



BAB II

TINJAUAN PUSTAKA

2.1. Teori Umum

2.1.1. Pengertian Komputer

Wahyudi (2012:3), komputer adalah peralatan (*device*) yang menerima data (*input*) dan menyimpan (*storage*) kemudian diproses (*process*) untuk menghasilkan data dalam bentuk lain.

Williams dan Sawyer (2010:4), komputer adalah mesin yang bisa diprogram dan memiliki beragam fungsi untuk menerima data, baik data mentah maupun angka, lalu memproses atau mengubahnya ke dalam bentuk informasi yang bisa kita manfaatkan, misal bentuk ringkasan, jumlah total (dalam penghitungan), dan juga laporan.

Jadi komputer adalah suatu alat elektronik yang bisa melakukan beberapa tugas tertentu seperti menerima data (*input*), menyimpan (*storage*), memproses (*process*), dan menyediakan keluaran (*output*) dalam bentuk informasi.

2.1.2. Pengertian Data

Sutabri (2012:1), data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuannyata. Data merupakan bentuk jamak dari bentuk tunggal datum.

Wahyudi (2012:3), data adalah suatu *file* atau *field* yang berupa karakter atau tulisan dan gambar. Data juga dapat dibedakan berdasarkan kebutuhan, ada data mentah, atau data yang telah jadi (informasi).

Williams dan Sawyer (2010:25), data adalah sesuatu yang terdiri dari fakta-fakta dan gambaran mentahan yang akan diproses menjadi informasi. Misalnya, suara untuk kandidat pemilu yang berbeda-beda.

Jadi data adalah suatu gambaran kenyataan yang mewakili suatu objek tertentu berupa karakter, simbol, tulisan, gambar, ataupun kombinasinya.



2.1.3. Pengertian Informasi

Sutabri (2012:1), informasi adalah data yang diproses ke dalam suatu bentuk yang mempunyai arti bagi penerima dan mempunyai nilai nyata dan terasa bagi keputusan saat itu atau keputusan mendatang.

Williams dan Sawyer (2010:25), informasi adalah data yang telah dirangkum atau dimanipulasi dalam bentuk lain untuk tujuan pengambilan keputusan.

Jadi informasi adalah data yang telah diproses ke dalam bentuk yang lebih berarti bagi penerimanya guna meningkatkan pengetahuan seseorang yang menggunakan data tersebut.

2.1.4. Pengertian Aplikasi

Hengki menyatakan aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti system perniagaan, *game* pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia.

Santoso menyatakan aplikasi adalah suatu kelompok file (*form, class, report*) yang bertujuan untuk melakukan aktivitas tertentu yang saling terkait.

2.1.5. Pengertian Pengolahan data

Ladjamudin (2013:9), pengolahan data adalah masa atau waktu yang digunakan untuk mendeskripsikan perubahan bentuk data menjadi informasi yang memiliki kegunaan.

Sutabri (2012:6), pengolahan data adalah suatu proses yang terdiri dari kegiatan-kegiatan penyimpanan data (*data storage*) dan penanganan data (*data handling*).

Jadi pengolahan data adalah bentuk pengolahan data menjadi informasi membuat data itu berguna sesuai hasil yang diinginkan agar bisa digunakan.



2.1.6. Pengertian Anggota

Kamus Besar Bahasa Indonesia (KBBI), anggota adalah orang yang menjadi bagian atau masuk ke dalam suatu golongan tertentu seperti organisasi, perserikatan, dewan, ataupun panitia.

2.1.7. Pengertian Tentara Nasional Indonesia (TNI)

Dinas Penerangan TNI-AD dalam situs resminya menyatakan bahwa Tentara Nasional Indonesia atau biasa disingkat TNI adalah nama sebuah angkatan perang dari negara Indonesia. Tentara Nasional Indonesia (TNI) terdiri dari tiga angkatan bersenjata, yaitu TNI Angkatan Darat, TNI Angkatan Laut, dan TNI Angkatan Udara. TNI dipimpin oleh seorang Panglima TNI, sedangkan masing-masing angkatan dipimpin oleh seorang Kepala Staf Angkatan.

