

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian pertama yakni menerapkan Sistem Pendukung Keputusan Penerimaan Karyawan pada PT. Pioneerindo Gourmet International Tbk. Cabang Rantauprapat. Sistem tersebut menggunakan metode SAW. Penelitian tersebut dilakukan pada bulan Februari tahun 2022. Setiap perusahaan biasanya melakukan perekrutan karyawan baru guna membantu perusahaan tersebut dalam menyelesaikan suatu proyek. Tujuan penelitian ini adalah membuat sistem yang terotomatisasi yang digunakan untuk membantu menentukan apakah calon karyawan diterima atau tidak di PT tersebut sehingga nantinya diperoleh karyawan yang berkualitas. Metode yang digunakan dalam penelitian ini adalah metode SAW. Metode tersebut dipilih karena dapat menentukan nilai bobot setiap atribut, kemudian dilanjutkan dengan proses perengkingan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif dan penilaian akan lebih tepat karena didasarkan pada nilai kriteria dan bobot prefrensi yang sudah ditentukan. Terdapat 5 kriteria yang digunakan dalam sistem ini. Sistem ini dibuat menggunakan bahasa pemrograman PHP dan MYSQL sebagai *databasenya*. Sistem ini berbasis *website*. Hasil dari penelitian tersebut adalah sistem memberikan hasil yang akurat dan valid sesuai dengan data-data yang dimiliki oleh calon karyawan sehingga dapat menentukan calon karyawan yang layak diterima atau tidak[3].

Penelitian selanjutnya yaitu menerapkan Sistem Pendukung Keputusan Penerimaan Pelatih Sepak Bola (Studi Kasus : PSDS Deli Serdang).. Sistem tersebut menggunakan Metode *Profile Matching*. Penelitian tersebut dilakukan pada bulan Desember 2021. Pelatih sepak bola adalah orang yang berada penuh melatih, membimbing, dan memberikan dukungan moral kepada pemain dalam melakukan latihan maupun pertandingan. Tujuan penelitian ini adalah membuat sebuah sistem yang dapat membantu PSDS Deli Serdang dalam menentukan calon pelatih terbaik. Metode yang digunakan dalam penelitian ini adalah metode *Profile Matching*. Dengan menggunakan metode *profile matching* mampu menyeleksi berupa pelatih baru melalui kriteria-kriteria yang sudah ditentukan, dengan menyeleksi alternatif terbaik dari

sejumlah alternatif kemudian dilakukan proses perankingan dalam mendapatkan alternative yang terbaik yaitu pelatih baru. Terdapat 16 kriteria yang digunakan dalam sistem ini.. Sistem ini menggunakan *Visual Basic Net.2008* sebagai bahasa pemrogramannya. Hasil dari penelitian ini adalah sistem dapat megolah data masukan berupa data calon pelatih dan data bobot variabel menjadi sebuah proses penilaian calon pelatih yang akan dipilih sehingga proses seleksi menjadi lebih cepat dan akurat[4].

Penelitian selanjutnya yaitu menerapkan Sistem Pendukung Keputusan Pemilihan Karyawan Terbaik di PT. Karya Anugrah Teknologi. Sistem tersebut menggunakan metode *Profile Matching*. Penelitian ini dilakukan pada bulan Februari 2022. Setiap tahun, PT. Karya Anugrah Teknologi mengadakan evaluasi kinerja karyawan dan pemilihan karyawan terbaik yang akan diberikan reward dan bonus. Evaluasi kinerja diperlukan oleh perusahaan untuk menjaga dan meningkatkan kualitas karyawan. Tujuan dari penelitian ini adalah membuat sistem penilaian kinerja karyawan dalam rangka pemberian reward dan bonus kenaikan gaji dengan menganalisa data-data pegawai sesuai dengan kriteria yang telah ditentukan perusahaan. Terdapat 14 kriteria yang digunakan dalam sistem ini. Metode yang digunakan dalam penelitian ini adalah metode *Profile Matching*. Tahapan proses yang dilakukan yakni Pemetaan GAP, Pembobotan, Perhitungan dan Pengelompokan *Coredan Secondary Factor*, Perhitungan Nilai total, dan Perankingan. Hasil yang diperoleh dari penelitian ini adalah sistem dapat menentukan karyawan terbaik dengan skor tertinggi yang nantinya karyawan tersebut memperoleh *reward* dan kenaikan gaji sehingga dapat meningkatkan produktifias dan kualitas dalam bekerja[5].

