **2.3 Teori Khusus**

### **2.3.1 Pengertian UML (*Unified Modeling Language*)**

Menurut Sukamto dan Shalahuddin (2018:140), “Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori”. Pembagian kategori dan macam-macam diagram Menurut Sukamto dan Shalahuddin tersebut dapat dilihat pada gambar dibawah:



**Gambar 2.1** Macam-macam Diagram UML

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.

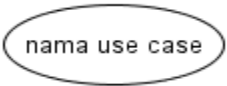
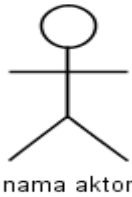



- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.


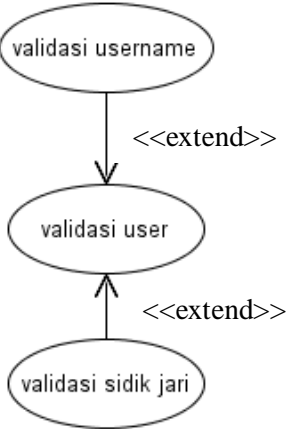
### 2.3.2 Pengertian Use Case Diagram

Menurut Sukamto dan Shalahuddin (2018:155) “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.”

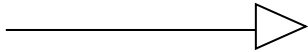
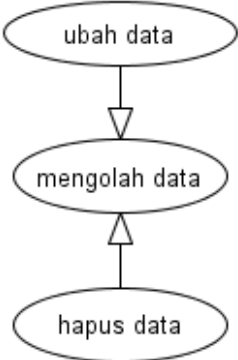

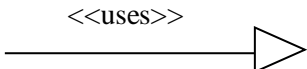
**Tabel 2.1** Simbol-simbol pada *Use case Diagram*

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i></p>
<p>aktor / <i>actor</i></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 actor</p>
<p>asosiasi / <i>association</i></p> 	<p>komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>.</p>

Lanjutan Tabel 2.1 Simbol-simbol pada *Use case Diagram*

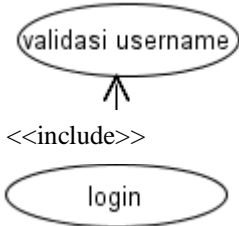
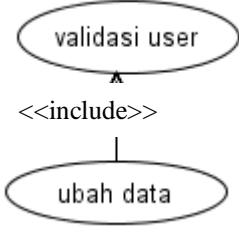
Simbol	Deskripsi
<p>ekstensi / <i>extend</i></p> <p></p>	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misalnya</p> <div data-bbox="790 862 1077 1288" style="text-align: center;">  <pre> graph TD     A([validasi user]) -.-&gt; &lt;&lt;extend&gt;&gt;  B([validasi username])     C([validasi sidik jari]) -.-&gt; &lt;&lt;extend&gt;&gt;  A           </pre> </div> <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>

Lanjutan Tabel 2.1 Simbol-simbol pada *Use case Diagram*

Simbol	Deskripsi
<p data-bbox="300 398 675 432">Generalisasi / <i>generalization</i></p> 	<p data-bbox="758 398 1364 600">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>  <p data-bbox="758 1055 1294 1144">arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
<p data-bbox="300 1167 687 1200">menggunakan / include / uses</p>  	<p data-bbox="758 1167 1337 1473">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>:</p> <ul data-bbox="758 1496 1369 1697" style="list-style-type: none"> <li>• <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:</li> </ul>



Lanjutan Tabel 2.1 Simbol-simbol pada *Use case Diagram*

Simbol	Deskripsi
	<div style="text-align: center;">  </div> <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul> <div style="text-align: center;">  </div> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Sukamto dan Shalahuddin (2018:156)

### 2.3.3 Pengertian *Activity Diagram*





Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.



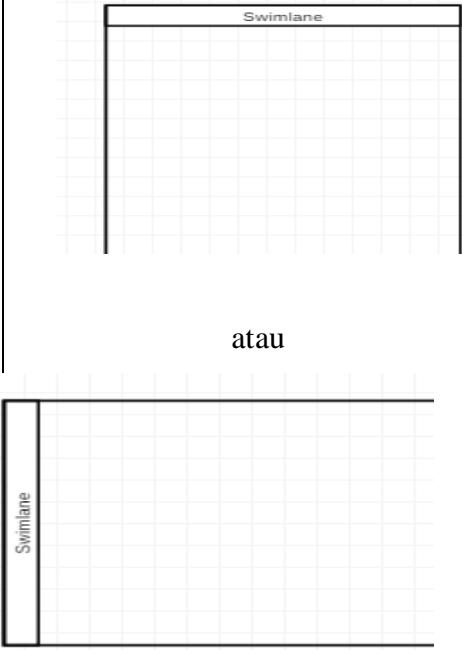
Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

**Tabel 2.2** Simbol-simbol pada *Activity Diagram*

<b>Simbol</b>	<b>Deskripsi</b>
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir



**Lanjutan Tabel 2.2** Simbol-simbol pada *Activity Diagram*

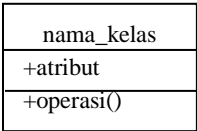
Simbol	Deskripsi
<p data-bbox="512 398 643 432">Swimlane</p>  <p data-bbox="549 786 608 819">atau</p>	<p data-bbox="884 398 1321 544">Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi</p>

Sumber: Sukamto dan Shalahuddin (2018:163)

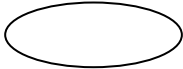


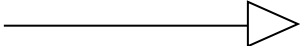

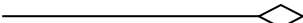
### 2.3.4 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), menyebutkan *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Diagram Class* dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.