2.1.7.1. TNI-AD

Dinas Penerangan TNI-AD dalam situs resminya menyatakan bahwa Tentara Nasional Indonesia Angkatan Darat (atau biasa disingkat TNI Angkatan Darat atau TNI-AD) adalah salah satu cabang angkatan perang dan merupakan bagian dari Tentara Nasional Indonesia (TNI) yang bertanggung jawab atas operasi pertahanan negara Republik Indonesia di darat. Tugas pokok sesuai dengan pasal 7 ayat (1) TNI AD adalah menegakkan kedaulatan negara, mempertahankan keutuhan wilayah Negara Kesatuan Republik Indonesia (NKRI) yang berdasarkan Pancasila dan UUD RI Tahun 1945, serta melindungi segenap bangsa dan seluruh tumpah darah Indonesia dari ancaman dan gangguan terhadap keutuhan bangsa dan negara. Perwira tersenior Angkatan Darat, Kepala Staf TNI Angkatan Darat, adalah perwira tinggi berbintang empat dengan pangkat Jenderal mengepalai Angkatan Darat di bawah Panglima TNI. Di TNI Angkatan Darat, kepangkatan terdiri dari Perwira, Bintara dan Tamtama. Adapun pangkat tertinggi di Angkatan Darat adalah Jenderal Besar dengan bintang lima.

2.2. Teori Khusus

2.2.1. Kamus Data (*Data Dictionary*)

Rosa dan M. Shalahuddin (2014:73), kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data biasanya berisi:

- Nama — nama dari data.
- Digunakan pada — merupakan proses-proses yang terkait data.
- Deskripsi — merupakan deskripsi data.
- Informasi tambahan — seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel 2.1. Simbol-simbol pada kamus data

No	Simbol	keterangan
1.	=	Disusun atau terdiri dari
2.	+	Dan
3.	[]	Baik...atau...
4.	{ } ⁿ	N kali diulang/bernilai banyak
5.	()	Data optional
6.	*...*	Batas komentar

2.2.2. Basis Data (*Database*)

Winarno dan Ali Zaki (2014:102), *database* merupakan suatu tempat untuk menyimpan data yang jenisnya beraneka ragam. Keuntungan menyimpan data di *database* adalah kemudahan dalam penyimpanan dan menampilkan data karena dalam bentuk tabel.



Rosa dan M. Shalahudin (2014:43) Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat.

Subekti, Ichsan Risnandar, dan Hafni S.S., (2013:90) basis data adalah kumpulan data yang tersimpan dalam tabel-tabel. Tabel-tabel itu disusun berdasarkan baris dan kolom.

2.2.3. Unified Modeling Language (UML)

2.2.3.1. Pengenalan Unified Modeling Language (UML)

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. *UML* merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung.

UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

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 (Rosa A.S dan M. Shalahudin, 2014:133).

2.2.3.2. Sejarah UML

Bahasa pemrograman berorientasi objek yang pertama dikembangkan dikenal dengan nama Simula-67 yang dikembangkan pada tahun 1967.

Perkembangan aktif dari pemrograman berorientasi objek mulai menggeliat ketika berkembangnya bahasa pemrograman Smalltalk pada awal 1980-an yang kemudian diikuti dengan perkembangan bahasa pemrograman berorientasi objek yang lainnya seperti C objek, C++, Eiffel, dan CLOS.

Sekitar lima tahun setelah Smalltalk berkembang, maka berkembang pula metode pengembangan berorientasi objek. Karena banyaknya metodologi-



metodologi yang berkembang pesat saat itu, maka muncullah ide untuk membuat sebuah bahasa yang dapat dimengerti semua orang. Maka dibuat bahasa yang merupakan gabungan dari beberapa konsep, seperti konsep *Object Modeling Technique (OMT)* dari Rumbaugh dan Booch (1991), konsep *The Classes, Responsibilities, Collaborators (CRC)* dari Rebecca Wirfs-Brock (1990), konsep pemikiran Ivar Jacobson, dan beberapa konsep lainnya dimana James R. Rumbaugh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan *Unified Modeling Language (UML)*.

