BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1 Pengertian Komputer

Irma dalam Pratiwi (2019:7) mengatakan, "Komputer adalah sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (proses) dan memberikan informasi (*output*) serta terkoordinasi dibawah kontrol program yang tersimpan dimemorinya".

Kadir dalam Pratiwi (2019:2) mengatakan, "Komputer adalah peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia".

Berdasarkan pengertian diatas, dapat disimpulkan bahwa komputer adalah alat bantu pemrosesan data secara elektronik mulai dari *input*, proses, dan *output* yang bermanfaat untuk melaksanakan berbagai pekerjaan manusia.

2.1.2 Pengertian Perangkat Lunak

Sukamto dan Shalahuddin (2011:2) mengatakan, "Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain dan cara penggunaan (*user manual*)".

Utami dan Asnawati dalam Pratiwi (2019:7) mengatakan, "*Software* adalah prangkat lunak yang berisikan sebuah intruksi yang diperintahkan dan diproses dengan bantuan perangkat keras sehingga tanpa prangkat lunak, perangkat keras tidak bisa dipakai sehingga *software* dan *hardware* tidak bisa dipisahkan".

Berdasarkan pengertian diatas, dapat disimpulkan bahwa perangkat Lunak adalah program komputer yang diperintahkan dan diproses dengan bantuan

perangkat keras yang terasosiasi dengan dokumentasi perangkat lunak untuk memudahkan pekerjaan manusia.

2.1.3 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2011:105), "RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*)". Menurut Sukamto dan Shalahuddin (2011:109-111) adapun tahap-tahap (*fase*) dalam metode pengembangan RUP adalah sebagai berikut:

1. *Inception* (Permulaan)

   Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*bussiness modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*). Berikut adalah tahap yang dibutuhkan pada tahap ini :

   1. Memahami ruang lingkup dari proyek (termasuk pada biaya, ekonomi (Ekonomi, mengendalikan biaya atau meningkatkan keuntungan), waktu, kebutuhan, penentuan skala prioritas, resiko, dan lain sebagainya).
   2. Membangun kasus bisnis yang dibutuhkan.

      Menurut Alkautsar dkk. (2013:4) mengatakan bahwa ada beberapa kriteria kelayakan yang digunakan yaitu kelayakan operasional berhubungan dengan people- oriented, kelayakan teknis berhubungan dengan computer-oriented, kelayakan ekonomis berhubungan dengan biaya dan keuntungan sistem informasi.

      Menurut Fitria dan Widowati (2017:30-31) mengatakan bahwa analisis persyaratan bertujuan untuk menentukan apa yang dapat dilakukan oleh sistem dan harus memenuhi tujuan dari sistem tersebut. *Requirement* yang ada akan dibagi menjadi dua bagian. Bagian pertama adalah *Functional Requirement* (Kebutuhan Fungsional) yaitu aktivitas dan service yang harus disediakan oleh sistem yang akan dikembangkan. Bagian kedua

adalah *Non functional Requirement* (Kebutuhan Non Fungsional) yaitu fitur-fitur lain yang diperlukan oleh sistem agar sistem lebih memuaskan. Non functional Requirement terbagi lagi menjadi, model tampilan, model penyimpanan data, model pengontrolan sistem, model efisiensi sistem.

2. *Elaboration* (Perluasan/perencanaan)

   Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi resiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada purwarupa sistem (*prototype*).

   Hasil yang diharapkan dari tahap ini adalah memenuhi *Lifecycle Architecture Milestone* dengan kriteria berikut :

   1. Model kasus yang digunakan (*use case*) di mana kasus dan aktor yang terlibat telah diidentifikasi dan sebagian besar kasus harus dikembangkan.
   2. Deskripsi dari arsitektur perangkat lunak dari proses pengembagan sistem perangkat lunak telah dibuat.
   3. Rancangan arsitektur yang dapat diimplementasikan dan mengimplementasikan *use case*, dan lain sebagainya.

3. *Construction* (Kontruksi)

   Tahap ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operasional awal.

   Menurut Hutahean dkk. (2019:5796) mengatakan bahwa setelah sistem di implementasikan, kemudian dilakukan pengujian sistem dengan *validation testing*, pengujian *validation tasting* ini menggunakan *black box testing* yang dilakukan untuk menguji apakah sistem dapat berjalan sesuai harapan pengguna.

