

BAB II

LANDASAN TEORI

2.1 PROFIL CV. MULIA JAMBI

CV. Mulia Jambi sebagai perusahaan yang bergerak di bidang konstruksi dan kesehatan. Awal CV. Mulia Jambi ini didirikan pada tanggal 18 Mei 2007 yang beralamat di Jln. Amangkurat No. 82, Kecamatan Jambi Timur, Jambi yang dipimpin oleh Bapak Ahmad Mulyadi dan memiliki beberapa orang tenaga kerja.

CV. Mulia Jambi didirikan dengan maksud untuk memenuhi kebutuhan konsumen dan untuk memperoleh keuntungan, tentunya dengan menggunakan strategi yang baik, salah satunya yaitu bagaimana menjalin hubungan yang baik kepada konsumen. Karena dengan menjalin hubungan yang baik kepada para konsumen akan meningkatkan omzet perusahaan. Dengan meningkatkan omzet perusahaan maka sudah pasti keuntungan yang diperoleh juga meningkat. Dengan begitu pemilik sangat memperhatikan pelayanan dan kualitas produk yang akan diberikan kepada pelanggan sehingga pelanggan dapat merasa puas akan fasilitas dan jasa yang diberikan.

Visi dan misi CV. Mulia Jambi, antara lain :

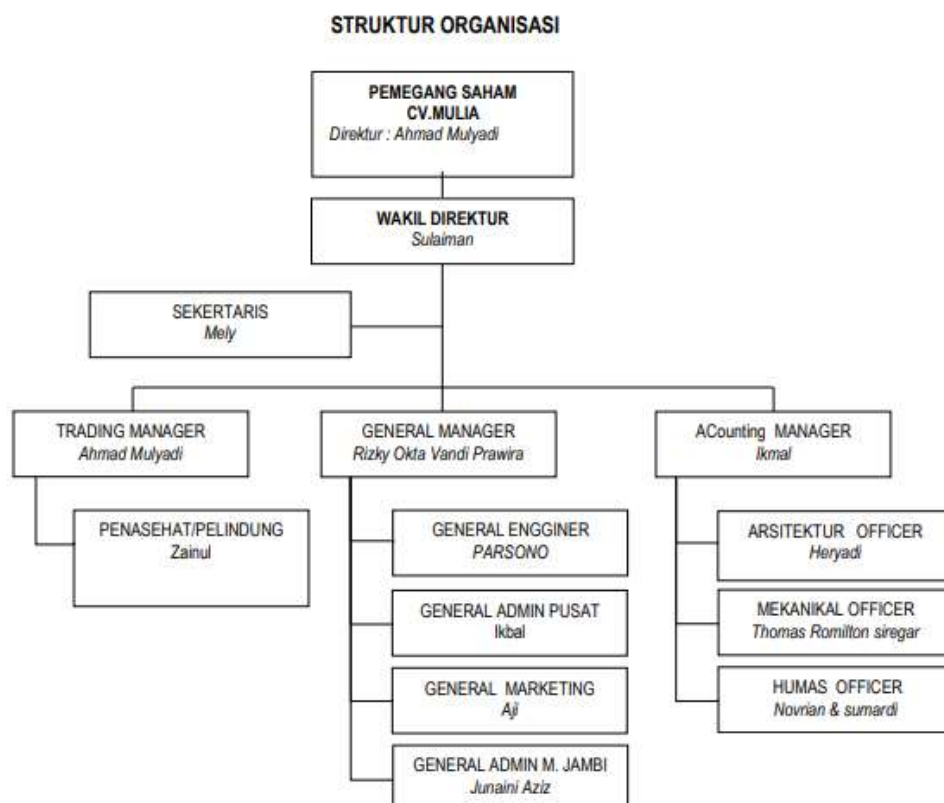
a. Visi

Menjadi suatu perusahaan yang bergerak dalam konstruksi dan kesehatan yang terdepan di Provinsi Jambi dan selalu mengutamakan mutu dan kualitas.

b. Misi

1. Memberikan pelayanan terbaik dan standard mutu pada pelanggan dengan menjalankan proses kerja terbaik sehingga tercapai kepuasan pelanggan.
2. Meningkatkan motivasi dan semangat kerja karyawan secara optimal melalui peningkatan dedikasi, disiplin, dan kemampuan kerja serta penghargaan yang memadai sesuai dengan kinerjanya.

Berikut ini merupakan gambar stuktur CV. Mulia Jambi yang dapat dilihat pada gambar 2.1.



Gambar 2.1 Struktur Organisasi CV. Mulia Jambi

CV. Mulia Jambi saat ini menyeleksi penerimaan karyawan baru pada CV. Mulia Jambi untuk jabatan admin pusat bagian personalia, *marketing* (pemasaran) dan *accounting* (keuangan) dikarenakan karyawan yang bekerja terkadang berhenti dan juga karyawan yang bekerja kurang berkompeten dalam pekerjaannya sehingga dibutuhkan sistem pendukung keputusan untuk seleksi karyawan.

2.2 SISTEM PENDUKUNG KEPUTUSAN

Dalam pengambilan keputusan, perlu diperhatikan beberapa hal mengenai apa itu keputusan dan bagaimana kriteria keputusan, tipologi keputusan, dan langkah-langkah pengambilan keputusan. Selain beberapa hal dalam pengambilan keputusan tersebut, perlu juga kita memahami apa tujuan sistem pendukung keputusan, karakteristik dan nilai guna sistem pendukung keputusan, komponen, langkah-langkah pemodelan dalam sistem pendukung keputusan.