Pada tahun 1996, *Object Management Group (OMG)* mengajukan proposal agar adanya standarisasi pemodelan berorientasi objek dan pada bulan September 1997 *UML* diakomodasi oleh *OMG* sehingga sampai saat ini *UML* telah memberikan kontribusinya yang cukup besar di dalam metodologi berorientasi objek dan hal-hal yang terkait di dalamnya (Rosa A.S dan M. Shalahudin, 2014:138).

2.2.3.3. Diagram *UML*

Rosa A.S dan M. Shalahudin (2014:140), pada *UML* terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Berikut ini penjelasan singkat dari pembagian kategori tersebut.

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. *Structure diagram* terdiri dari *class diagram*, *object diagram*, *component diagram*, *composite structure diagram*, *package diagram* dan *deployment diagram*.
2. *Behavior diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Behavior diagram terdiri dari *Use case diagram*, *Activity diagram*, *State Machine System*.
3. *Interaction diagram* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence*

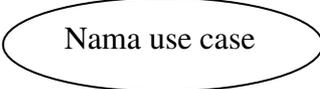
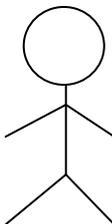
Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.

2.2.4. Use Case Diagram

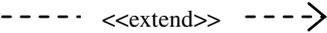
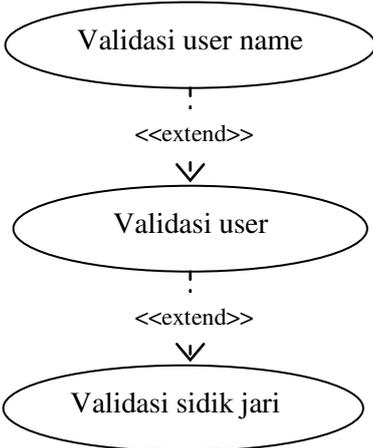
Rosa dan M. Shalahudin (2014:155), *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

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

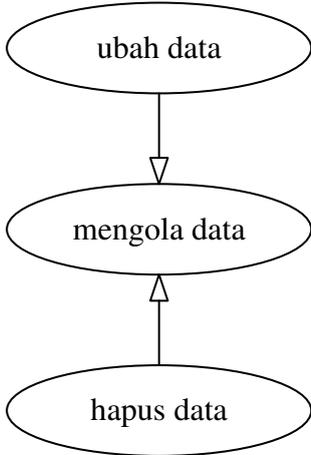
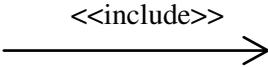
Tabel 2.2. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
1.	<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> .
2.	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.

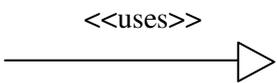
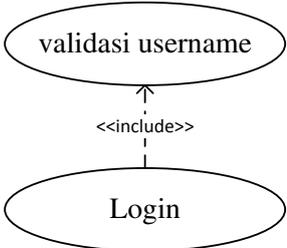
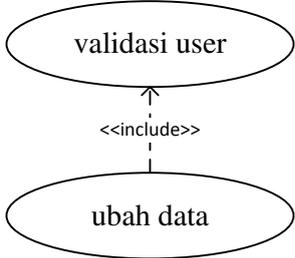
Lanjutan Tabel 2.2. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
3.	Assosiasi/ <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.	Exstensi/ <i>extend</i> 	<p>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, misal</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 Tabel 2.2. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
5.	<p>Generalisasi/<i>generalization</i></p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua <i>buah use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya :</p>  <p>arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>
6.	<p>Menggunakan / <i>include / uses</i></p> 	<p>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</p> <p>Ada dua sudut pandang yang cukup besar mengenai <i>include</i> 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, missal pada kasus berikut :

Lanjutan Tabel 2.2. Simbol-simbol diagram *use case*

No.	Simbol	Deskripsi
		<div style="text-align: center;">  <pre> graph BT Login((Login)) -.-> <<include>> ValidasiUsername((validasi username)) </pre> </div> <p>- <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 di jalankan, misal pada kasus berikut :</p> <div style="text-align: center;">  <pre> graph BT UbahData((ubah data)) -.-> <<include>> ValidasiUser((validasi user)) </pre> </div> <p>Kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Rosa A.S dan M. Shalahudin (2014:156)