Menurut Jaya (2018:45-46) mengatakan bahwa *black box testing* merupakan teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak. *black box testing* bekerja dengan mengabaikan struktur control sehingga perhatiannya difokuskan pada informasi domain.

4. *Transition* (Transisi)

   Tahap ini lebih pada deployment atau instalasi sistem agar dapat dimengerti oleh user. Tahap ini menghasilkan produk perangkat lunak dimana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tongggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.2 Teori Khusus

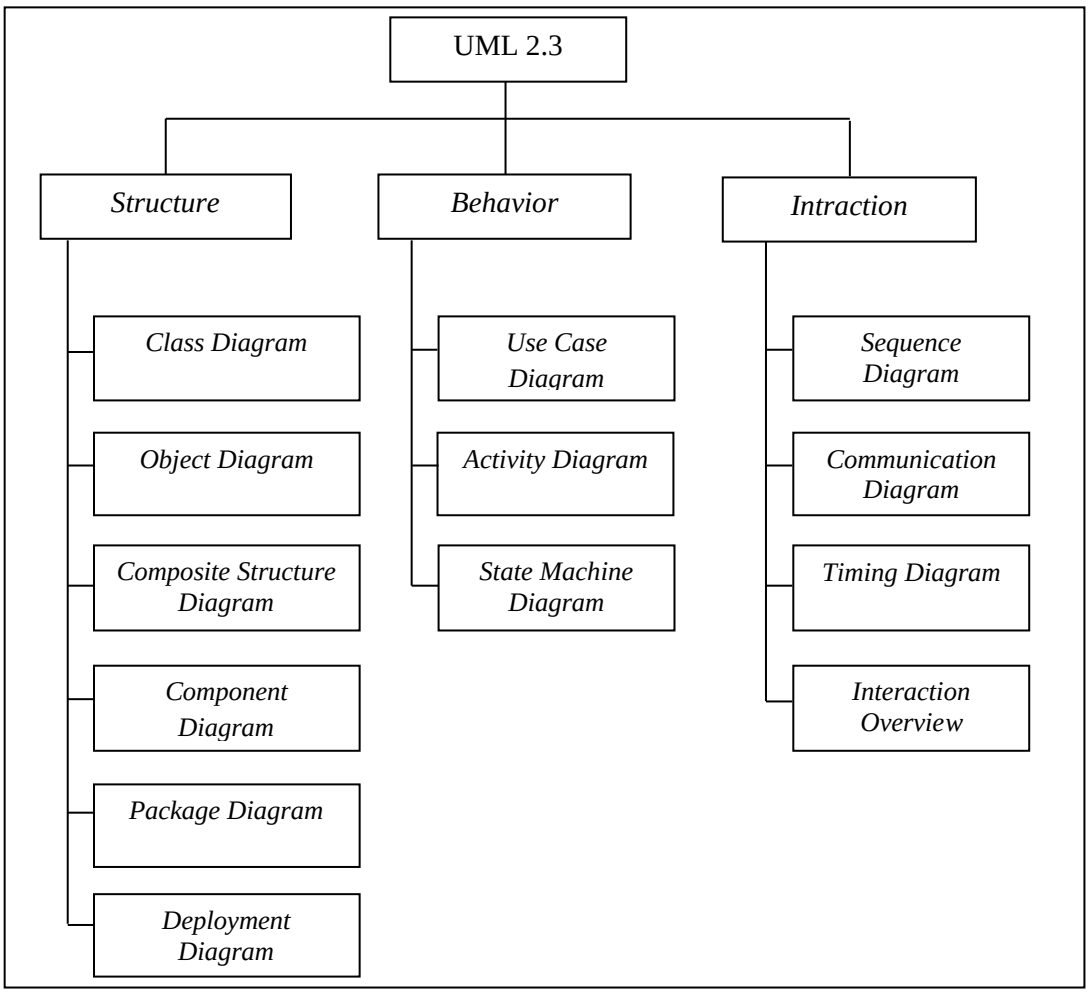
2.2.1 Pengertian *Unified Modeling Language* (UML)

Sukamto dan Shalahuddin (2011:118), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

"Unified Modeling Language (UML) adalah salah satu standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek".

