



## BAB II TINJAUAN PUSTAKA

### 2.1. Teori Umum

#### 2.1.1 Pengertian Komputer

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

Sedangkan menurut Rosdiana (2016:1), “Secara definisi komputer diterjemahkan sebagai sekumpulan alat elektronik yang saling berkeja sama, dapat menerima data (*input*), mengolah data (*proses*) dan memberikan informasi (*output*) serta terkoordinasi dibawah *control* program yang tersimpan di memorinya”.

Dari pernyataan yang telah ditemukan oleh kedua pengarang diatas, maka penulis menarik kesimpulan bahwa komputer adalah sebuah alat elektronik yang memiliki fungsi mengelola data dengan berbagai proses.

#### 2.1.2 Pengertian Perangkat Lunak (*Software*)

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 penggunaa (*user manual*)”.

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

Jadi, dapat disimpulkan bahwa perangkat lunak merupakan program yang berisikan sebuah instruksi yang terasosiasi dengan dokumentasi perangkat lunak yang diperintahkan dan diproses dengan bantuan perangkat keras.



### 2.1.3 Pengertian Basis Data (*Database*)

Menurut Lubis (2016:2), “Basis data merupakan gabungan *file* yang dibentuk dengan hubungan/relasi yang logis dan dapat diungkapkan dengan catatan serta bersifat *independent*”.

Sedangkan menurut Fathansyah (dikutip Sagita dan Sugiarto, 2016:51), “Basis data adalah himpunan kelompok data atau arsip yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah”.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa basis data adalah kumpulan data yang berbentuk informasi.

### 2.1.4 Pengertian Internet (*Interconnected Network*)

Internet merupakan jaringan luas yang menghubungkan komputer-komputer di seluruh dunia, dengan internet sebuah komputer bisa mengakses informasi atau data yang terdapat pada komputer lain di benua yang berbeda (Hidayatullah dan Kawistara dikutip Ayu dan Permatasari, 2018:19).

Menurut Pibriana dan Ricoida (2017:105), “Internet adalah penghubung antara organisasi dan pelanggannya, sehingga tercipta sebuah organisasi baru secara *visual*”.

Dari kedua pendapat di atas dapat disimpulkan bahwa internet adalah jaringan yang menghubungkan komputer satu dengan yang lain untuk saling bertukar informasi.

## 2.2 Teori Judul

### 2.2.1 Pengertian Metode *Technique for Order Preference by Similarity to Ideal* (TOPSIS)

Menurut Sari (2018:96), “TOPSIS menggunakan prinsip bahwa alternatif yang terpilih harus mempunyai jarak terdekat dari solusi ideal positif dan terjauh dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak



Euclidean untuk menentukan kedekatan relatif dari suatu alternatif dengan solusi optimal”.

Menurut Nofriansyah dan Defit (2020:41), “Metode ini merupakan salah satu metode yang digemari oleh peneliti di dalam merancang sebuah Sistem Pendukung Keputusan, selain konsepnya yang sederhana tetapi kompleksitas dalam pemecahan masalah baik itu ditandai dengan konsep penyelesaian metode ini yaitu dengan memilih alternatif terbaik yang tidak hanya memiliki jarak terpendek dari solusi ideal positif tetapi juga memiliki jarak terpanjang dari solusi ideal negatif”. Adapun algoritma penyelesaian metode ini yaitu :

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan dijadikan tolak ukur penyelesaian masalah.
2. Langkah 2 : Menormalisasi setiap nilai alternatif (matriks ternormalisasi) dan matriks ternormalisasi berbobot.
3. Langkah 3 : Menghitung nilai Solusi Ideal Positif atau negatif.
4. Langkah 4 : Menghitung *Distance* nilai berbobot setiap alternatif terhadap solusi ideal positif dan negatif.
5. Langkah 5 : Menghitung nilai preferensi dari setiap alternatif.
6. Langkah 6 : Melakukan perangkingan.

