



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Teori Umum

##### 2.1.1 Pengertian Perangkat Lunak

Menurut Sukanto dan Shalahuddin (2018:2), “Perangkat Lunak (*Software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*)”. Sedangkan menurut Pratama (2014:15) “Elemen perangkat lunak (*software*) berfungsi untuk membantu sistem informasi didalam proses pengoperasian, pengolahan data, pengambilan keputusan, analisis, manajemen data, dan lain – lain”.

Dari pernyataan diatas dapat disimpulkan bahwa perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan untuk membantu proses sistem informasi.

##### 2.1.2 Metode Pengembangan Perangkat Lunak

Penelitian ini menggunakan metode pengembangan perangkat lunak *Waterfall*. Menurut Sukanto dan Shalahuddin (2018:28) metode air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) atau alur hidup klasik (*classic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut mulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*).

1. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk menspesifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user. Spesifikasi kebutuhan perangkat lunak ini perlu didokumentasikan.



## 2. Desain

Desain perangkat lunak adalah proses multi langkah yang fokus pada desain pembuatan program sistem termasuk struktur data, arsitektur sistem, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan sistem dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak ini yang dihasilkan pada tahap ini juga perlu didokumentasikan.

## 3. Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain. Desain perangkat lunak ini juga perlu didokumentasikan.

## 4. Pengujian

Pengujian fokus pada perangkat lunak dari segi logika dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

## 5. Pendukung (*support*) atau Pemeliharaan (*maintenance*)

Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke user. Perubahan bisa terjadi karena adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian atau perangkat lunak harus beradaptasi dengan lingkungan baru. Tahap pendukung dan pemeliharaan dapat mengulangi proses pengembangan mulai dari analisis spesifikasi untuk perubahan perangkat lunak yang sudah ada, tapi tidak untuk perangkat lunak yang baru.

## 2.2 Teori Khusus

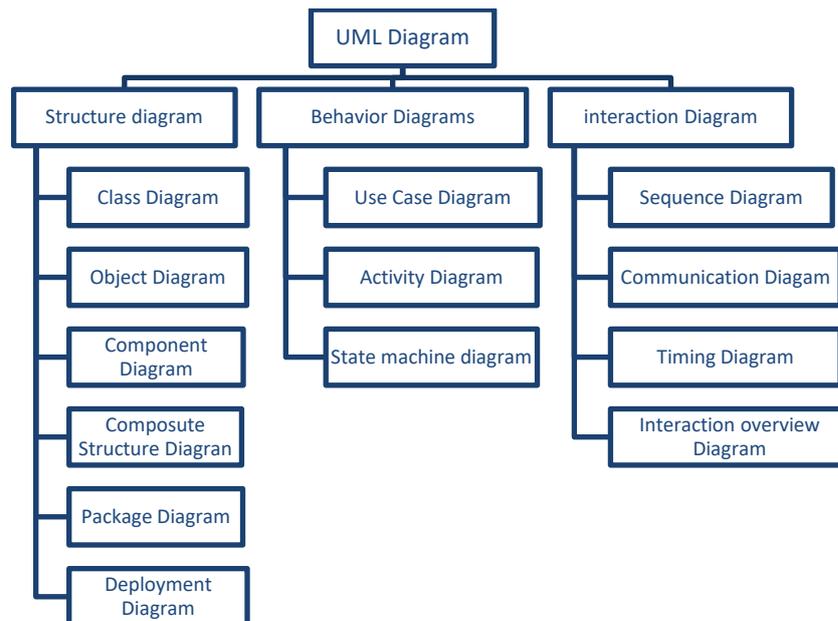
### 2.2.1 Pengertian *Unified Modeling Language* (UML)

Sukanto dan Shalahuddin (2018:137), menjelaskan *Unified Modeling Language* (UML) merupakan bahasa visual untuk pemodelan dan komunikasi



mengenai sebuah sistem dengan menggunakan diagram dan teks – teks pendukung.”

Pada UML terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam – macam diagram tersebut yaitu :



**Gambar 2.1 Diagram UML**

Berikut merupakan penjelasan singkat dari pembagian kategori tersebut.

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. *Interactions diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun sistem interaksi antar subsistem pada suatu sistem.



