



BAB II

TINJAUAN PUSTAKA

2.1 Teori Umum

2.1.1 Pengertian Komputer

Kadir (2017:2), “Komputer merupakan kumpulan peralatan elektronik yang bermanfaat untuk melaksanakan berbagai pekerjaan yang dilakukan oleh manusia”.

Irma (2016:1), “Secara definisi komputer merupakan sekumpulan alat elektronik yang saling bekerja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) yang terkoordinasi dibawah kontrol program yang tersimpan dimemorinya”.

Penulis menyimpulkan bahwa komputer adalah sebuah alat elektronik yang dapat mengelola data menjadi informasi yang berguna bagi manusia, komputer tidak hanya dipergunakan sebagai alat yang mampu melakukan proses perhitungan saja tetapi dapat digunakan disemua bidang yang berguna dalam mempermudah pekerjaan manusia.

2.1.2 Pengertian Perangkat Lunak

Utami dan Asnawati (2015:2), “*Software* adalah perangkat lunak yang berisikan sebuah intruksi yang diperintahkan dan diproses dengan bantuan perangkat keras sehingga tanpa perangkat lunak, perangkat keras tidak bisa dipakai sehingga *software* dan *hardware* tidak bisa dipisahkan”.

Kadir (2017:2), “Mengatakan bahwa perangkat lunak adalah sebuah perintah yang diberikan oleh manusia untuk menjalankan sebuah tugas-tugas tertentu didalam komputer”.

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



Jadi didapatkan kesimpulan bahwa Perangkat Lunak adalah program komputer yang dibuat dan diproses dengan bantuan perangkat keras untuk membantu mempermudah pekerjaan manusia dalam mengerjakan tugas-tugas tertentu.

2.1.3 Pengertian Basis Data

Fathansyah (2018:2), “Basis data merupakan kumpulan dari data (arsip) yang berhubungan dan diorganisasikan sedemikian rupa supaya dapat digunakan dengan cepat dan mudah”.

Sukamto dan Shalahuddin (2018:43), “Basis data adalah sistem komputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”.

Abdulloh (2018:7), “Basis data (*data base*) adalah kumpulan informasi yang disimpan dalam komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi”.

Dari beberapa pendapat mengenai definisi 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

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukamto dan Shalahuddin (2018:125), “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*)”. Adapun tahap-tahap (*fase*) dalam metode pengembangan RUP menurut Sukamto dan Shalahuddin (2018:128-131) adalah sebagai berikut:



1. *Inception* (permulaan)

Tahap ini lebih pada memodelkan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

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*).

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.

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/tonggak kemampuan operasional awal. Aktifitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.

2.2 Teori Judul

2.2.1 Pengertian Sistem

Menurut Subtari dalam Herliana dan Rasyid (2016:42), “Sistem didefinisikan sebagai sekumpulan unsur-unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu”.

Pratama (2014:7), “Sistem merupakan kumpulan prosedur yang saling berkaitan dan berhubungan untuk melakukan tugas secara bersama-sama”.



Taufiq (2018:2), “Sistem adalah kumpulan dari sub-sub sistem baik abstrak maupun fisik yang saling terintegrasi dan berkolaborasi untuk mencapai suatu tujuan tertentu”.

Dari definisi diatas penulis menyimpulkan bahwa sistem adalah sekumpulan komponen yang saling berhubungan satu sama lain dan bekerja sama dalam melaksanakan tugas-tugas tertentu untuk mencapai suatu tujuan.

2.2.2 Pengertian Informasi

Taufiq (2018:2), “Informasi adalah data-data yang diolah sehingga memiliki nilai tambah dan bermanfaat bagi pengguna”.

Muslihudin dan Oktafianto (2016:9), menjelaskan bahwa “Informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan”.

Herliana dan Rasyid (2016:42-43), “Informasi dapat didefinisikan sebagai suatu hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata yang digunakan untuk pengambilan keputusan. Dalam arti yang lebih singkat, informasi merupakan hasil olah dari data sehingga menjadi sebuah bentuk yang lain, yang lebih berguna bagi penerimanya”.