Kadir [4] menyatakan bahwa : “Sistem pendukung keputusan (SPK) atau *Decision Support System* (DSS) adalah sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data yang digunakan untuk membantu pengambilan keputusan pada situasi yang semi terstruktur dan situasi yang tidak terstruktur dimana tak seorang pun tahu secara pasti bagaimana keputusan itu dibuat”.

Nofriansyah dan Defit [5] menyatakan bahwa “Sistem pendukung keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu

manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur”.

Sari [6] menyatakan bahwa : “Sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model”.

Sehingga dapat disimpulkan bahwa sistem penunjang keputusan merupakan sistem informasi yang mendukung manajemen level menengah dalam mengambil keputusan setengah terstruktur atau semi *structured* dengan menggunakan pemodelan analitis dan data yang ada.

2.2.1 Aspek Utama Sistem Pendukung Keputusan

Dalam Sistem Pendukung Keputusan memiliki 3 aspek utama [1], yaitu :

1. Data, data yang digunakan dalam SPK adalah data yang diambil dari data *warehouse* pada organisasi yang telah dikategorikan berdasarkan kebutuhan.
2. Model matematis, merupakan bagian untuk menganalisis data dan berfungsi untuk merubah data menjadi informasi dan *knowledge* yang berguna untuk pengambilan keputusan.
3. Antarmuka pengguna (*user interface*). Aspek ini merupakan aspek yang secara langsung dilihat dan berinteraksi dengan *end user* atau dalam hal ini pemegang keputusan. Data yang ditampilkan harus memberikan informasi yang *valid*, *reliable* dan dapat mendukung untuk pengambilan keputusan.

2.2.2 Tujuan Sistem Penunjang Keputusan

Sistem Penunjang Keputusan merupakan suatu sistem informasi untuk membantu manajer level menengah untuk proses pengambilan keputusan setengah terstruktur (*semi structured*) supaya lebih efektif dengan menggunakan model-model analitis dan data yang tersedia. Sistem pendukung keputusan memiliki tujuan – tujuan [7], yang diperlukan, yaitu

1. Membantu manajer dalam pengambilan keputusan atas masalah semi terstruktur.
2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
3. Meningkatkan efektifitas keputusan yang di ambil manajer lebih daripada perbaikan efisiensinya.
4. Kecepatan komputasi.
5. Peningkatan produktifitas.
6. Dukungan kualitas.
7. Berdaya saing.
8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

2.2.3 Karakteristik Sistem Pendukung Keputusan

Karakteristik dari sistem pendukung keputusan [1], antara lain :

1. Dukungan untuk pengambil keputusan, terutama pada situasi semi terstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
2. Dukungan untuk keputusan independen dan sekuensial.
3. Dukungan disemua fase proses pengambilan keputusan: inteligensi, desain, pilihan dan implementasi.
4. Dukungan diberbagai proses dan gaya pengambilan keputusan.
5. Ada aktivitas sepanjang waktu.
6. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, *timelines*, kualitas) ketimbang pada efisiensinya (biaya).
7. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan masalah.
8. Pengguna akhir (*end user*) dapat mengembangkan dan memodifikasi sendiri sistem sederhana.

2.2.4 Fase-Fase Proses Pengambilan Keputusan

Ada 3 fase proses pengambilan keputusan [1], yaitu :

1. *Intelligence* : pengumpulan informasi untuk mengidentifikasi permasalahan.
2. *Design* : tahap perancangan solusi dalam bentuk alternatif pemecahan masalah.

3. *Choice* : melakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan.

2.3 METODE TOPSIS (*TECHIQUE FOR OTHERS REFERENCE BY SIMILARITY TO IDEAL SOLUTION*)

TOPSIS adalah salah satu metode pengambilan keputusan multi kriteria yang pertama kali diperkenalkan oleh Yoon dan Hwang (1981). 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. Solusi ideal positif didefinisikan sebagai jumlah dari seluruh nilai terbaik yang dapat dicapai untuk setiap atribut, sedangkan solusi negatif-ideal terdiri dari seluruh nilai terburuk yang dicapai untuk setiap atribut.

TOPSIS mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif dengan mengambil kedekatan relatif terhadap solusi ideal positif. Berdasarkan perbandingan terhadap jarak relatifnya, susunan prioritas alternatif bisa dicapai.

Ismail dan Hariansyah [8] menyatakan bahwa : “TOPSIS adalah salah satu metode pengambilan keputusan multikriteria atau *alternative* pilihan yang merupakan *alternative* yang mempunyai jarak terkecil dari solusi ideal positif

dan jarak terbesar dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *euclidean*”.

Yani, dkk [2] menyatakan bahwa : “*Technique for Order Performance by Similarity to Ideal Solution* (TOPSIS) merupakan salah satu metode dalam pengambilan keputusan akan memilih alternatif yang tidak hanya paling mendekati solusi ideal positif, akan tetapi juga paling jauh dari solusi ideal negatif”.

