



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Perangkat Lunak

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

Pressman (2012:5), “Perangkat lunak adalah: (1) intruksi-intruksi (program komputer) yang ketika dijalankan menyediakan fitur-fitur, fungsi-fungsi, dan kinerja-kinerja yang dikehendaki; (2) struktur data yang memungkinkan program-program memanipulasi informasi, dan (3) informasi deskriptif pada salinan tercetak dan bentuk-bentuk maya yang menggambarkan pengoperasian dan penggunaan program-program”.

##### 2.1.2 Data Base (*Basis Data*)

Sukamto dan Shalahuddin (2013:43), ”Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan.pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Fathansyah dal am Marhada dan Giap (2011:160), “Basis data adalah kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan”.

Berdasarkan penjelasan di atas maka penulis menyimpulkan bahwa basis data (*database*) merupakan tempat atau media yang digunakan untuk menyimpan data-data agar lebih terorganisir.



### 2.1.3 Data

Murhada dan Giap (2011:160), “Data adalah representasi dari fakta dunia nyata yang mewakili suatu objek yang sedang ditinjau (manusia, barang, peristiwa, hewan, konsep, keadaan dan sebagainya), dan direkam dalam bentuk huruf, kata, angka, simbol, gambar, bunyi, atau kombinasinya.”

Sujatmiko (2012:76), ”Data adalah Kumpulan dari angka-angka maupun karakter-karakter yang tidak memiliki arti. Data dapat diolah sehingga menghasilkan informasi”.

Berdasarkan penjelasan di atas maka penulis menyimpulkan bahwa data adalah suatu bahan (nilai, keadaan, sifat, kejadian, atau fakta) yang bentuknya masih mentah atau belum diolah.

### 2.1.4 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan perangkat lunak dengan RUP (*Rational Unified Process*). Menurut Sukanto dan Shalahuddin (2016:125), “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 Sukanto dan Shalahuddin (2016:128-131) 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*).

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 Khusus

### 2.2.1 Kamus Data

Menurut Sukamto (2018:73), “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)”.

Simbol-simbol yang ada dalam kamus data adalah sebagai berikut.

**Tabel 2.1** Simbol-Simbol Kamus Data

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

(Sumber : Sukamto dan Shalahuddin, 2018:74).



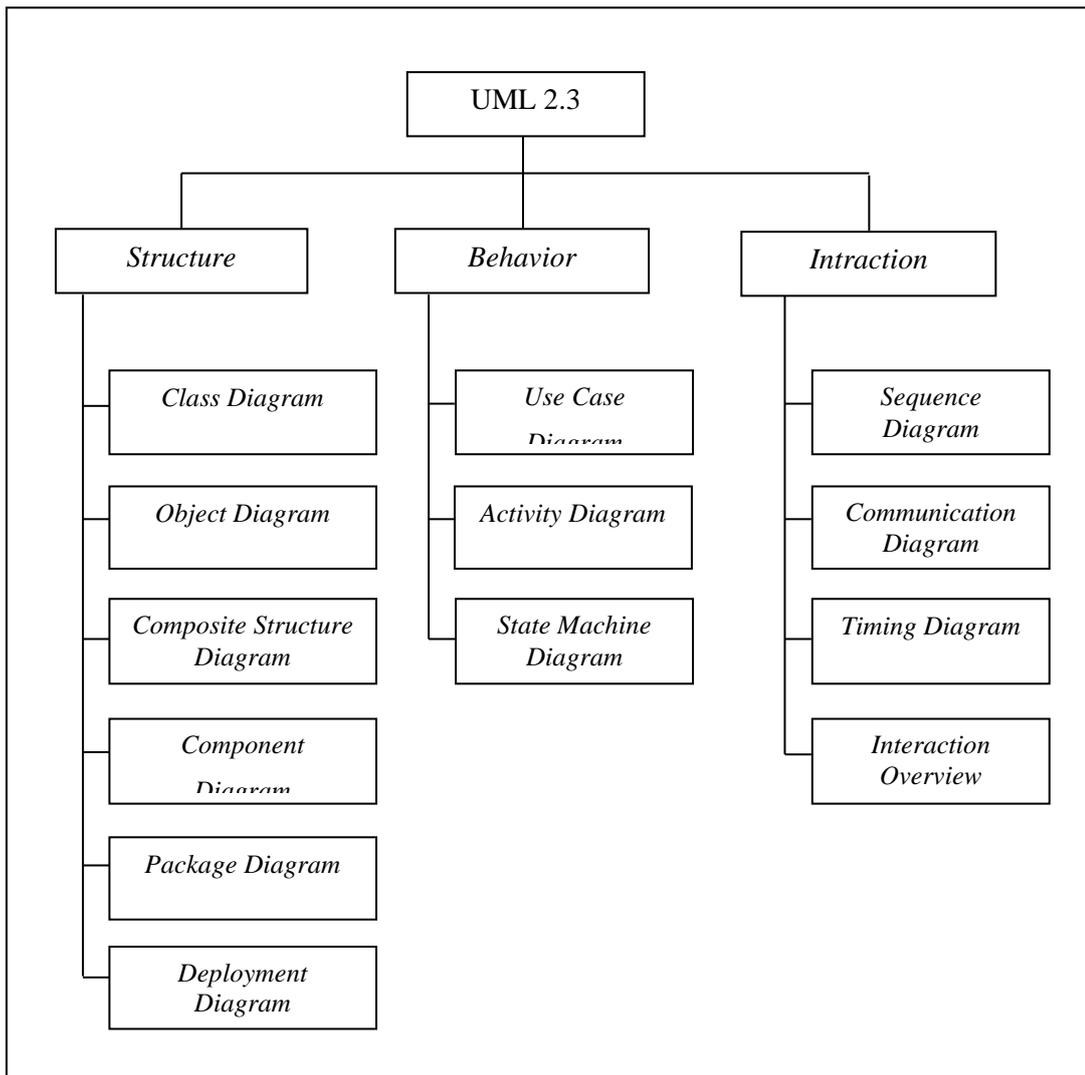
### **2.2.2 *Undifed Modeling Language (UML)***

