



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Komputer

Wangdra dan Elfi (2016:2), komputer merupakan alat untuk mengolah data sesuai perintah yang sudah dirumuskan.

Fitra dan Gusteti (2014:27), komputer adalah mesin penghitung elektronik yang cepat dan dapat menerima informasi *input digital*, kemudian memprosesnya sesuai dengan program yang tersimpan di memorinya dan menghasilkan *output* berupa informasi.

Sanders dalam Sutabri (2016: 96), komputer adalah sistem elektronik untuk memanipulasi data yang cepat dan tepat serta dirancang dan diorganisasikan secara otomatis menerima dan menyimpan data *input*, memprosesnya dan menghasilkan *output* di bawah pengawasan suatu langkah instruksi program yang tersimpan di memori (*stored program*).

Dari penjelasan beberapa pendapat para ahli diatas maka dapat disimpulkan bahwa komputer adalah suatu alat elektronik yang digunakan untuk mempermudah pekerjaan manusia dalam mengelola data-data dan menghasilkan output berupa informasi.

##### 2.1.2 Pengertian Aplikasi

Kamus Besar Bahasa Indonesia (2014:81), aplikasi adalah program computer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu.

Sujatmiko (2012:23), aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel*.

Dari penjelasan diatas, dapat disimpulkan bahwa aplikasi adalah adalah suatu perangkat lunak yang memanfaatkan kemampuan komputer langsung untuk membantu manusia mengerjakan tugas dengan perangkat komputer.



### 2.1.3 Pengertian Perangkat Lunak

Rusdiana dan Irfan (2014:212), *software* (perangkat lunak) adalah rangkaian prosedur dan dokumentasi program yang berfungsi untuk menyelesaikan berbagai masalah yang dikehendaki.

Sujatmiko (2012:256), *software* (perangkat lunak) adalah kumpulan beberapa perintah yang dieksekusi oleh mesin komputer dalam menjalankan pekerjaannya.

Dari penjelasan beberapa pendapat para ahli maka dapat disimpulkan bahwa perangkat lunak adalah kumpulan perintah yang digunakan dalam media komputer dengan menggunakan bahasa komputer untuk mengolah ataupun memproses data di computer.

### 2.1.4 Pengertian Data

Indrajani (2014:2), data adalah fakta atau observasi mentah yang biasanya mengenai fenomena fisik atau transaksi bisnis.

Rusdiana dan Irfan (2014:71), data adalah fakta atau bagian dari fakta yang mengandung arti sehubungan dengan kenyataan, simbol-simbol, gambar-gambar, angka-angka, huruf atau simbol yang menunjukkan suatu ide, objek, kondisi atau situasi dan lainnya yang didapatkan melalui suatu observasi atau secara data diartikan sebagai keterangan tentang sesuatu.

Dari penjelasan beberapa pendapat para ahli diatas maka dapat disimpulkan bahwa data adalah fakta atau deskripsi tentang sesuatu yang belum diolah lebih lanjut yang merupakan keterangan tentang sesuatu.