Dari kedua definisi diatas, penulis menyimpulkan bahwa *Technique for Order Preference by Similarity to Ideal* (TOPSIS) adalah metode yang digunakan pada sistem pendukung keputusan dengan memilih solusi terpendek untuk memilih solusi positif, tetapi juga memiliki solusi terpanjang dari ideal negatif

### **2.2.2 Pengertian Implementasi Metode *Technique for Order Preference by Similarity to Ideal* (TOPSIS) untuk Peningkatan Perguruan Tinggi Wilayah II Pada Lembaga Layanan Pendidikan Tinggi Wilayah II Daerah Sumatera Bagian Selatan**

Implementasi Metode *Technique for Order Preference by Similarity to Ideal* (TOPSIS) untuk Peningkatan Perguruan Tinggi Wilayah II Pada Lembaga



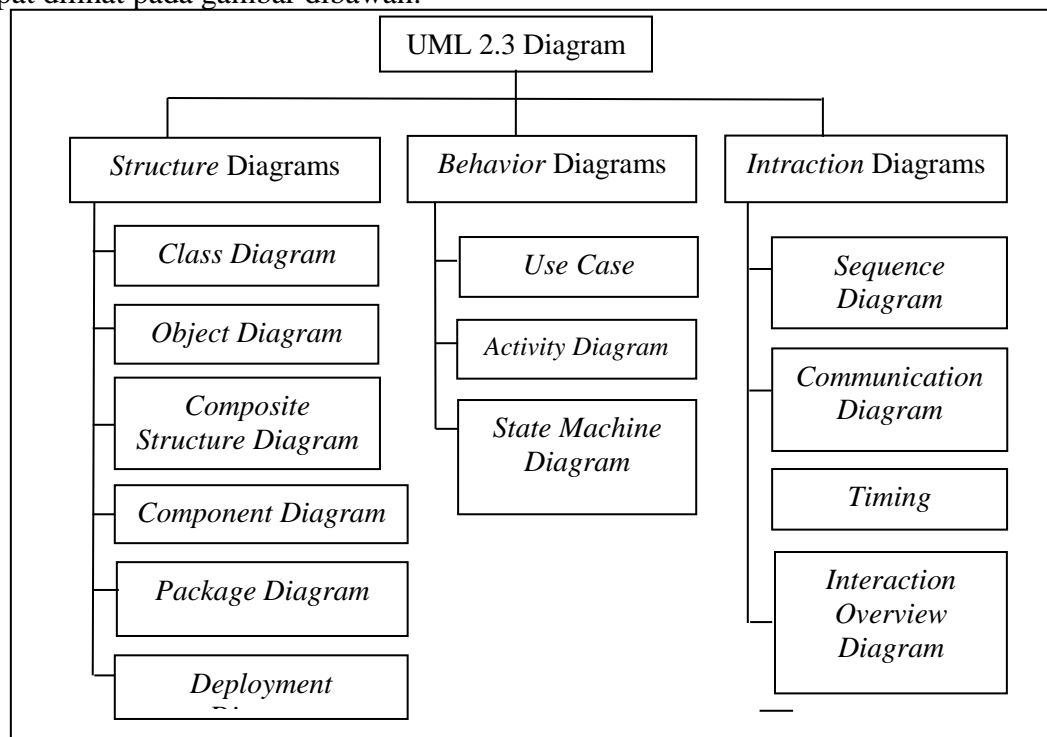
Layanan Pendidikan Tinggi Wilayah II Daerah Sumatera Bagian Selatan adalah sebuah program komputer yang dibuat untuk membantu tim penilai dalam melakukan kegiatan pemeringkatan perguruan tinggi untuk wilayah II.

## 2.3 Teori Khusus

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

Menurut Gata dan Grace (dikutip Hendini 2016:108), “*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem”.

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:



Sumber : Sukamto dan Shalahuddin (2018:140)

**Gambar 2.1** Macam-macam Diagram UML



Penjelasan singkat dari pembagian kategori pada diagram UML menurut Sukamto dan Shalahuddin (2016:141) :

- 1) *Structure diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
- 2) *Behavior diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
- 3) *Interaction diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

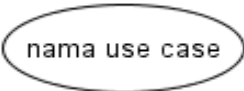


### **2.3.2 Jenis-jenis Diagram *Unified Modelling Language* (UML)**

#### **2.3.2.1 Pengertian *Use Case Diagram***


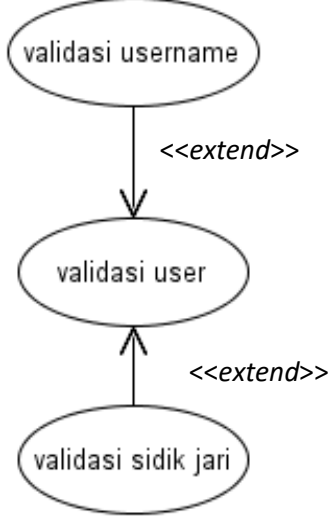
Menurut Sukamto dan Shalahuddin (2018:155), “*Use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem”. Adapun simbol-simbol yang digunakan dalam *use case* adalah sebagai berikut:



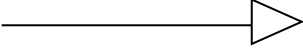
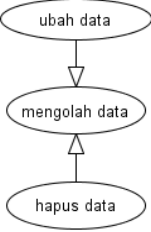
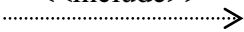
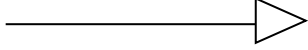
**Tabel 2.1** Simbol-simbol pada *Use case* Diagram

No	Simbol	Deskripsi
1	<p data-bbox="555 533 667 566"><i>Use case</i></p>  <p data-bbox="517 647 719 674">nama use case</p>	<p data-bbox="871 501 1362 808">fungsionalitas 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 data-bbox="549 853 708 887">aktor / <i>actor</i></p>  <p data-bbox="555 1077 687 1099">nama aktor</p>	<p data-bbox="871 853 1350 1323">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 data-bbox="491 1406 767 1440">asosiasi / <i>association</i></p> 	<p data-bbox="871 1397 1337 1480">komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i>.</p>

Lanjutan **Tabel 2.1** Simbol-simbol pada *Use case* Diagram

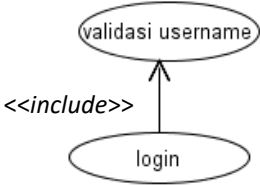
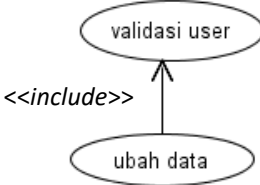
No	Simbol	Deskripsi
4	ekstensi / <i>extend</i> 	<p>relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang di tambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <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>

Lanjutan **Tabel 2.1** Simbol-simbol pada *Use case* Diagram

No	Simbol	Deskripsi
5	Generalisasi / <i>generalization</i> 	hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:  arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)
6	menggunakan / <i>include</i> / <i>uses</i> <<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</li> </ul>



Lanjutan **Tabel 2.1** Simbol-simbol pada *Use case* Diagram

No	Simbol	Deskripsi
		<p>dijalankan, misalnya pada kasus berikut:</p>  <pre> graph BT     login([login]) -- "&lt;&lt;include&gt;&gt;" --&gt; validasi_username([validasi username])   </pre> <p><i>Include</i> berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang di tambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</p>  <pre> graph BT     ubah_data([ubah data]) -- "&lt;&lt;include&gt;&gt;" --&gt; validasi_user([validasi user])   </pre> <p>kedua interpretasi di atas dapat dianut salah satu atau keduanya tergantung pada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber : Sukamto dan Shalahuddin (2018:156-158)

### 2.3.2.2 Pengertian *Activity Diagram*





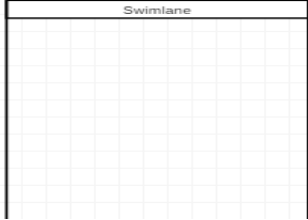
Sukamto dan Shalahuddin (2018:161), menjelaskan tentang *activity diagram* sebagai berikut :

*Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas

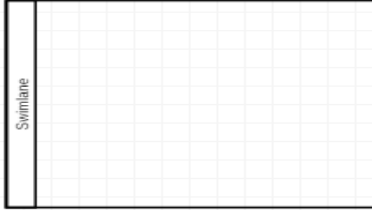


dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:

**Tabel 2.2** Simbol-simbol pada *Activity Diagram*

No	Simbol	Deskripsi
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> 	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	<i>Swimlane</i> 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

Lanjutan Tabel 2.2 Simbol-simbol pada *Activity Diagram*

No	Simbol	Deskripsi
	<p>atau</p> 	

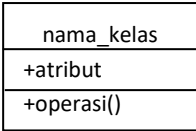

Sumber : Sukamto dan Shalahuddin (2018:162-163)

### 2.3.2.3 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 Class* 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 pada *Class Diagram*

No	Simbol	Deskripsi
1	<p>kelas</p> 	Kelas pada struktur sistem
2	<p>antarmuka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

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

No	Simbol	Deskripsi
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 (2018:146-147)