Sukamto 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.”

*UML* menyediakan serangkaian gambar dan diagram yang sangat baik. Beberapa diagram memfokuskan diri pada ketangguhan teori *object-oriented* dan sebagian lagi memfokuskan pada detail rancangan dan konstruksi. Semua dimaksudkan sebagai sarana komunikasi antar *team programmer* maupun dengan pengguna.

### **2.2.3 *Katagori Undifed Modeling Language (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 2.1** 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.

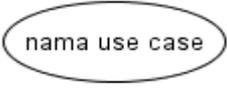
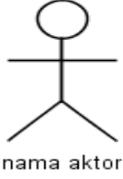


## 2.2.4 Jenis-Jenis Diagram UML

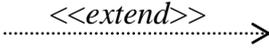
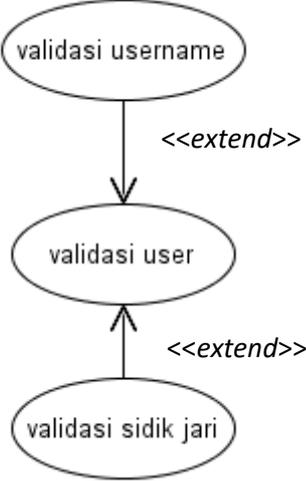
### 2.2.4.1 Use Case Diagram

Menurut Sukanto dan Shalahuddin (2018:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

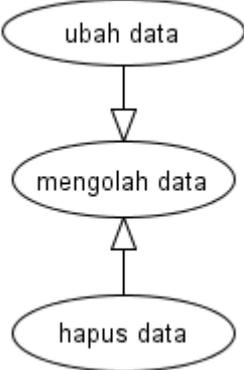
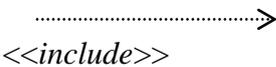
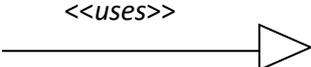
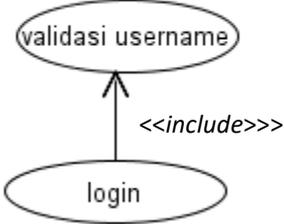
**Tabel 2.2** Simbol-simbol *Use case* Diagram