Langkah-langkah yang harus dilakukan untuk memperoleh hasil penilaian metode TOPSIS [8] adalah :

1. Membangun *normalized decision matrix*

Elemen r_{ij} hasil dari normalisasi *decision matrix* R dengan metode *Euclidean length of a vector* adalah:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \dots\dots\dots(2.1)$$

2. Membangun *weighted normalized decision matrix*

Dengan bobot $W = (w_1, w_2, \dots, w_n)$, maka normalisasi bobot matriks V adalah:

$$V = \begin{bmatrix} w_1 r_{11} & \dots & w_n r_{1n} \\ \vdots & \ddots & \vdots \\ w_1 r_{m1} & \dots & w_n r_{m1} \end{bmatrix} \dots\dots\dots(2.2)$$

3. Menentukan solusi ideal dan solusi ideal negatif.

Solusi ideal dinotasikan A^* , sedangkan solusi ideal negatif dinotasikan A^- :

$$A^* = \{(\max v_{ij} \mid j \in J, (\min v_{ij} \mid j \in J'), i = 1, 2, 3, \dots, m) = v_{1z}, v_{2z} \dots \dots v_{nz}\}$$

$$A^- = \{(\min v_{ij} \mid j \in J, (\max v_{ij} \mid j \in J'), i = 1, 2, 3, \dots, m) = v_{1z}, v_{2z} \dots \dots v_{nz}\} \dots\dots\dots(2.3)$$

Dimana :

$J = \{j = 1, 2, 3, \dots, n \text{ dan } j \text{ merupakan } \textit{benefit criteria}\}$

$J' = \{j = 1, 2, 3, \dots, n \text{ dan } j \text{ merupakan } \textit{cost criteria}\}$

4. Menghitung separasi

S_{i*} adalah jarak (dalam pandangan *Euclidean*) alternatif dari solusi ideal didefinisikan sebagai:

$$S_{i*} = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^*)^2}, \text{ dengan } i = 1, 2, 3 \dots m \dots \dots \dots (2.4)$$

Dan jarak terhadap solusi negatif-ideal didefinisikan sebagai:

$$S_{i-} = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \text{ dengan } i = 1, 2, 3 \dots m \dots \dots \dots (2.5)$$

Menghitung kedekatan relatif terhadap solusi ideal

$$C_{i*} = \frac{S_{i-}}{S_{i*} + S_{i-}}, \text{ dengan } 0 < C_{i*} < 1 \text{ dan } i = 1, 2, 3, \dots, m \dots \dots \dots (2.6)$$

5. Merangking Alternatif

Alternatif dapat dirangking berdasarkan urutan C_{i*} . Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal

Kelebihan dari metode TOPSIS, yaitu :

1. Konsepnya sederhana dan mudah dipahami,
2. Komputasinya efisien, dan
3. Memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana.

Kekurangan dari metode TOPSIS, yaitu :

1. Harus adanya bobot yang dihitung menggunakan AHP untuk melanjutkan hitungan data selanjutnya dengan memakai TOPSIS.

2.4 KARYAWAN

Tiap-tiap perusahaan memerlukan karyawan jadi tenaga yang menggerakkan tiap-tiap kesibukan yang ada pada organisasi perusahaan. Karyawan adalah aset terutama yang mempunyai dampak begitu besar pada keberhasilan satu perusahaan. Tanpa ada mesin mutakhir, perusahaan bisa selalu beroperasi dengan manual, walau demikian tanpa ada karyawan, perusahaan akan tidak bisa jalan sekalipun. Berikut ini pengertian karyawan menurut para ahli antara lain :

Sholihaningtias [9] menyatakan bahwa : “Karyawan adalah setiap orang yang mampu melakukan pekerjaan guna menghasilkan barang dan jasa baik untuk memenuhi kebutuhan sendiri maupun masyarakat, baik didalam maupun diluar hubungan kerja”.

Siagian [1] menyatakan bahwa : “Karyawan adalah seorang pekerja yang bekerja dibawah perintah orang lain dan mendapat kompensasi serta jaminan”.

Berdasarkan penjelasan dapat disimpulkan bahwa karyawan bisa disimpulkan adalah orang yang bekerja pada suatu lembaga “kantor, perusahaan, dan sebagainya” dengan mendapat gaji “upah”, pegawai, pekerja.

2.5 DATABASE

Database dapat diartikan sebagai suatu kumpulan file atau data yang saling terhubung yang disimpan pada suatu media tertentu, dan diorganisasikan dengan cara tertentu. Sistem database merupakan sistem yang memuat data yang terorganisasi dengan baik sehingga memudahkan penyimpanan dan pengambilan kembali secara elektronis.

A. S dan Shalahuddin [10] menyatakan bahwa “Basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat”.

Chan [11] menyatakan bahwa : “*Database* (basis data) adalah tempat menyimpan koleksi data yang terorganisir yang terdiri dari skema, tabel, *view*, *query*, *store procedure*, dan objek-objek lainnya”.

Berdasarkan dari definisi para pakar dapat disimpulkan bahwa *database* merupakan suatu kumpulan data yang saling berhubungan yang dapat diolah dan bisa menghasilkan informasi secara terstruktur.

