# BAB II LANDASAN TEORI



## Penelitian Terdahulu

Dalam penulisan tugas akhir penulis mencoba mencari informasi dari hasil penelitian sebelumnya sebagai bahan perbandingan, baik mengenai kelebihan atau kekurangan dari peneliti sebelumnya. Selain itu, penulis juga mendapatkan inforasi dari buku maupun tugas akhiryang sudah ada, teori yang berkaitan dengan judul tugas akhir ini antara lain:

### **Penelitian Eko Darmanto, Noor Latifah dan Nanik Susanti (2015)**

Penelitian ini membahas tentang Penerapan Metode AHP *(ANALYTHIC HIERARCHY PROCESS)* Untuk Menentukan Kualitas Gula Tumbu.

Dari hasil penelitian dengan menggunakan metode AHP dapat disimpulkan Aplikasi SPK Menentukan Kualitas Gula Tumbu ini, sudah dapat melakukan perhitungan dengan metode AHP (Analytic Hierarchy Process) lebih cepat dibandingkan perhitungan secara manual sehingga bias lebih efisien dan tingkat keakuratan data sudah mendekati sempurna, dan Seluruh pendataan yang berhubungan dalam Menentukan Kualitas Gula Tumbu meliputi data warna, data rasa, data kekerasan dan data perhitungan metode dapat diolah seluruhnya di dalam Aplikasi SPK Menentukan Kualitas Gula Tumbu ini dan dapat terorganisir dengan baik.

### **Penelitian Bagus Prasetyo, Wawan Laksito Y.S. dan Sri Siswanti**

Penelitian ini membahas tentang Sistem Penunjang Keputusan Pemilihan Paket Internet Operator Telekomunikasi Dengan Metode AHP *(ANALYTICAL HIERARCHY PROCESS).*

Dari hasil penelitian dengan menggunakan metode AHP dapat disimpulkan ¹Metode AHP (Analytical Hierarchy Process) dapat diterapkan dalam sistem pendukung keputusan pemilihan paket internet operator telekomunikasi dengan menentukan prioritas utama dari beberapa kriteria serta alternatif yang ada untuk mengambil sebuah keputusan, dan ²Hasil perhitungan menggunakan AHP untuk menentukan prioritas pilihan sangat bergantung pada pemberian bobot nilai terhapdap kriteria dan sub kriteria yang ada pada tahap penilaian kriteria dan sub kriteria yang akan menghasilkan nilai prioritas.

### **Penelitian Bambang Yuwono, Frans Richard Kodong, Hendy Ayusta Yudha (2011)**

Penelitian ini membahas tentang Sistem Pendukung Keputusan menggunakan metode PROMETHEE (Studi Kasus : Stasiun Pengisian Bahan Bakar Umum).

Dari hasil penelitian dengan menggunakan metode PROMETHEE dapat disimpulkan hasil analisis, perancangan, dan implementasi, telah berhasil dibangun sebuah sistem pendukung keputusan sehingga dapat ditentukan keputusan yang objektif. Keputusan (Rangking atau prioritas) yang dihasilkan dari sistem pendukung keputusan untuk menentukan lokasi pembangunan stasiun pengisian bahan bakar umum dengan metode promethee bukanlah suatu keputusan (Rangking prioritas) yang mutlak dimana keputusan akhir tetap ditentukan sendiri oleh User. Jadi sistem pendukung keputusan merupakan suatu program bantu dalam mempertimbangkan suatu pengambilan keputusan.

### **Penelitian Wahid Azhari Fauzi (2017)**

Penelitian ini membahas tentang Sistem Pendukung Keputusan Pemilihan Jenis Kamera CCTV Menggunakan Metode PROMETHEE.

Dari hasil penelitian dengan menggunakan metode PROMETHEE dapat disimpulkan ¹Sistem pendukung keputusan pemilihan kamera CCTV dapat mempermudah dan mempercepat penyeleksian dalam pemilihan jenis kamera CCTV berdasarkan kriteria yang sudah ditetapkan, sehingga membantu para user yang masih awam tentang kamera CCTV, dan ²Aplikasi ini dapat diakses dimana saja, karena Sistem Pendukung Keputusan ini berbasis Web.

