



## BAB II TINJAUAN PUSTAKA

### 2.1 Teori Umum

#### 2.1.1 Pengertian Komputer

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

Menurut Wahyudin dan Munir (2018:1), “Komputer adalah suatu peralatan elektronik yang dapat menerima *input*, mengolah *input*, memberikan informasi, menggunakan suatu program yang tersimpan di memori komputer, dapat menyimpan program dan hasil pengolahan, serta bekerja secara otomatis.”

Menurut Rohman (2017:1), “Komputer merupakan teknologi yang memiliki kemampuan seperti manusia yaitu menghitung, berlogika, mengawas, mengolah, berkomunikasi dan lain-lain.”

Dari pendapat para ahli diatas dapat disimpulkan bahwa, komputer adalah sebuah teknologi alat bantu yang digunakan untuk mengolah, memberikan informasi, serta memproses data-data secara elektronik dengan kemampuan seperti manusia karena dapat melakukan berbagai pekerjaan yang dilakukan manusia.

#### 2.1.2 Pengertian Perangkat Lunak

Menurut Kadir (2017:2), “Perangkat lunak adalah instruksi-instruksi yang ditujukan kepada komputer agar dapat melaksanakan tugas sesuai kehendak pemakai. Sistem operasi seperti *Windows*, *Mac OS*, dan *Linux*, dan aplikasi *Microsoft Word* dan *Microsoft Excel* adalah contoh perangkat lunak.”

Menurut Sukamto dan Shalahuddin (2016:2), “Perangkat Lunak (*software*) adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan, model desain, dan cara penggunaan (*user manual*).”



Sedangkan menurut Kristanto (2018:1), “Perangkat lunak (*software*) adalah instruksi (program komputer) yang ketika dijalankan menyediakan fungsi dan tampilan yang diinginkan, struktur data yang memberi kesempatan program untuk memanipulasi informasi dan dokumen yang mendeskripsikan operasi dan penggunaan program.”

Dari beberapa pengertian diatas penulis menyimpulkan bahwa *software* atau perangkat lunak adalah sekumpulan perintah atau instruksi yang dijalankan oleh komputer untuk melaksanakan tugas dari *user*.

### 2.1.3 Pengertian Basis Data

Menurut Yudho dan Helmi (2019:16), “Basis data (*database*) adalah kumpulan data yang disimpan secara sistematis di dalam komputer yang dapat diolah atau dimanipulasi menggunakan perangkat lunak (program aplikasi) untuk menghasilkan informasi.”

Menurut Sukanto dan Shalahuddin (2016:43), “Sistem Basis Data adalah sistem yang terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi teredia saat dibutuhkan.”

Menurut Fathansyah (2015:2), “Basis Data (*database*) adalah himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.”

Dari beberapa pengertian diatas penulis menyimpulkan bahwa basis data atau *data base* adalah kumpulan data yang berhubungan dan disimpan secara sistematis yang diorganisasi, diolah, atau dimanipulasi menggunakan sebuah program komputer untuk membuat dan menghasilkan informasi yang dibutuhkan.

### 2.1.4 Pengertian Internet

Menurut Winarto (2017:125), “*Internet* merupakan jaringan komputer yang dapat memungkinkan manusia saling berinteraksi secara maya atau virtual.”

Menurut Sofana (2015:5), “*Internet* adalah interkoneksi jaringan komputer skala besar (miri WAN), yang dihubungkan menggunakan protocol khusus.”



Dari beberapa pengertian diatas penulis menyimpulkan bahwa internet atau *interconnected network* merupakan suatu jaringan pada komputer yang saling terhubung satu sama lain menggunakan *protocol* khusus dan memungkinkan manusia dapat saling berinteraksi secara *virtual*.

### **2.1.5 Pengertian Website**

Menurut Sa'ad (2020:4), "*Website* adalah sebuah cara untuk menampilkan diri anda di internet. Dapat diibaratkan *website* adalah sebuah tempat di *internet*, siapa saja di dunia ini dapat mengunjunginya, kapan saja mereka dapat mengetahui tentang anda, memberi pertanyaan kepada, memberi anda masukan atau bahkan mengetahui dan membeli produk anda."

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

Dari beberapa pengertian diatas penulis menyimpulkan bahwa *website* adalah kumpulan halaman yang memberikan informasi digital yang dapat berupa teks, gambar, animasi, suara atau *video* yang disediakan secara perorangan, kelompok, atau organisasi yang dapat diakses melalui internet.