Gambar 2.1 Tampilan Logo UML

Menurut Sukamto dan Shalahuddin (2011:120-121) mengatakan, "Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori". Pembagian macam-macam diagram tersebut dapat dilihat pada gambar dibawah:



Gambar 2.2 Kategori dan Macam-macam Diagram UML

Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2011:121) :

1. *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagrams* yaitu kumpulam diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahana 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.

2.2.2 Jenis-Jenis Diagram UML

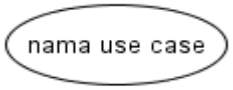
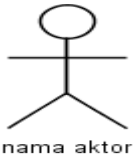
2.2.2.1 Pengertian *Use case* Diagram

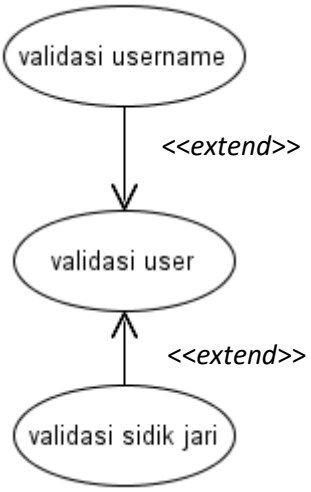
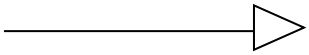
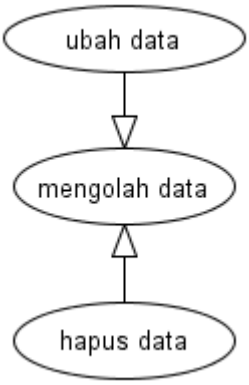
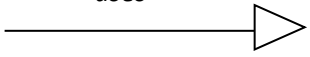
Sukamto dan Shalahuddin (2011:130), menjelaskan tentang *use case* diagram sebagai berikut :

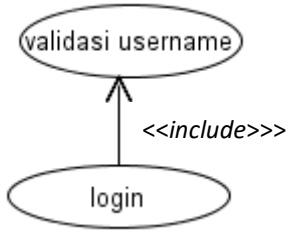
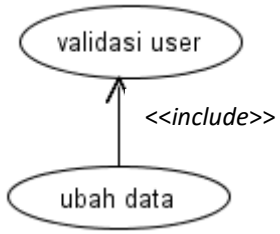
"Use case merupakan pemodelan untuk kelakuan *(behavior)* sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat".

Adapun simbol-simbol yang digunakan dalam *Use case* adalah sebagai berikut:

Tabel 2.1. Simbol-simbol Use case Diagram

Simbol	**Deskripsi**
Use case 	fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama *use case*
aktor / *actor* 	orang, proses, atau siatem 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
asosiasi / *association* _____	komunikasi antar aktor dan *use case* yang berpartisipasi pada *use case* atau *use case* memiliki ingeraksi dengan actor
ekstensi / *extend* 	relasi *use case* tambahan ke sebuah *use case* dimana *use case* yang di tambahkan dapat berdiri sendiri walau tanpa *use case* tambahan itu; mirip dengan prinsip *inheritance* pada pemrograman berorientasi objek; biasanya *use case* tambahan memiliki nama depan yang sama dengan *use case* yang ditambahkan, misalnya

	 arah panah mengarah pada *use case* yang ditambahkan; biasanya *use case* yang menjadi *extend*-nya merupakan jenis yang sama dengan *use case* yang menjadi induknya
Generalisasi / *generalization* 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah *use case* dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,  misalnya: arah panah mengarah pada *use case* yang menjadi generalisasinya (umum)
menggunakan / include / uses <<*include*>> <<*uses*>> 	relasi tambahan ke sebuah *use case* dimana *use case* yang ditambahkan memerlukan *use case* ini untuk menjalankan fungsinya atau sebagai syarat dijalankan *use case* ini ada dua sudut pandang yang cukup besar mengenai include di *use case*: 1. *Include* berarti *use case* yang

	ditambahkan akan selalu di panggil saat *use case* tambahan dijalankan, misalnya pada kasus berikut:  2. *Include* berarti *use case* yang tambahan akan selalu melakukan pengecekan apakah *use case* yang di tambahkan telah dijalankan sebelum *use case* tambahan dijalankan, misal pada kasus berikut:  kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

Sumber: Sukamto dan Shalahuddin (2011:131-133)

Sukamto dan Shalahuddin (2011:131) mengatakan, ada dua hal utama pada *use case* yaitu:

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

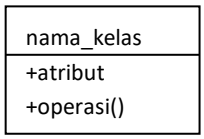
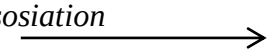
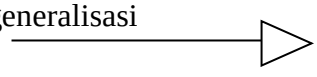
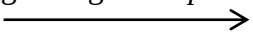
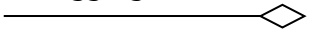
2.2.2.2 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2011:122), menjelaskan tentang *class diagram* sebagai berikut :

"Class Diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron".