### **Penelitian Julianto Lemantara, Noor Akhmad Setiawan dan Marcus Nurtiantara Aji (2013)**

Penelitian ini membahas tentang Rancang Bangun Sistem Pendukung Keputusan Pemilihan Mahasiswa Berprestasi Menggunakan Metode AHP dan PROMETHEE.

Dari hasil penelitian dengan menggunakan metode AHP dan PROMETHEE dapat disimpulkan ¹Sistem yang dibuat dapat mempercepat proses pemilihan mahasiswa karena proses penilaian sudah dilakukan secara terkomputerisasi. Pada proses seleksi, kemahasiswaan tidak perlu memasukkan kembali nilai dari juri. Juri dapat langsung memasukkan nilai melalui aplikasi, ²Sistem yang dibuat dapat menghasilkan keputusan pemilihan mahasiswa yang lebih tepat dan objektif karena proses pemilihan dilakukan dengan metode ilmiah dan ³Hasil pemeringkatan dari kombinasi AHP dan Promethee terbukti lebih baik daripada hasil pemeringkatan dari metode AHP atau Promethee saja. Hal ini dikarenakan kelebihan dari masing-masing metode digunakan.

### **Penelitian Ahmad Abdul Chamida, Bayu Surarso dan Farikhin (2015)**

Penelitian ini membahas tentang Implementasi Metode AHP dan PROMETHEE Untuk Pemilihan Supplier.

Dari hasil penelitian dengan menggunakan metode AHP dan PROMETHEE dapat disimpulkan ¹Sistem ini dapat menghasilkan prioritas alternatif pemasok obat dari nilai Net Flow tertinggi. ²Semakin banyak penilai yang melakukan pembobotan kriteria akan menghasilkan bobot kriteria yang objektif. ³Sistem ini bekerja secara dinamis, pengguna dapat menambah kriteria yang diinginkan berapapun jumlahnya, serta dapat menambah alternatif berapapun jumlahnya. Pengguna dapat mengatur indikator setiap kriteria. Dan Sistem ini menyediakan fasilitas pembobotan kriteria secara manual, serta menyediakan pembobotan kriteria berdasarkan penilaian dari penilai yang profesional.

### **Penelitian Supriyadi dan Rinto Priambodo (2019)**

Penelitian ini membahas tentang Penentuan Klien Prioritas Dengan Metode ANALYTIC HIERARCHY PROCESS (AHP) DAN PROMETHEE DI PT XYZ.

Dari hasil penelitian dengan menggunakan metode AHP dan PROMETHEE dapat disimpulkan ¹Dengan adanya sistem pendukung keputusan penentuan klien prioritas dapat membantu Divisi Sales dalam menentukan klien prioritas yang sesuai dengan mempertimbangkan kriteria-kriteria yang sudah di tentukan. ² Dengan adanya sistem ini proses penentuan yang awalnya masih menggunakan manual dan membutuhkan waktu cukup lama, kini sudah terkomputerisasi hanya mengklik sesuai dengan periode tahun yang diinginkan sudah mendapatkan rekomendasi klien prioritas.

## Dasar Teori

Dalam dasar teori ini berisi tentang dasar-dasar yang akan dibuat sebagai acuan pembuatan laporan serta dasar-dasar dalam pembuatan aplikasi.