## **2.2 Teori Judul**

### **2.2.1 Pengertian Aplikasi**

Menurut Marjito dan Tesaria (2016:42), "Secara umum pengertian aplikasi adalah suatu program yang siap untuk digunakan yang dibuat untuk melaksanakan suatu fungsi bagi pengguna jasa aplikasi serta penggunaan aplikasi lain yang dapat digunakan oleh suatu sasaran yang akan dituju."

Berdasarkan pengertian diatas penulis menyimpulkan bahwa, aplikasi adalah suatu program yang dipakai untuk menjalankan fungsi dan perintah dari pengguna aplikasi yang bertujuan untuk mengerjakan tugas tertentu dengan tujuan mendapatkan hasil yang lebih akurat.



### 2.2.2 Pengertian *Private Tutor*

Menurut Hasan (dalam Putra, dkk., 2017:22), “Les adalah pelajaran tambahan di luar jam sekolah, sedangkan privat berarti pribadi, tersendiri. Berdasarkan definisi tersebut maka pengajar les privat seseorang yang mengajar atau memberi bimbingan pelajaran tambahan pada mata pelajaran tertentu di luar jam belajar sekolah yang diadakan secara pribadi di rumah bagi siswa TK, SD, SLTP, maupun SMA. Jasa yang diberikan oleh guru les berupa waktu dan intelektual. Guru les memberikan bimbingan, pemahaman, pengertian, penjelasan, dan aspek – aspek kognitif yang lain kepada siswa les.”

Berdasarkan Kamus Besar Bahasa Indonesia, guru adalah orang yang pekerjaannya (mata pencariannya, profesinya) mengajar. (<https://kbbi.web.id/guru>. Diakses pada tanggal 9 Juni 2020 pukul 09.32 WIB)

Dari pengertian diatas penulis menyimpulkan bahwa *private tutor* atau guru *private* merupakan guru yang memberi bimbingan pelajaran tambahan di luar jam sekolah secara *private* atau pribadi.

### 2.2.3 Pengertian *Profile Matching*

Menurut Yulyantari dan ADH (2019:156), “*Profile Matching* merupakan proses penting dalam manajemen SDM, di mana kompetensi (kemampuan) yang diperlukan oleh suatu jabatan ditentukan terlebih dahulu.

Menurut Handojo (dalam Nurdianto, dkk., 2016:2), “Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara nilai data aktual dari suatu profile yang akan dinilai dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*), semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar.”

Menurut Kusriani (dalam Chairi, dkk., 2018:2062), “Metode *Profile Matching* atau pencocokan profil adalah sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati.”



Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa *profile matching* adalah suatu metode sistem pendukung keputusan yang memiliki proses dalam membandingkan antara nilai data aktual dari suatu profile yang akan dinilai dengan nilai profile yang diharapkan sehingga diketahui nilai *gap*nya.

#### 2.2.4 Metode *Profile Matching*

*Profile Matching* merupakan proses membandingkan antara nilai data aktual dari suatu profile yang akan dinilai dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*), semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar (Handoyo, 2011).

Langkah pertama dalam metode *profile matching* adalah menghitung nilai *gap* untuk masing-masing kriteria. Proses penghitungan nilai *gap* ini dilakukan dengan menentukan selisih antara profil *private tutor* dengan profil calon siswa.

$$\text{Gap} = \text{Value Attribut} - \text{Value Target}$$

#### 2.2.5 Cara Menghitung Metode *Profile Matching*

Berikutnya adalah beberapa tahapan dan perumusan perhitungan dengan metode *profile matching*:

##### 1. Pembobotan

Pada tahap ini, akan dilakukan pembobotan nilai masing-masing aspek dengan menggunakan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Inputan dari proses pembobotan ini adalah selisih dari profil *private tutor* dan calon siswa dan akan diberikan bobot nilai sesuai dengan tabel berikut :

**Tabel 2.1** Bobot Nilai Gap

No	Selisih Gap	Bobot Nilai	Keterangan
1	0	5	Kompetensi sesuai dengan yang dibutuhkan
2	1	4.5	Kompetensi individu kelebihan 1 tingkat/level
3	-1	4	Kompetensi individu kurang 1 tingkat/level
4	2	3.5	Kompetensi individu kelebihan 2 tingkat/level
5	-2	3	Kompetensi individu kurang 2 tingkat/level
6	3	2.5	Kompetensi individu kelebihan 3 tingkat/level
7	-3	2	Kompetensi individu kurang 3 tingkat/level
8	4	1.5	Kompetensi individu kelebihan 4 tingkat/level
9	-4	1	Kompetensi individu kurang 4 tingkat/level