Istilah-istilah dasar database sebagai berikut :

1. Entitas, yaitu sekumpulan obyek yang mempunyai karakteristik sama dan bisa dibedakan dari lainnya.
2. *Atribut*, yaitu deskripsi data yang bisa mengidentifikasi entitas.
3. *Field*, yaitu lokasi penyimpanan untuk salah satu elemen data atribut.
4. *Record*, yaitu kumpulan dari field yang berhubungan satu sama lain.
5. *File*, yaitu kumpulan dari record yang menggambarkan himpunan entitas.

6. Basis Data, yaitu kumpulan file yang digunakan untuk membentuk hubungan tertentu di antara record-record di file-file tersebut.
7. *Key*, yaitu elemen record yang dipakai untuk menemukan record tersebut pada waktu akses.

2.6 ALAT BANTU PEMODELAN SISTEM

2.6.1 *Use Case Diagram*

Untuk menganalisis kebutuhan sistem penulis menggunakan alat bantu yaitu dengan *use case diagram*. Tujuan pembuatan *use case* adalah untuk mendapatkan dan menganalisis informasi persyaratan yang cukup untuk mempersiapkan model yang mengkomunikasikan apa yang diperlukan dari perspektif pengguna, tetapi bebas dari detail fisik tentang bagaimana sistem akan dibangun dan diimplementasikan. Dan ada pula beberapa pengertian menurut para ahli antara lain :

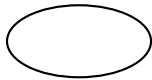
A. S dan Shalahuddin [10] menyatakan bahwa “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat”.

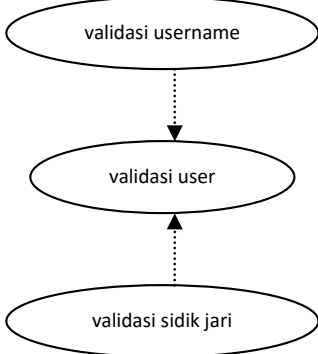
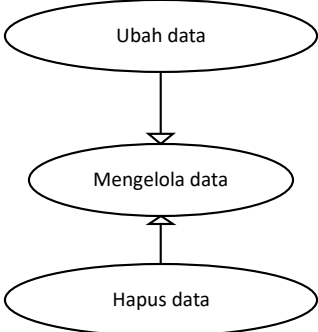
Sholih [12] menyatakan bahwa “*Use case* adalah fungsionalitas atau persyaratan-persyaratan sistem yang harus dipenuhi oleh sistem yang dikembangkan tersebut menurut pandangan pemakai sistem”.

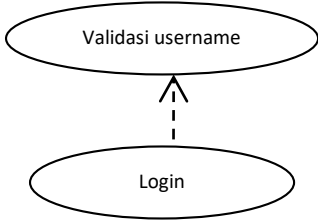
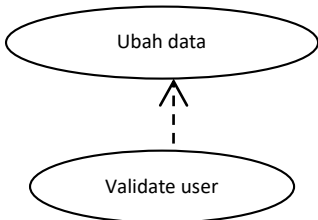
Sehingga dari pendapat para ahli tersebut dapat disimpulkan bahwa *use case diagram* adalah pemodelan deskripsi fungsi yang digunakan untuk mendeskripsikan sistem informasi yang akan dibuat secara keseluruhan.

Use case dapat menjadi basis pertama dalam melakukan kegiatan analisis. Pada UML, representasi *use case* adalah diagram *use case*. Diagram *use case* berisi *actor* dan *use case*. Diagram *use case* dibuat dengan menggunakan simbol-simbol yang dapat dilihat pada tabel 2.1

Tabel 2.1 Simbol Dan Fungsi *Use Case Diagram* [10]

No.	Simbol	Nama Simbol	Kegunaan
1		<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama <i>use case</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang : biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
3		<i>Association</i>	Komunikasi antara aktor dan <i>Use Case</i> yang berpartisipasi pada <i>Use Case</i> memiliki interaksi dengan aktor
4		<i>Extend</i>	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 inheritance pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal  Arah panah mengarah pada <i>use case</i> yang ditambahkan.
5	→	Generalisasi	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	---→	Include	Sebagai syarat dijalankan <i>use case</i> ini Ada 2 sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : 1. Include berarti <i>use case</i>

			yang ditambahkan akan selalu dipanggil saat <i>use case</i> tambahan dijalankan, misalnya  <pre> graph BT Login((Login)) ValidasiUsername((Validasi username)) Login -.-> ValidasiUsername </pre> 2. Include berarti <i>use case</i> yang ditambah akan selalu melakukan pengecekan apakah <i>use case</i> yang ditambahkan telah dijalankan sebelum <i>use case</i> tambahan dijalankan, misal  <pre> graph BT ValidateUser((Validate user)) UbahData((Ubah data)) ValidateUser -.-> UbahData </pre>
--	--	--	---

2.6.2 Activity Diagram

Pada dasarnya, diagram aktivitas merupakan diagram *flowchart* yang diperluas yang menunjukkan aliran kendali dari satu aktivitas ke aktivitas lain. Diagram aktivitas mendeskripsikan aksi-aksi dan hasil aksinya. Diagram aktivitas berupa operasi-operasi dan aktivitas-aktivitas pada *use case*.

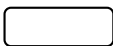
A. S dan Shalahuddin [10] menyatakan bahwa “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.

Sholiq [12] menyatakan bahwa “Diagram aktivitas adalah cara untuk memodelkan alur kerja (*workflow*) dari *use case* bisnis dalam bentuk grafik”.