### **Tanaman Kedelai**

Menurut Salim (2013:11), Kedelai merupakan salah satu tanaman polong-polongan yang telah dibudidayakan sejak 3.500 tahun yang lalu di Asia Timur. Tanaman kedelai telah lama diusahakan di Indonesia sejak tahun 1970. Kedelai merupakan tanaman subtropis yang multiguna dan sudah sejak dahulu dimanfaatkan sebagai salah satu sumber pemenuhan kebutuhan protein nabati di berbagai negara, termasuk Indonesia (Warisno dan Dahana, 2010:2). Kedelai merupakan tanaman pangan jenis kacang-kacangan yang biasa diolah masyarakat menjadi berbagai bentuk pangan olahan. Menurut Warisno dan Dahana (2010:2), Di Indonesia, konsumsi kacang-kacangan menempati urutan ke-3 setelah padi-padian dan ikan. Sebagai bahan makanan kedelai banyak mengandung protein, lemak dan vitamin, sehingga tidak mengherankan bila kedelai mendapat julukan *gold from the soil* (emas yang muncul dari tanah). Berdasarkan warna kulitnya, kedelai dapat dibedakan atas kedelai putih, kedelai hitam, kedelai coklat dan kedelai hijau. Menurut Salim (2013:12) Kedelai yang umumnya dibudidayakan adalah spesies *Glycine max* (biji kedelai berwarna putih kekuningan) dan *Glycine soya* (biji kedelai berwarna hitam. Kedelai putih kekuningan umumnya digunakan sebagai bahan baku pembuatan tempe dan tahu. Sementara itu, kedelai hitam lebih banyak digunakan dalam pembuatan kecap.

### **Sistem Pendukung Keputusan (SPK)**

Konsep sistem pendukung keputusan (SPK) pertama kal dikenalkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System.* Sistem tersebut adalah suatu sistem yang berbasis komputer yang ditunjukan untuk membantu pengambilan keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur (Endratama, 2015).

*Decision Support System* (DSS) atau Sistem Pendukung Keputusan (SPK) adalah suatu sistem informasi berbasis komputer yang dapat menghasilkan berbagai alternatif keputusan untuk membantu permasalaha yang terstruktur ataupun tidak terstruktur (Putra, Andreswari, dan Susilo, 2015). Tujuan SPK adalah untuk mendukung pengambilan keputusan yang sifatnya memilih alternatif hasil pengolahan informasi dengan model-model pengambilan keputusan untuk menyelesaikan masalah yang terstruktur dan tidak terstruktur.

SPK digunakan untuk membantu pengambilan keputusan dalam memecahkan masalah. SPK dibuat untuk membantu masyarakat untuk dapat memilih hasil dari pelolahan informasi yang sesuai dengan metode yang digunakan.

### ***Analytical Hierarchy Process* (AHP)**

Metode *Analytical Hierarchy Process* (AHP) merupakan salah satu model untuk pengambilan keputusan yang dapat membantu kerangka berpikir manusia. Metode ini mula-mula dikembangkan oleh Thomas L. Saaty pada tahun 70-an. Dasar berpikirnya metode AHP adalah proses membentuk skor secara numerik untuk menyusun rangking setiap alternatif keputusan berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan (Sonatha, 2010). Metode *Analytical Hierarchy Process* (AHP) ini membantu memecahkan persoalan yang kompleks dengan menstruktur suatu hirarki kriteria, pihak yang berkepentingan, hasil dan dengan menarik berbagai pertimbangan guna mengembangkan bobot atau prioritas (Irawan, 2014).

#### **Prinsip Metode Analytical Hierarchy Process (AHP)**

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah (Kusrini, 2007) :

1. Membuat hierarki sistem yang kompleks bisa dipahami dengan memecahnya menjadi elemen-elemen pendukung, menyusun elemen secara hierarki dan menggabungkannya atau mensintesisnya.
2. Penilaian kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty, untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur menggunakan gambar tabel 2.1 analisis berikut.