## 2.2.2 Jenis-Jenis Diagram UML

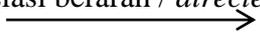
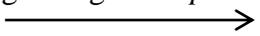
### 2.2.2.1 Pengertian *Class Diagram*

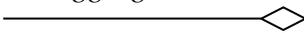
Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *class diagram* sebagai berikut :

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

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

**Tabel 2.1** 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

No	Simbol	Deskripsi
7.	agregasi / <i>aggregation</i> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> )

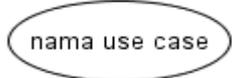
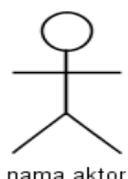
Sumber: Sukamto dan Shalahuddin (2018:146)

### 2.2.2.2 Pengertian *Use case Diagram*

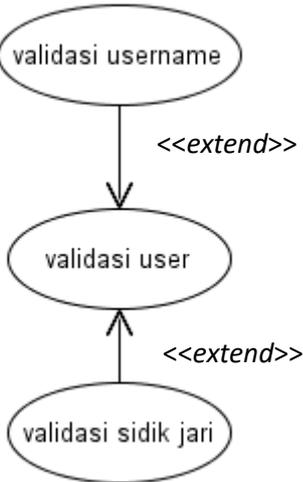
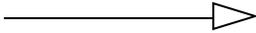
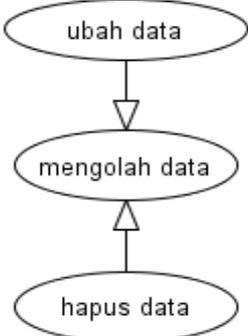
Sukamto dan Shalahuddin (2018:155), menjelaskan tentang *use case diagram* sebagai berikut :

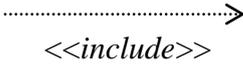
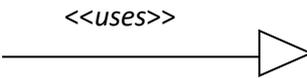
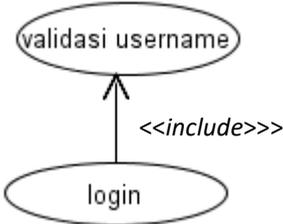
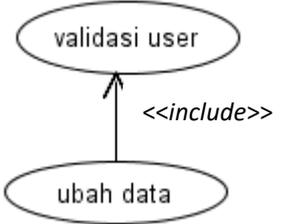
“*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem.” Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:

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

No	Simbol	Deskripsi
1.	<i>Use case</i> 	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> .
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 <i>actor</i> .
3.	asosiasi / <i>association</i> 	komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.



No	Simbol	Deskripsi
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>
5.	Generalisasi / <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,</p>  <p>misalnya: arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum).</p>

No	Simbol	Deskripsi
6.	menggunakan / include / uses    	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 ada dua sudut pandang yang cukup besar mengenai include di <i>use case</i> : <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> <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> kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.

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.

### 2.2.2.3 Pengertian *Activity Diagram*

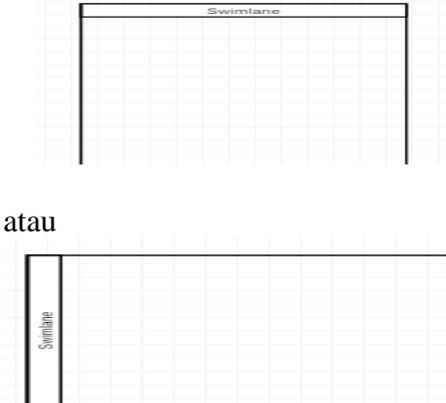
Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

“*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.”

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

**Tabel 2.3** Simbol-simbol *Activity Diagram*

No	Simbol	Deskripsi
1.		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2.		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
3.		Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu
4.		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5.		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

No	Simbol	Deskripsi
6.	Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Sumber: Sukamto dan Shalahuddin (2018:162)

#### 2.2.2.4 Pengertian *Sequence Diagram*

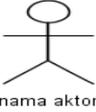
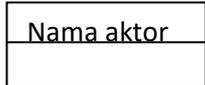
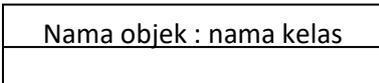
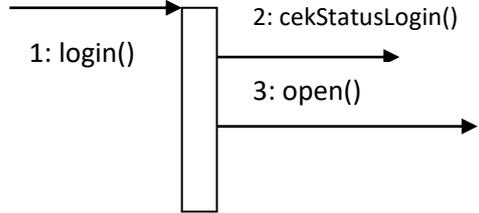
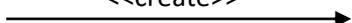
“Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah beserta metode-metode yang dimiliki kelas yang diinstansikan menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case” (Sukamto dan Shalahuddin, 2018:165).