**Tabel 2.3** Simbol-simbol pada *Class Diagram*

Simbol	Deskripsi
<p data-bbox="600 1706 671 1740">kelas</p> 	<p data-bbox="999 1706 1254 1794">Kelas pada struktur sistem</p>

Lanjutan Tabel 2.3 Simbol-simbol pada *Class Diagram*

Simbol	Deskripsi
antarmuka / interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
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>
Generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

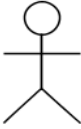

Sumber: Sukamto dan Shalahuddin (2018:146)



### 2.3.5 Pengertian *Sequence Diagram*


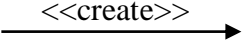
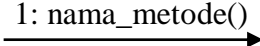
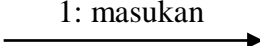
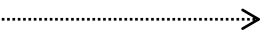
Sukanto dan Shalahuddin (2018:165), menyebutkan *Sequence Diagram* menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka haru diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

**Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

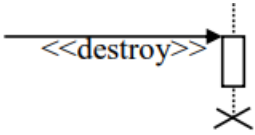
Simbol	Deskripsi
<p>aktor</p>  <p>nama aktor</p> <p>atau</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>nama aktor</u> </div> <p>tanpa waktu aktif</p>	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi 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 fase nama actor.</p>
<p>Garis hidup / <i>Lifeline</i></p> 	<p>menyatakan kehidupan suatu objek.</p>
<p>Objek</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;"> <u>nama objek : nama kelas</u> </div>	<p>menyatakan kehidupan suatu objek.</p>



**Lanjutan Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
<p>Waktu aktif</p> 	<p>menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah tahapan yang dilakukan di dalamnya.</p>
<p>Pesan tipe <i>create</i></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
<p>Pesan tipe <i>call</i></p> 	<p>menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.</p>
<p>Pesan tipe <i>send</i></p> 	<p>menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.</p>
<p>Pesan tipe <i>return</i></p> 	<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.</p>

**Lanjutan Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Pesan tipe <i>destroy</i> 	menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .

Sumber: Sukamto dan Shalahunddin (2018:165)

## 2.4 Teori Program

### 2.4.1 Pengertian Basis Data (*Database*)

Menurut Kristanto (2018:79), “Basis data adalah kumpulan data, yang dapat digambarkan sebagai aktifitas dari satu atau lebih organisasi yang berelasi”. Sedangkan menurut Nugroho (2017:175), “Database/Basis Data merupakan sekumpulan data yang sangat kompleks, kemudian data tersebut memiliki hubungan antara data yang satu dengan yang lainnya”.

### 2.4.2 Pengertian Codeigniter (CI)



**Gambar 2.2** Logo *CodeIgniter*

Menurut Sulistiono (2018:7), “*CodeIgniter* adalah sebuah aplikasi open source yang berupa kerangka kerja atau framework untuk membangun website menggunakan bahasa pemrograman PHP.”



### 2.4.3 Pengertian CSS (*Cascading Style Sheet*)

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

Menurut Abdulloh (2016:3), ) CSS adalah singkatan dari *Cascading Style Sheets* yaitu skrip yang digunakan untuk mengatur desain website.”

Contoh skrip CSS :

```
P{
    color: red;
    font-weight: bold;
    text-transform: uppercase;
}
```

### 2.4.4 Pengertian HTML (*Hypertext Markup Language*)

Menurut Hidayatullah dan Kawistara (2017:17), “*Hypertext Markup Language* (HTML) adalah bahasa standard yang digunakan untuk menampilkan halaman web”.

Menurut Abdulloh (2018:7), “HTML singkatan dari Hypertext Markup Language yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (World Wide Web Consortium) berupa tag-tag yang menyusun setiap elemen dari website”.

Contoh skrip HTML:

```
<!DOCTYPE HTML>
<html>
    <head>
        <title>Judul Webstie</title>
    </head>
    <body>
```





Isi Website

```
</body>  
</html>
```

#### 2.4.5 Pengertian *PHP (Hypertext Preprocessor)*



**Gambar 2.3** Logo PHP

Menurut Raharjo (2018:38), “PHP salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web”.

Menurut Abdulloh (2018:127), “PHP kependekan dari PHP Hypertext Preprocessor yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server”.

#### 2.4.6 Pengertian *MySQL (My Structure Query Language)*

Menurut Nugroho (2019:133), “MySQL merupakan database yang paling digemari dikalangan Programmer Web, dengan alasan bahwa program ini merupakan database yang sangat kuat dan cukup stabil”.

Menurut Sidik (2017:301), “MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bisa dikatakan paling cepat, dan jarang bermasalah”.



### 2.4.7 Pengertian *Xampp*



**Gambar 2.4** Logo XAMPP

Menurut Enterprise (2018:3), *XAMPP* merupakan server yang paling banyak digunakan untuk para *programmer PHP*, khususnya level pemula, fiturnya lengkap dan gampang digunakan oleh *programmer PHP* tingkat awam karena yang perlu anda lakukan hanyalah mengunduh, menginstal, dan menjalankan salah satu module bernama *Apache* yang dapat memproses *PHP*.

Menurut Dantes, dkk (2019:98), *XAMPP* merupakan sebuah paket *software* yang berisi *Apache HTTP* server dan *MySQL* dan mendukung program *PHP* dan *Perl* dan berjalan di beberapa Sistem Operasi (*Windows*, *MAC*, *Linux*, *BSD*) sehingga dengan meng-install *XAMPP* maka kita bisa menjalankan sebuah *web* server di komputer kita.