Berdasarkan pengertian diatas penulis menyimpulkan bahwa informasi adalah kumpulan data yang diolah menjadi sebuah bentuk yang baru dan berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi.

2.2.3 Pengertian Sistem Informasi

Pratama (2014:10) “Menyatakan bahwa sistem informasi merupakan gabungan dari empat bagian utama. Bagian tersebut mencakup perangkat lunak (*software*), perangkat keras (*hardware*), infastruktur, dan sumber daya manusia (SDM) yang terlatih. Bagian tersebut saling berkaitan untuk mengolah data menjadi informasi yang bermanfaat”.



Subtari dalam Pramana dan Dewantara (2018:22), “ Sistem Informasi merupakan suatu sistem dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi dari organisasi agar dapat menyediakan laporan-laporan yang diperlukan pihak luar”.

Dari beberapa pendapat tersebut dapat disimpulkan bahwa sistem informasi adalah gabungan dari sub0sub sistem yang terbagi menjadi empat bagian utama yaitu perangkat lunak (*software*), perangkat keras (*hardware*), infrastruktur, dan sumber daya manusia (SDM) yang terlatih dan saling berhubungan untuk mengelola informasi serta menyediakan laporan-laporan yang bermanfaat bagi pengguna.

2.2.4 Pengertian Karyawan

Siagian (2018:65), “Karyawan adalah aset utama perusahaan yang menjadi perencanaan dan pelaku aktif dari aktivitas organisasi dan merupakan seorang pekerja yang bekerja dibawah perintah orang lain dan mendapat kompensasi suatu jaminan”.

Menurut Fajarianto, dkk (2017:49), “Karyawan merupakan salah satu bagian yang sangat penting bagi perusahaan karena tanpa adanya karyawan tidak akan terjadi produktivitas didalam perusahaan tersebut”.

Dari definisi diatas penulis menyimpulkan bahwa karyawan adalah bagian dari sebuah perusahaan atau organisasi yang sangat berpengaruh kegiatan aktivitasnya untuk kemajuan produktivitas didalam perusahaan atau organisasi tersebut.

2.2.5 Pengertian Baru

Dikutip dari Kamus Besar Bahasa Indonesia (2015:137) baru adalah belum pernah ada (dilihat) sebelumnya, belum pernah didengar sebelumnya, belum lama selesai (dibuat, didirikan) , belum pernah dipakai dan belum lama dibeli



2.2.6 Pengertian Rekrutmen (*Recruitment*)

Bagan dikutip Permana dan Dewantara (2018:21), “Rekrutmen adalah penarikan karyawan berkaitan dengan pemenuhan kebutuhan karyawan sesuai dengan jumlah dan kualitasnya”.

Menurut Handoko dikutip Potale, dkk (2016:454), “Rekrutmen merupakan proses pencarian dan “pemikatan” para calon karyawan (pelamar) yang mampu untuk melamar sebagai karyawan”.

Penulis menyimpulkan bahwa pengertian dari rekrutmen (*recruitment*) adalah proses pencarian dan penarikan seseorang sesuai dengan kebutuhan untuk mengisi jabatan atau pekerjaan yang kosong pada sebuah perusahaan atau organisasi.

2.2.7 Sistem Informasi Penerimaan Karyawan Baru (*E-Recruitment*) pada PT Nusantara Jaya Patria

“Sistem Informasi Penerimaan Karyawan Baru (*E-Recruitment*) pada PT Nusantara Jaya Patria adalah Sistem Informasi yang digunakan untuk mempermudah calon karyawan dalam proses perekrutan mulai dari proses pendaftaran, penyeleksian administrasi dan akademik, hingga pengumuman peserta yang lulus menjadi karyawan pada PT Nusantara Jaya Patria”