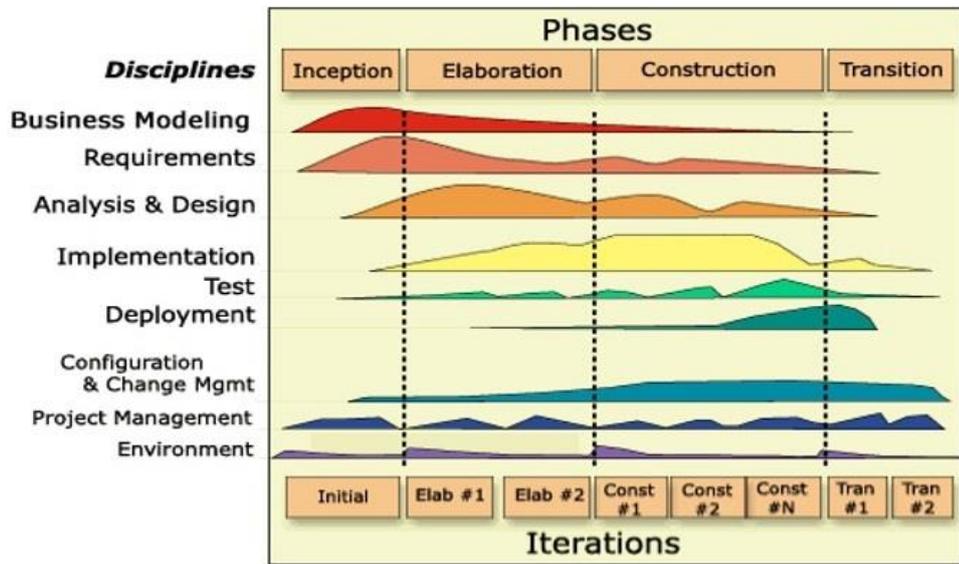
### 2.1.5 Metode Pengembangan Sistem

Sukamto dan Shalahuddin (2015:124) *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*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan

---



penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. Berikut ini gambar proses pengulangan atau iteratif RUP:



Sumber: Sukamto dan Shalahuddin, (2015:129)

**Gambar 2.1** Proses Iteratif RUP

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar dibawah dapat dilihat bahwa RUP memiliki, yaitu:

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari fase selanjutnya. Setiap fase dapat berdiri dari satu beberapa iterasi. Dimensi ini terdiri atas *Inception, Elaboration, Construction, dan Transition*.
2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang



dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing, what, how dan when*.

Pada penggunaan kedua standar tersebut diatas berorientasi obyek (*object oriented*) memiliki manfaat yakni:

1. *Improve productivity*

Standar ini dapat memanfaatkan kembali komponen-komponen yang telah tersedia/dibuat sehingga dapat meningkatkan produktifitas.

2. *Deliver high quality system*

Kualitas sistem informasi dapat ditingkatkan sebagai sistem yang dibuat pada komponen-komponen yang telah teruji (*well-tested dan well-proven*) sehingga dapat mempercepat *delivery* sistem informasi yang dibuat dengan kualitas yang tinggi.

3. *Lower maintenance cost*

Standar ini dapat membantu untuk menyakinkan dampak perubahan yang terlokalisasi dan masalah dapat dengan mudah terdeteksi sehingga hasilnya biaya pemeliharaan dapat dioptimalkan atau lebih rendah dengan pengembangan informasi tanpa standar yang jelas.

4. *Facilitate reuse*

Standar ini memiliki kemampuan yang mengembangkan komponen-komponen yang dapat digunakan kembali untuk pengembangan aplikasi yang lainnya.

5. *Manage complexity*

Standard ini mudah untuk mengatur dan memonitor semua proses dari semua tahapan yang ada sehingga suatu pengembangan sistem informasi yang amat kompleks dapat dilakukan dengan aman dan sesuai dengan harapan semua manajer proyek IT/IS yakni *deliver good quality software within cost and schedule time and the users accepted*.



Sukamto dan Shalahuddin (2015:129) Berikut ini penjelasan untuk setiap fase RUP:

1. *Inception* (Analisis kebutuhan)

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.

3. *Construction* (Konstruksi)

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.

4. *Transition* (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Tahap ini menghasilkan produk perangkat lunak di mana menjadi syarat dari *Initial Operational Capability Milestone* atau batas/tonggak kemampuan operational awal.

## 2.2 Teori Khusus

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

Sukamto dan Shalahuddin (2015:133), *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.

### **2.2.2 Sejarah UML**

Sukamto dan Shalahuddin (2014:138), 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 Small talk 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 Small talk 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. Rumbaigh, Grady Booch, dan Ivar Jacobson bergabung dalam sebuah perusahaan yang bernama Rational Software Corporation menghasilkan bahasa yang disebut dengan Unified Modeling Language (UML).