**Tabel 2.1** Penilaian Kriteria

|  |  |
| --- | --- |
| **Intensitas Kepentingan** | **Keterangan** |
| 1 | Kedua Elemen sama pentingnya |
| 3 | Elemen yang satu sedikit lebih penting dari pada elemen yang lain |
| 5 | Elemen yang satu lebih penting dari pada elemen yang lainnya |
| 7 | Satu elemen jelas lebih mutlak penting dari pada elemen lainnya |
| 9 | Satu elemen mutlak penting dari pada elemen lainnya |
| 2,4,6,8 | Nilai-nilai antara dua nilai pertimbangan yang berdekatan |

1. *Synthesis of priority* (menentukan prioritas) Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwise Comparisons*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.
2. *Logical Consistency* (Konsistensi Logis) Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu.

#### **Langkah-langkah Metode *Analytical Hierarchy Process* (AHP)**

Metode *Analytical Hierarchy Process* (AHP) memiliki langkah-langkah, yaitu (Kusrini, 2007):

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hierarki dari permasalahan yang dihadapi. Penyusunan hierarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menetukan prioritas elemen
   * langkah pertama dalam menentukan prioritas elemen adalah membuat perbandingan pasangan, yaitu membandingkan elemen secara berpasangan sesuai kriteria yang diberikan.
   * matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresentasikan kepentingan relatif dari suatu elemen terhadap elemen lainnya.
3. Sintesis pertimbangan-pertimbangan terhadap perbandingan berpasangan di sintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah :
   * menjumlahkan nilai-nilai dari setiap kolom pada matriks
   * membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks
   * menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata
4. Mengukur konsistensi Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah :
   * kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya
   * jumlahkan setiap baris
   * hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan
   * jumlahkan hasil bagi di atas dengan banyaknya elemen yang ada, hasilnya disebut λ maks
5. Hitung *Consistency Index* (CI) dengan rumus :

CI = (λ maks – n)/ n **(2.1)**

dimana n = banyaknya elemen

1. Hitung Rasio Konsistensi/ *Consistency Ratio* (CR) dengan rumus :

CR = CI/ IR **(2.2)**

dimana CR = *Consistency Ratio*

CI = *Consistency Index*

IR = *Indeks Random Consistency*

1. Memeriksa konsistensi hierarki. Jika nilainya lebih dari 10%, maka penilaian data judgement harus diperbaiki. Namun jika rasio konsistensi (CI/ IR) kurang atau sama dengan 0,1 maka hasil perhitungan bisa dinyatakan benar. Berikut daftar *Indeks Random Consistency* (IR) pada tabel 2.2 berikut.

**Tabel 2.2** Daftar Indeks Random Consistency

|  |  |
| --- | --- |
| **Ukuran Matriks** | **Nilai IR** |
| 1,2 | 0,00 |
| 3 | 0,58 |
| 4 | 0,90 |
| 5 | 1,12 |
| 6 | 1,24 |
| 7 | 1,32 |
| 8 | 1,41 |
| 9 | 1,45 |
| 10 | 1,49 |
| 11 | 1,51 |
| 12 | 1,48 |
| 13 | 1,56 |
| 14 | 1,57 |
| 15 | 1,59 |

### **Preference Ranking Organization Method for Enrichment Evaluation (PROMETHEE**)

Metode Promethee merupakan salah satu metode yang dapat digunakan adalah menentukan urutan (prioritas) dalam analisis multikriteria. Metode Promethee pertama kali dikembangkan oleh JP.Brans dan dipublikasikan pada tahun 1982 pada sebuah konferensi yang diorganisasikan R.Nadeau dan M.Landry di Universitas Laval, Quebec Canada. Metode Promethee dapat dijalankan melalui beberapa tahap, yaitu (Brans & Mareschal, 2009):

1. Menentukan kriteria-kriteria yang akan digunakan beserta bobot dari masing-masing kriteria.
2. Menentukan semua alternatif yang ada.
3. Menentukan tipe preferensi untuk tiap-tiap kriteria secara tepat.Tipe preferensi yang digunakan dalam metode Promethee adalah fungsi keanggotaan himpunan fuzzy. Fungsi keanggotaan himpunan fuzzy memetakan setiap anggota himpunan domain ke anggota himpunan bilangan real yang memiliki interval dari 0 sampai dengan 1. Tipe preferensi ditentukan berdasarkan karakteristik dari kriteria tersebut. Ada enam bentuk tipe preferensi yang sering digunakan, yaitu:

**Tabel 2.3** Tipe dari Fungsi Preferensi Kriteria

|  |  |  |
| --- | --- | --- |
| **Tipe Preferensi Kriteria** | | **Parameter** |
| Kriteria Biasa (*Usual Criterion*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\1.png | - |
| Kriteria Quasi (*Quasi Criterion* atau *U-Shape*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\2.png | Q |
| Kriteria dengan preferensi Linier (*Criterion with Linear Preference* atau *V-Shape*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.png | P |
| Kriteria Level (*Level Criterion*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\5.png | q.p |
| Kriteria dengan preferensi Linier dan area yang tidak berbeda -- Linear Quasi (*Criterion with Linear Preference and Indifference Area*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\4.png | q.p |
| Kriteria gaussian (*Gaussian Criterion*) | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\6.png | σ |

1. Menghitung prefensi dari tiap-tiap kriteria Preferensi dari tiap – tiap kriteria dihitung berdasarkan perbandingan antara setiap pasang alternatif yaitu selisih antara nilai evaluasi dari dua buah alternatif terhadap kriteria tertentu. Nilai preferensi berkisar dari nol sampai satu. Preferensi bernilai nol apabila tidak ada perbedaan antara kedua alternatif yang dibandingkan. Preferensi akan bernilai satu apabila alternatif yang satu lebih baik dari alternatif lainnya.
2. Menghitung arah preferensi berdasarkan nilai indeks leaving flow dan entering flow. Untuk setiap alternatif, nilai leaving flow dapat dihitung menggunakan persamaan 2.3, sedangkan nilai entering flow dihitung dengan menggunakan persamaan 2.4.

**(2.3)**

**(2.4)**

Keterangan :

= leaving flow alternatif a.

= entering flow alternatif a.

n = banyaknya kriteria yang ada.

P(a, b) = preferensi dari dua buah alternatif a dan b, a, b Î A

A = himpunan alternatif yang ada

1. Net flow dihitung dengan menggunakan persamaan 2.5.

ф (a) = - **(2.5)**

Keterangan :

ф (a) = netflow.

= leaving flow.

= entering flow.

1. Urutkan alternatif berdasarkan net flow (rangking). Hasil net flow dari semua alternatif diurutkan dari yang nilai yang paling besar sampai dengan nilai terkecil. Alternatif yang terbaik adalah alternatif yang mempunyai nilai net flow terbesar.

### **Website**

Menurut Irawan (2009:180) “Website (situs) berbeda dengan Web Page, website merupakan kumpulan dari Web Page ”. Website atau di sebut juga dengan situs merupakan kumpulan halaman yang menampilkan informasi data, teks, gambar, animasi, suara, dan gabungan dari semuanya yang membentuk suatu rangkaian yang saling terkait dengan jaringan-jaringan halaman.

Ada dua macam website, yakni website statis dan website dinamis. Website statis merupakan website yang informasinya merupakan informasi satu arah, yakni hanya berasal dari pemilik software-nya saja. Umumnya website ini bersifat tetap, jarang berubah, dan hanya bisa di update oleh pemiliknya saja. Sementara website dinamis merupakan website yang mempunyai arus informasi dua arah, yakni berasal dari pengguna dan pemilik, sehingga aktivitas update dapat di lakukan oleh penggu na dan juga oleh pemilik website.