2.3 Teori Khusus

2.3.1 Pengertian *Unified Modeling Language* (UML)

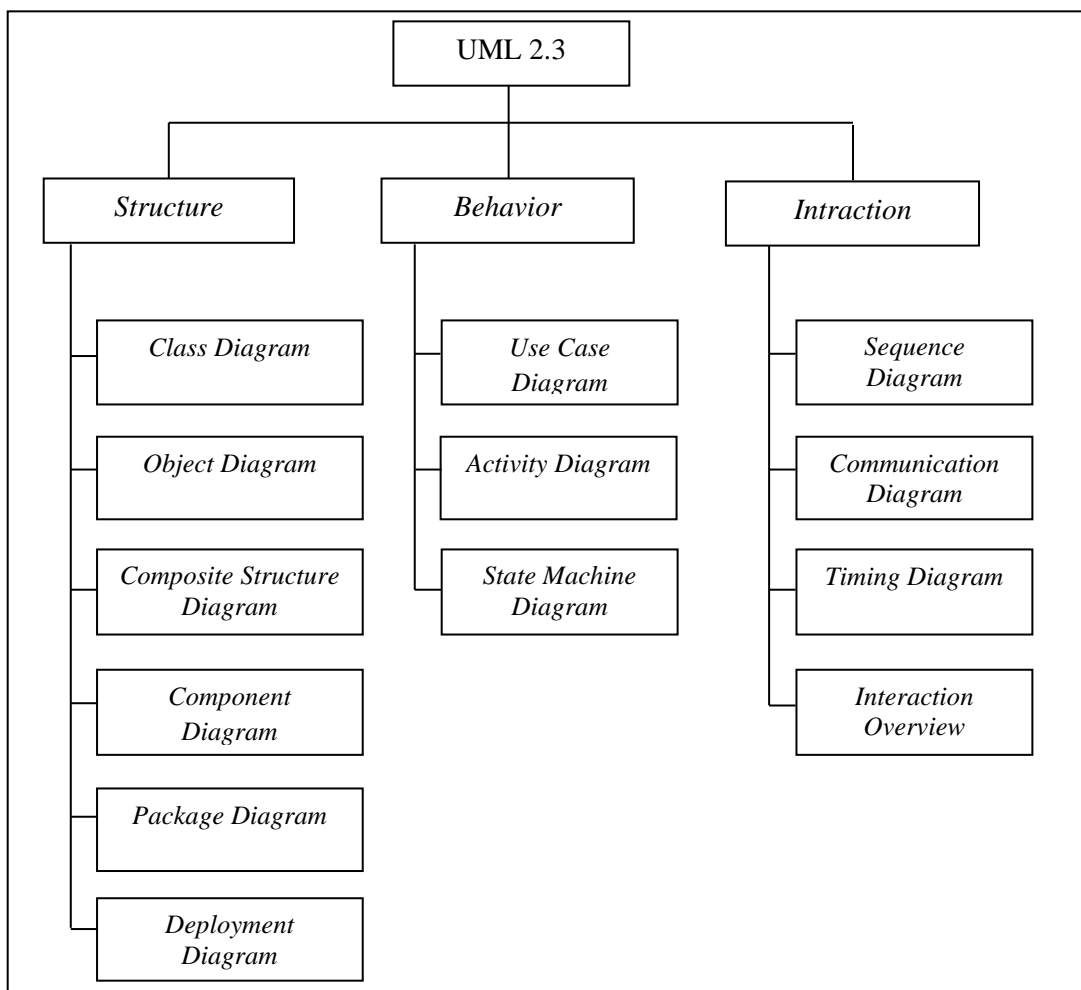
Sukanto dan Shalahuddin (2018:133), menjelaskan tentang pengertian *Unified Modeling Language* sebagai berikut :

“*Unified Modeling Language* (UML) adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.”