## 2. Pengelompokan *Core* dan *Secondary Factor*

Setelah menentukan bobot nilai *gap* kriteria yang dibutuhkan, kemudian tiap kriteria juga dikelompokkan menjadi dua kelompok yaitu *core factor* dan *secondary factor*.



a. *Core Factor* (Faktor Utama)

*Core factor* merupakan aspek yang menonjol atau dibutuhkan yang diperkirakan dapat menghasilkan kinerja optimal. Untuk menghitung *core factor* digunakan rumus:

$$NCF = \frac{\sum NC}{\sum IC}$$

Keterangan :

NCF = Nilai rata-rata *core factor*

NC = Jumlah total nilai *core factor*

IC = Jumlah item *core factor*

b. *Secondary factor* (Faktor Pendukung)

*Secondary factor* adalah aspek yang lain yang tidak ada pada *core factor*. Untuk menghitung *secondary factor* digunakan rumus :

$$NSF = \frac{\sum NS}{\sum IS}$$

Keterangan :

NSF = Nilai rata-rata *secondary factor*

NS = Jumlah total nilai *secondary factor*

IS = Jumlah item *secondary factor*

Dari perhitungan *core factor* dan *secondary factor* dari masing-masing aspek, kemudian dihitung nilai total dan masing-masing aspek yang diperkirakan berpengaruh pada kinerja masing-masing *profile*. Untuk menghitung nilai total dari masing-masing aspek digunakan rumus :

$$Ni = (X) \% NCF + (X) \% NSF$$



Keterangan :

Ni= Nilai Total Tiap Aspek

NCF = Nilai *Core Factor*

NSF = Nilai *Secondary Factor*

(X)% = Nilai Persentase

3. Langkah terakhir dari *profile matching* adalah penentuan nilai akhir atau *ranking* dari setiap *private tutor*. Rumus perhitungannya adalah sebagai berikut :

$$\text{Ranking} = \sum (x)\%Ni$$

Keterangan :

Ni = Nilai setiap aspek penilaian

(x)% = Nilai persen *ranking*

### 2.2.6 Pengertian Aplikasi Rekomendasi *Private Tutor* Menggunakan Metode *Profile Matching* pada Bimbingan Belajar Rumah Ilmu Palembang Berbasis Web

Pengertian Aplikasi Rekomendasi *Private Tutor* Menggunakan Metode *Profile Matching* pada Bimbingan Belajar Rumah Ilmu Palembang Berbasis Web adalah sebuah *website* yang dapat memudahkan siswa dalam mencari rekomendasi guru *private* atau *private tutor* secara *online*.

## 2.3 Teori Khusus

### 2.3.1 Model *Rapid Application Development* (RAD)

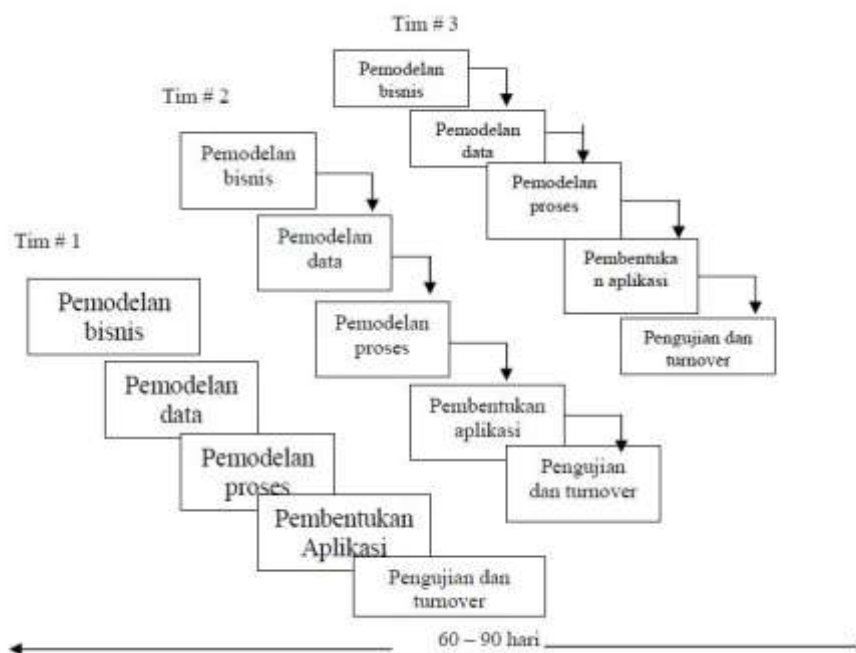
Menurut Sukamto dan Shalahuddin (2016:34), “*Rapid Application Development* (RAD) adalah model proses pengembangan perangkat lunak yang bersifat inkremental terutama untuk waktu pengerjaan yang pendek. Model RAD adalah adaptasi dari model air terjun untuk pengembangan setiap komponen perangkat lunak.”