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

*Unified Modelling Language* (UML) merupakan alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi obyek (Munawar, 2005:18). Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi rancangan mereka dengan yang lain (Gushelmi, 2012)

#### **Use Case Diagram**

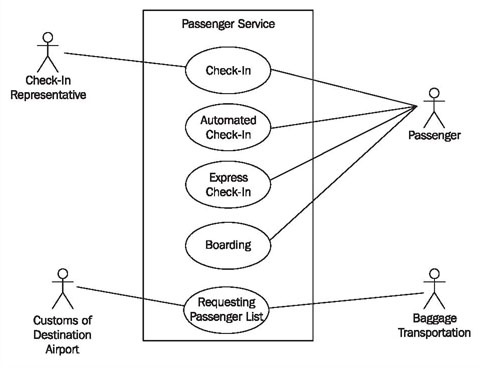
Menurut Rosa & Shalahudin (2014) *use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Untuk mendukung dalam pembuatan *Use Case*, berikut adalah penjelasan mengenai komponen yang terdapat pada *Use Case* dijelaskan pada tabel 2.4.

**Tabel 2.4** Komponen Pada Use Case Diagram

|  |  |  |
| --- | --- | --- |
| **No** | **Gambar** | **Keterangan** |
| 1 | C:\Users\Aries\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_1.png | Aktor : mewakili peran orang, system yang lain, atau alat ketika berkomunikasi dengan *use case.* |
| 2 |  | *Use case* : abstraksi dan interaksi antara sistem dengan aktor. |
| 3 |  | *Association* : abstraksi dari penghubung antara aktor dengan use case. |
| 4 |  | *Generalisasi* : menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan use case. |
| 5 | <<include>> | Menunjukkan bahwa suatu use case seluruhnya merupakan fungsionalitas dengan use case. |
| 6 | <<extend>> | Menunjukkan bahwa suatu use case merupakan tambahan fungsionallitas dari use case lainnya jika suatu kondisi terpenuhi. |

Contoh *use case diagram* dapat dilihat pada gambar 2.1, yang merupakan *use case diagram* aplikasi penerbangan.



**Gambar 2.1** Contoh Use Case Diagram

#### **Activity Diagram**

***Activity diagram* adalah teknik untuk menggambarkan logika procedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram hampir sama dengan sebuah aliran data, akan tetapi perbedaan prinsip antara activity diagram lebih mendukung kepada *behavior parallel*. (Rijal, 2010)**

**Sukamto dan Shalahuddin (2013:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”.**

***Activity diagram*** merupakan diagram yang menggambarkan tentang aktifitas pada suatu sistem. Dari pertama sampai akhir, diagram ini menunjukkan langkah apa saja yang terjadi dalam proses kerja sistem yang dibuat. Sehingga dengan menggunakan *activity* diagram dapat memudahkan dalam proses pembuatan sistem.

**Tabel 2.5** Simbol Activity Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Gambar** | **Nama** | **Keterangan** |
| 1 |  | *Actifity* | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | *Action* | State dari sistem yang mencerminkan eksekusi dari suatu aksi. |
| 3 |  | *Initial Node* | Bagaimana objek dibentuk atau diawali. |
| 4 |  | *Actifity Final*  *Node* | Bagaimana objek dibentuk dan dihancurkan . |
| 5 |  | *Fork Node* | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran. |

#### **Class Diagram**

Class diagram menggambarkan struktur kelas statis dalam sistem. Kelas merepresentasikan hal-hal yang ditangani oleh sistem. Kelas dapat dikaitkan satu sama lain dalam berbagai cara: asosiasi (terhubung satu sama lain), subordinasi (satu kelas ke kelas lain), khusus (satu kelas adalah spesialisasi dari yang lain) atau paket (digabungkan sebagai satu unit) ( Code Politan, 2019). Suatu sistem memiliki beberapa diagram kelas, dan diagram kelas elemen ditunjukkan pada Tabel 2.4 (pccontrol, 2013).

