

BAB II

LANDASAN TEORI

2.1 *Inventory* (Persediaan)

Persediaan merupakan barang atau bahan yang disimpan dan akan digunakan pada saat tertentu dan dengan tujuan tertentu. Misalnya untuk proses produksi, untuk dijual kembali atau sebagai cadangan dari peralatan yang digunakan. Untuk lebih jelasnya berikut adalah beberapa penjelasan mengenai persediaan menurut para ahli : Dalam bukunya Sofyan Assauri (2004;169) berpendapat, bahwa : Persediaan merupakan sejumlah bahan-bahan, parts yang disediakan dan bahan-bahan dalam proses yang terdapat dalam perusahaan untuk proses produksi, serta barang-barang jadi/produk yang disediakan untuk memenuhi permintaan dari komponen atau langganan setiap waktu. Hani Handoko mengemukakan bahwa persediaan atau *inventory* adalah sebuah istilah yang digunakan untuk menunjukkan segala sesuatu atau sumber daya yang disimpan sebagai antisipasi pemenuhan terhadap munculnya permintaan (Hani Handoko, 2000;333).

Secara sederhana diungkapkan bahwa "*Inventory* atau persediaan merupakan simpanan material yang berupa bahan mentah, barang dalam proses dan barang jadi." (Lalu Sumayang, 2003;189) Menurut Warren (2005;452) menyatakan "Persediaan (*inventory*) digunakan untuk mengindikasikan (1) barang dagang yang disimpan untuk kemudian dijual dalam bisnis perusahaan dan (2) bahan yang digunakan dalam proses produksi atau yang disimpan untuk tujuan itu" Berdasarkan definisi yang dikemukakan para ahli tersebut dapat ditarik kesimpulan bahwa persediaan

merupakan simpanan barang yang digunakan untuk memenuhi permintaan para konsumen.

2.2 TOPSIS (*Technique For Orders Reference by Similarity to Ideal*)

Solution)

Metode ini adalah salah satu metode pengambilan keputusan multikriteria yang pertama kali diperkenalkan oleh yoon dan hwang pada tahun 1981, metode ini salah satu metode yang banyak di gunakan untuk menyelesaikan pengambilan keputusan secara praktis. TOPSIS memiliki konsep dimana alternatif yang terpilih merupakan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan jarak terjauh dari solusi ideal negatif. Semakin banyak faktor yang harus di pertimbangkan dalam proses pengambilan keputusan maka semakin relatif sulit juga untuk mengambil keputusan terhadap suatu permasalahan. Apalagi jika upaya pengambilan keputusan dari suatu permasalahan tertentu, selain mempertimbangkan berbagai faktor atau kriterianya yang beragam, juga melibatkan beberapa orang pengambil keputusan. Permasalahan demikian juga di kenal dengan permasalahan *multipel criteeria decison making* (MCDM). Metode TOPSIS di gunakan sebagai suatu upaya untuk menyelesaikan permasalahan *multipel Criteeria decison making*. Hal ini di sebabkan konsepnya sederhana dan mudah di pahami, komputasinya efisien dan memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan.

Langkah-langkah Metode TOPSIS

Langkah-langkah yang dilakukan dalam menyelesaikan suatu permasalahan menggunakan metode TOPSIS adalah sebagai berikut:

- g. Menggambarkan alternatif (m) dan kriteria (n) ke dalam sebuah matriks, dimana X_{ij} adalah pengukuran pilihan dari alternatif ke- i dan kriteria ke- j . Matriks ini dapat dilihat pada persamaan satu.

$$D = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{13} \\ X_{21} & X_{22} & \dots & X_{23} \\ \vdots & \vdots & \ddots & \vdots \\ X_{i1} & X_{i2} & \dots & X_{i3} \end{bmatrix} \quad (1)$$

- h. Membuat matriks R yaitu matriks keputusan ternormalisasi
Setiap normalisasi dari nilai r_{ij} dapat dilakukan dengan perhitungan

menggunakan

persamaan

dua.