Sedangkan menurut Sagala (2018:88), “*Rapid Application Development* (RAD) atau *rapid prototyping* adalah model proses pembangunan perangkat lunak yang tergolong dalam teknik inkremental (bertingkat). *Rapid Application Development* (RAD) menekankan pada siklus pembangunan pendek, singkat, dan cepat.”

Berikut adalah gambar model RAD:



**Gambar 2.1 Model RAD**

### 2.3.2 Pengertian UML (*Unified Modelling Language*)

Menurut Sukanto & Shalahuddin (2016:133) “*UML (Unified Modelling Language)* adalah salah satu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek”.

Pada perkembangan Teknik pemrograman berorientasi objek, munculah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language (UML)*.



Menurut Mujilan (2017:47), “UML merupakan diagram sistem menggunakan orientasi objek (*object oriented*) dalam analisis dan perancangan sistem. UML saat ini menjadi standar berbagai tipe solusi piranti lunak dalam pengembangan sistem.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa UML atau *Unified Modelling Language* adalah standar bahasa pemodelan yang digunakan untuk pembangunan perangkat lunak dalam pemrograman berorientasi objek atau OOP.

### 2.3.3 Pengertian Use Case Diagram

Menurut Munawar (2018:89), “*Use case* adalah deskripsi fungsi dari sebuah *system* dari perspektif pengguna. *Use case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai.”

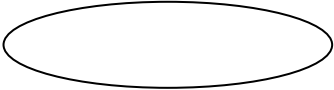
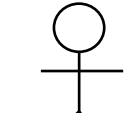

Menurut Sukanto dan Shalahuddin (2016:155), “*Use case* atau diagram *use case* merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.”

Menurut Mujilan (2017:75), “*Use Case Diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.”

Dari pengertian-pengertian diatas penulis menyimpulkan bahwa *use case diagram* adalah model yang bekerja dengan cara mendeskripsikan interaksi pengguna (*user*) dalam sebuah sistem dengan actor yang terdiri dari satu atau lebih.

Menurut Sukanto & Shalahuddin (2016:156-158) simbol-simbol yang ada pada diagram *use case* yaitu:

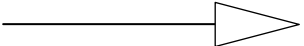
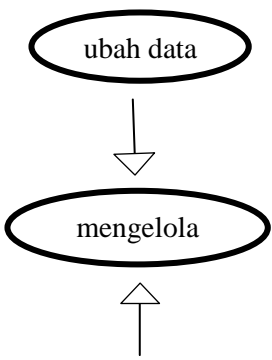
**Tabel 2.2** Simbol-Simbol *Use Case* Diagram

No.	Simbol	Deskripsi
1.	<i>Uses case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
2.	Actor / <i>actor</i>  nama aktor	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 actor adalah gambar orang, tapi actor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.
3.	Asosiasi / <i>association</i> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case Diagram*


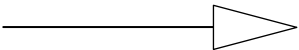
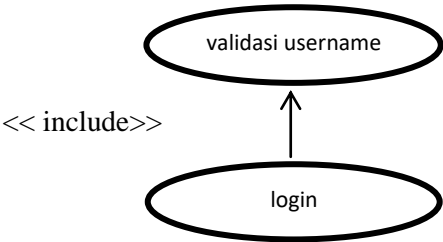
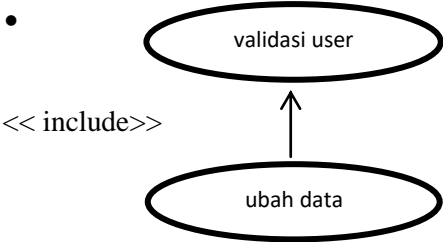
No.	Simbol	Deskripsi
4.	<p data-bbox="395 416 612 450">Ekstensi / <i>extend</i></p> <p data-bbox="443 584 612 618">&lt;&lt; extend &gt;&gt;</p> <p data-bbox="416 640 663 674">-----&gt;</p>	<p data-bbox="812 416 1359 887">Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal</p> <div data-bbox="812 920 1243 1357"> <pre> graph TD     A([validasi admin]) -.-&gt; &lt;&lt; extend &gt;&gt;  B([validasi username])     C([validasi sidik jari]) -.-&gt; &lt;&lt; extend &gt;&gt;  A           </pre> </div> <p data-bbox="812 1357 1359 1603">arah panah mengarah pada <i>use case</i> yang ditambahkan; biasanya <i>use case</i> yang menjadi <i>extend</i>-nya merupakan jenis yang sama dengan <i>use case</i> yang menjadi induknya.</p>