### **2.2.3 Diagram UML**

Sukamto dan Shalahuddin (2014:140), 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 sub sistem pada suatu sistem. *Interaction diagram* terdiri dari *Sequence Diagram*, *Communication Diagram*, *Timing Diagram*, *Interaction Overview Diagram*.

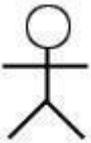
#### 2.2.4 Use Case Diagram

Sukamto dan Shalahuddin (2015:155), “*Use Case* atau diagram *usecase* 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”. Berikut adalah simbol-simbol yang ada pada diagram *use case* menurut Sukamto dan Shalahuddin (2013:156):

**Tabel 2.1** Simbol-simbol Diagram *Use Case*

No	Simbol	Keterangan
1.	 <p>Nama <i>use case</i></p>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anatar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i> .



2.	<p>Aktor/<i>actor</i></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 menggunakan kata benda diawal frase nama aktor.
3.	<p>Asosiasi/ <i>association</i></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.
4.	<p>Ekstensi/ <i>extend</i></p>  <p>&lt;&lt;<i>extend</i>&gt;&gt;</p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
5.	<p>Generalisasi/</p>  <p><i>generalization</i></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.
6.	<p>Menggunakan/</p> <p><i>include/use</i></p>  <p>&lt;&lt;<i>include</i>&gt;&gt;</p>  <p>&lt;&lt;<i>uses</i>&gt;&gt;</p>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

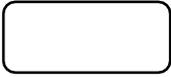
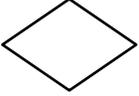
Sumber : Sukamto dan Shalahuddin (2013:156)



### 2.2.5 Activity Diagram

Sukamto dan Shalahuddin (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 2.2** Simbol-simbol *Activity Diagram*

No	Simbol	Keterangan
1.	Status awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas  aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.	Percabangan/ <i>decision</i> 	Percabangan/ <i>decision</i>
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

Sumber: Sukamto dan Shalahuddin (2014:162)



### 2.2.6 Class Diagram

Sukamto dan Shalahuddin (2015:141), “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 metode atau operasi.

1. Atribut merupakan varibel-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut:

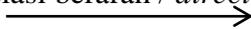
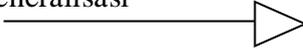
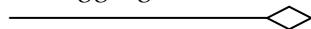
1. Kelas main adalah kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem (*view*) adalah kelas yang mendefinisikan dan mengatur tampilan ke pemakai.
3. Kelas yang diambil dari pendefinisian *use case* (*controller*) adalah kelas yang menangani fungsi-fungsi yang harus ada diambil dari pendefinisian *use case*, kelas ini biasanya disebut dengan kelas proses yang menangani proses bisnis pada perangkat lunak.
4. Kelas yang diambil dari pendefinisian data (*model*) adalah kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

Berikut adalah simbol-simbol yang ada pada diagram kelas menurut Sukamto dan Shalahuddin (2015:146):

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

No	Simbol	Deskripsi
1.	<div style="border: 1px solid black; padding: 5px; width: fit-content;"> <div style="border-bottom: 1px solid black; margin-bottom: 2px;">nama_kelas</div> <div style="margin-bottom: 2px;">+atribut</div> <div>+operasi()</div> </div>	Kelas pada struktur system.



2	antarmuka / <i>interface</i>   nama_ <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	asosiasi / <i>association</i> 	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 (2015:146)

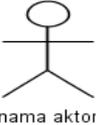
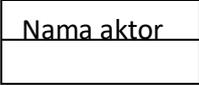
### 2.2.7 Sequence Diagram

Sukamto dan Shalahuddin (2015:165), “*Sequence diagram* atau diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirim dan diterima antar objek”.