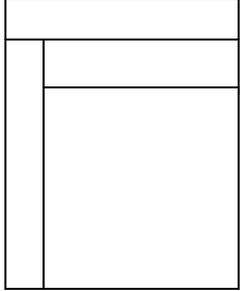
2.2.5. Activity Diagram

Rosa dan M. Shalahudin (2014:161), diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang

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

Berikut adalah simbol-simbol yang ada pada diagram aktivitas :

Tabel 2.3. Simbol-simbol *activity diagram*

No.	Simbol	Deskripsi
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Percabangan/ <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4.	Penggabungan/ <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status akhir 	Status akhir yang dilakukan oleh sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber : Rosa A.S dan M. Shalahudin (2014:162)

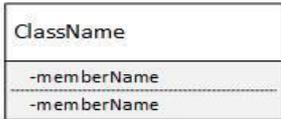
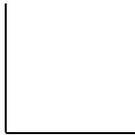
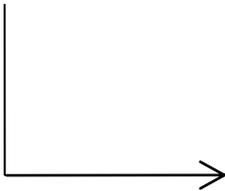
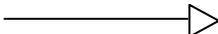
2.2.6. Class Diagram

Rosa dan M. Shalahudin (2014:141), 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 *method* :

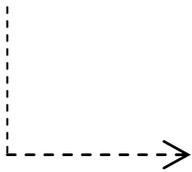
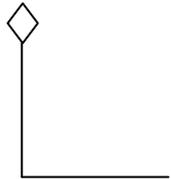
1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.4 Simbol-simbol *class diagram*

No.	Simbol	Deskripsi
1.		Kelas pada struktur sistem
2.	<p>Antarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek
3.	<p>Asosiasi/<i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
4.	<p>Asosiasi berarah/<i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>
5.	<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)

Lanjutan Tabel 2.4 Simbol-simbol *class diagram*

No.	Simbol	Deskripsi
6.	Kebergantungan/ <i>dependensi</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas
7.	Agrgasi/ <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)

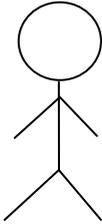
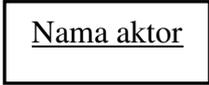
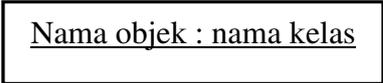
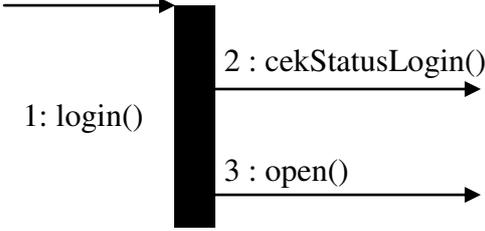
Sumber : Rosa A.S dan M. Shalahudin (2014:146)

2.2.7. Sequence Diagram

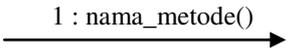
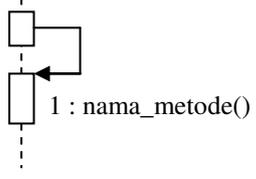
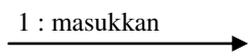
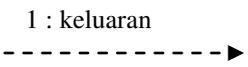
Rosa dan M. Shalahudin (2014:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksinya jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

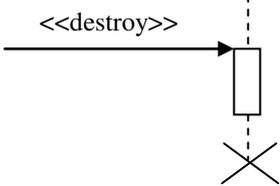
Tabel 2.5. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>Atau</p>  <p>Tanpa waktu aktif</p>	<p>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, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.</p>
2.	<p>Garis hidup/<i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
3.	<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
4.	<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya</p> 

Lanjutan Tabel 2.5. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
		Maka cekStatusLogin() dan open() dilakukan didalam metode login(). Aktor tidak memiliki waktu aktif
5.	<p>Pesan tipe <i>create</i></p> 	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6.	<p>Pesan tipe <i>call</i></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>
7.	<p>Pesan tipe <i>send</i></p> 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim
8.	<p>Pesan tipe <i>return</i></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