Lanjutan Tabel 2.2 Simbol-Simbol *Use Case* Diagram

No.	Simbol	Deskripsi
5.	<p>Generalisasi / generalization</p> 	<p>Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya, misalnya:</p>  <p>Arah panah mengarah pada <i>use case</i> yang menjadi generalisasinya (umum)</p>



Lanjutan Tabel 2.2 Simbol-Simbol Use Case Diagram

No.	Simbol	Deskripsi
6.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> <p><code>&lt;&lt; include &gt;&gt;</code></p>  <p><code>&lt;&lt; uses &gt;&gt;</code></p> 	<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. Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i>:</p> <ul style="list-style-type: none"> <li>• Include berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul>  <ul style="list-style-type: none"> <li>• Include berarti <i>use case</i> yang tambahan akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal pada kasus berikut:</li> </ul>  <p>Kedua interpretasi diatas dapat dianut salah satu atau keduanya tergantung ada pertimbangan dan interpretasi yang dibutuhkan.</p>

Sumber: Sukanto & Salahudin, 2016.



### 2.3.4 Pengertian Activity Diagram




Sukanto dan Shalahuddin (2016:161) mengemukakan, “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 diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.”

Menurut Ericson (dalam Mujilan, 2017:81), “*Activity diagram* adalah suatu ilustrasi sederhana akan apa yang terjadi dalam aliran kerja, aktivitas apa yang dapat dilakukan secara paralel, dan apakah terdapat jalur alternatif dalam aliran kerja.”



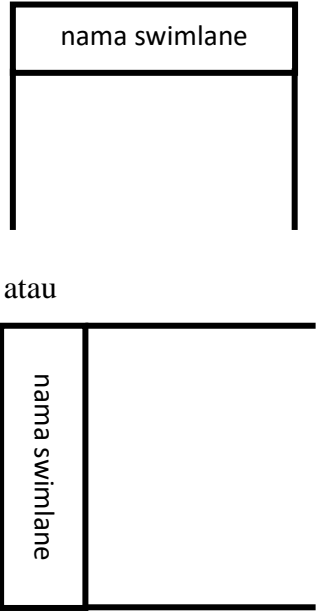
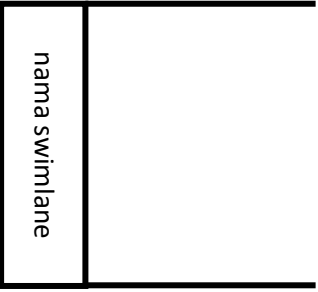
Dari pengertian diatas penulis menyimpulkan bahwa *activity diagram* adalah sebuah diagram yang menggambarkan aliran kerja atau aktivitas didalam sebuah sistem.

Menurut Sukanto & Shalahuddin (2016:162—163) simbol-simbol yang ada pada diagram aktivitas:

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

No.	Simbol	Deskripsi
1.	Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.	Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.	Pecabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.

Lanjutan Tabel 2.3 Simbol-simbol *Activity Diagram*

No.	Simbol	Deskripsi
4.	Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5.	Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6.	Swimlane  atau 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Sumber: Sukanto & Salahudin, 2016.





### 2.3.5 Pengertian *Class Diagram*

Menurut Sukamto dan Shalahuddin (2016:141), “Diagram kelas atau *class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.”



Menurut Munawar (2018:101), “*Class diagram* adalah diagram statis. Ini memiliki pandangan statis dari suatu aplikasi. *Class diagram* tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi perangkat lunak.”

Menurut Kuske (dalam Mujilan, 2017:87), “Suatu diagram klas (*class diagram*) adalah suatu gambar dengan simpul-simpul (*nodes*) yang menunjukkan klas-klas dan gambaran asosiasi (*asociations*), perumuman (*generalizations*), atau ketergantungan (*dependencies*).”






Dari pengertian-pengertian diatas penulis menyimpulkan bahwa *class diagram* adalah sebuah diagram yang menggambarkan struktur dalam sistem dan membangun kode eksekusi dalam pembangunan sebuah sistem.