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^m X_{ij}}} \quad (2)$$

- i. Membuat pembobotan pada matriks yang telah dinormalisasi Setelah dinormalisasi, setiap kolom pada matriks R dikalikan dengan bobot bobot (w_j) untuk menghasilkan matriks pada persamaan tiga.

$$D = \begin{bmatrix} W_1 r_{11} & W_1 r_{12} & \dots & W_n r_{1n} \\ W_2 r_{21} & \dots & \dots & \dots \\ W_j r_{m1} & W_j r_{m2} & \dots & W_j r_{mm} \end{bmatrix} \quad (3)$$

- j. Menentukan nilai solusi ideal positif dan solusi ideal negatif. Solusi ideal dinotasikan A^+ , sedangkan solusi ideal negatif dinotasikan A^- . Persamaan untuk menentukan solusi ideal dapat dilihat pada persamaan empat.

$$A^+ = \{(\max V_{ij} \mid j \in J), (\min V_{ij} \mid j \in J')\}, \\ i = 1, 2, 3, \dots, m \} = \{V_1^+, V_2^+, \dots, V_n^+\}$$

$$A^- = \{(\max V_{ij} \mid j \in J), (\min V_{ij} \mid j \in J')\}, \\ i = 1, 2, 3, \dots, m \} = \{V_1^-, V_2^-, \dots, V_n^-\}$$

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

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

- k. Menghitung separation measure. Separation measure ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif.

– Perhitungan solusi ideal positif dapat dilihat pada persamaan lima :

$$S_i^+ = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^+)^2} \quad (5)$$

Dengan $i=1,2,3,\dots,m$

– Perhitungan solusi ideal negatif dapat dilihat pada persamaan

$$S_i^- = \sqrt{\sum_{j=1}^n (V_{ij} - V_j^-)^2} \quad (6)$$

Dengan $i=1,2,3,\dots,m$

enam :

1. Menghitung nilai preferensi untuk setiap alternatif. Untuk menentukan ranking tiap-tiap alternatif yang ada maka perlu dihitung terlebih dahulu nilai preferensi dari tiap alternatif. Perhitungan nilai preferensi dapat dilihat melalui persamaan tujuh.

$$C_i^+ = \frac{S_i^-}{S_i^+ + S_i^-} \quad (7)$$

Dimana $0 < C_i^+ < 1$ dan $i=1,2,3,\dots,m$

Setelah didapat nilai C_i^+ , maka alternatif dapat diranking berdasarkan

urutan C_i^+ . Dari hasil perankingan ini dapat dilihat alternatif terbaik yaitu alternatif yang memiliki jarak terpendek dari solusi ideal dan berjarak terjauh dari solusi ideal negatif.

2.3 Pemodelan Objek

Pemodelan objek merupakan suatu metode untuk menggambarkan struktur sistem yang memperlihatkan semua objek yang ada pada sistem. (Nugroho, 2005, hal:37)

2.3.1 Definisi UML

“*Unified Modeling Language (UML)* adalah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek (Rosa A.S,2013).

2.3.2 Diagram-diagram dalam UML

Asosiasi bisa digunakan untuk memodelkan relasi diantara obyek-obyek. Nama dari asosiasi memberi informasi secara tepat tentang relasi apa yang sedang dimodelkan. “Di *UML*, ada relasi dengan perlakuan khusus yang disebut dengan “bagian dari (*part of*)” yang menangani antar obyek-obyek dimana salah satunya adalah bagian dari yang lain” (Rosa, dkk: 2011).

Berikut ini adalah beberapa jenis diagram yang digunakan dalam pembuatan diagram *UML (Unified Modeling Language)*:

a. *Use Case Diagram*

“*Use Case* adalah deskripsi sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat” (Rosa, dkk: 2011). *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *Scenario*. Setiap *Scenario* mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan *use case*

adalah serangkaian *scenario* yang digabungkan bersama-sama oleh tujuan umum pengguna.

Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami (Rosa, dkk: 2011). Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

a) *Aktor* merupakan orang, proses, atau sistem lain yang

berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan di buat itu sendiri.

b) *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *use case diagram*, dapat kita lihat pada tabel 2.1 berikut ini :

Tabel 2.1 Simbol-simbol *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu <i>actor</i> .
2		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
3		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
4		<< <i>extend</i> >>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber

			pada suatu titik yang diberikan.
5		<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>).
6		<< <i>include</i> >>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .

Sumber: Rosa,dkk (2011)

b. Activity Diagram

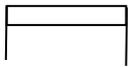
Activity diagram menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis (Rosa, dkk: 2011). Yang perlu diperhatikan disini adalah bahwa *diagram activity* menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *activity diagram*, dapat kita lihat pada tabel 2.2 berikut ini

:

Tabel 2.2 Simbol-simbol *Activity Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		Status Awal	Bagaimana objek dibentuk atau diawali.
2		<i>Aktivitas</i>	Aktifitas yang dilakukan sistem, aktifitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktifitas lebih dari satu.