Gambar 2.1 Tampilan Logo *UML*

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 2.2 Kategori dan Macam-macam Diagram UML



Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2018:141) :

- 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 Jenis-Jenis Diagram UML

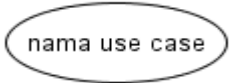
2.3.2.1 Pengertian *Use case Diagram*

Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case* diagram sebagai berikut :

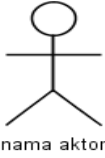

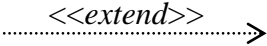
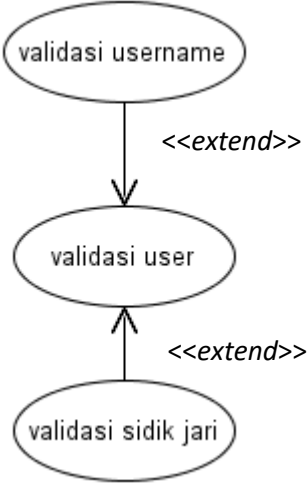
“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.”

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

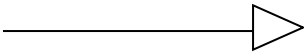
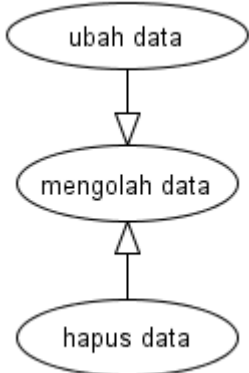
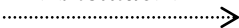
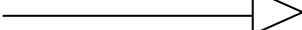
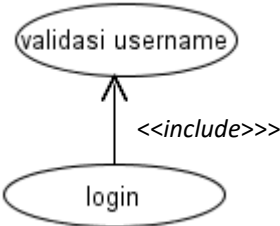
Tabel 2.1. Simbol-simbol *Use case Diagram*

| 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> |

Lanjutan Simbol-simbol *Use case* Diagram

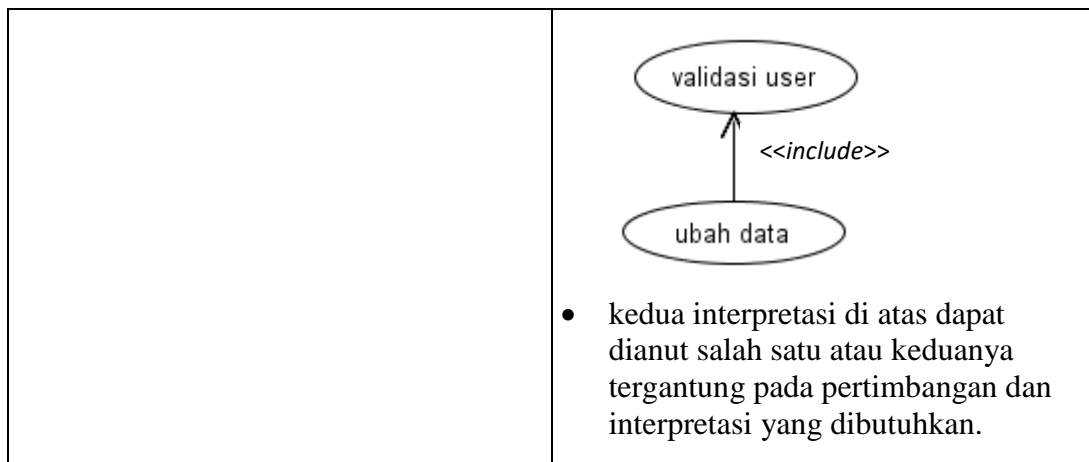
| | |
|---|--|
| <p>aktor / <i>actor</i></p>  <p>nama aktor</p> | <p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan 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> atau <i>use case</i> memiliki interaksi dengan actor</p> |
| <p>ekstensi / <i>extend</i></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>  <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 Simbol-simbol *Use case Diagram*

| | |
|---|--|
| <p>Generalisasi / <i>generalization</i></p>  | <p>hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya,</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p> |
| <p>menggunakan / <i>include / uses</i></p> <p><code><<include>></code> </p> <p><code><<uses>></code> </p> | <p>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</p> <p>ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i>:</p> <ul style="list-style-type: none"> • <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:  <ul style="list-style-type: none"> • <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut: |



Lanjutan Simbol-simbol *Use case Diagram*



Sumber: Sukamto dan Shalahuddin (2018:156)

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.