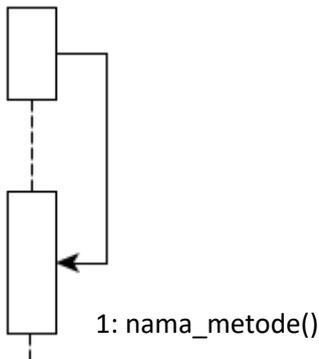
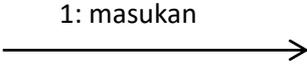
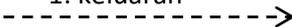
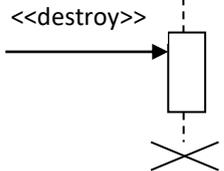
Dapat penulis simpulkan bahwa Sequence diagram adalah penggambaran skenario dari sebuah objek yang ada pada use case yang meliputi rangkaian langkah-langkah aktivitas dari objek berdasarkan waktu hidup objek dan pesan-pesan yang diterima maupun yang dikirimkan objek kepada objek lainnya.



Berikut simbol-simbol pada Sequence Diagram :

**Tabel 2.4** Simbol-simbol pada Sequence Diagram

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>
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 <code>cekStatusLogin ()</code> dan <code>open()</code> dilakukan di dalam metode <code>login()</code> 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>

No	Simbol	Deskripsi
6.	Pesan tipe call 1: nama metode() 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri,  arah panah mengarah pada objek yang memiliki operasi/metode, karena ini memanggil operasi/metode maka operasi/metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi
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 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 (2018:165-167)



## **2.2.3 Metode Pengujian Perangkat Lunak**

### **2.2.3.1 Pengertian Metode Pengujian**

Sukamto dan Shalahuddin (2016:272) menyatakan, “Pengujian adalah satu set aktifitas yang direncanakan dan sistematis untuk menguji atau mengevaluasi kebenaran yang diinginkan. Aktifitas pengujian terdiri dari satu set atau sekumpulan langkah dimana dapat menempatkan desain kasus uji yang spesifik dan metode pengujian.”

### **2.2.3.2 Metode Pengujian**

Secara umum pola pengujian perangkat lunak adalah sebagai berikut:

1. Pengujian dimulai dari level komponen hingga integrasi antar komponen menjadi sebuah sistem.
2. Teknik pengujian berbeda-beda sesuai dengan berbagai isi atau unit uji dalam waktu yang berbeda-beda pula bergantung pada pengujian pada bagian mana yang dibutuhkan.
3. Pengujian dilakukan oleh pengembang perangkat lunak, dan jika untuk proyek besar, pengujian bisa dilakukan oleh tim uji yang tidak terkait dengan tim pengembang perangkat lunak (independent test group (ITG)).
4. Pengujian dan penirkutan (debugging) merupakan aktivitas yang berbeda tetapi penirkutan (debugging) harus diakomodasikan pada berbagai strategi pengujian.

### **2.2.3.3 Back-Box Testing (Pengujian Kotak Hitam)**

Sukamto dan Shalahuddin (2018:275) menjelaskan “*Black-box testing* (pengujian kotak hitam) yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan”.



Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misalkan untuk kasus proses login maka kasus uji coba yang dilakukan adalah:

1. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika user memasukkan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalkan nama pemakai benar tapi kata sandi salah, atau sebaliknya atau keduanya salah.

## **2.3 Teori Judul**

### **2.3.1 Pengertian Aplikasi**

Menurut Mutohari et al. (2016:158) “Aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data.” Sedangkan menurut Deslianti dan Imam Muttaqin (2016:27) “Aplikasi adalah suatu subkelas perangkat lunak *computer* yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media”

Berdasarkan pendapat diatas dapat disimpulkan bahwa aplikasi adalah perangkat lunak yang digunakan untuk melakukan suatu tugas atau pekerjaan berupa penerapan, penggunaan, pengolahan, penambahan data dan pemutar media.

### **2.3.2 Pengertian Beasiswa**

Menurut Sulistyو dan Sri Winiarti (2015:965) “Beasiswa merupakan pemberian berupa bantuan keuangan maupun pendidikan yang diberikan perorangan, mahasiswa atau pelajar yang digunakan demi keberlangsungan pendidikan yang ditempuh bagi yang memiliki prestasi dibidang akademik, non akademik dan kemampuan ekonominya lemah, serta telah memenuhi syarat-syarat yang ditentukan pihak pemberi beasiswa.”



Menurut Junaidi dan Fany Visella (2017:119), Beasiswa dapat dikatakan sebagai pembiayaan yang tidak bersumber dari pendanaan sendiri atau orang tua, akan tetapi diberikan oleh pemerintah, perusahaan swasta, kedutaan, universitas, serta lembaga pendidik atau peneliti, atau juga kantor tempat bekerja yang karena prestasi seorang karyawan dapat diberikan kesempatan untuk meningkatkan kapasitas sumber daya manusianya melalui pendidikan.