Sehingga dapat disimpulkan bahwa *activity diagram* adalah diagram yang menggambarkan aliran kerja untuk memodelkan *event-event* yang terjadi dalam suatu *use case*.

Activity Diagram dibuat dengan menggunakan simbol-simbol yang dapat dilihat pada tabel 2.2

Tabel 2.2 Simbol Dan Fungsi *Activity Diagram* [10]

No.	Simbol	Nama Simbol	Keterangan
1.		Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2.		Aktivitas	Aktivitas yang biasa dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3.		Percabangan	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
5.		Penggabungan	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
6.		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki status akhir
7.		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.6.3 *Class Diagram*

Diagram kelas menunjukkan struktur informasi serta juga mendeskripsikan perilaku. Diagram kelas membantu dalam visualisasi struktur

kelas-kelas dari suatu sistem dan hubungan antar kelas (*inheritance*, *aggregation*, dan *association*) dan penjelasan detail tiap kelas (*methode/function/behavior* dan *atribut/property/data*). Dan ada pula beberapa pengertian menurut para ahli antara lain :

A. S dan Shalahuddin [10] menyatakan bahwa “Diagram kelas atau *Class diagram* menggambarkan stuktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”.

Sholiq [12] menyatakan bahwa “Diagram kelas digunakan untuk menampilkan kelas-kelas atau paket-paket dalam sistem dan relasi antar mereka”.

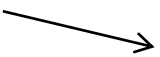
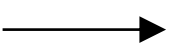
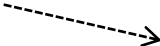
Sehingga dapat disimpulkan bahwa diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Diagram kelas menunjukkan struktur informasi serta juga mendeskripsikan perilaku. Diagram kelas membantu dalam visualisasi struktur kelas-kelas dari suatu sistem dan hubungan antar kelas (*inheritance*, *aggregation*, dan *association*) dan penjelasan detail tiap kelas (*methode/function/behavior* dan *atribut/property/data*).

Pemodelan kelas merupakan paling utama di pendekatan berorientasi objek permodelan kelas menunjukkan kelas-kelas yang ada di pendekatan berorientasi objek permodelan kelas menunjukkan kelas-kelas yang ada di sistem dan hubungan antar kelas-kelas itu, atribut-atribut dan operasi-operasi di kelas-

kelas. Diagram *class* dibuat dengan menggunakan simbol-simbol yang dapat dilihat pada tabel 2.3.

Tabel 2.3 Simbol Dan Fungsi *Class Diagram* [10]

No.	Simbol	Nama Simbol	Keterangan
1.	Nama Kelas + Atribut + Operasi	Kelas	Kelas Pada Struktur Sistem.
2.		Antarmuka / <i>Interface</i>	Sama dengan konsep <i>interface</i> dalam promgraman berorientasi objek.
3.		Asosiasi / <i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
4.		Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi, biasanya juga disertasi 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.

2.6.4 *Sequence Diagram*

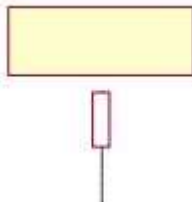
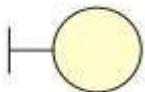
Sequence diagram digunakan untuk menjelaskan perilaku pada sebuah skenario dan menggambarkan bagaimana entitas dan sistem berinteraksi, termasuk pesan yang dipakai saat interaksi. Semua pesan digambarkan dalam urutan pada eksekusi. Dan ada pula beberapa pengertian menurut para ahli antara lain :

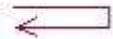
A. S dan Shalahuddin [10] menyatakan bahwa “*Sequence diagram* merupakan suatu diagram yang menjelaskan interaksi objek dan menunjukkan (memberi tanda atau petunjuk) komunikasi diantara objek-objek tersebut”.

Sholiq [12] menyatakan bahwa “*Sequence diagram* merupakan sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci”.

Sehingga dapat disimpulkan bahwa *Sequence diagram* merupakan salah satu yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya.

Tabel 2.4 Simbol Dan Fungsi *Sequence Diagram* [10]

No	Simbol	Nama	Keterangan
1		Aktor	Merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem.
2		Lifeline	Menghubungkan objek selama sequence (message dikirim atau diterima dan aktifasinya).
3		General	Merepresentasikan entitas tunggal dalam sequence diagram.
4		Boundary	Berupa tepi dari sistem, seperti user interface atau suatu alat yang berinteraksi dengan sistem yang

			lain.
5		Control	Element mengatur aliran dari informasi untuk sebuah skenario. Objek ini umumnya mengatur perilaku dan perilaku bisnis.
6		Entitas	Elemen yang bertanggung jawab menyimpan data atau informasi. Ini dapat berupa beans atau model object.
7		Activation	Suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah sequence yang menunjukkan kapan sebuah objek mengirim atau menerima objek.
8		Message Entry	Berfungsi untuk menggambarkan pesan/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi.
9		Message to Self	Simbol ini menggambarkan pesan/hubungan objek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.
10		Message Return	Menggambarkan hasil dari pengiriman message dan digambarkan dengan arah dari kanan ke kiri.

