

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

Pengertian Sistem Menurut para ahli :

Menurut Romney (2015) sistem adalah serangkaian dua atau lebih komponen yang saling terkait dan berinteraksi untuk mencapai suatu tujuan.

Menurut Mulyadi (2016) Sistem pada dasarnya adalah sekelompok unsur yang erat berhubungan satu dengan lainnya, yang berfungsi bersama-sama untuk mencapai tujuan tertentu.

Menurut Febriani O.M (2018) Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu.

2.2 Metode Pengembangan Sistem Pendukung Keputusan

Menurut Simon (1960) pada buku Teori Dan Teknik Pengambilan Keputusan Kualitatif Dan Kuantitatif (2016) menguraikan proses pengambilan keputusan berlangsung melalui 4 tahap yaitu:

1. *Intelligence*

Intelligence adalah proses pengumpulan informasi yang bertujuan mengidentifikasi permasalahan.

2. *Design*

Design adalah tahap perencanaan solusi terhadap masalah. Biasanya pada tahap ini dikaji berbagai macam alternatif pemecahan masalah.

3. *Choice*

Choice adalah tahap mengkaji kelebihan dan kekurangan dari berbagai macam alternatif yang ada dan memilih yang terbaik.

4. Implementasi

Implementasi adalah tahap pengambilan keputusan dan melaksanakannya.

2.3 Definisi Sistem Pendukung Keputusan

Sistem pendukung keputusan dapat didefinisikan sebagai sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan tidak terstruktur. SPK dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka. Selain itu juga sistem pendukung keputusan ditujukan untuk keputusan-keputusan yang memerlukan penilaian atau pada keputusan-keputusan yang sama sekali tidak dapat didukung oleh algoritma.

2.4 TOPSIS (*Technique For Others Reference by Similarity to Ideal Solution*)

TOPSIS adalah metode multi kriteria yang digunakan untuk mengidentifikasi solusi dari himpunan alternatif berdasarkan minimalisasi simultan dari jarak titik ideal dan memaksimalkan jarak dari titik terendah. TOPSIS dapat menggabungkan bobot relatif dari kriteria penting. TOPSIS memiliki konsep yang sederhana dan mudah dipahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana. Metode ini banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis.

2.5 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemrograman dan perangkat lunak pendukung yang digunakan dalam penulisan ini adalah antara lain *Website, XAMPP, PHP, HTML, CSS, Javasript*

2.5.1 Website

Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa *text*,

gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Abdulloh, 2015).

2.5.2 HTML

HTML singkatan dari *Hypertext Markup Language*, yaitu script yang berupa tag-tag untuk membuat dan mengatur struktur website. (Rohi Abdulloh, 2015).

Beberapa tugas HTML dalam membangun website diantaranya, sebagai berikut :

- a. Menentukan layout website.
- b. Memformat text dasar seperti pengaturan paragraph, dan format font.
- c. Membuat list dan formulir.
- d. Membuat tabel, gambar, video, audio, dan link

2.5.3 CSS

Cascading Style Sheets yaitu skrip yang digunakan untuk mengatur desain *website*.”Walaupun HTML mempunyai kemampuan untuk mengatur tampilan *website*, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur *website* yang dibuat dengan HTML terlihat lebih indah. (Rohi Abdulloh, 2015).

2.5.4 javascript

Javascript merupakan bahasa pemrograman web yang pemrosesanya dilakukan disisi client. Kerna berjalan disisi client, javascript dapat dijalankan hanya dengan menggunakan browser. Berbeda dengan PHP yang bekerja disisi server, untuk menjalankan skrip javascript tidak memerlukan referesh pada *browser*.

Java script biasanya dijalankan ketika ada event tertentu yang terjad pada halaman web. Baik event yang dilakukan oleh user, maupun event yang terjadi karena adanya perubahan pada halaman *website*. (Rohi Abdullah,2018)




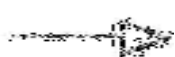


2.6 UML (*Unified Modeling Language*)


UML (*Unified Model Language*) merupakan alat bantu yang sangat handal didunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku. Mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain. (Munawar,2018).

2.6.1 Use Case Diagram

Menurut Rosa dan Shalahuddin (2015) *Use case* atau *diagram use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut Aktor dan *use case*.