### 2.3.2.4 Pengertian *Sequence Diagram*

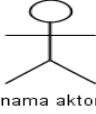
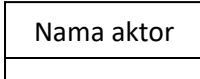
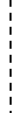
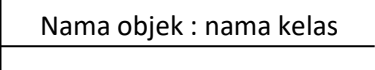

Sukamto dan Shalahuddin (2018:141), menjelaskan tentang *Sequence diagram* sebagai berikut :

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



Banyaknya diagram *sequence* yang harus digambarkan adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram *sequence* sehingga semakin banyak *use case* yang didefinisikan maka diagram *sequence* yang harus dibuat juga semakin banyak. Berikut simbol-simbol pada *Sequence Diagram* :

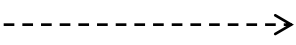
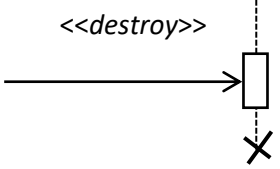
**Tabel 2.4** Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
1	<p style="text-align: center;"><i>Actor</i></p>  <p style="text-align: center;">nama aktor</p> <p style="text-align: center;">atau</p>  <p style="text-align: center;">Nama aktor</p> <p style="text-align: center;">tanpa waktu aktif</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 aktor
2	<p style="text-align: center;">Garis hidup / <i>lifeline</i></p> 	menyatakan kehidupan suatu objek
3	<p style="text-align: center;">Objek</p>  <p style="text-align: center;">Nama objek : nama kelas</p>	menyatakan objek yang berinteraksi pesan
4	<p style="text-align: center;">Waktu aktif</p> 	menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya, misalnya

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

	Simbol	Deskripsi
		<p>maka cekStatusLogin () dan open() dilakukan di dalam metode login() aktor tidak memiliki waktu aktif</p>
5	Pesan tipe <i>create</i> 	menyatakan suatu objek membuat objek yang lain, arah panah mengarah kepada objek yang dibuat
6	Pesan tipe <i>call</i> 	menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri, <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 <i>send</i> 	menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim

Lanjutan Tabel 2.4 Simbol-simbol pada *Sequence Diagram*

No	Simbol	Deskripsi
8	<p>Pesan tipe <i>return</i></p> <p>1: keluaran</p> 	menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
9	<p>Pesan tipe <i>destroy</i></p> 	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

Sumber : Sukamto dan Shalahuddin (2018:165-167)

## 2.4 Teori Program

### 2.4.1 Pengertian Web

Web merupakan layanan yang paling banyak digunakan di internet untuk menyampaikan informasi karena sifatnya mendukung multimedia tidak hanya disampaikan melalui teks, tapi juga gambar, video dan suara (Swara dan Pebriadi, 2016:32).

Menurut Hariyanto (dikutip Destianingrum 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*)”

Dari kedua pendapat di atas, dapat diartikan bahwa web adalah sekumpulan halaman yang berisi berbagai informasi dalam bentuk teks, gambar, suara, video dan lain-lain yang terhubung melalui jaringan.



#### 2.4.2 Pengertian HTML (*Hyper Text Markup Language*)

Kustiyahningsih dan Anamisa (dikutip Sasongko, 2017:147), Menjelaskan bahwa HTML kependekan dari *Hyper Text Markup Language* yang *file* teks murni yang dapat dibuat dengan editor teks sembarang. Dokumen ini dikenal sebagai web *page*. *File-file* HTML ini berisi instruksi-instruksi yang kemudian diterjemahkan oleh browser yang ada dikomputer *client (user)* sehingga isi formasinya dapat ditampilkan secara visual dikomputer pengguna (*user*).

Hidayatulah dan Kawistara (dikutip Ayu dan Permatasari, 2018:20), menjelaskan tentang HTML sebagai berikut :

*Hypertext Markup Language* (HTML) adalah Bahasa standard yang digunakan untuk menampilkan halaman web. Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dan halaman web dan isinya.
2. Membuat tabel dalam halaman web.
3. Mempublikasikan dalam halam web secara *online*.
4. Membuat *form* yang bisa digunakan untuk menangani registrasi dan transaksi via web.

Contoh: Setiap dokumen HTML diawali dan diakhiri dengan tag HTML.

Dari penjelasan di atas dapat disimpulkan bahwa *Hyper Text Markup Language* (HTML) adalah bahasa standar yang merupakan teks murni yang berisi instruksi yang diterjemahkan oleh browser dan ditampilkan secara *visual* di komputer pengguna.