Menurut Sukamto & Shalahuddin (2016:146-147), simbol-simbol yang ada pada diagram kelas :

**Tabel 2.4** Simbol-Simbol *Class Diagram*

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

Lanjutan **Tabel 2.4** Simbol-Simbol *Class Diagram*

No.	Simbol	Deskripsi
3.	Asosiasi / association 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.	Asosiasi berarah / <i>directed association</i> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus).
6.	Kebergantungan / <i>dependency</i> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
7.	Agregasi / <i>aggregation</i> 	Relasi antar kelas dengan makna semua-bagian (whole-part).

Sumber: Sukanto & Salahudin, 2016.

### 2.3.6 Pengertian *Sequence Diagram*

Menurut Sukanto dan Shalahuddin (2016:165), “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 menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*.”

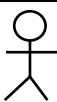
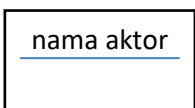

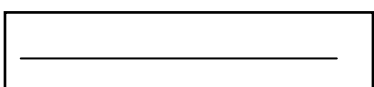


Menurut Mujilan (2017:52), “*Sequence diagram*, secara khusus, menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek di dalam *use case*.”

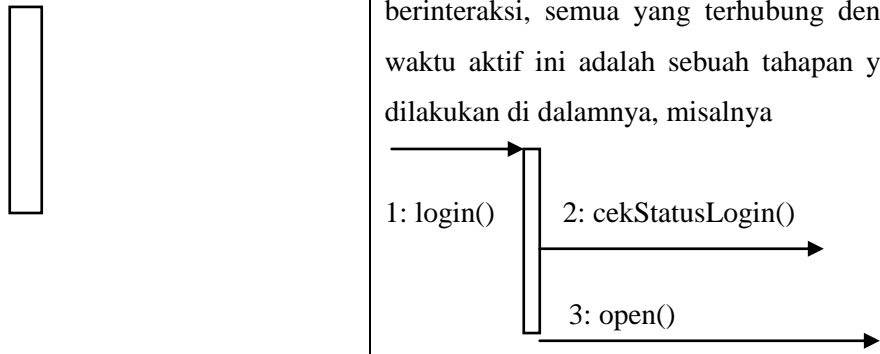

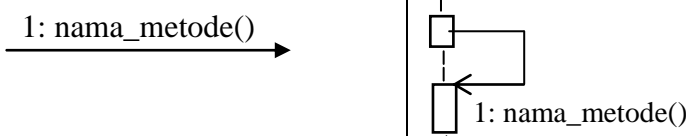
Dari pengertian diatas penulis menyimpulkan bahwa *sequence diagram* merupakan urutan diagram yang menggambarkan aksi dari suatu objek dalam *use case* dengan mendeskripsikan waktu sehingga objek-objek yang berkaitan dengan proses berjalannya suatu operasi diurutkan dari kiri ke kanan.

Menurut Sukanto & Shalahuddin (2016:165-167) simbol-simbol yang ada pada *sequence diagram* yaitu:


**Tabel 2.5** Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
1.	<p>Aktor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang kan 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.
2.	<p>Garis hidup / lifeline</p> 	Menyatakan kehidupan suatu objek.
3.	<p>Objek</p> 	Menyatakan objek yang berinteraksi pesan.

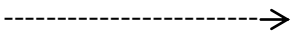
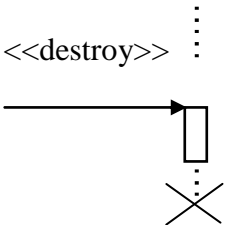
Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
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 cekStatusLogin() dan open() dilakukan di dalam metode login(). Aktor tidak memiliki waktu aktif.</p>
5.	<p>Pesan tipe create</p> <p>&lt;&lt;create&gt;&gt;</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.</p>
6.	<p>Pesan tipe call</p> <p>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.
----	--	--

Lanjutan Tabel 2.5 Simbol-Simbol *Sequence Diagram*

No.	Simbol	Deskripsi
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>> : 	Meyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy.

Sumber: Sukanto & Salahudin, 2016.

## 2.4 Teori Program

### 2.4.1 Pengertian MySQL

Menurut Yudho dan Helmi (2019:16), “MySQL yang merupakan singkatan dari “*My Structured Query Language*” adalah database yang paling *favorite* saat ini. Program ini berjalan sebagai server yang menyediakan *multi-user*, mengakses ke sejumlah database baik *multithread* maupun *multi-user*, dan telah diinstal oleh sekitar 6 juta kali di seluruh dunia.”