Penelitian selanjutnya yaitu menerapkan Sistem Pendukung Keputusan dalam menentukan kenaikan jabatan pegawai di Dinas Pengelolaan Sumber Daya Air Provinsi Lampung. Sistem tersebut menggunakan metode *Profile Matching*. Penelitian ini dilakukan pada tahun 2022. Dinas Pengelolaan Sumber Daya Air (DPSDA) Provinsi Lampung merupakan instansi vertikal dibawah naungan Direktur Jenderal Sumber Daya Air (DJSDA). Setiap instansi tentunya membutuhkan pegawai yang berkualitas dan memperoleh pegawai yang diharapkan oleh instansi. Dinas Pengelolaan Sumber Daya Air Provinsi Lampung melakukan penilaian dan kenaikan jabatan terhadap pegawainya. Dengan adanya penilaian, pegawai dapat dinilai melalui keahlian yang dimilikinya dan dapat memberi keahlian maksimal

terhadap pencapaian tujuan sehingga memberikan jaminan terhadap kestabilan, kelancaran, dan efektivitas kerja. Dinas Pengelolaan Sumber Daya Air terdapat 4 bidang yaitu Bidang Pembangunan dan Rehabilitas Sumber Daya Air, Bidang Operasi dan Pemeliharaan Sumber Daya Air, Bidang Penatagunaan Sumber Daya Air, serta Bidang Perencanaan Sumber Daya Air. Jumlah keseluruhan pegawai pada dinas tersebut yakni berjumlah 80 pegawai, dengan jumlah masing-masing bidang 20 pegawai. Dinas Pengelolaan Sumber Daya Air melakukan penilaian dan melakukan kenaikan jabatan terhadap pegawai-pegawainya. Hasil dari penelitian ini, Dinas Pengelolaan Sumber Daya Air dapat dengan mudah dalam menentukan penilaian pegawai dan meningkatkan akurasi pilihannya secara tepat dan otomatis[6].

Penelitian selanjutnya yaitu menerapkan Sistem Pendukung Keputusan untuk menentukan teknologi energi dengan emisi paling rendah di Polandia. Penelitian ini dilakukan pada tahun 2018. Penelitian ini bertujuan menentukan lima energy dengan emisi paling rendah untuk mengurangi polusi yang ada di Polandia. Metode yang dipakai dalam penelitian ini yaitu menggunakan metode Fuzzy AHP-TOPSIS. Dalam penelitian ini terdapat tiga kriteria utama yaitu ekonomi, social, dan lingkungan serta enam belas subkriteria. Penelitian dibagi menjadi dua tahap. Tahap yang pertama menggunakan metode Fuzzy-AHP untuk mencapai bobot kriteria. Tahap yang kedua yakni penentuan rank dengan metode Fuzzy-TOPSIS. Hasil dari penelitian tersebut menghasilkan lima energy dengan emisi terendah yakni biomass, biogas, *wind on shore*, *wind off shore*, dan nuklir[7].

Pada penelitian ini, peneliti akan membangun sebuah sistem berupa rotasi kerja pegawai yang akan diterapkan di Kantor Imigrasi Kelas 1 Cilacap. Metode yang dipakai di sistem ini adalah metode *Profile Matching*. Sistem ini dibuat menggunakan PHP sebagai bahasa pemrogramannya dan MYSQL sebagai *databasenya*. Terdapat perbedaan dalam sistem yang peneliti buat. Perbedaannya adalah sistem ini melakukan proses perhitungan beberapa kali untuk menempatkan pegawai ke dalam bidang yang sesuai. Model pengembangan perangkat lunak yang digunakan di sistem ini menggunakan model *Waterfall*. Model tersebut dipilih karena proses pengembangannya dilakukan secara *step by step* sehingga dapat meminimalisir kesalahan yang mungkin terjadi.