#### 2.4.3 Pengertian PHP (*Hypertext Preprocessor*)

Menurut Madcoms (dikutip Ayu dan Permatasari, 2018:20) “PHP (*Hypertext Preprocessor*) adalah bahasa *script* yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs web dinamis”.

Sedangkan Menurut Hastanti dkk. (2015:4), “PHP adalah bahasa pemrograman skrip sederhana yang digunakan untuk pemrosesan HTML form di dalam halaman web. Strukturnya sangat sederhana sehingga PHP dapat





dengan mudah dipelajari *programmer* pemula bahkan orang tanpa latar belakang teknologi informasi”.

Dari kedua pendapat tersebut dapat disimpulkan bahwa *Hypertext Preprocessor*) adalah bahasa pemrograman yang biasanya digunakan untuk membuat web dinamis.

#### 2.4.4 Pengertian CSS (*Cascading Style Sheet*)

Menurut Saputra dan Agustin (dikutip Sagita dan Sugiarto 2016:51), “CSS atau yang memiliki kepanjangan *Cascading Style Sheet*, merupakan suatu bahasa pemrograman web yang digunakan untuk mengendalikan dan membangun berbagai komponen dalam web sehingga tampilan web akan lebih rapi, terstruktur, dan seragam”.

Sedangkan Djaelangkara dkk. (2015:88), menjelaskan bahwa *Cascading Style Sheet* (CSS) adalah aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Pada umumnya CSS dipakai untuk memformat tampilan halaman web yang dibuat dengan bahasa HTML dan XHTML.

Dari kedua penjelasan di atas dapat disimpulkan bahwa *Cascading Style Sheet* (CSS) adalah sebuah bahasa pemrograman web atau aturan untuk mengendalikan dan membangun beberapa komponen dalam sebuah web agar terstruktur, lebih rapih dan seragam.

#### 2.4.5 Pengertian *Javascript*

Menurut Hidayatullah dan Kawistara (dikutip Ayu dan Permatasari, 2018:21), “*JavaScript* (*js*) ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu web”.

Kustiyahningsih dan Anamisa (dikutip Lavarino dan Yustanti, 2016:74), menjelaskan tentang *JavaScript* sebagai berikut :

“*JavaScript* adalah bahasa yang berbentuk kumpulan skrip berjalan pada suatu dokumen HTML. Bahasa ini adalah bahasa pemrograman untuk



memberikan kemampuan tambahan terhadap HTML dengan mengizinkan pengeksekusian perintah-perintah disisi *user* variabel atau fungsi dengan nama *TEST* berbeda dengan variabel dengan nama *test* dan setiap instruksi diakhiri dengan artinya disisi browser bukan disisi server web. *JavaScript* adalah bahasa yang “*case sensitive*” artinya membedakan penamaan variabel dan fungsi yang menggunakan huruf besar dan huruf kecil, contoh karakter titik koma”.

Dari kedua pendapat di atas, maka dapat disimpulkan bahwa *javascript* adalah suatu bahasa yang berbentuk kumpulan *script* yang merupakan kemampuan tambahan sebagai fungsionalitas dalam membuat suatu web.

#### 2.4.6 Pengertian *Sublime Text*

Menurut Bos (dikutip Pahlevi dkk., 2018:29), “*Sublime Text* merupakan salah satu *text editor* yang sangat *powerful* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi”.

Sedangkan menurut Annisak dkk. (2017:2), “*Sublime text* adalah teks *editor* berbasis Python, sebuah teks *editor* yang elegan, kaya fitur, *cross platform*, mudah dan *simple* yang cukup terkenal di kalangan *developer* (pengembang), penulis *source code* dan *desaigner*”.

Berdasarkan kedua pendapat di atas, dapat disimpulkan bahwa *sublime text* merupakan sebuah teks *editor* berbasis *Python* yang sangat *powerful*, elegan, kaya fitur, *cross platform*, mudah dan *simple* yang dapat meningkatkan produktivitas dan mengembangkan kualitas kode yang tinggi dan berguna bagi kalangan pengembang, penulis *source code* dan *desaigner*.