Adapun simbol-simbol yang digunakan dalam *class* diagram adalah sebagai berikut:

Tabel 2.2 Simbol-simbol *Class Diagram*

Simbol	Deskripsi
kelas 	Kelas pada struktur sistem
antarmuka / interface  nama_interface	Sama dengan konsep interface dalam pemrograman berorientasi objek
asosiasi / association	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai *multiplicity*
asosiasi berarah / *directed assosiation* 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*
generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
kebergantungan / *dependency* 	Relasi antarkelas dengan makna kebergantungan antar kelas
agregasi / *aggregation* 	Relasi antarkelas dengan makna semua-bagian (*whole-part*)

Sumber: Sukamto dan Shalahuddin (2011:123-124)

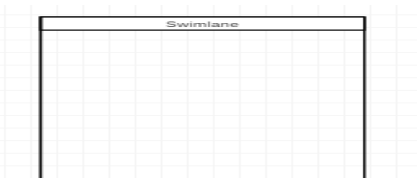
2.2.2.3 Pengertian *Activity Diagram*

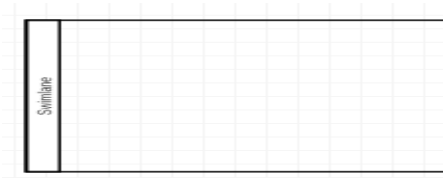
Sukamto dan Shalahuddin (2011:134), menjelaskan tentang *activity diagram* sebagai berikut :

"A*ctivity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem".

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

Tabel 2.3 Simbol-simbol *Activity Diagram*

Simbol	Deskripsi
Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / *decision* 	Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
Penggabungan / *join* 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

atau 	

Sumber: Sukamto dan Shalahuddin (2011:134-135)

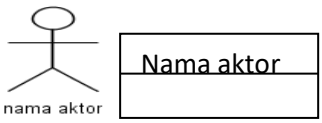
2.2.2.4 Pengertian *Sequence Diagram*

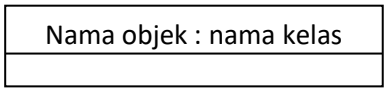
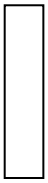
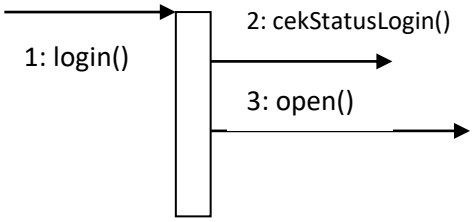
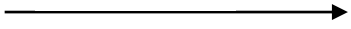
Sukamto dan Shalahuddin (2011:137) mengatakan, "Diagram sekuen menggambarkan kelakuan objek pada *Use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu".

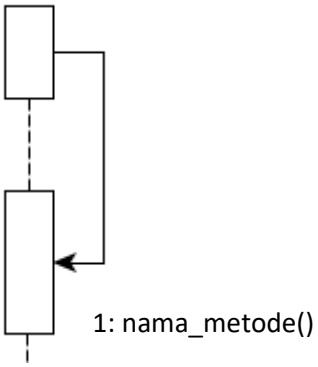
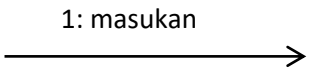
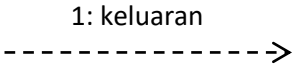
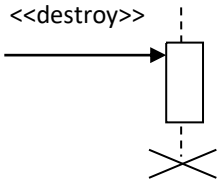
Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa *Sequence diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya. Berikut simbol-simbol pada *Sequence Diagram* :

Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

Simbol	Deskripsi
Actor 	orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu

Atau tanpa waktu aktif	merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor
Garis hidup / *lifeline* 	menyatakan kehidupan suatu objek
Objek 	menyatakan objek yang berinteraksi pesan
Waktu aktif 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya  maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif
Pesan tipe *create* <<create>> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call* <<create>> 	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 operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
Pesan tipe *send* 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi
Pesan tipe *return* 	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
Pesan tipe destroy 	menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada *create* maka ada destroy

Sumber: Sukamto dan Shalahuddin (2011:138-139)

2.3 Teori Judul

2.3.1 Pengertian Aplikasi

Ibisa dalam Dewi (2019:1) mengatakan, “Aplikasi adalah alat bantu untuk mempermudah dan mempercepat proses pekerjaan dan bukan merupakan beban bagi penggunanya”.