2.2 Landasan Teori

2.2.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan merupakan sebuah sistem yang telah dirancang dan dapat diimplementasikan untuk mendukung keputusan yang sudah disepakati dalam pemilihan suatu objek. Ada beberapa definisi dan fungsi-fungsi utama dalam penerapan pendukung keputusan. Pengguna dapat mendefinisikan pemahaman yang berbeda-beda berdasarkan kebutuhan dan diimplementasikan dengan menerapkan teknologi. Perspektif yang dianjurkan dan diuraikan untuk lebih lanjut terpusat pada pengguna dan masalah yang terjadi. Pada umumnya kontribusi yang diperlukan pada perangkat lunak berbeda dalam pengembangan SPK untuk aplikasi yang berkelanjutan. Pengambilan keputusan pada dasarnya harus sesuai dengan kondisi pada permasalahan yang sudah terjadi. Penentuan sebuah keputusan terdiri dari tahapan dan melalui suatu proses, adanya penentuan sebuah keputusan terdiri dari empat tahapan yang saling terhubung dan berurut[4].

Berdasarkan pengertian di atas diperoleh informasi bahwa Sistem Pendukung Keputusan bukan merupakan alat yang dijadikan dalam pengambilan keputusan, melainkan sebuah sistem yang digunakan untuk membantu seseorang dalam proses pengambilan keputusan.

2.2.2 Metode *Profile Matching*

Profile Matching merupakan mekanisme pengambil keputusan terutama dalam manajemen sumber daya manusia untuk menentukan suatu jabatan dengan kualifikasi yang telah ditetapkan. Proses *profile matching* diawali dengan pemilihan kriteria yang dibutuhkan dan memberikan nilai target pada masing-masing aspek. Tahap berikutnya perbandingan dilakukan antara kemampuan individu dengan kualifikasi yang telah ditetapkan sehingga didapatkan gap dimana semakin kecil nilai yang didapatkan maka bobot nilai semakin besar. *Core Factor* dan *Secondary Factor* menjadi perhitungan berikutnya dengan memberikan besaran persentase untuk kedua faktor tersebut[4]

2.2.3 Perhitungan Metode *Profile Matching*

Adapun langkah-langkah perhitungan dari metode *Profile Matching* sebagai berikut :

1. Menentukan variabel data-data yang diperlukan.
2. Menentukan aspek-aspek yang digunakan untuk penelitian

3. Pemetaan Gap Profil
Gap = profil minimal-profil data tes
4. Memberikan bobot untuk masing-masing nilai Gap.
5. Perhitungan dan pengelompokan *CoreFactor* dan *Secondary Factor*.
- a. *CoreFactor* merupakan kriteria yang paling penting atau paling dibutuhkan oleh suatu penilaian.

$$NCF = \frac{\sum NC}{\sum IC} \quad (1)$$

Keterangan :

NCF : Nilai rata-rata *CoreFactor*

NC : Jumlah total nilai *CoreFactor*

IC : Jumlah item *CoreFactor*.

- b. *Secondary Factor* merupakan item-item selain yang ada di *CoreFactor*. Dengan kata lain, merupakan *factor* pendukung yang dapat menunjang suatu penilaian.

$$NSF = \frac{\sum NS}{\sum IS} \quad (2)$$

Keterangan :

NSF : Nilai rata-rata *Secondary Factor*

NS : Jumlah total nilai *Secondary Factor*

IS : Jumlah item *Secondary Factor*.