#### 2.4.7 Pengertian MySQL

Kurniawan (dikutip Pahlevi dkk., 2018:28), menjelaskan mengenai MySQL sebagai berikut:

“MySQL adalah salah satu jenis *database* yang banyak digunakan untuk membuat aplikasi berbasis web yang dinamis. MySQL termasuk jenis RDBMS (*Relational Database Management System*). MySQL ini mendukung Bahasa

---



pemrograman PHP. MySQL juga mempunyai *query* atau bahasa SQL (*Structured Query Language*) yang *simple* dan menggunakan *escape character* yang sama dengan PHP”.

Sedangkan menurut Lavarino dan Yustanti (2016:73), “MySQL ( *MY Structure Query Language* ) adalah salah satu *Database Management System* ( *DBMS* ) dari sekian banyak *DBMS* seperti Oracle, MS SQL, Postagre SQL, dan lainnya. MySQL berfungsi untuk mengolah Basis Data menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bisa menggunakannya secara gratis”.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa MySQL merupakan salah satu jenis *Database Management System* (*DBMS*) yang digunakan untuk membuat aplikasi berbasis web dinamis yang bersifat *open source*.

#### **2.4.8 Pengertian Laragon**

Menurut Utama (dikutip Harianto dkk, 2019:14), “Lavarel merupakan sebuah *framework* PHP yang akan membantu para *web developer* untuk menciptakan aplikasi istimewa secara simpel, kode yang bersih dan menyenangkan”.

Menurut Putra dkk (2019:3) Laragon ialah perangkat lunak bebas yang didalamnya terdapat banyak sistem operasi sebagai *localhost* atau server mandiri. Laragon menyediakan banyak layanan, peralatan, dan fitur yang terdiri dari *Apache*, *PHP*, *Server*, *PHPMysqlAdmin*, *MySQL*, *Memchaced Redis*, *Composer*, *Xdebug*, *Cmdr* dan *Lavarel*.

Berdasarkan penjelasan di atas, dapat disimpulkan bahwa Laragon adalah perangkat lunak yang menyediakan *framework* PHP secara simpel.



## 2.5 Penelitian Terdahulu

**Tabel 2.5** Penelitian Terdahulu

No.	Nama Penulis	Judul Penelitian	Hasil Penelitian
1.	Miftakhur Rohman	Pemeringkatan Perguruan Tinggi Swasta Di Malang Menggunakan <i>Automated Website Evaluation Dan Linear Weightage Model</i> , p-ISSN 2460-8181 dan e- ISSN 2502-8995	Penelitian ini bertujuan untuk mengetahui capaian kampus swasta lainnya, maka perlu dilakukan penilaian independen sehingga dapat diketahui capaian masing-masing website PTS lainnya.
2.	Donny Alianto, dkk	Pemeringkatan Perguruan Tinggi Menggunakan Metode <i>Probabilistic Latent Semantic Analysis (pLSA)</i> untuk Mengukur Tingkat Kesiapterapan Teknologi di Indonesia, Vol. 6, No. 2 (2017), 2337-3520 (2301- 928X Print)	Penelitian tersebut menjelaskan bahwa mengukur kualitas perguruan tinggi dapat dilihat dari berbagai indikator penilaian, salah satunya adalah reputasi akademik perguruan tinggi yang memiliki peran penting karena memiliki tingkat hubungan yang tinggi dengan indikator penilaian lainnya.

Lanjutan **Tabel 2.5** Penelitian Terdahulu

No.	Nama Penulis	Judul Penelitian	Hasil Penelitian
3.	Mahdalena Simanjorang	Sistem Pendukung Keputusan Pemilihan Dosen Terbaik dengan menggunakan Metode TOPSIS Studi Kasus: STMIK Pelita Nusantara Medan, p-ISSN : 2548- 6985 dan e-ISSN : 2599-3089	Dalam penelitian ini didasarkan pada konsep bahwa alternatif terpilih yang terbaik tidak hanya memiliki jarak terpendek dari solusi ideal positif tetapi juga memiliki jarak terpanjang dari solusi ideal negatif.
4.	Gunawan Wibisono, dkk	Penelitian Metode TOPSIS Dalam Penentuan Dosen Terbaik, p-ISSN 2087-1716 dan e- ISSN 2548-7779	Penelitian ini menjelaskan bahwa TOPSIS dipilih karena beberapa kelebihanannya, antara lain konsep yang sederhana dan mudah dipahami, komputasi yang efisien, dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif alternatif keputusan dalam bentuk matematis sederhana.