2.6.5 *Flowchart*

Flowchart merupakan diagram yang menggambarkan aliran sistem dimana *flowchart* membantu perancang sistem untuk melihat aliran sistem yang dirancang dan mengetahui sistem mana yang akan dibuat. Dan ada pula beberapa pengertian menurut para ahli antara lain :

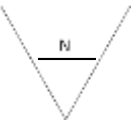
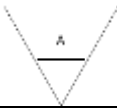
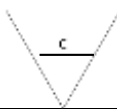
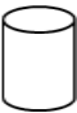
Indrajani [13] mengungkapkan bahwa : “*Flowchart* merupakan penggambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program.”

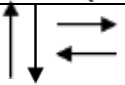
Kursini dan Koniyo [14] mengungkapkan bahwa, “Bagan Alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan aliran (*flow*) di dalam program atau prosedur sistem secara logika, digunakan terutama sebagai alat bantu komunikasi dan untuk dokumentasi”.

Berdasarkan beberapa pendapat yang dikemukakan diatas dapat ditarik kesimpulan bahwa, *flowchart* atau diagram alur adalah suatu alat yang banyak digunakan untuk membuat algoritma, yakni bagaimana rangkaian pelaksanaan suatu kegiatan. Suatu diagram alur memberikan gambaran dua dimensi berupa simbol-simbol grafis. Masing-masing simbol telah ditetapkan terlebih dahulu fungsi dan artinya.

Tabel 2.5 Simbol Bagan Alir Dokumen [14]

No	Simbol	Keterangan
1		Dokumen, menunjukkan input dan <i>output</i> baik untuk proses manual, mekanik dan komputer.

2		Manual , Menunjukkan pekerjaan manual
3		Simpanan <i>offline</i> , file <i>non</i> -komputer yang diarsip urut angka.
4		Simpanan <i>offline</i> ,file <i>non</i> -komputer yang diarsip urut huruf.
5		Simpanan <i>Offline</i> , file <i>non</i> -komputer yang diarsip urut tanggal.
6		Kartu punc, menunjukkan I/O yang menggunakan kartu punc.
7		Proses, menunjukkan kegiatan proses dari operasi program komputer.
8		Operasi luar, menunjukkan operasi yang dilakukan di luar operasi komputer.
9		Sort <i>offline</i> , menunjukkan proses pengurutan data di luar proses komputer.
10		Pita magnetik, menunjukkan i/o menggunakan pita magnetik
11		Disk, menunjukkan i/o menggunakan harddisk.
12		Disket, menunjukkan i/o menggunakan disket
13		Drum magnetik, menunjukkan I/O menggunakan drum magnetik
14		Pita kertas berlubang, menunjukkan I/O menggunakan pita kertas berlubang

15		Keyboard, menunjukkan input yang menggunakan online keyboard
16		Display, menunjukkan output yang ditampilkan di monitor
17		Hubungan Komunikasi, menunjukkan proses transmisi data melalui saluran komunikasi
18		Garis alir, menunjukkan aliran proses
19		Penjelasan, menunjukkan penjelasan dari suatu proses.
20		Penghubung, menunjukkan penghubung ke halaman yang sama atau halaman lain
21		Pita Kontrol, menunjukkan penggunaan pita kontrol dalam batch control untuk pencocokan di proses batch processing

2.7 ALAT BANTU PEMBUATAN PROGRAM

2.7.1 *Hyper Text Markup Language (HTML)*

HTML (*HyperText Markup Language*) adalah standar dipakai pada halaman Web. Berdasarkan standar inilah *browser* bisa memahami isi suatu dokumen yang berasal dari *Web Server*. HTML berkerja menggunakan HTTP (*HyperText Transfer Protokol*), yaitu protokol komunikasi yang memungkinkan *Web Server* berkomunikasi dengan *Web Browser*. Berikut ini beberapa definisi HTML antara lain sebagai berikut :

Sidik dan Pohan [15] menyatakan bahwa “HTML kependekan dari *Hyper Text Markup Language*. Dokumen HTML adalah file teks murni yang dapat dibuat dengan editor teks sembarang”.

Enterprise [16] menyatakan bahwa : “HTML adalah *Hypertext Markup Language*, yang artinya adalah sebuah teks berbentuk link yang saat di-klik akan membawa kita berpindah dari satu dokumen ke dokumen lainnya”.

Berdasarkan uraian diatas dapat disimpulkan bahwa HTML (*HyperText Mark up Language*) merupakan metoda untuk mengimplementasikan konsep *hypertext* dalam suatu naskah atau dokumen. HTML bukan tergolong bahasa pemrograman, karena sifatnya yang hanya memberikan tanda (*marking up*) pada suatu naskah teks dan bukan sebagai program.

2.7.2 PHP

Sampai saat ini, PHP adalah bahasa pemrograman yang paling sering digunakan oleh semua kalangan untuk membuat website, selain bahasa pemrogramannya mudah, PHP juga dapat dihubungkan dengan berbagai macam jenis *database* dengan kelengkapan yang berbeda-beda yaitu sebagai berikut : *MySQL, MSQL, Oracle, Inges, Ms Access, InterBase, DBM, Sybase*, dan *PostgreSQL*.

Saputra [17] menyatakan bahwa “PHP atau yang memiliki kepanjangan *PHP Hypertext Preprocessor* merupakan suatu bahasa pemograman yang difungsikan untuk membangun suatu *website* dinamis”.