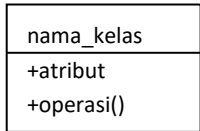
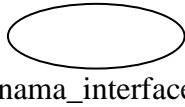

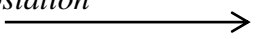
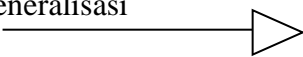
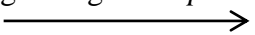
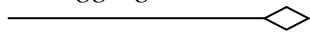
2.3.2.2 Pengertian *Class Diagram*

Sukamto dan Shalahuddin (2018:141), 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  | 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)






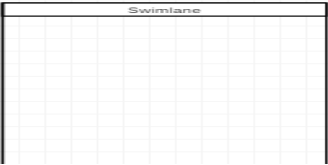
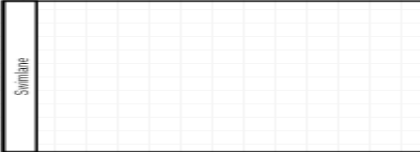
2.3.2.3 Pengertian *Activity Diagram*

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.”

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 / <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 |
| Swimlane  atau  | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi |

Sumber: Sukamto dan Shalahuddin (2018:164)

2.3.2.4 Pengertian *Sequence Diagram*

(Sukamto dan Shalahuddin, 2018:165) “Diagram sekuen menggambarkan kelakuan objek pada *Use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk





menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *Use case*”

Banyaknya diagram sekuen yang harus digambarkan adalah minimal 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.


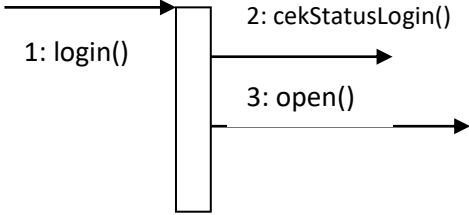


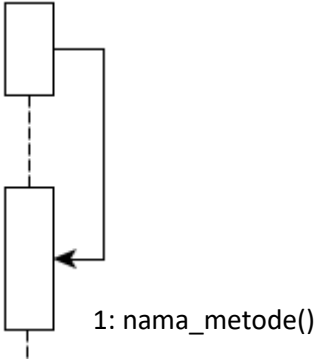
Dapat penulis simpulkan bahwa *Sequence diagram* adalah penggambaran skenario dari sebuah objek yang ada pada *use case* yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.

Berikut simbol-simbol pada *Sequence Diagram* :


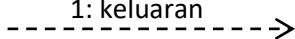
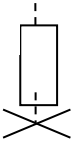
Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

| Simbol | Deskripsi | | |
|--|---|--|---|
| <p>Actor</p>  <p>nama aktor</p> <p>atau</p> <table border="1" data-bbox="392 1464 592 1554"> <tr> <td>Nama aktor</td> </tr> <tr> <td> </td> </tr> </table> <p>tanpa waktu aktif</p> | Nama aktor | | <p>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 merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama actor</p> |
| Nama aktor | | | |
| | | | |
| <p>Garis hidup / <i>lifeline</i></p>  | <p>menyatakan kehidupan suatu objek</p> | | |
| <p>Objek</p> <table border="1" data-bbox="408 1883 788 1962"> <tr> <td>Nama objek : nama kelas</td> </tr> <tr> <td> </td> </tr> </table> | Nama objek : nama kelas | | <p>menyatakan objek yang berinteraksi pesan</p> |
| Nama objek : nama kelas | | | |
| | | | |

Lanjutan Simbol-simbol pada *Sequence Diagram*

| | |
|--|--|
| <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, misalnya</p>  <p>maka <code>cekStatusLogin ()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> aktor tidak memiliki waktu aktif</p> |
| <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> |
| <p>Pesan tipe <i>call</i></p> <p><code><<create>></code></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 ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi</p> |