Tabel 2.1 Use Case Diagram



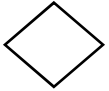



| Simbol | Nama | Keterangan |
|---|----------------------------|---|
|  | Actor | Mewakili peran orang, sistem yang lain, atau alat ketika berhubungan dengan use case. |
|  | Use case | Abstraksi dari penghubung antara aktor dengan use case. |
|  | association | Abstraksi dari penghubung antara aktor dengan use case. |
|  | Generalization | Menunjukkan spesialis actor untuk dapat berpartisipasi dengan use case. |
|  | Relalization | Hubungan elemen yang ada di bagian tanda panah akan merealisasikan pernyataan elemen yang ada pada bagian tanda panah. |
|  | Ekstensi/ <i>Extend</i> | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walau tanpa use case tambahan memiliki |

| | | |
|---|--------------------------------|--|
| | | nama depan yang sama dengan use case yang di tambahkan. |
|  | Menggunakan/ <i>Include</i> | Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsional atau sebagai syarat dijalankan use case ini. |

2.6.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak (Rosa dan Shalahuddin, 2015).

Tabel 2.2 Activity Diagram

| Simbol | Nama | Keterangan |
|---|-----------------|--|
|  | Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
|  | Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
|  | Percabangan | Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu. |
|  | Penggabungan | Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu. |
|  | <i>Swimlane</i> | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |
|  | Status akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir. |

2.6.3 Class Diagram


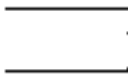

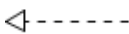
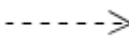
Menurut Rosa dan Shalahuddin (2015) diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan

dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut pola dan metode atau operasi ;

- Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas
- Operasi atau metode adalah fungsi yang dimiliki oleh suatu kelas

Berikut adalah simbol-simbol yang ada pada diagram kelas :

Tabel 2.3 Class Diagram

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-------------------------|---|
| 1 | | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 2 |  | <i>Nary Association</i> | Upaya untuk menghindari asosiasi dengan lebih dari 2 objek. |
| 3 |  | <i>Class</i> | Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama. |
| 4 |  | <i>Collaboration</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 5 |  | <i>Realization</i> | Operasi yang benar-benar dilakukan oleh suatu objek. |
| 6 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri |
| 7 | | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya |

2.7 Tahapan Metode TOPSIS

1. Membuat matriks keputusan yang ternormalisasi.

Sebelum membuat matriks keputusan ternormalisasi terlebih dahulu harus mengetahui bobot preferensi (W) yang diambil dari penilai setiap alternatif terhadap setiap kriteria.

Setelah itu, membuat matriks keputusan yang ternormalisasi, dengan rumus dibawah ini:

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

dengan : $i=1,2,\dots,m$; dan $j=1,2,\dots,n$

keterangan

r_{ij} = rating kinerja ternormalisasi dari alternatif pada setiap kriteria.

x_{ij} = nilai setiap alternatif pada setiap

$$\sqrt{\sum_{i=1}^m x_{ij}^2}$$

Dimana i = baris dan j = kolom

kemudian setelah mendapatkan hasil matriks keputusan ternormalisasi dengan lambang (R) dilanjutkan pada langkah ke 2.

2. Membuat matriks keputusan yang ternormalisasi terbobot. Dengan rumus berikut ini:

$$Y_{ij} = W_{ij} \times R_{ij} \quad (\text{Persamaan 2.2})$$

Dimana

y_{ij} = rating bobot ternormalisasi

w_i = nilai setiap bobot preferensi

r_{ij} = nilai dari matriks ternormalisasi

Setelah mendapatkan nilai matriks yang ternormalisasi terbobot dengan lambang (Y) melanjutkan kelangkah 3.

3. Menentukan ideal positif dan negatif.

Pada tahap ini setiap kriteria akan di ambil nilai yang paling terbesar dan nilai yang terkecil.

4. Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif dan matriks solusi ideal negatif.

Berikut ini adalah rumus untuk menentukan jarak antara alternatif dengan solusi ideal positif :

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_{ij}^+ - y_j^+)^2} \quad ; i=1, 2, \dots, m \quad (\text{Persamaan 2.3})$$

Kemudian untuk menentukan jarak antara alternatif dengan solusi ideal negatif, diberikan rumus:

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_j^-)^2} \quad ; i=1, 2, \dots, m \quad (\text{Persamaan 2.3})$$

Dimana untuk menentukan jarak antara alternatif dengan solusi ideal negatif ini dilakukan perhitungan jarak antara nilai terbobot setiap alternatif terhadap solusi ideal negatif yang dihasilkan dari langkah sebelumnya, yaitu dari hasil perhitungan solusi ideal negatif (A^-).

Selanjutnya untuk langkah yang terakhir yaitu Menentukan nilai preferensi untuk setiap alternatif.

5. Menentukan nilai preferensi untuk setiap alternatif. Rumus menentukan nilai preferensi untuk setiap alternatif, sebagai berikut:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad ; i=1, 2, \dots, m \quad (\text{Persamaan 2.4})$$

Dimana

V_i = nilai preferensi

D_i^+ = jarak alternatif dengan solusi ideal positif

D_i^- = jarak alternatif dengan solusi ideal negatif