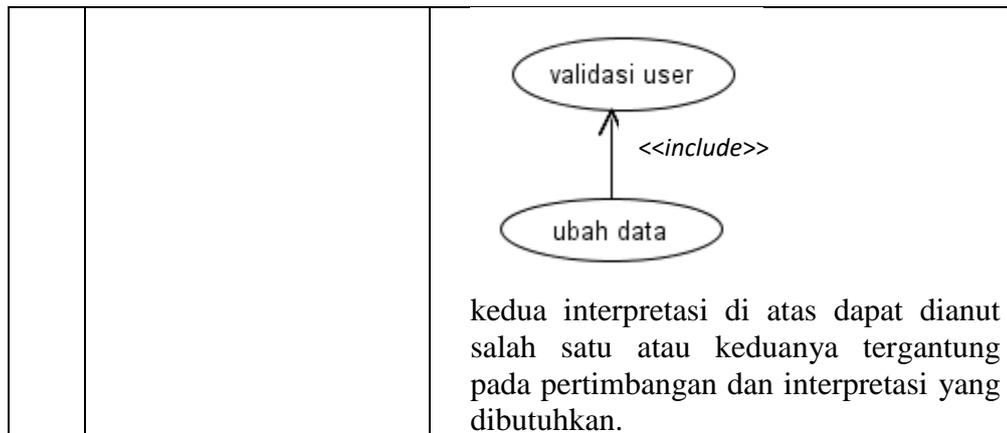
No	Simbol	Deskripsi
1.	<p><i>Use case</i></p> 	<p>fungsi yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal-awal frase nama <i>use case</i>.</p>
2.	<p>aktor / <i>actor</i></p> 	<p>orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
3.	<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 aktor.</p>

Lanjutan Tabel 2.2 Simbol-simbol *Use case Diagram*

4.	ekstensi / <i>extend</i>  	<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 Tabel 2.2 Simbol-simbol *Use case Diagram*

5.	<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>
6.	<p>menggunakan / include / uses</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"> <li>• <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu di panggil saat <i>use case</i> tambahan dijalankan, misalnya pada kasus berikut:</li> </ul>  <ul style="list-style-type: none"> <li>• <i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul>



**Lanjutan Tabel 2.2** Simbol-simbol *Use case* Diagram

Sumber : Sukamto dan Shalahuddin (2018:156)

Ada dua hal utama pada *use case* yaitu:

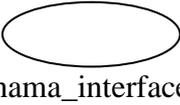
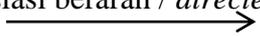
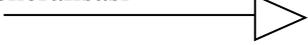
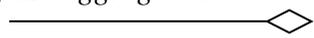
1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. *Use case* merupakan fungsionalitas yang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

#### 2.2.4.2 Class Diagram

Sukamto dan Shalahuddin (2018:141) menjelaskan, “*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 2.3 Simbol-simbol *Class Diagram*

No	Simbol	Deskripsi
1.	kelas 	Kelas pada struktur sistem
2	antarmuka / interface 	Sama dengan konsep interface dalam pemrograman berorientasi objek
3.	asosiasi / association 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i>
4.	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>
5.	generalisasi 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus)
6.	kebergantungan / <i>dependency</i> 	Relasi antarkelas dengan makna kebergantungan antar kelas
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

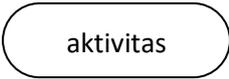
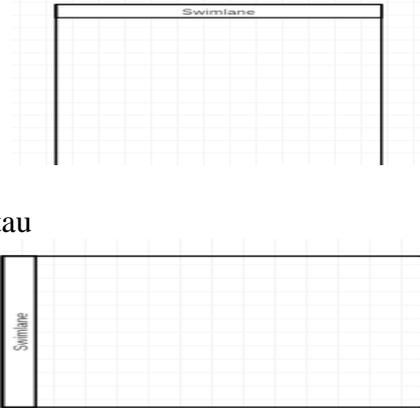
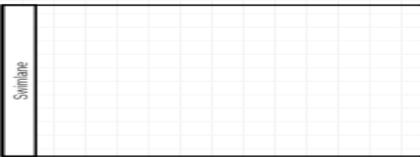
Sumber: Sukamto dan Shalahuddin (2018:146)

### 2.2.4.3 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), “*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 2.4 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 di mana jika ada pilihan aktivitas lebih dari satu
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.	Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Rosa dan Shalahuddin (2018:162)



#### 2.2.4.4 Sequence Diagram

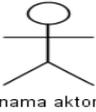
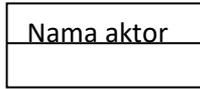
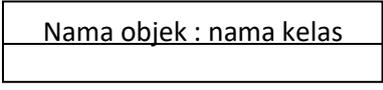
Menurut Sukanto dan Shalahuddin (2018:165), “Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam 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”.

Adapun simbol-simbol yang digunakan dalam *Sequence Diagram* adalah sebagai berikut:

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.