Menurut Mundzir (2018:217), “MySQL adalah sistem manajemen *database* yang sifatnya *open source* (terbuka) dan paling banyak digunakan saat ini. Sistem *database MySQL* mampu mendukung beberapa fitur seperti *multithreaded*, *multiuser*, dan *SQL database management system (DBMS)*.”



Menurut Rusmawan (2019:97), “MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di dunia.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa MySQL adalah sebuah sistem manajemen basis data yang berifat *open source*. MySQL berjalan sebagai server yang menyediakan *multi-user*, mengakses ke sejumlah database baik *multithread* maupun *multi-user*.

#### 2.4.2 Pengertian PHP

Menurut Subagia (2018:1), “PHP merupakan sebuah bahasa pemrograman yang berjalan dalam sebuah *web server (server side)*. PHP diciptakan oleh *programmer Unix* dan *Perl*, bernama Rasmus Lerdoft pada bulan Agustus-September 1994.”

Menurut Mundzir (2018:3), “PHP berasal dari kata “*Hypertext Preprocessor*”, yaitu bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML.”

Menurut Pratama (2019:3), “Dalam pengertian paling sederhana, PHP adalah bahasa pemrograman web yang digunakan untuk men-generate atau menghasilkan kode HTML.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa PHP merupakan sebuah bahasa pemrograman *script server-side* yang digunakan sebagai pembangun dan pengembang situs *web*.

#### 2.4.3 Pengertian Sublime Text

Menurut Mulyani (2020:106), “*Sublime Text* adalah *text editor* yang sangat populer karena fiturnya sangat lengkap. *Text editor* ini merupakan salah satu *text editor* yang canggih dan *powerful*.”

Menurut Rerung (2018:25), “*Sublime Text* adalah aplikasi *editor* untuk kode dan teks yang dapat berjalan di berbagai *platform operating system* dengan



menggunakan teknologi API. Terciptanya aplikasi ini terinspirasi dari aplikasi Vim. Aplikasi ini sangatlah fleksibel dan *powerful*.”

Berdasarkan pengertian diatas penulis menyimpulkan bahwa *Sublime Text* adalah aplikasi yang digunakan sebagai *editor* kode dan teks yang memiliki kekuatan teknologi API dan dapat berjalan di berbagai *platform*.

#### 2.4.4 Pengertian *Bootstrap*

Menurut Abdulloh (2018:261), “*Bootstrap* merupakan salah satu *framework* CSS paling populer dari sekian banyak *framework* CSS yang ada. *Bootstrap* memungkinkan desain sebuah *web* menjadi responsif sehingga dapat dilihat dari berbagai macam ukuran *device* dengan tampilan tetap menarik.”

Menurut Rozi (dalam Christian, dkk., 2018:22), “*Bootstrap* adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah *website*. Bisa dikatakan, *bootstrap* adalah template desain web dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan *bootstrap*.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa *Bootstrap* adalah sebuah sistem *framework* CSS yang siap pakai untuk membuat *front-end website* dan menjadikan *web* lebih responsif.

#### 2.4.5 Pengertian *PhpMyAdmin*

Menurut Madcoms (2016:186), “*PhpMyAdmin* adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan *PhpMyAdmin*, anda dapat membuat *database*, membuat *table*, meng-*insert*, menghapus dan meng-*update* data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual.”



Hikmah *et al* (2015:2), “*PHPMYAdmin* merupakan aplikasi yang dapat digunakan untuk membuat *database*, pengguna (*user*), memodifikasi *table*, maupun mengirim *database* secara cepat dan mudah tanpa harus menggunakan perintah (*command*) SQL.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa *PHPMYAdmin* adalah aplikasi PHP yang sering disebut sebagai *tool* pengelola atau pengatur data pada MySQL.

#### 2.4.6 Pengertian XAMPP

Menurut Yudho dan Helmi (2019:17), “*XAMPP* adalah kompilasi program aplikasi gratis terfavorit di kalangan *developer/programmer* yang berguna untuk pengembangan *website* berbasis *PHP* dan *MySQL*.”

Menurut Madcoms (2016:186), “*XAMPP* adalah sebuah paket kumpulan *software* yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla*, dan lain-lain.”