Lanjutan Tabel 2.5. Simbol-simbol *sequence diagram*

No.	Simbol	Deskripsi
9.		Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Rosa A.S dan M. Shalahudin (2014:165)

2.3. Teori Program

2.3.1. Hyper Text Markup Language (HTML)

Winarno dan Ali Zaki (2014:1) *Hyper Text Markup Language (HTML)* adalah sebuah bahasa untuk menampilkan konten di *web*. *HTML* sendiri adalah bahasa pemrograman yang bebas, artinya tidak dimiliki oleh siapapun, pengembangannya dilakukan oleh banyak orang di banyak negara dan bisa dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global.

2.3.1.1. Struktur Dasar HTML

Winarno dan Ali Zaki (2014:3), Elemen *HTML* dimulai dengan *tag* awal, yang diikuti dengan isi elemen dan *tag* akhir. *Tag* terakhir termasuk simbol/diikuti oleh tipe elemen, misalnya `</HEAD>`. Sebuah elemen *HTML* dapat bersarang di dalam elemen lainnya. Sebuah dokumen *HTML* standar terlihat seperti ini :

```
<html>
  <head>
    <title> Ini judul dokumen HTML </title>
  </head>
  <body>
    Teks ini adalah teks yang muncul di Body dari dokumen
  </body>
</html>
```



2.3.2. PHP

2.3.2.1. Pengertian PHP

Winarno dan Ali Zaki (2014:49), *PHP* adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode *PHP* dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*). *PHP* adalah bahasa *script* yang sangat cocok untuk pengembangan *web* dan dapat dimasukkan ke dalam *HTML*.

Subekti, Ichsan Risnandar, Hafni S.S., dan Teguh N (2013:57), *PHP* (*Hypertext Preprocessing*) merupakan bahasa pemrograman yang biasa digunakan untuk membuat halaman *HTML*. File *.php* yang dibuat akan diproses di dalam *server*, sedangkan halaman yang akan dikirim ke *browser* pengunjung hanyalah tampilan *HTML*-nya. Dengan *PHP*, halaman *website* yang dibuat akan menjadi dinamis, yakni dapat selalu berubah tanpa harus mengubah isi *website* secara manual. Informasi akan diproses ulang oleh *web server* sehingga akan didapatkan isi paling mutakhir dari halaman *web*.

2.3.2.2. Kelebihan PHP

Winarno dan Ali Zaki (2014:49), *PHP* banyak dipakai oleh banyak orang adalah perangkat lunak bebas (*open source*) yang dirilis di bawah lisensi *PHP*. Artinya untuk menggunakan bahasa pemrograman ini gratis, bebas, dan tidak terbuka. Untuk *web*, *PHP* adalah bahasa *scripting* yang bisa dipakai untuk tujuan apapun. Di antaranya cocok untuk pengembangan aplikasi *web* berbasis *server* (*server-side*) dimana *PHP* nantinya dijalankan di *server web*. Setiap kode *PHP* akan dieksekusi oleh *runtime PHP*, hasilnya adalah kode *PHP* yang dinamis tergantung kepada *script PHP* yang dituliskan. *PHP* dapat digunakan di banyak *server web*, sistem operasi, dan *platform*. Selain itu digunakan juga di sistem manajemen *database relasional (RDBMS)*. Semuanya ini bisa diperoleh gratis, dan Grup *PHP* menyediakan kode sumber lengkap bagi pengguna untuk membangun, menyesuaikan, dan mengotak-atik sesuai fungsi yang mereka inginkan.



2.3.2. MySQL

MySQL adalah sebuah *software database*. *MySQL* merupakan tipe data relasional yang artinya *MySQL* menyimpan datanya dalam bentuk tabel-tabel yang saling berhubungan (Edy Winarno dan Ali Zaki, 2014:102).