6. Perhitungan Nilai Total
Nilai total diperoleh dari presentase *corefactor* dan *Secondary Factor*.

$$N = (x) \% NCF + (x) \% NSF \quad (3)$$

Keterangan :

N : Nilai total dari kriteria

NCF : Nilai rata-rata *CoreFactor*

NSF : Nilai rata-rata *Secondary Factor*

(x) % : Nilai persen yang diinputkan

2.2.4 Rotasi Kerja

Rotasi kerja merupakan perubahan posisi atau jenis pekerjaan atau wilayah tertentu. Banyak perusahaan yang sudah menerapkan sistem rotasi kerja bagi para pegawainya karena memiliki beberapa keuntungan. Keuntungan tersebut diantaranya memiliki pengetahuan baru, pengalaman baru, wawasan yang luas, dan melatih keterampilan pada dirinya[8]. Rotasi kerja juga dapat mengurangi rasa bosan / jenuh

bagi pegawainya. Salah satu perusahaan yang menerapkan rotasi kerja yakni Kantor Imigrasi Kelas 1 Cilacap.

2.2.5 Sistem Informasi

Sistem informasi adalah seperangkat komponen yang mengelola data sehingga data yang diolah dapat dijadikan sebagai informasi yang bermakna dan dapat membantu mencapai tujuan suatu perusahaan atau organisasi[9]. Perusahaan bisnis dan organisasi lain menggunakan sistem informasi untuk menjalankan dan mengelola operasi mereka, berinteraksi dengan pelanggan, dan masih banyak lagi.

2.2.6 Website

Website adalah kumpulan informasi yang dapat diakses kapan saja dan dimana saja oleh setiap orang selama terhubung secara *online* di jaringan internet[10]. Peneliti memilih sistem informasi berbasis *web* karena memiliki beberapa kelebihan diantaranya mudah dikembangkan oleh seorang *web developer*, mudah diakses, informasi mudah didistribusikan, dan fleksibel.

2.2.7 Codeigniter

Codeigniter merupakan *framework* PHP yang memiliki *library* yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh aplikasi berbasis *web*. *Codeigniter* juga menjadi satu-satunya *framework* dengan dokumentasi yang lengkap dan jelas. *Source code* CI yang dilengkapi dengan *comment* di dalamnya sehingga lebih memperjelas fungsi sebuah kode program. *Codeigniter* juga dapat memudahkan *developer* dalam membuat aplikasi *web* berbasis PHP karena *framework* sudah memiliki kerangka kerja sehingga tidak perlu menulis semua kode program dari awal. Selain itu, struktur dan susunan semakin teratur dan dapat fokus pada fitur-fitur apa yang akan dibutuhkan dalam pembuatan aplikasi tersebut[11].

2.2.8 PHP (Hypertext Preprocessor)

PHP merupakan bahasa pemrograman yang digunakan untuk membuat *website* yang bersifat *server-side scripting*. PHP bersifat dinamis. PHP dapat berjalan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *Mac OS*. PHP juga mendukung beberapa *web server* diantaranya *Apache*, *Microsoft ISS*, *Caudium*, dan sebagainya. Sistem manajemen *database* yang sering digunakan bersama PHP yaitu *MYSQL*. Ketika dipanggil dari *web browser*, program yang ditulis

dengan PHP akan diparsing di dalam *web server* oleh *interpreter* PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali *web server* karena pemrosesan program PHP dilakukan di dalam lingkungan *web browser*[12].

2.2.9 XAMPP

XAMPP merupakan perangkat lunak yang mendukung banyak sistem operasi dan merupakan kompilasi dari beberapa program seperti *Apache*, *MYSQL*, *PHP*, dan *Perl*. *XAMPP* adalah tool yang menyediakan paket perangkat lunak dalam satu buah paket. Paket *XAMPP* tersebut berisi *Apache (Web Server)*, *MYSQL (Database)*, *PHP (Server Side Scripting)*, *Perl*, *FTP Server*, dan berbagai pustaka bantu lainnya. Kepanjangan dari *XAMPP* sebagai berikut :

- X :** Program ini dapat berjalan di berbagai sistem operasi seperti *Windows*, *Linux*, dan *Mac Os*.
- A :** *Apache* merupakan aplikasi *web server*. Tugas dari *Apache* adalah menghasilkan halaman *web* yang benar kepada *User* berdasarkan kode PHP yang dituliskan oleh pembuat aplikasi.
- M :** *MYSQL* merupakan aplikasi data server. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah *database*.
- P :** *PHP* merupakan bahasa pemrograman berbasis *web*, dimana *User* dapat menggunakan bahasa pemrograman ini untuk membuat *web* yang bersifat server-side scripting.
- P :** *Perl* merupakan bahasa pemrograman tingkat tinggi. *Perl* bersifat dinamis yang bisa anda gunakan untuk mengembangkan berbagai macam pemrograman atau aplikasi[12].