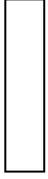
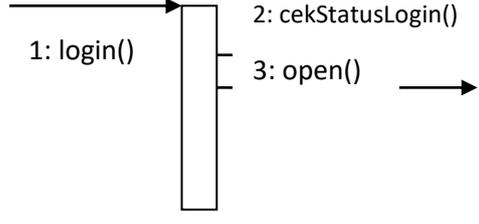
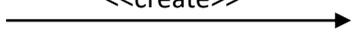
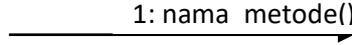
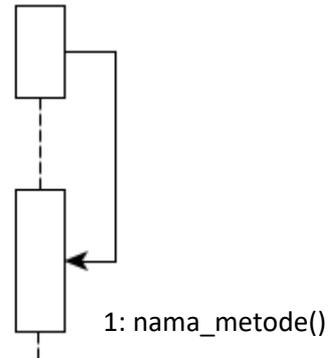
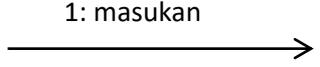
Berikut simbol-simbol pada Sequence Diagram :

**Tabel 2.5** Simbol-simbol pada Sequence Diagram

No	Simbol	Deskripsi
1.	<p>Actor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<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>
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>

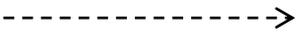


**Lanjutan Tabel 2.5** Simbol-simbol pada Sequence Diagram

4.	<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 cek Status Login () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe create</p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe call</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 send</p> 	<p>menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>



**Lanjutan Tabel 2.5** Simbol-simbol pada Sequence Diagram

8.	Pesan tipe return 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
9.	Pesan tipe destroy <<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 (2018:166)

## 2.3 Teori Judul

### 2.3.1 Aplikasi

Sujatmiko (2012:23) menyatakan, “Aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas–tugas tertentu, misalnya MS-Word, MS-Excel”.

Menurut Budiharto (2013:5) menyatakan bahwa, “*Applications* merupakan program yang dapat berjalan di komputer tersendiri (*stand alone computer*) ,dari mulai program yang simpel sampai dengan program yang besar dan rumit”.

Berdasarkan penjelasan di atas maka penulis menyimpulkan bahwa aplikasi merupakan suatu program yang berisi perintah-perintah tertentu untuk menyelesaikan suatu pekerjaan tertentu.

### 2.3.2 Cuti

Rohendi (2015:63), “Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu. Cuti diberikan dalam rangka usaha menjamin kesegaran jasmani dan rohani”.



Setiyanto (2013:381), “Cuti merupakan salah satu hak pegawai. cuti dapat digunakan oleh pegawai untuk tidak masuk kerja dengan alasan tertentu, misalnya refreshing, sakit, melahirkan, menunaikan kewajiban agama, dan keperluan lain sesuai dengan ketentuan cuti pada masing-masing organisasi”.

Berdasarkan pengertian di atas maka dapat di simpulkan bahwa cuti adalah keadaan tidak masuk kerja yang diizinkan.

### **2.3.2.1 Jenis Cuti**

Menurut Peraturan Pemerintah Nomor 24 Tahun 1976, Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu. Cuti terdiri dari :

#### **1. Cuti Tahunan**

Setiap Pegawai Negeri Sipil yang telah bekerja sekurang-kurangnya satu tahun secara terus menerus berhak atas cuti tahunan. Lamanya cuti tahunan adalah 12 (dua belas) hari kerja. Cuti tahunan tersebut dapat diambil secara terpecahpecah, dengan ketentuan setiap bagian tidak boleh kurang dari 3 (tiga) hari kerja. Cuti tahunan yang tidak diambil dalam tahun yang bersangkutan dapat diambil dalam tahun berikutnya untuk paling lama 18 (delapan belas) hari kerja termasuk cuti tahunan dalam tahun yang sedang berjalan. Cuti tahunan yang tidak diambil dalam kurun waktu 2 (dua) tahun berturut-turut atau lebih, dapat diambil dalam tahun berikutnya untuk paling lama 24 (dua puluh empat) hari kerja, termasuk cuti tahunan dalam tahun yang sedang berjalan.

#### **2. Cuti Besar**

adalah cuti yang dapat diambil oleh seorang PNS setelah bekerja 6 tahun berturut-turut. Lama cuti besar adalah maksimal 90 hari kalender.

#### **3. Cuti Sakit**

adalah cuti yang bisa diambil oleh seorang PNS ketika sakit dan membutuhkan waktu istirahat untuk pemulihan kondisinya.



#### 4. Cuti Bersalin

Cuti bersalin adalah cuti yang dapat diambil oleh PNS wanita untuk melahirkan anak pertama, kedua, dan ketiga. Lama cuti bersalin adalah maksimal 90 hari kalender.