Berdasarkan pengertian tersebut, beasiswa adalah bantuan pendanaan yang bersumber dari pemerintah, perusahaan swasta, kedutaan, universitas, serta lembaga pendidik atau peneliti, atau juga dari kantor tempat bekerja untuk meningkatkan kapasitas pendidikan.

### **2.3.3 Pengertian CSR (*Corporate Social Responsibility*)**

Menurut Parengkuan(2017:566) “*Corporate Social Responsibility (CSR)* atau pertanggung-jawaban sosial perusahaan adalah mekanisme bagi suatu organisasi untuk secara sukarela mengintegrasikan perhatian terhadap lingkungan dan sosial ke dalam operasinya dan interaksinya dengan *stakeholders*, yang melebihi tanggung jawab organisasi di bidang hukum.” Sedangkan menurut Awuy, Vinta Paulinda, dan Yosefa Sayekti dan Indah Purnamawati (2016:17) “*Corporate Social Responsibility (CSR)* merupakan komitmen sosial dari perusahaan sebagai bentuk kepedulian perusahaan atas masyarakat dan juga kelestarian lingkungan serta tanggung jawab terhadap kesejahteraan tenaga kerja perusahaan di dalam praktek bisnis perusahaan.

Dari pengertian diatas, dapat disimpulkan bahwa *Corporate Social Responsibility (CSR)* adalah suatu tanggung jawab sosial perusahaan sebagai wujud kepedulian terhadap aspek lingkungan dan sosial di dalam praktik bisnis perusahaan.



### 2.3.4 Pengertian Website

Menurut Abdullah (2015:1) “Website dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet.”

Sedangkan menurut Abdullah (2015:42) dalam jurnalnya menyebutkan :

“Terminologi website adalah kumpulan dari halaman-halaman situs, yang biasanya terangkum dalam sebuah domain atau subdomain, yang tempatnya berada didalam World Wide Web (WWW) di Internet. Sebuah web page adalah dokumen yang ditulis dalam format HTML (*Hyper Text Markup Language*), yang hampir selalu bisa diakses melalui HTTP, yaitu protokol yang menyampaikan informasi dari server website untuk ditampilkan kepada para pemakai melalui web browser.”

Dapat disimpulkan, website adalah kumpulan dari beberapa halaman yang berisi informasi terangkum dalam domain dan dapat di akses melalui koneksi internet.

### 2.3.5 Pengertian Aplikasi Pendaftaran Beasiswa Biaya Pendidikan Sekitar Bukit Asam (Bidiksiba) pada CSR (*Corporate Social Responsibility*) pada PT Bukit Asam Tbk

Berdasarkan pengertian yang telah dijelaskan diatas dapat disimpulkan bahwa aplikasi pendaftaran beasiswa biaya pendidikan sekitar Bukit Asam (bidiksiba) pada CSR (*Corporate Social Responsibility*) merupakan suatu perangkat lunak yang dibuat untuk melakukan pendaftaran bantuan Biaya Pendidikan sekitar Bukit Asam (Bidiksiba) pada CSR (*Corporate Social Responsibility*) atau pertanggungjawaban sosial perusahaan PT Bukit Asam Tbk di bidang pendidikan.

## 2.4 Teori Program

### 2.4.1 Pengertian PHP

Menurut Abdullah (2015:3) “PHP (*Hypertext Processor*) merupakan *server-hide programming*, yaitu bahasa pemrograman yang di proses di sisi



server”. Pendapat lain oleh Abdullah (2015:42) “PHP adalah bahasa *scripting* yang menampilkan output HTML ataupun output lain sesuai dengan keinginan pemrogram (misalnya: PDF dan lain-lain) yang dijalankan pada *server side*. Artinya semua sintak yang diberikan akan sepenuhnya dijalankan pada server sedangkan yang dikirim pada browser hanya menghasilkan output saja.”

Dapat disimpulkan, PHP merupakan bahasa pemrograman yang dijalankan dengan *server hide programming* yang menghasilkan output pada browser.

#### **2.4.2 Pengertian HTML**

Menurut Abdulloh (2015:2) “HTML singkatan dari *Hyper Text Markup Language* yaitu script yang berupa tag – tag untuk membuat dan mengatur struktur website.” Menurut Harison dan Ahmad Syarif (2016:43) “*HyperText Markup Language* (HTML) adalah sebuah bahasa markup yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan formating hypertext sederhana yang ditulis kedalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.”