2.2.10 Database

Database adalah sebuah sistem untuk menyimpan kumpulan data yang terorganisir. *Database* berfungsi sebagai tempat penyimpanan informasi yang dapat diakses, dikelola, dan diperbarui dengan efisiensi. Data dalam *database* biasanya diorganisasi dalam bentuk tabel atau struktur lainnya, yang memungkinkan penggunaan yang mudah untuk menyimpan, mengambil, memperbarui, dan menghapus informasi. Tujuan utamanya adalah memberikan kerangka terstruktur untuk menyimpan dan mengelola data agar bisa diakses dan dimanipulasi dengan efisien oleh pengguna atau aplikasi. Selain itu, *database* juga memungkinkan pengguna yang berbeda untuk berbagi dan mengakses data secara bersamaan, dengan pengaturan kontrol akses dan keamanan

yang sesuai dengan kebutuhan. MYSQL dipilih sebagai *database* karena memiliki beberapa keunggulan diantaranya *software open source* yang artinya dapat digunakan oleh siapa saja, *Multi User* yang artinya dapat digunakan oleh beberapa *User* secara bersamaan, dan antar muka yang fleksibel pada beberapa aplikasi dan pemrograman[13].

Sebuah *DBMS (Database Management System)* memiliki komponen diantaranya DDL dan DML yang digunakan untuk mengontrol akses database[14].

1. DDL (*Data Definition Language*)

DDL (*Data Definition Language*) merujuk pada segmen dari bahasa kueri basis data yang berfungsi untuk menggambarkan kerangka dan skema dari basis data. DDL berperan dalam pembuatan, modifikasi, dan penghapusan objek-objek dalam basis data, seperti tabel-tabel, indeks, tampilan, serta pembatasan-pembatasan. Instruksi-instruksi DDL memungkinkan pengguna untuk merinci entitas-entitas dan menggarisbawahi keterhubungan di antara mereka dalam basis data. Beberapa perintah DDL yang sering digunakan antara lain:

- a) *CREATE*: Digunakan untuk membuat objek database baru, seperti tabel, indeks, atau tampilan.
- b) *ALTER*: Digunakan untuk mengubah struktur objek database yang sudah ada, misalnya menambahkan kolom baru ke dalam tabel.
- c) *DROP*: Digunakan untuk menghapus objek database, seperti menghapus tabel atau indeks dari database.

2. DML (*Data Manipulation Language*)

DML (*Data Manipulation Language*) merujuk pada bagian dari bahasa kueri basis data yang digunakan untuk mengelola informasi di dalam tabel. DML berperan dalam memasukkan (*insert*), memodifikasi (*update*), menghapus (*delete*), serta mengambil (*select*) data dari tabel. DML mengizinkan pengguna untuk mengelola data dengan mengubah nilai-nilai di dalam tabel, mengambil subkelompok data yang diinginkan, dan menjalankan operasi-operasi lain yang berkaitan dengan data. Beberapa perintah DML yang umum digunakan meliputi:




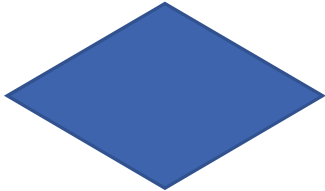
- a) *INSERT*: Digunakan untuk menyisipkan data baru ke dalam tabel.
- b) *UPDATE*: Digunakan untuk memperbarui data yang sudah ada dalam tabel.








- c) DELETE: Digunakan untuk menghapus data dari tabel.
- d) SELECT: Digunakan untuk mengambil data yang spesifik dari tabel menggunakan kriteria tertentu.



2.2.11 Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah. *Flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah. *Flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja satu tim dalam suatu proyek. *Flowchart* membantu memahami urutan-urutan yang rumit dan panjang[15].