**Tabel 2.6** Tabel Simbol Class Diagram

|  |  |
| --- | --- |
| **Simbol** | **keterangan** |
| |  | | --- | | Nama *Class* | | Atribut | | Method |   Class | Bagian atas menunjukkan nama dari *class*, bagian tengah mengindikasikan atribut dari *class* dan bagian bawah mendefinisikan *method* dari sebuah *class*. |
| Directed association | Menunjukkan *relationship* atau hubungan antar *class*. |
| 0..1  Relationship Multiplicity | Menunjukkan jumlah suatu objek 0 atau 1 yang berhubungan dengan objek lain. |
| 1  Relationship Multiplicity | Menunjukkan jumlah suatu objek hanya satu yang berhubungan dengan objek lain. |
| 0..\*  Relationship Multiplicity | Menunjukkan jumlah suatu objek nol atau lebih yang bisa berhubung dengan objek lain. |
| 1..\*  Relationship Multiplicity | Menunjukkan jumlah suatu objek satu atau lebih yang bisa berhubungan dengan objek lain. |

### **Basis Data (*Database*)**

Basis data berasal dari 2 kata yaitu basis yang berarti tempat penyimpanan atau gudangsebagai tempat berkumpul atau menetap. Sedangkan data merupakan fakta yang diperoleh. Jadi basis data bisa dikatakan sebagai kumpulan data yang saling terkoneksi dan tersimpan secara berkelompok untuk dapat digunakan atau dimanfaatkan dengan cepat dan mudah untuk memenuhi suatu kebutuhan dan disimpan pada suatu media penyimpanan elektronik (Fathansyah,2012:2).

Basis data (Database) adalah sebuah kumpulan dari *files*yang saling berkaitan. (Whitten & Bently, 2007).Teknologi *database* memiliki kelebihan dalam cara penyimpanan data dengan format yang fleksibel. Hal ini memungkinkan dikarenakan *database* terpisah dari sistem informasi dan aplikasi yang menggunakannya. Database juga dapat berkembang sesuai dengan perubahan kebutuhan organisasi maupun perkembangan lainnya.

Whitten juga menjelaskan bahwa komponen pembentuk atau penunjang dari *Database* adalah sebagai berikut :

1. Field

*field* adalah bagian paling kecil dari data yang memiliki arti atau makna didalam sebuah *file* atau *database*.

1. Records

*Records* adalah sebuah kumpulan dari *field-field* yang tersusun dan terikat berdasarkan format tertentu yang sudah ditentukan sebelumnya.

1. File atau Tables

*File* dalam konteks *database* dalah himpunan semua kejadian yang terjadi pada struktur *record*. *Tables* adalah *database* yang mempunyai relasi setara dengan sebuah *files*.

#### **Istilah Umum Basis Data**

Berikut beberapa istilah dalam basis data yang perlu diketahui diantaranya sebagai berikut :

1. *Entity*

*Entity* adalah objek dalam enterprise yang akan disajikan di dalam database. Contoh : supplier, gudang, pegawai, part, dll.

1. Atribut (elemen data).

Setiap *entity* memiliki atribut-atribut yang mencirikan *entity* tersebut. Contoh: entity pegawai terdiri atas atribut : kode pegawai, alamat, dll.

1. Relasi

*Relasi* adalah suatu asosiasi hubungan antara dua atau lebih *entity*. Contoh : mahasiswa mengambil mata kuliah, pasien dirawat dokter, dan lain-lain.Dalam hubungan antar *entity* dikenal ada dua kunci data penghubung, yaitu :

1. *Primary Key* (Kunci Utama)

Satu attribute *field* atau satu set attribute yang mengidentifikasi secara unik suatu kejadian yang spesifik pada *entity*.

1. *Foreign Key* (Kunci Tamu)

Satu *attribute* atau satu set attribute yang melengkapi suatu hubungan yang menunjukkan ke entity induknya. Kunci tamu berada pada *entity* anak.

Setiap proyek dalam sistem database dapat dipresentasikan dengan menggunakan beberapa *entity* dan setiap *entity* dapat terdiri dari beberapa elemen data (atribut). Dalam database model relasi dikenal beberapa tipe relasi antara entity dan relasi antara elemen data. Tipe relasi tersebut yaitu:

1. Relasi satu ke satu (*One to One*)

Menggambarkan hubungan satu ke satu, yaitu satu record pada entity pertama berhubungan dengan satu *record* pada *entity* kedua atau sebaliknya. Misalnya setiap pasien yang dirawat akan menempati satu tempat tidur untuk satu periode perawatan.