Berdasarkan pendapat diatas, (*HyperText Markup Language*) HTML adalah bahasa markup berupa tag – tag yang digunakan untuk membuat halaman web dan menampilkan informasi didalam sebuah web internet.

#### **2.4.3 Pengertian CSS**

CSS merupakan sebuah pemrograman yang mempunyai kontrol terhadap semua tampilan dan desain dari sebuah halaman website (Johan dkk, 2015:134). Pendapat lain dikemukakan oleh Abdulloh (2015:2) “CSS (*Cascading Style Sheets*) yaitu skrip yang digunakan untuk mengatur desain website.” Jadi CSS atau *Cascading Style Sheets* adalah skrip yang digunakan untuk mengatur tampilan dan desain dari halaman website.



#### 2.4.4 Pengertian Basis Data

Menurut Abdullah (2015:42) “Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang, dan lain-lain. Data dinyatakan dengan nilai (angka, deretan karakter, atau simbol).”

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut :

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan secara cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan (*redudancy*) yang tidak perlu, untuk memenuhi kebutuhan.
- c. Kumpulan *file*/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

Database adalah sebuah tempat penyimpanan yang besar dimana terdapat kumpulan data yang tidak hanya berisi data operasional tetapi juga deskripsi data. Sedangkan Menurut Rahmad dan Dedi Setiadi (2014:1333) “Basis data adalah kumpulan file-file yang saling berelasi, relasi tersebut biasa ditunjukkan dengan kunci dari tiap file yang ada.”

Dapat disimpulkan bahwa basis data atau database adalah sekumpulan data yang berisi deskripsi dari data yang dinyatakan dalam bentuk angka, karakter dan symbol yang dibutuhkan dalam sebuah organisasi.

#### 2.4.5 Pengertian MySQL

Menurut Abdullah (2015:42) “MySQL merupakan sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user. MySQL sangat populer pengembang web (*web developers*) karena memiliki kecepatan dan ukuran yang kecil membuat MySQL lebih ideal untuk website ditambah lagi dengan fakta bahwa MySQL adalah *open source* yang berarti



gratis.” Dalam pengertian lain oleh Yuliansyah (2014:826) “MySQL adalah sebuah database manajemen system (DBMS) populer yang memiliki fungsi sebagai *relational database manajemen system* (RDBMS).”

Jadi, dapat disimpulkan, MySQL adalah sebuah perangkat lunak basis data yang memiliki fungsi sebagai *relational database manajemen system* RDBMS yang multithread dan multi-user.

#### **2.4.5 Pengertian XAMPP**

XAMPP merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web server Apache, PHP dan MySQL secara manual. XAMPP akan menginstallasi dan mengkonfigurasikannya secara otomatis untuk anda atau auto konfigurasi. (Lestanti dan Ardina, 2016:69)

Sedangkan menurut Poipessy dan Mirna Umasangadji (2018:40) “XAMPP merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP.” Dapat disimpulkan XAMPP merupakan paket konfigurasi Apache, PHP dan MySQL yang digunakan untuk membantu pengembangan aplikasi.

#### **2.4.6 Pengertian JQuery**

Menurut Harison (2016:42) “JQuery merupakan sebuah *Javascript Library* atau bisa disebut juga sebagai perpustakaan dari kumpulan kode/*listing Javascript* yang siap pakai.” Sedangkan menurut Prabowo dan Mamay (2016:72) “JQuery adalah kumpulan kode/fungsi *Javascript* siap pakai, sehingga memudahkan dan mempercepat dalam membuat kode *Javascript*.”

Dari pengertian tersebut dapat disimpulkan bahwa, JQuery merupakan kumpulan *Javascript* yang berisi kode *listing* siap pakai yang memudahkan pengguna untuk membuat kode *Javascript*.



### 2.4.7 Pengertian *Sublime Text*

Menurut Syifani dan Ardiansyah (2018:25) “Sublime Text adalah aplikasi editor untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API”. Sedangkan menurut Putra (2016:181) “Sublime text adalah teks editor berbasis Python, sebuah teks editor yang elegan, kaya fitur, cross platform, mudah dan simpel yang cukup terkenal dikalangan *developer* (pengembang) dan *desainer*”.

Dapat disimpulkan bahwa, sublime text merupakan aplikasi editor kode dan teks yang kaya fitur, *cross platform*, mudah dan *simple* yang berbasis Phyton.