Tabel 2. 1 *Flowchart*

No	Gambar Simbol	Nama Simbol	Keterangan
1		<i>Terminal point</i>	Menunjukkan mulai atau berakhirnya suatu proses.
2		<i>Flow direction</i>	Simbol yang digunakan untuk menghubungkan antar simbol, berfungsi juga untuk menunjukkan alur sebuah proses.
3		Proses	Menunjukkan sebuah kegiatan yang dilakukan oleh komputer atau sistem.
4		<i>Decision</i>	Simbol yang digunakan untuk memilih sebuah keputusan atau proses sesuai dengan kondisi yang ada.

5		<i>Input output</i>	Menunjukkan sebuah <i>input</i> atau <i>output</i> yang ada tidak bergantung pada jenis peralatannya.
6		<i>Predefined process</i>	Menunjukkan pelaksanaan suatu bagian prosedur, bagian prosedur yang terinformasi belum detail dan akan diperinci ditempat lain.
7		<i>Connector (On-page)</i>	Menghubungkan suatu simbol pada satu halaman yang letaknya berjauhan.
8		<i>Connector (Off-page)</i>	Menghubungkan simbol yang berada dalam halaman yang berbeda.
9		<i>Preparation</i>	Simbol ini menunjukkan persiapan penyimpanan kedalam <i>storage</i> .
10		<i>Manual input</i>	Menunjukkan proses <i>input</i> yang dilakukan secara manual menggunakan <i>online keyboard</i>
11		<i>Manual operation</i>	Menunjukkan proses/kegiatan yang dilakukan tanpa menggunakan komputer.

12		<i>Document</i>	Menunjukkan <i>input</i> berupa dokumen dalam bentuk kertas atau <i>output</i> yang harus dicetak.
13		<i>Multiple document</i>	Sama seperti simbol <i>document</i> hanya saja dokumen yang digunakan lebih dari satu.

2.2.12 UML (*Unified Modelling Language*)



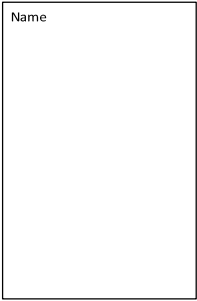

UML adalah salah satu standar bahasa yang banyak digunakan untuk mendefinisikan *requirement*, membuat analisis, dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. Tahapan awal yang dilakukan dalam pembuatan sistem informasi adalah perencanaan. Penelitian ini, penulis melakukan perencanaan menggunakan *Unified Modelling Language* (UML). UML mampu merepresentasikan rancangan sistem informasi yang akan dibuat sehingga menjadi sebuah sistem yang siap digunakan oleh *User* atau pengguna[16].

2.2.13 *Usecase Diagram*

Usecase diagram adalah satu dari berbagai jenis diagram UML yang menggambarkan hubungan interaksi antara sistem dan aktor. *Use case* dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. *Use case* menspesifikasikan perilaku sistem dan merupakan deskripsi dari sekumpulan aksi-aksi yang diinginkan oleh pengguna perangkat lunak yang akan kita kembangkan. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara *User* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai[17]