Berdasarkan pengertian-pengertian diatas penulis menyimpulkan bahwa *XAMPP* adalah *server* atau perangkat yang menggabungkan tiga aplikasi kedalam satu paket yaitu *Apache*, *PHP*, dan *MySQL* yang berperan dalam menginstalasi dan mengkonfigurasi ketiga aplikasi tersebut dan digunakan dalam pengembangan *website*.

#### 2.5 Referensi Penelitian Terdahulu

Penelitian ini menggunakan 4 (empat) jurnal dari peneliti terdahulu yang dilakukan oleh beberapa orang dengan menggunakan metode *Profile Matching*, yaitu :

Menurut penelitian Chairi *et al* pada tahun 2018 dengan judul Rekomendasi Tempat Wisata Kota Malang Menggunakan Metode *Profile Matching* Dan Saran Rute Menggunakan *Floyd Warshall* Berbasis Android, e-ISSN : 2548-964X, yang menjelaskan bahwa *Profile Matching* merupakan sebuah mekanisme pengambilan keputusan dengan mengasumsikan bahwa





terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Selain itu untuk membantu dalam penentuan jalur akan digunakan metode *floyd-warshall*. Sehingga proses penentuan jalur dari maps akan dibantu oleh *floyd-warshall*. Untuk *Core Factor* merupakan kriteria (kompetensi) yang paling penting atau menonjol atau paling dibutuhkan oleh suatu penilaian yang diharapkan dapat memperoleh hasil yang optimal sedangkan untuk *secondary factor* merupakan faktor pendukung yang kurang dibutuhkan oleh suatu penilaian. Untuk perenkingan menggunakan 60 % untuk *core factor* dan 40 % untuk *secondary factor* menghasilkan tingkat akurasi sebesar 75%.

Menurut penelitian Agustin dan Fransiskus pada tahun 2018 dengan judul Penerapan Metode Profile Matching pada Penilaian Kinerja Penelitian dan Pengabdian kepada Masyarakat, ISSN : 2338 - 6649 menjelaskan bahwa tahapan dalam perhitungan menggunakan metode *Profile Matching* yaitu menentukan aspek-aspek yang digunakan untuk penilaian, pemetaan gap profil dengan mencari selish antara profil minimal atau profil ideal dengan profil data tes, dan setelah diperoleh nilai *gap* selanjutnya diberikan bobot untuk masing-masing nilai *gap*.

Menurut jurnal Fauzi pada tahun 2019 dengan judul Implmentasi Metode *Profile Matching* Pada Sistem Pendukung Keputusan Dalam Pemberian Jabatan Kelas Pada Mahasiswa, ISSN : 2548 - 9712 menjelaskan bahwa *profile matching* atau pencocokan profil adalah sebuah mekanisme pengambilan keputusan dengan mengansumsikan terdapat tingkat variabel prediktor yang ideal yang harus dimiliki oleh pelamar, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Dalam proses profile matching, akan dilakukan proses perbandingan antara kompetensi individu ke dalam kompetensi standar, dalam hal ini profil asisten praktikum yang ideal sehingga dapat diketahui perbedaan kompetensinya (disebut juga *gap*). Semakin kecil *gap* yang dihasilkan maka bobot nilainya semakin besar. Calon



yang memiliki bobot nilai yang besar berarti memiliki peluang lebih besar untuk dapat menempati posisi sebagai asisten praktikum. Dan berikut merupakan langkah-langkah perhitungan dalam *profile matching*.

Menurut penelitian Ernawati *et al* pada tahun 2017 dengan judul Rancang Bangun Sistem Pendukung Keputusan Kenaikan Jabatan Pegawai Dengan Metode Profile Matching (Studi Kasus: Kementerian Agama Kantor Wilayah DKI Jakarta), ISSN : 1979-0767 menjelaskan bahwa Dalam jurnal berjudul Perbandingan Metode Profile Matching dan Simple Additive Weighting pada Penentuan Jurusan SiswaKelas X SMAN 2 Ngaglik, diketahui bahwa kedua metode tersebut menunjukkan hasil akhir yang sama dan diperoleh berdasarkan kriteria dan data yang sama pula. Namun, metode SAW adalah metode penjumlahan antar bobot dengan proses menentukan rating kriteria. Dalam arti setiap kriteria memiliki bobot persen yang harus ditentukan. Sedangkan dalam studi kasus kenaikan jabatan di Kementerian Agama Kanwil DKI Jakarta, diketahui bahwa aspek kompetensi memiliki nilai persentase yang sama. Jadi metode ini tidak sesuai dengan studi kasus kenaikan jabatan di Kementerian Agama Kanwil DKI Jakarta