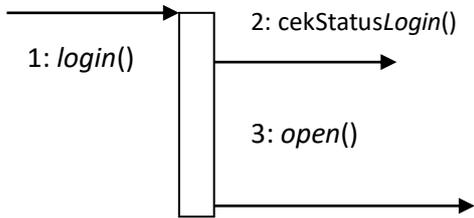
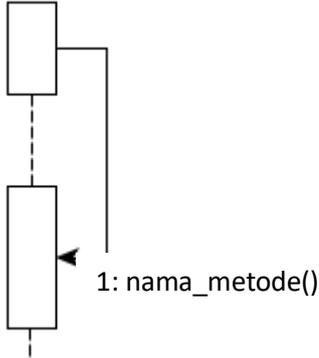
Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka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

Berikut adalah simbol-simbol yang ada pada diagram sekuen menurut Sukamto dan Shalahuddin (2013:146):

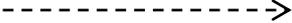
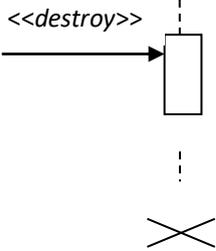
Tabel 2.4 Simbol-simbol *Sequence Diagram*

No	Simbol	Deskripsi
1.	<p data-bbox="395 510 469 539"><i>Actor</i></p>  <p data-bbox="427 703 523 725">nama aktor</p> <p data-bbox="395 790 453 819">atau</p>  <p data-bbox="395 1010 624 1039">tanpa waktu aktif</p>	<p data-bbox="871 443 1366 725">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 <i>actor</i>.</p>
2.	<p data-bbox="395 1048 663 1077">Garis hidup / <i>lifeline</i></p> 	<p data-bbox="871 1048 1331 1077">menyatakan kehidupan suatu objek.</p>
3.	<p data-bbox="395 1384 475 1413">Objek</p>  <p data-bbox="437 1458 740 1487">Nama objek : nama kelas</p>	<p data-bbox="871 1384 1366 1451">menyatakan objek yang berinteraksi pesan.</p>
4.	<p data-bbox="395 1753 552 1783">Waktu aktif</p> 	<p data-bbox="871 1753 1366 1933">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 <i>cekStatusLogin ()</i> dan <i>open()</i> dilakukan di dalam metode <i>login()</i></p> <p>aktor tidak memiliki waktu aktif</p>
5.	<p>Pesan tipe <i>create</i></p> <p style="text-align: center;"><i>&lt;&lt;create&gt;&gt;</i></p> 	<p>menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
6.	<p>Pesan tipe <i>call</i></p> <p style="text-align: center;">1: nama metode()</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.	Pesan tipe send 1: masukan 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
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 <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 (2015:165)

## 2.3 Teori Judul

### 2.3.1 Pengertian Aplikasi

Kamus Besar Bahasa Indonesia (2014:81), aplikasi adalah program komputer atau perangkat lunak yang didesain untuk mengerjakan tugas tertentu.

Sujatmiko (2012:23), aplikasi adalah program komputer yang dibuat oleh suatu perusahaan komputer untuk membantu manusia dalam mengerjakan tugas-tugas tertentu, misalnya *Microsoft Word* dan *Microsoft Excel*.

Dari penjelasan diatas, dapat disimpulkan bahwa aplikasi adalah adalah suatu perangkat lunak yang memanfaatkan kemampuan komputer langsung untuk membantu manusia mengerjakan tugas dengan perangkat komputer.



### 2.3.2 Pengertian *Virtual Reality*

*Virtual Reality* (VR) atau realitas maya adalah teknologi yang membuat pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer (*computer-simulated environment*), suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi (Sihite, 2013).

Konsep VR mengacu pada sistem prinsip-prinsip, metode dan teknik yang digunakan untuk merancang dan menciptakan produk-produk perangkat lunak untuk digunakan oleh bantuan dari beberapa sistem komputer multimedia dengan sistem perangkat khusus (Lacrama, 2007).