Tabel 2. 2 *Use Case Diagram*

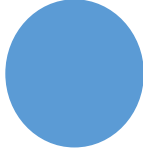

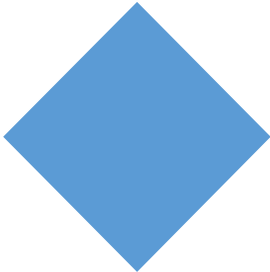

No	Simbol	Fungsi
----	--------	--------

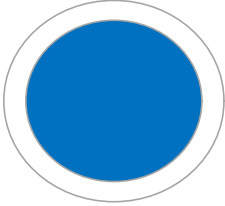
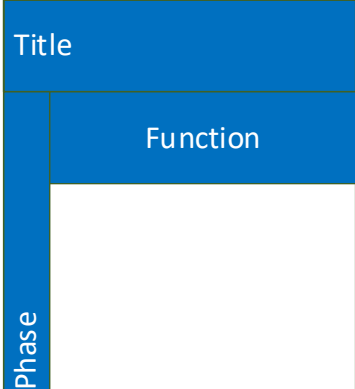
1		<p>Segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi <i>actor</i> ini bisa berupa orang, perangkat keras atau mungkin juga obyek lain dalam sistem yang sama.</p>
2	<pre> -- -- -- <<include>> -- -- --> -- -- -- <<extend>> -- -- --> </pre>	<p>Menspesifikasikan bahwa perilaku <i>use case</i> merupakan bagian dari <i>use case</i> lain.</p>
3		<p>Menggambarkan navigasi antar <i>class</i>, berupa banyak obyek lain yang berhubungan dengan satu obyek, dan apakah suatu <i>class</i> menjadi bagian dari <i>class</i> lainnya.</p>
4		<p><i>System Boundary</i> yaitu batasan sebuah sistem.</p>
5		<p><i>Use case</i> menjelaskan urutan kegiatan yang dilakukan aktor dan sistem untuk mencapai suatu tujuan tertentu. Walaupun menjelaskan kegiatan, <i>use case</i> hanya menjelaskan apa yang dilakukan oleh aktor dan sistem, bukan bagaimana aktor dan sistem melakukan kegiatan.</p>

2.2.14 Activity Diagram

Activity Diagram merupakan diagram yang dapat memodelkan proses-proses yang terjadi pada suatu sistem. Runtutan proses pada *Activity Diagram* digambarkan secara vertikal. *Activity Diagram* merupakan pengembangan dari use case yang memiliki alur aktivitas. *Activity Diagram* tidak menjelaskan kelakuan aktor, melainkan hanya dipakai untuk menggambarkan alur kerja sebuah sistem saja[18]

Tabel 2. 3 *Activity Diagram*

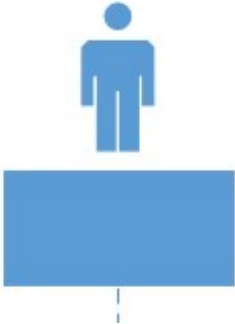
No	Simbol	Nama Simbol	Keterangan
1		<i>Initial Node</i>	Biasanya digunakan untuk menandakan status awal, tindakan awal, atau titik awal aktivitas untuk setiap <i>Activity Diagram</i> .
2		<i>Activity</i>	Aktivitas yang sedang dilakukan atau sedang terjadi dalam sistem.
3		<i>Decision</i>	Suatu titik atau point yang mengindikasikan suatu kondisi dimana adanya kemungkinan dalam perbedaan transisi.
4		<i>Join Node</i>	Menghubungkan kembali <i>Activity</i> dengan <i>action</i> secara paralel.

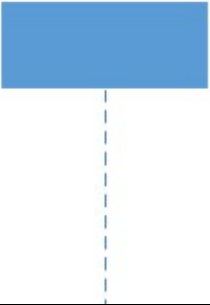


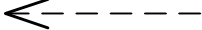
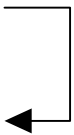
5		<i>Final Node</i>	Menandakan proses <i>Activity</i> berakhir.
		<i>Swimlane</i>	Memecah <i>Activity Diagram</i> menjadi kolom dan baris untuk membagi tanggung jawab objek-objek yang melakukan suatu aktivitas.

2.2.15 Sequence Diagram

Sequence diagram merupakan diagram yang menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Semakin banyak use case yang didefinisikan, maka *Sequence* diagram yang harus dibuat juga semakin banyak[16].

Tabel 2. 4 *Sequence Diagram*

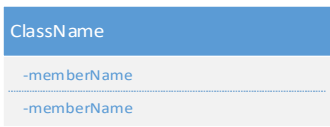

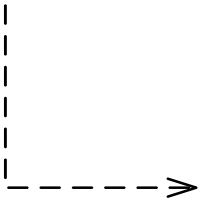

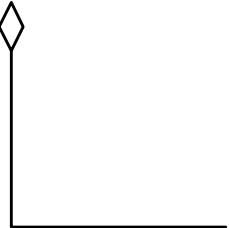
No	Simbol	Nama Simbol	Keterangan
1		<i>Actor Lifeline</i>	Aktor memperlihatkan sebuah entitas yang terdapat di luar dari sistem serta dapat berinteraksi dengan sistem. Aktor ini dapat berinteraksi baik dengan perangkat lunak maupun dengan perangkat keras.