Subekti, Ichsan Risnandar, dan Hanafi S.S (2013:92), *MySQL* merupakan basis data yang bersifat *open source* sehingga banyak digunakan di dunia. Ada dua cara untuk menggunakan *MySQL* yaitu:

- a. Menggunakan *console/command line* atau *PHP*. Kita harus hafal perintah-perintah dalam *MySQL*. *Interface* yang digunakan hanya *text-based*.
- b. Menggunakan *PHPMysqlAdmin*, dapat diakses dengan alamat <http://alamatwebsite/phpmyadmin>. apabila digunakan *localhost* maka alamat <http://localhost/phpmyadmin> atau di 127.0.0.1/phpmyadmin.

Penggunaan *PHPMysqlAdmin* lebih mudah digunakan karena menggunakan *interface* yang lebih mudah dipahami.

2.3.4. CSS

Subekti dan Teguh N (2013:39), *CSS (Cascading Style Sheets)* adalah kumpulan aturan-aturan pemformatan yang mengontrol tampilan konten dalam sebuah halaman *web*. Terdapat tiga jenis *CSS*, yaitu :

1. *Inline style sheet* : kita cukup menambahkan atribut *style* di *tag* yang ingin kita berikan pemformatan.
2. *Internal style sheet* : kita meletakkan aturan pemformatan dengan *CSS* dibagi `<head>` dari *html* dengan tambahan *tag* `<style>`.
3. *External style sheet* : kita memisahkan antar file *CSS* dengan file *HTML*-nya.

2.3.5. XAMPP

Petrus F.S. dan Teguh N (2013:53), *XAMPP* adalah suatu program yang digunakan sebagai *server* untuk mengeksekusi fungsi yang ada dalam halaman *website* yang kita buat sekaligus menampilkan halaman *website* tersebut agar bisa diakses oleh *user*.



Hidayatullah dan Jauhari K.K. (2014:127), *XAMPP* adalah fasilitas untuk banyak sistem operasi seperti Windows, Linux, Mac, dan Solaris yang memungkinkan sebuah *web* dinamis bisa diakses secara local menggunakan *web server* local. Kata *XAMPP* sendiri terdiri dari :

- a. X yang berarti *Cross Platform* karena *XAMPP* bisa dijalankan di Windows, Linux, Mac, dan Solaris.
- b. A yang berarti Apache sebagai *web-server*-nya.
- c. M yang berarti *MySQL* sebagai *Database Management System (DBMS)*
- d. PP yang berarti *PHP* dan Perl sebagai bahasa yang didukungnya.

2.3.6. Adobe Dreamweaver CS3

Prabantini (2010:1), *dreamweaver* adalah *HTML* editor professional untuk mendesain *web* secara visual dan mengelola situs atau halaman *web*. *Dreamweaver CS3* memiliki kemampuan untuk mendesain *web*, menyunting kode, serta pembuatan aplikasi *web* dengan menggunakan berbagai bahasa pemrograman *web*, antara lain : *JPS*, *PHP*, *ASP*, dan *ColdFusion*. Ruang kerja, fasilitas dan kemampuan *dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun membangun suatu situs *web*. *Dreamweaver CS3* memiliki peningkatan kemampuan *toolbar* yang dapat digunakan untuk memodifikasi tampilan *toolbar* atau menambahkan fungsi baru. *Dreamweaver CS3* juga memiliki kemampuan dalam melakukan *print* kode pada jendela *code view*, dan memiliki fasilitas kode *hints* yang membantu dalam urusan *tag-tag*, serta *tag inspector* yang berguna untuk menangani *tag-tag HTML*.

2.3.7. JavaScript

Winarno dan Ali Zaki (2014:129), *JavaScript* adalah bahasa *scripting client side* yang sangat populer, karena bisa dipakai di *HTML*, *web*, untuk *server*, *PC*, laptop, tablet, ponsel, dan lainnya. Hampir semua programmer *web* menggunakan *JavaScript* untuk memberi efek pemrograman di halaman. *JavaScript* tidak hanya berdiri sendiri, tapi *JavaScript* juga menjadi dasar yang bisa digunakan untuk teknologi lainnya seperti *Ajax*, *jQuery*, dan *jQuery Mobile*.