Dari penjelasan diatas, dapat disimpulkan bahwa *virtual reality* adalah teknologi yang memungkinkan pengguna berinteraksi dengan suatu lingkungan 3D yang disimulasikan oleh sistem komputer dan perangkat khusus.

### 2.3.3 Pengertian Proses Pembelajaran

Menurut pendapat Bafadal (2005:11), pembelajaran dapat diartikan sebagai “segala usaha atau proses belajar mengajar dalam rangka terciptanya proses belajar mengajar yang efektif dan efisien”.

Menurut Jogiyanto (2007:12) juga berpendapat bahwa pembelajaran dapat didefinisikan sebagai suatu proses yang mana suatu kegiatan berasal atau berubah lewat reaksi suatu situasi yang dihadapi dan karakteristik-karakteristik dari perubahan aktivitas tersebut tidak dapat dijelaskan berdasarkan kecenderungan-kecenderungan reaksi asli, kematangan atau perubahan-perubahan sementara.

Dari penjelasan diatas, dapat disimpulkan bahwa proses pembelajaran adalah segala usaha belajar mengajar lewat situasi yang dihadapi berdasarkan kecenderungan reaksi asli atau perubahan-perubahan sementara.



#### **2.3.4 Pengertian Tempat Bersejarah**

Pitana (dalam Hermawan, 2017:66) mengemukakan pengertian destinasi yaitu tempat yang dikunjungi dengan waktu yang signifikan selama perjalanan seseorang dibandingkan dengan tempat lain yang dilalui selama perjalanan 39 (misalnya daerah transit)

Badarab, Trihayuningtyas, dan Suryadana (2017:99) menjelaskan bahwa destinasi pariwisata ialah suatu entitas yang mencakup wilayah geografis tertentu yang didalamnya terdapat komponen produk wisata seperti atraksi wisata, amenitas dan aksesibilitas serta layanan maupun pendukung lainnya seperti masyarakat, pelaku industri pariwisata dan institusi pengembangan yang membentuk sinergis dalam menciptakan suatu motivasi kunjungan serta totalitas pengalaman berkunjung wisatawan.

Dari penjelasan diatas, dapat disimpulkan bahwa tempat bersejarah adalah tempat yang dikunjungi yang mencakup wilayah bersejarah tertentu yang didalamnya terdapat komponen dan layanan untuk masyarakat.

#### **2.3.5 Pengertian Bukit Siguntang**

Terdapat penjelasan pengertian Bukit Siguntang dalam situs [https://id.wikipedia.org/wiki/Bukit\\_Seguntang](https://id.wikipedia.org/wiki/Bukit_Seguntang) adalah sebuah bukit kecil setinggi 29—30 meter dari permukaan laut yang terletak sekitar 3 kilometer dari tepian utara sungai musi dan masuk dalam wilayah kota Palembang, Sumatera Selatan. Secara administratif situs ini termasuk kelurahan Bukit Lama, Kecamatan Ilir Barat I, Palembang. Bukit ini berjarak sekitar 4 kilometer di sebelah barat daya pusat kota Palembang, dapat dicapai dengan menggunakan angkutan umum menuju jurusan Bukit Besar. Di lingkungan sekitar bukit ini ditemukan beberapa temuan purbakala yang dikaitkan dengan kerajaan Sriwijaya yang berjaya sekitar kurun abad ke-6 sampai ke-13 masehi. Di puncak bukit ini terdapat beberapa makam yang dipercaya sebagai leluhur warga Palembang. Oleh masyarakat setempat, kompleks ini dianggap keramat dan menjadi tempat tujuan ziarah.