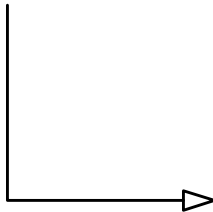
2		<i>Object Lifeline</i>	menunjukkan eksekusi sebuah objek dalam <i>Sequence</i> .
3		<i>Activation</i>	mengetahui waktu yang diperlukan untuk suatu objek menyelesaikan tugasnya.
4		<i>Message</i>	<i>message</i> memiliki fungsi untuk komunikasi antar objek yang akan dilakukan.
5		<i>Return Message</i>	balasan untuk sebuah panggilan tertentu.
6		<i>Self Message</i>	menggambarkan hubungan antar objek itu sendiri untuk diurutkan sesuai dengan aktivitas yang dihasilkan.

2.2.16 Class Diagram

Class Diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi *class*, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi. Diagram kelas ini sesuai jika diimplementasikan ke proyek yang menggunakan konsep object-oriented karena gambaran dari *Class Diagram* cukup mudah untuk digunakan[18].

Tabel 2. 5 *Class Diagram*

No	Simbol	Nama Simbol	Keterangan
1		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
2		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
3		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung pada elemen yang tidak mandiri.
4		<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya
5		<i>Agregation</i>	Hubungan antara dua <i>class</i> di mana salah satu <i>class</i> merupakan bagian dari <i>class</i> lain, tetapi dua <i>class</i> ini dapat berdiri masing-masing.

6		<i>Inheritance</i>	mewarisi seluruh atribut dan metode dari <i>class</i> asalnya (<i>superclass</i>) ke <i>class</i> lain (<i>subclass</i>).
---	---	--------------------	---



2.2.17 ERD (*Entity Relationship Diagram*)

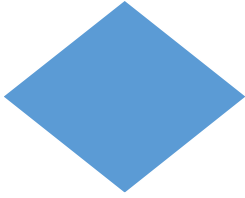
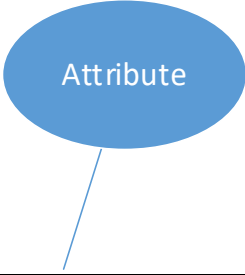
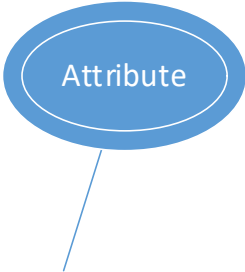
ERD (*Entity Relationship Diagram*) merupakan diagram berbentuk notasi grafis yang berada dalam pembuatan *database* yang menghubungkan antara data satu dengan data lainnya. ERD berfungsi sebagai alat bantu dalam pembuatan *database* dan memberikan gambaran bagaimana kerja *database* yang akan dibuat[19].

ERD memiliki 3 relasi yaitu :

1. *One to One*
Satu anggota entitas dapat berelasi dengan satu anggota entitas lain.
2. *One to Many*
Satu anggota entitas dapat berelasi dengan beberapa anggota entitas lain.
3. *Many to Many*
Beberapa anggota entitas dapat berelasi dengan beberapa anggota entitas lain.

Tabel 2. 6 *Entity Relationship Diagram*

No	Simbol	Nama Simbol	Keterangan
1		<i>Entity</i>	objek atau konsep yang ingin Anda simpan informasinya.
2		<i>Weak Entity</i>	entitas yang harus ditentukan oleh hubungan kunci asing dengan entitas lain karena tidak dapat diidentifikasi secara unik dengan

			atributnya sendiri.
3		<i>Relationship</i>	menunjukkan bagaimana dua entitas berbagi informasi dalam <i>database</i> .
4		<i>Attribute</i>	karakteristik entitas yang unik dan berbeda.
5		<i>Multi-Valued Attribute</i>	Atribut multivalai dapat memiliki lebih dari satu nilai.