#### 5. Cuti Karena Alasan Penting

PNS dapat cuti karena alasan penting untuk paling lama 60 hari kalender. Lamanya cuti karena alasan penting hendaknya ditetapkan sedemikian rupa, sehingga benar-benar hanya untuk waktu yang diperlukan saja. Dalam kondisi tertentu, cuti alasan penting hanya bisa diambil setelah hak cuti pegawai yang bersangkutan tidak mencukupi lagi/habis.

#### 6. Cuti di Luar Tanggungan Negara

Cuti diluar tanggungan negara dapat diberikan kepada PNS yang telah bekerja sekurang-kurangnya 5 tahun secara terus-menerus dan adanya alasan-alasan pribadi yang penting dan mendesak

### 2.3.2.2 Proses Cuti

Sebelum melaksanakan cuti, pegawai yang bersangkutan perlu melaksanakan urutan tata cara untuk mendapatkan cuti. Secara umum proses/prosedur pelaksanaan cuti adalah sebagai berikut :

#### 1. Proses Pengajuan Permohonan Cuti

Pegawai yang akan melaksanakan cuti membuat surat permohonan cuti

#### 2. Proses pemeriksaan hak cuti

Bagian kepegawaian akan mengevaluasi jumlah hari kerja yang masih menjadi hak cuti pegawai yang bersangkutan

#### 3. Proses persetujuan atasan langsung

Pegawai yang bersangkutan mengajukan cuti kepada atasan langsungnya untuk mendapatkan persetujuan.

#### 4. Proses pembuatan surat izin cuti

Setelah disetujui oleh atasan langsung yang bersangkutan, bagian kepegawaian membuat surat izin cuti untuk ditandatangani pejabat yang berwenang memberikan cuti



#### 5. Proses percetakan surat cuti

Surat izin cuti dibuat rangkap 2 (dua), asli untuk yang bersangkutan, tembusan untuk arsip kepegawaian.

#### 6. Proses laporan kepada atasan langsung

Setelah menjalankan cuti pegawai yang bersangkutan wajib melapor kepada atasannya.

(Sumber: <https://www.google.com/url?sa=t&source=web&rct=j&url=https://digilib.uns.ac.id/dokumen/download/15478/MzA0ODg%3D/Prosedur-pelaksanaan-cuti-pegawai-pada-PT-Bank-Tabungan-Negara-PERSERO-kantor-cabang-Surakarta-abstrak.pdf> )

### 2.3.3 Web

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

Hariyanto dalam Destiningrum dan Adrian (2017:32), “Web dapat diartikan sebagai kumpulan halaman yang menampilkan informasi data teks, data gambar, data animasi, suara, video dan gabungan dari semuanya. Baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, dimana masing-masing dihubungkan dengan jaringan-jaringan halaman (*hyperlink*)”. Jadi, dapat disimpulkan website adalah aplikasi internet yang berisi sekumpulan halaman berbentuk data teks, gambar, video, audio, maupun animasi.

### 2.3.4 Aplikasi Permohonan Cuti Pegawai Pada Kantor Kejaksaan Negeri Palembang Berbasis Web

Aplikasi Permohonan Cuti Pegawai Pada Kantor Kejaksaan Negeri Palembang Berbasis Web merupakan suatu program atau perangkat lunak yang



berisi perintah-perintah khusus untuk mempermudah proses permohonan cuti pegawai pada kantor kejaksaan negeri Palembang. Aplikasi ini digunakan untuk mengelolah data permohonan cuti pegawai agar lebih teratur dan mudah dalam proses pencarian jika sewaktu-waktu dibutuhkan dan untuk mempermudah pegawai dalam membuat laporan cuti pegawai.

## **2.4 Teori Program**

### **2.4.1 XAMPP**

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

Menurut Andi ( 2015 : 55 ), “ XAMPP merupakan singkatan dari x (empat sistem operasi apa pun), Apache, Mysql, PHP, Perl, XAMPP adalah *tool* yang menyediakan paket perangkat lunak dalam satu buah paket.

### **2.4.2 PHP**

Sidik (2017:4), “PHP adalah kependekan dari Hyper Text Preprocessor (rekursif, mengikut gaya penamaan di unix), merupakan bahasa utama script serverside yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop”.

Menurut Abdulloh (2018:127), PHP merupakan kependekan dari PHP Hypertext preprocessor yaitu bahasa pemrograman web yang dapat disisipkan dalam skrip HTML dan bekerja di sisi server.” Berdasarkan pendapat tersebut, dapat disimpulkan bahwa PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web”.