Pratama dalam Dewi (2019:1) mengatakan, “Aplikasi adalah suatu unit perangkat lunak komputer yang dibuat untuk melayani kebutuhan akan beberapa aktivitas”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa aplikasi adalah sebuah unit perangkat lunak computer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas atau kebutuhan pengguna menjadi lebih ringan.

2.3.2 Pengertian Korespondensi

Asnawi (2017:10) mengatakan, “Korespondensi adalah sarana untuk mengirim atau memberi infromasi tertulis kepada atasan atau pihak lain, baik sebagai laporan, pemberitahuan, permintaan, ataupun pertanyaan”.

Sugiarto dalam Asnawi (2017:18) mengatakan, “Korespondensi adalah sebagai suatu kegiatan saling berkirim surat oleh perorangan atau oleh organisasi”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa korespondensi adalah suatu sarana untuk berkomunikasi dalam bentuk surat yang berisi sutau infromasi, pertanyaan, ataupun permintaan kepada pihak lain baik individu atau organisasi.

2.3.3 Pengertian Website

Abdulloh (2018:1) mengatakan, “Website dapat diartikan sebagai kumpulan halaman yang berisi informasi data digital baik berupa teks, gambar, animasi, suara dan video atau gabungan dari semuanya yang disediakan melalui jalur koneksi internet sehingga dapat diakses dan dilihat oleh semua orang di seluruh dunia”.

2.3.4 Pengertian Inspektorat Pemerintah Kota Palembang

Saputra (2014:2) mengatakan, "Inspektorat Pemerintah Kota Palembang merupakan perangkat daerah kota Palembang sebagai unsur pengawas penyelenggaraan Pemerintah Daerah. Inspektorat dipimpin oleh seorang inspektur yang berada dibawah dan bertanggung jawab kepada Walikota melalui Sekretaris Daerah".

2.3.5 Pengertian Aplikasi Korespondesni (AKPON) Berbasis Web Pada Inspektorat Pemerintah Kota Palembang Sumatera Selatan

"Aplikasi Korespondesni (AKPON) Berbasis Web Pada Inspektorat Pemerintah Kota Palembang Sumatera Selatan adalah Suatu aplikasi yang dibuat dengan tujuan sebagai saran berkomunikasi dalam bentuk surat pada Inspektorat Pemerintah Kota Palembang kepada pihak lain dan juga dalam aplikasi ini bisa untuk pengarsipan surat".

2.4 Teori Program

2.4.1 Pengertian XAMPP

Dadan dan Kerendi dalam Pratiwi (2019:21) mengatakan, "*XAMPP* adalah salah satu aplikasi *web server apache* yang terintegrasi dengan *mysql* dan *phpmyadmin*".

Madcoms dalam Pratiwi (2019:21) mengatakan, "*XAMPP* adalah sebuah paket kumpulan software yang terdiri dari *Apache, MySQL, PhpMyAdmin, Perl, Filezilla* dan lain-lain".

Berdasarkan pengertian diatas, dapat disimpulkan bahwa *XAMPP* adalah sebuah aplikasi perangkat lunak pemrograman dan *database* yang di dalamnya terdapat berbagai macam aplikasi pemrograman yang terdiri dari Apache, *MySQL, PhpMyAdmin, Perl, Filezilla* dan lain-lain.

2.4.2 Pengertian *MySQL*

Kadir (2019:428) mengatakan, "MySQL adalah salah satu jenis database server yang sangat terkenal, MySQL mengguanakan SQL sebagai bahasa dasar untuk mengakses database-nya".

Sukamto dan Shalahuddin (2011:46), "SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus".