1. Relasi satu ke banyak (*One to Many*)

Menggambarkan hubungan satu ke banyak, yaitu satu *record* pada entity pertama berhubungan dengan banyak *record* pada *entity* kedua atau sebaliknya. Misalnya antara *file* data karyawan dengan *file* pendapatan hariannya. Seorang karyawan memepunya beberapa *record* data pendapatan hariannya, yang keseluruhannya menunjukkan gaji karyawan tersebut pada periode tertentu.

1. Relasi banyak ke banyak (*Many to Many*)

Menggambarkan hubungan banyak ke banyak, yaitu lebih dari satu *record* pada entity pertama berhubungan dengan lebih dari satu *record* pada *entity* kedua atau sebaliknya. Dibutuhkan *associativeentity* atau *entity* perantara yang berisi *field* kunci dari masing-masing *entity*.

### **MySQL**

MySQL adalah bagian dari perangkat lunak RDBMS untuk mengolah database dengan cepat, juga dapat menyimpan data dalam jumlah besar, sehingga bisa diakses banyak pengguna, sehingga dapat melakukan proses secara teratur dan bersamaan.

Adapun beberapa alasan mengapa menggunakan MySQL sebagai server *database* untuk aplikasi yang akan dikembangkan ini (Raharjo,2011):

1. Fleksibel

MySQL dapat digunakan sebagai pengembangan aplikasi dekstop maupun website dengan menggunakan teknologi yang bervariasi.

1. Performa Tinggi

MySQL memiliki mesin *query* dengan performa yang tinggi, dengan begitu proses transaksional dapat dilakukan dengan cara cepat.

1. Gratis

MySQL dapat digunakan secara gratis dan juga ada yang bersifat komersial. Bersifat komersial biasanya telah ditambahi dengan kemampuan spesifik dan mendapat pelayanan dari *technicial support.*

1. Proteksi data yang handal

MySQL menyediakan mekanisme yang *powerfull* untuk menangani tentang keamanan data dengan menyediakan fasilitas manajemen *user*, enkripsi data, dll.

1. Komunitas Luas

Pengguna MySQL sangatlah banyak dan memilik komunitas yang luas. Hal ini dapat berguna jika terjadi suatu permasalahan dalam pengolahan data menggunakan MySQL. Dengan mengikuti komunitas tertentu, maka dapat menanyakan atau mendiskusikan permasalahan tersebut melalui forum.

### ***Perl Hypertext Preprocessor (PHP)***

Menurut Arief (2011), PHP merupakan bahasa *server side scripting* yang menyatu dengan HTML. *Server side scripting* merupakan sebuah teknologi *scripting* atau pemograman *web* dimana *scrip* dikompilasi atau diterjemahkan di*server* yang memunkinkan untuk menghasilkan halaman *web* yang dinamis. Karena PHP merupakan *server side scripting* maka kode dan perintah-perintah PHP akan dieksekusi di*server* kemudian hasilnya akan dikirimkan ke browser dengan format HTML. Contoh script PHP :

<?php

Echo “Halo nama saya Alief”;

?>

### **Framework**

Framework atau dalam bahasa indonesia dapat diartikan sebagai “kerangka kerja” merupakan kumpulan dari fungsi-fungsi atau prosedur-prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga mempermudah dan mempercepat pekerjaan seorang programer, tanpa harus membuat fungsi atau class dari awal.

#### ***Codeigniter***

*Codeigniter* merupakan framework php yang berjalan pada versi php 4, dan php 5. Tujuan utama dari *CodeIgniter* adalah untuk memudahkan dan mempercepat pekerjaan seorang programmer dalam mengembangkan aplikasi secara cepat tanpa harus melakukan pemrograman dari awal (Saputra, 2009).