Raharjo [18] menyatakan bahwa “PHP adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*”.

Enterprise [19] menyatakan bahwa : “PHP merupakan bahasa pemrograman yang digunakan untuk membuat aplikasi berbasis *website*”.

Berdasarkan uraian diatas dapat disimpulkan bahwa PHP atau PHP *Hypertext Preprocessor* merupakan suatu bahasa pemrograman berbentuk sebuah skrip yang ditempatkan dan di proses dalam *server*. Keluaran dari proses ini dikirim ke *client*, dan dibuka menggunakan *browser*.

Beberapa keunggulan PHP antara lain :

1. Cepat. Karena ditempelkan (*embedded*) di dalam kode HTML, sehingga waktu tanggap menjadi pendek.
2. Tidak mahal – gratis. Pada kenyataanya PHP adalah gratis dan anda bisa mendapatkannya tanpa harus membayarnya.
3. Mudah untuk digunakan. PHP berisi beberapa fitur khusus dan fungsi yang dibutuhkan untuk membuat halaman *web* dinamis, bahasa PHP dirancang untuk dimasukkan dengan mudah di dalam file HTML.
4. Berjalan pada beberapa sistem operasi. Dia bejalan pada sitem operasi yang beragam, Windows, Linu, Mac OS, dan kebanyakan variasi dari Unix.
5. Dukungan teknis tersedia secara luas. Karena PHP menyediakan dukungan gratis via daftar diskusi *e-mail*.
6. Aman. Pengguna tidak melihat kode PHP, karena kode yang ditampilkan pada *browser* adalah kode HTML.

2.7.3 MySQL

MySQL merupakan *database* yang dapat menyimpan berbagai informasi dengan membagi berdasarkan kategori-kategori tertentu. Dan ada pula beberapa pengertian menurut para ahli antara lain :

Raharjo [20] menyatakan bahwa “MySQL merupakan *software* RDMS (atau *server database*) yang dapat mengelola *database* dengan sangat cepat, dapat menampung data dalam jumlah besar, dan dapat diakses oleh banyak *user* (*multi user*), dan dapat melakukan suatu proses secara sinkron atau bersamaan (*multi threaded*)”.

Puspitosari [21] menyatakan bahwa “MySQL merupakan salah satu *software* untuk *database* server yang banyak digunakan, MySQL bersifat *open source* dan menggunakan SQL”.

Arief [22] menyatakan bahwa “MySQL adalah salah satu *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya”.

Selain itu, MySQL dapat berjalan diberbagai platform, antara lain Linux dan Windows. Dan kini PHP telah menyediakan fasilitas untuk koneksi hampir ke semua program database populer, baik yang bersifat komersial maupun gratis, dan ini sangat mudah untuk digunakan.

Berdasarkan uraian diatas dapat disimpulkan bahwa MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL

2.7.4 *Visual Studio Code*

Visual Studio Code adalah *editor source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, *shortcut keyboard*, dan preferensi. *Visual Studio Code* gratis dan *open-source*, meskipun unduhan resmi berada di bawah *lisensi proprietary*. Berikut adalah beberapa definisi tentang *Visual Studio Code* :

Aprilia dan Anwar [23] menyatakan “*Visual Studio Code* yakni sebuah *framework* PHP yang bersifat *open source* dengan menggunakan model MVC (*Model, View, Controller*) untuk membangun sebuah website dinamis dengan menggunakan PHP”.

Ramdhan dan Nufriana [24] menyatakan “*Visual Studio Code* merupakan sebuah aplikasi *editor code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS”.

Berdasarkan penjelasan beberapa ahli dapat disimpulkan *visual studio code* merupakan adalah *editor source code* yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk *debugging*, GIT *Control* yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode *refactoring*.

2.7.5 XAMPP

Untuk membuat suatu aplikasi berbasis *Web* dengan menggunakan bahasa PHP , tentu saja diperlukan sebuah *server* dan interpreter PHP. *Server* tidak harus sebuah komputer khusus dengan kinerja tinggi dan berukuran sangat besar, tetapi bisa di buat dari PC yang mempunyai fungsi selayaknya sebuah *Web server*, yaitu dengan menginstal XAMPP. Berikut adalah beberapa definisi tentang XAMPP :

Nugroho [25] menyatakan bahwa “XAMPP adalah program *web* lengkap yang dapat ada pakai untuk belajar pemograman *web*, khususnya PHP dan MySQL, paket ini dapat didownload secara gratis dan legal”.

Enterprise [16] menyatakan bahwa XAMPP merupakan *server* yang paling banyak digunakan. Fiturnya lengkap, tetap gampang digunakan oleh *programer* PHP pemula.

Berdasarkan pendapat para ahli tersebut dapat disimpulkan bahwa, XAMPP adalah suatu *software* yang berbasis *open source* yang di dalamnya terdapat *software-software* pembantu seperti *Apache*, MySQL, PHP, dan *PhpMyAdmin*, yang dapat digunakan sebagai alat bantu pengembangan aplikasi berbasis PHP.