Lanjutan Simbol-simbol pada *Sequence Diagram*

| | |
|---|--|
| Pesan tipe <i>send</i> 1: masukan  | menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim |
| Pesan tipe <i>return</i> 1: 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 |
| Pesan tipe <i>destroy</i> <<destroy>>  | 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: Sukamto dan Shalahuddin (2018:165-167)

2.4 Teori Program

2.4.1 Pengertian XAMPP

Haqi dan Setiawan (2019:8), “Xampp adalah perangkat lunak bebas (*free software*) yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsi xampp sendiri sebagai server yang berdiri sendiri (*localhost*), yang terdiri dari beberapa program, antara lain: *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman *PHP* dan *Perl*”.

Suryatiningsih dikutip Putra (2019:11), “*XAMPP* merupakan *tool* yang menyediakan paket perangkat lunak kedalam satu buah paket. Dengan menginstal *XAMPP* maka tidak perlu lagi melakukan instalasi dan konfigurasi web server *Apache*, *PHP* dan *MySQL* secara manual. *XAMPP* akan menginstal dan mengkonfigurasi secara otomatis”.

Jadi, penulis simpulkan pengertian *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

Sukamto dan Shalahuddin (2018:46), “SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus.”

Haqi dan Setiawan (2019:8) “MySQL adalah sebuah perangkat lunak sistem manajemen basis data *SQL (database management system)* atau DBMS yang *multithreaded, multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia”.

Menurut Mundzir (2018:217) “MySQL adalah sistem manajemen *database SQL* yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database MySQL* mampu mendukung beberapa fitur seperti *multithreaded, multi-user*, dan *SQL database management system (DBMS)*”.

Jadi dapat penulis simpulkan pengertian MySQL adalah bahasa yang digunakan untuk mengelola data SQL (*database management system*) atau DBMS yang *multithreaded, multi-user*.

2.4.3 Pengertian PHP

Menurut Raharjo (2016:38) “*Php* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web”.

Sidik (2017:4) “*Php* merupakan bahasa utama *script server-side* yang disisipkan pada *HTML* yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi dekstop”.

Nurmalasari, dkk (2019:10), “*Php* adalah bahasa pemrograman yang digunakan bersamaan dengan *HTML* dijalankan di sisi web browser”.

Jadi, dapat penulis simpulkan pengertian PHP adalah bahasa pemrograman berbasis *server-side* yang bisa kita gunakan untuk membuat aplikasi web yang disisipkan pada *HTML*.



Gambar 2.3. Tampilan Logo PHP

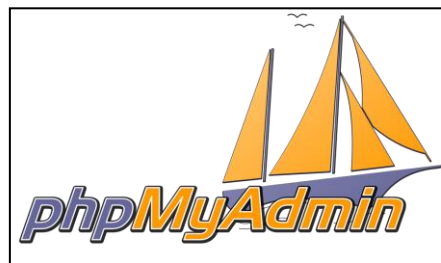


2.4.4 Pengertian *PHPMYAdmin*

Nurmalasari, dkk (2019:9), “*PhpMyadmin* adalah sebuah aplikasi pemrograman yang digunakan untuk manajemen database melalui browser (web) untuk mengontrol data mereka dan isi web yang akan ditampilkan dalam sebuah *website* yang mereka buat tanpa harus menggunakan perintah (*command*) *SQL*”.

Pradana, dkk (2017:39), “*PhpMyAdmin* adalah sebuah aplikasi/perangkat lunak bebas (*opensource*) yang ditulis dalam bahasa pemrograman *PHP* yang digunakan untuk menangani administrasi *database MySQL* melalui jaringan lokal maupun internet. *phpMyAdmin* mendukung berbagai operasi *MySQL*, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain)”.

Jadi, penulis simpulkan pengertian *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*