---



## **Pengertian Aplikasi Berbasis *Virtual Reality* untuk Mendukung Proses Pembelajaran Tempat Bersejarah di Kota Palembang (Studi Kasus Bukit Siguntang)**

Aplikasi Berbasis *Virtual Reality* untuk Mendukung Proses Pembelajaran Tempat Bersejarah di Kota Palembang (Studi Kasus Bukit Siguntang) adalah aplikasi berbasis *virtual reality* untuk mensimulasikan lingkungan Bukit Siguntang Palembang melalui sistem komputer dengan media VR Box dan Bluetooth Joystick Controller untuk mendukung proses belajar mengajar tempat bersejarah di Kota Palembang sehingga lebih menarik, efektif, efisien dan inovatif.

### **2.4 Teori Program**

#### **2.4.1 Pengertian C#**

Terdapat penjelasan pengertian C# dalam situs <https://www.codepolitan.com/pengenalan-bahasa-pemrograman-c-587effa1cb95b> yaitu C# atau yang dibaca C sharp adalah bahasa pemrograman sederhana yang digunakan untuk tujuan umum. Dalam artian, bahasa pemrograman ini dapat digunakan untuk berbagai fungsi, misalnya untuk pemrograman server-side pada website, membangun aplikasi desktop ataupun mobile, pemrograman game dan sebagainya. Selain itu C# juga bahasa pemrograman yang berorientasi objek, jadi C# juga mengusung konsep objek seperti *inheritance*, *class*, *polymorphism* dan *encapsulation*.

Dari penjelasan diatas, dapat disimpulkan bahwa C# adalah bahasa pemrograman yang berorientasi objek dengan berbagai fungsi seperti untuk membangun *server-side* pada *website*, aplikasi, desktop maupun mobile, pemrograman *game* dan sebagainya.



### 2.4.2 Pengertian Unity Game Engine

Unity3D merupakan salah satu game engine dengan lisensi open source (gratis) dan proprietary (biasanya berbayar). Unity3D tidak membatasi publikasi aplikasi, pengguna Unity3D dengan lisensi gratis dapat mempublikasikan aplikasi yang dibuat tanpa harus membayar biaya lisensi kepada unity3D. Seperti kebanyakan game engine lainnya. Unity3d dapat mengolah beberapa data seperti objek tiga dimensi, suara, tekstur, dan lain sebagainya (Sihite et al., 2013).

Unity dibuat dengan menggunakan bahasa perogram C#, Unity memiliki kemiripan dengan *game engine* lainnya seperti, Blender, Virtools, *Gamestudio*, adapapun kelebihan dari Unity dapat dioperasikan pada *platform Windows* dan *Mac Os* dan dapat menghasilkan *Game* untuk *Windows, Mac, Linux, Wii, iPad, iPhone, google Android* dan juga *browser*. *Game* Unity juga mendukung dalam pembuatan *Game* untuk console *Game Xbox 360* dan *PlayStation*. (Yulianto,2012:36)

Dari penjelasan diatas, dapat disimpulkan bahwa Unity Game Engine adalah *software* yang digunakan untuk membuat game berbasis 2 dimensi atau 3 dimensi dengan bahasa pemrograman C# dan dapat dioperasikan dalam perangkat Windows atau Mac OS untuk menghasilkan game dalam berbagai *platform*.

### 2.4.3 Pengertian Android

Menurut Desmira (2016:6). Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan *platform* terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc, pendatang baru yang membuat peranti lunak untuk ponsel. Android adalah sistem operasi berbasis *linux* yang digunakan untuk ponsel (telepon seluler) mencakup sistem operasi, *middleware* dan aplikasi.

Menurut Giyartono dkk (2015:2), “Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi,



*middleware* dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka.”

Menurut Sari dalam M.Ichwan (2015:151), “Android adalah sistem operasi berbasis kernel Linux yang pada awalnya dikembangkan oleh Android Inc, yang didukung *Google* finansial dan kemudian dibeli pada tahun 2005. Android ini diresmikan pada tahun 2007 seiring dengan berdirinya *Open Handset Alliance* konsorsium *hardware, software*, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar perangkat selular.”