2.8 PENELITIAN SEJENIS

Penelitian sejenis merupakan tinjauan penelitian yang sejenis dengan penelitian yang diambil oleh penelitian sebagai acuan atau referensi untuk perancangan sistem. Dan penelitian sejenis dapat dilihat pada tabel 2.6

Tabel 2.6 Penelitian Sejenis

No.	Penulis (Tahun)	TOPSIS	Pembahasan
1.	Zulfitri Yani, Devi Gusmita, dan Nurmaliana Pohan [2]	TOPSIS	1. Masalah : proses penerimaan karyawan dan perusahaan masih terlihat subjektif sehingga salah dalam mengambil keputusan untuk meloloskan karyawan, dan karyawan yang diterima nantinya tidak memenuhi standar kualifikasi perusahaan 2. Pembahasan : Sistem menggunakan metode TOPSIS yang terdiri dari 7 kriteria, yaitu tingkat pendidikan, pengalaman kerja, skill / keahlian, penampilan, bisa bekerja tim, hasil tes tertulis dan hasil tes wawancara yang digunakan oleh HRD perusahaan 3. Hasil : aplikasi yang telah dibuat dan sesuai dengan rancangan dapat memudahkan pengguna dalam menentukan calon karyawan yang telah lulus seleksi. Dan cukup userfriendly bagi pengguna untuk menggunakan aplikasi in
2.	Yessica Siagian [1]	TOPSIS	1. Permasalahan : Terjadinya kesalahan dalam memilih orang yang tepat bagi perusahaan dan proses rekrutmen dan seleksi yang menyita waktu, biaya dan tenaga 2. Pembahasan : Sistem penerimaan pegawai baru menggunakan metode TOPSIS dalam pemilihannya dengan 8 kriteria dengan proses perhitungan dengan TOPSIS yang mudah, memberikan kemudahan bagi perusahaan dalam memahaminya 3. Hasil : memberikan alternatif keputusan bagi HRD perusahaan dalam mengambil keputusan untuk menentukan pegawai baru yang sesuai dengan kebutuhan perusahaan.
3.	Muhammad Fahrur Rozi, Edy Santoso, dan Muhammad Tanzil Furqon [3]	AHP dan TOPSIS	1. Permasalahan : penerimaan pegawai baru sering terjadi kesalahan yang tidak kompeten dalam berkerja dan tidak sesuai dengan harapan perusahaan

			2. Pembahasan : sistem penerimaan pegawai baru menggunakan metode AHP dan TOPSIS dengan 4 kriteria, Dan akan memberikan hasil akhir dengan perangkingan. 3. Hasil : Hasil akhir dalam bentuk perangkingan maka memberikan hasil yang optimal dalam mencari pelamar yang terbaik sesuai dengan seleksi berdasarkan kriteria.
4.	Enggar Novianto dan Sri Winiarti [26]	SAW	1. Permasalahan : calon pelamar mengirimkan berkas persyaratan melalui kantor pos atau mengirimkan secara langsung yang ditujukan kepada bagian Human Resource Development (HRD). Model pengiriman berkas seperti tersebut menyulitkan calon pelamar dalam mengirimkan berkas yang ditujukan kepada bagian HRD, terutama bagi calon pelamar yang berasal dari luar daerah dan dapat menyebabkan berkas tersebut rusak ataupun hilang selama proses pengiriman 2. Pembahasan : sistem penunjang keputusan menggunakan metode SAW dengan 6 kriteria yang digunakan 3. Hasil : memberikan hasil keputusan sebagai referensi perusahaan untuk memilih karyawan yang ditempatkan di sesuai posisinya dengan kriteria yang telah digunakan.
5.	Ramos Somya dan Andre Wahyudi [27]	TOPSIS	1. Permasalahan : banyaknya kriteria penilaian yang harus dipertimbangkan serta melibatkan beberapa orang dalam pengambilan keputusan. Untuk mengatasi permasalahan tersebut, perlu untuk dibuat sebuah sistem pendukung keputusan perekrutan karyawan baru di PT Visionet Data Internasional agar proses perekrutan karyawan baru menjadi lebih baik 2. Pembahasan : sistem penunjang keputusan menggunakan metode TOPSIS dengan 5 kriteria yang digunakan 3. Hasil : sistem pendukung keputusan perekrutan karyawan pada PT Visionet

			Data Internasional dapat membantu perusahaan dalam menyediakan alternatif pengambilan keputusan saat perekrutan karyawan baru. Sistem bekerja dengan mengolah data calon karyawan baru berdasarkan kriteria dengan menentukan alternatif terbaik menggunakan metode TOPSIS.
--	--	--	---

Dan dari tabel 2.6 penelitian sejenis yang membedakan penelitian sendiri dengan penelitian lain yaitu kriteria yang digunakan antara lain :

1. Sistem akan dirancang secara *online* untuk pendaftaran calon karyawan baru dan menampilkan informasi kriteria apa saja yang akan dinilai sehingga calon karyawan dapat meningkatkan kemampuan untuk proses pemilihannya.
2. Kriteria yang digunakan, antara lain : jenjang pendidikan, nilai tes, pengalaman bekerja, kesehatan, umur, dan catatan hukum.
3. Kontent dari sistem pendukung keputusan seleksi penerimaan karyawan baru menggunakan metode *TOPSIS (Technique For Others Reference by Similarity to Ideal Solution)* yang dapat melakukan pengolahan data calon karyawan, data admin, data kriteria, data sub kriteria, data penilaian calon karyawan dan mencetak laporan hasil seleksi penerimaan karyawan baru.