### 2.4.3 Sintaks Dasar PHP

Kode (Script) PHP yang sering disebut dengan istilah embedded script yaitu script PHP yang disisipkan di antara script HTML. Jadi dapat dikatakan script PHP hanya ditulis atau disisipkan ketika dibutuhkan saja, seperti menampilkan data dari database meng-upload file, delete data, edit data dan lain sebagainya. Contoh script :

```
<HTML>
    <HEAD>
        <TITLE>CONTOH PHP</TITLE>
    </HEAD>
<BODY>
    <?php
        Echo"Belajar Bahasa Pemrograman PHP";
    ?>
</BODY>
</HTML>
```

### 2.4.4 MySQL

Sidik (2017:301), "MySQL merupakan software database yang termasuk paling populer di lingkungan Linux, kepopuleran ini karena ditunjang karena performansi query dari databasenya yang saat itu bisa dikatakan paling cepat, dan jarang bermasalah. MySQL telah tersedia juga di lingkungan Windows. berawal dari software yang shareware MySQL populer, kini mulai versi 3.23 MySQL menjadi software open source yang berarti free. MySQL dapat digunakan untuk kepentingan komersial ataupun personal (non profit)".

Sukanto 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".

Raharjo dalam Ghazali (2016:21), "*MySQL* merupakan *software* RDBMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak *user*



(*multi-user*), dan dapat melakukan suatu proses secara sinkron atau berbarengan (*multi-threaded*)”.

Dapat disimpulkan bahwa MySQL merupakan software DBMS yang bersifat Open Source dan digunakan untuk menyimpan data atau server database yang mendukung Bahasa database pencarian SQL.

#### **2.4.5 HTML (*HyperText Markup Language*)**

Menurut Pamungkas (2017:1-2), “*HyperText Markup Language (HTML)* merupakan bahasa pemrograman web berupa suatu format data yang digunakan untuk membuat dokumen hypertext yang dapat dibaca, diinterpretasikan dari satu platform komputer ke platform komputer lainnya tanpa perlu melakukan suatu perubahan apapun”.

Abdulloh (2018:127) menyatakan bahwa, “HTML merupakan singkatan dari Hypertext Markup Language yaitu bahasa standar web yang dikelola penggunaannya oleh W3C (World Wide Web Consortium) berupa tag-tag yang menyusun setiap elemen dari website”.

#### **2.4.6 CSS (*Cascading Style Sheets*)**

Menurut Abdulloh (2016:2), “CSS singkatan dari Cascading Style Sheet, yaitu skrip yang digunakan untuk mengatur desain website. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar terstruktur website yang dibuat dengan HTML terlihat lebih rapih dan elegan.

Menurut Sulistiono (2018:2) menjelaskan, “CSS (*Cascading Style Sheet*) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS bukan merupakan bahasa pemrograman. Sama halnya seperti styles dalam aplikasi pengolahan kata, Microsoft Word, yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada Umum nya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

---



### 2.4.7 JavaScript

Abdulloh (2018:193), JavaScript merupakan bahasa pemrograman web yang pemrosesannya dilakukan di sisi client. Karena berjalan di sisi client, JavaScript dapat dijalankan hanya dengan menggunakan browser. Karena berjalan di sisi client, JavaScript dapat dijalankan hanya dengan menggunakan browser. Berbeda dengan PHP yang bekerja di sisi server, untuk menjalankan skrip JavaScript tidak memerlukan refresh pada browser.

Suryana dan Koesheryatin dalam Ghazali (2016:21), “*JavaScript* adalah bahasa *script* berdasar pada objek yang memperbolehkan pemakai untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML, di mana objek tersebut dapat berupa suatu *window*, *frame*, URL, dokumen, *form*, *button*, atau item yang lain”.

Menurut Winarno dkk (2014:129), “JavaScript adalah bahasa *scripting client side* yang sangat populer karena javascript bisa dipakai di HTML, *web*, untuk *server*, PC, laptop, tablet, ponsel dan lainnya serta menjadi dasar yang bisa digunakan untuk teknologi lainnya seperti Ajax, jQuery dan *jQuery Mobile*”.

### 2.4.8 Cara Penulisan Kode Javascript pada HTML

#### 1. Tag <script>

Cara yang umum yang dipakai adalah menuliskannya dalam tag <script>.

Tag <script> bisa dibuat di dalam tag <head>, maupun di dalam tag <body>.