Berdasarkan pengertian diatas, dapat disimpulkan bahwa MySQL adalah bahasa yang digunakan untuk mengelola data SQL (*database management system*) atau DBMS yang *multitheard, multi-user.*

2.4.3 Pengertian *PHP*

Kadir (2019:1) mengatakan, "PHP singkattan dari *Hypertext Preprocessor*. Ia Merupakan bahasa berbentuk skrip tang ditempatkan dalam server dan diproses di server. Hasilnya yang dikirimkan ke klien, tempat pemakai menggunakan browser".

Madcoms dalam Pratiwi (2019:22) mengatakan, "PHP (*Hypertext preprocessor)* adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML".

Berdasarkan pengertian diatas, dapat disimpulkan bahwa PHP adalah bahasa pemograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada HTML.

Gambar 2.3. *Tampilan Logo PHP*

2.4.3.1 Sintaks Dasar PHP

Yuana dalam Pratiwi (2019:22), menjelaskan kode-kode PHP dituliskan diantara tanda berikut ini:

```
<?php
  …
  …
  …
?>
```

Atau

```
<?
  …
  …
  …
?>
```

Apabila membuat kode PHP dan berencana akan mendistribusikan ke pihak/orang lain, maka usahakan menggunakan sintaks <?php … ?>. Hal ini dikarenakan untuk penggunaan kode yang menggunakan <? … ?> terkadang tidak bisa dijalankan dalam *server* tertentu.

2.4.3.2 Tipe Data PHP

Murya dalam Pratiwi (2019:23) mengatakan, tipe data merupakan jenis dari suatu data yang akan diproses oleh bahasa pemrograman. Beberapa tipe data dalam PHP, sebagai berikut :

1. ***Integer*** merupakan tipe data yang berguna untuk menyimpan bilangan bulat. *Range* bilangan *integer* adalah antara -2.147.4833.647 sampai dengan 2.147.483.647.
2. ***Double Floating*** adalah tipe data yang berguna untuk menyimpan bilangan desimal. *Range* bilangan floating point antara 1e308 sampai dengan 1e308.
3. ***Boolean*** adalah tipe data yang paling sederhana, hanya berupa ***TRUE*** dan ***FALSE***.

4. ***String*** adalah tipe data yang terdiri dari kata, bias berupa kata tunggal maupun kalimat. Penulisan *string* harus diapit dengan tanda petik, baik berupa petik tunggal (‘…’) maupun petik ganda (“...”).
5. ***Objek*** adalah tipe data dibuat dengan tujuan agar para *programmer* terbiasa dengan OOP. Tipe data ini bias berupa bilangan.
6. ***Array*** merupakan ***Tipe Compound Primitif,*** terdapat pada bahasa pemrograman lain.
7. ***Null*** adalah tipe data yang tidak memuat apapun. Setiap variabel yang diset menjadi tipe data Null, ini akan menjadikan variabel tersebut kosong.
8. ***Resources*** tipe data spesial yang satu ini dikhususkan untuk menyimpan *resources*, sumber atau alamat.

2.4.4 Pengertian PHPMyAdmin

Hikmah, dkk dalam Pratiwi (2019:24) mengatakan, “PHPMyAdmin merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi tabel, maupun mengirim database secara cepat dan mudah tanpa harus menggunakan perintah (*command*) *SQL*”.

Rahman dalam Pratiwi (2019:24) mengatakan, “PHPMyAdmin adalah aplikasi PHP sebagai administrator *MySQL*, PHPMyAdmin mendukung berbagai aktivitas *MySQL* seperti pengelolaan data, *table*, relasi antar *table* dan lain sebagainya”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa PHPMyAdmin adalah aplikasi PHP sebagai administrator *MySQL* yang digunakan untuk membuat *database*, mengelola tabel, mengelola data, relasi antar tabel, dan mengirim database secara praktis tanpa harus menggunakan perintah (*command*) *SQL*.

Gambar 2.4. Tampilan logo PHPMyAdmin

2.4.5. Pengertian Basis Data

Abdulloh (2018:103) mengatakan, “Database atau basis data adalah kumpulan informasi yang disimpan dalam computer secara sistematik sehingga dapat diperiksa menggunakan suatu program computer untuk memperoleh informasi”.

Sukamto dan Shalahuddin (2011:44) mengatakan, “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau infromasi dan membuat infromasi tersedia saat dibutuhkan”.

Berdasarkan pengertian diatas, dapat disimpulkan bahwa basis data adalah sistem komputerisasi yang saling berhubungan berfungsi sebagai media penyimpanan data dan menyediakan informasi saat dibutuhkan.