Dari penjelasan diatas, dapat disimpulkan bahwa Android adalah sistem operasi berbasis kernel Linux untuk perangkat lunak yang dikembangkan oleh Android sebagai sistem operasi pada ponsel.

#### 2.4.4 Pengertian Blender

Blender adalah *software open source* yang dibuat oleh Blender Foundation yang berfungsi untuk membuat suatu karya dalam 3D. *Software* ini telah mendukung hampir seluruh *pipeline* atau tahapan dalam membuat film animasi 3D, urutan tersebut meliputi *modeling, rigging, animate, simulasi, rendering, compositing* dan *motion tracking* bahkan *video editing* dan pembuatan *game*. Karena ini merupakan *software open source*, jadi untuk pengguna yang sudah dalam tingkatan ahli dapat melakukan *developing* pada *software* yang mungkin nanti akan bisa dirilis pada versi Blender selanjutnya. (Blender Foundations 2017).

Menurut Baihaqi (2019:Vol.2 No.1) Blender adalah sebuah *software* pemodelan animasi 3D yang memiliki fitur *game engine*, blender pada awalnya dikembangkan oleh perusahaan animasi Belanda Neogeo sebagai program animasi internal[5], blender tumbuh dan berkembang bersama proyek yang dikerjakan NeoGeo. Kemudian tidak lama setelah versi gratis di publik di internet, NeoGeo tidak lagi menjalankan bisnisnya. Pada saat itu Ton Roosendaal, bapak dari blender dan programmer utama, membuat perusahaan yang bernama NOT a number untuk mengembangkan blender lebih jauh. Blender bisa menjadi sebuah



produk, versi gratis dari blender bukanlah versi demo namun berfungsi penuh, dan lisensinya memungkinkan penggunaan tak terbatas untuk produk komersial.

Dari penjelasan diatas, dapat disimpulkan bahwa Blender adalah sebuah software yang berfungsi untuk pembuatan karya dalam format 3D dan telah mendukung seluruh tahapan untuk membuat film animasi 3D yang disertai lisensi tak terbatas untuk produk komersial.

#### 2.4.5 Pengertian Adobe Photoshop

Menurut Hidayat (2011: 9) Adobe Photoshop adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolahan gambar/foto.

Menurut Khaeruddin (Dewi 2012:263) Aplikasi program adobe photoshop adalah program yang berorientasi untuk mengedit, modifikasi, maupun memanipulasi gambar atau foto, pengaturan warna yang semakin akurat, penggunaan efek filter yang semakin canggih, dan fasilitas modifikasian tipografi teks yang semakin unik dan fantastik.

Dari penjelasan diatas, dapat disimpulkan bahwa aplikasi adobe photoshop adalah program pengeditan maupun memanipulasi gambar atau foto disertai efek dan modifikasi tipografi teks yang unik dan fantastik.

#### 2.4.6 Pengertian Map Magic

Terdapat penjelasan pengertian Map Magic dalam situs <http://www.denispahunov.ru/MapMagic/> yaitu MapMagic adalah *platform* untuk pembuatan prototipe tingkat cepat. Ini juga mendukung generasi medan tak terbatas. Ini menggunakan antarmuka berbasis node untuk menentukan logika pembuatan: setiap generator (seperti noise, voronoi, campuran, kurva, erosi, pencar, hutan, dll) diwakili oleh node pada grafik. Setelah input dan output node terhubung keajaiban terjadi: peta permainan akan dibuat tanpa keterlibatan



manusia. Tidak perlu susah payah memahat dan menggambar medan yang luas, menempatkan ribuan objek, mengecat bidang rumput: cukup berikan perintah Anda ke plugin dalam bentuk grafik node dan itu akan membuat peta secara otomatis.

Dari penjelasan diatas, dapat disimpulkan bahwa Map Magic adalah platform untuk membuat medan tak terbatas dengan generator grafik sehingga menjadi suatu peta yang bisa disimulasikan dalam suatu aplikasi secara otomatis.