4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.
5		<i>Status Akhir</i>	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki akhir.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang ada

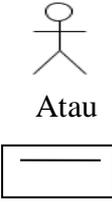
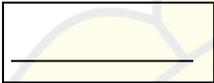
Sumber: Rosa,dkk (2011)

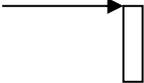
c. Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario* yang diterapkan pada sistem untuk model penerapan pada bahasa pemrograman.

Berikut ini simbol-simbol yang sering digunakan pada saat pembuatan *sequence diagram*, dapat kita lihat pada tabel 2.5 berikut ini :

Tabel 2.3 Simbol-simbol *Sequence Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1	 <p>Atau</p>	Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem.
2	Lifeline	Menyatakan kehidupan suatu objek
3		Objek	Merupakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan beriteraksi pesan.
5		Pesan tipe <i>create</i>	Menyatakan suatu objek memuat objek yang lain, arah panah mengarah pada objek yang dibuat.
6		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil oprasi/metode yang ada pada objek lain atau sendirinya sendiri.
7		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data /masukan/informasi ke objek lainnya.

8		Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembali ke objek tertentu.
9		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek yang lain.

Sumber: Rosa,dkk (2011)

2.4 Pengertian Aplikasi

Menurut Nazrudin Safaat H (2012 : 9) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna.

Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite).

2.5 Pengertian Sistem Informasi

Menurut Laudon dan Laudon (2010, p46) sistem informasi merupakan komponen yang saling bekerja sama untuk mengumpulkan, mengolah, menyimpan dan menyebarkan informasi untuk mendukung

pengambilan keputusan, koordinasi, pengendalian, analisis masalah dan visualisasi dalam sebuah organisasi.

Aktifitas dasar dari Sistem Informasi menurut Laudon dan Laudon (2010, p46-47) adalah sebagai berikut :

1. *Input*

Melibatkan pengumpulan data mentah dari dalam organisasi atau dari lingkungan eksternal untuk pengolahan dalam suatu sistem informasi.

2. *Process*

Melibatkan proses mengkonversi input mentah ke bentuk yang lebih bermakna.

3. *Output*

Mentransfer proses informasi kepada orang yang akan menggunakannya atau kepada aktivitas yang akan digunakan.

4. *Feedback*

Output yang di kembalikan ke anggota organisasi yang sesuai untuk kemudian membantu mengevaluasi atau mengkoreksi tahap Input.

2.6 Perangkat lunak yang digunakan dalam membuat aplikasi

2.6.1 Website

Menurut (Yeni Kustiyahningsih, 2011) *website* (situs web) adalah kumpulan dari beberapa halaman web. *Home page* sering juga disebut dengan *website*. Untuk bisa mengakses informasi yang terdapat didalam home page diperlukan suatu software tool yang disebut web browser untuk membaca *HTML* yang terdapat dalam suatu komputer.

Web terdiri atas beberapa jenis yaitu sebagai berikut :

1. Komersial, merupakan jenis web yang berekstensi .com dan .co.id
2. Pendidikan, merupakan jenis web yang berekstensi .edu.id dan .ac.id

3. Pemerintahan, merupakan jenis web yang berekstensi .go
4. Organisasi, merupakan jenis web yang berekstensi .go.id, .or. Id dan .ac.id

Website merupakan salah satu layanan yang didapat oleh komputer yang terhubung ke internet dengan fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan multimedia lainnya.

2.6.2 HTML

Menurut (Yeni Kustiyahningsih, 2011) *HTML* adalah *HyperText Markup Language*, yaitu suatu format data yang digunakan untuk membuat dokumen hypertext yang dapat dibaca dari satu platform computer keplatform computer lainnya, tanpa perlu melakukan suatu perubahan apapun. *HTML* tidak hanya mampu menampilkan teks tapi juga dapat menampilkan format – format lain dari teks tersebut, misalnya table, list, form, frame serta dapat digabungkan dengan objek suara, gambar, maupun video.

Dokumen yang berisi script HTML merupakan dokumen yang disajikan dalam bentuk *website*. Dokumen HTML disebut *markup language* karena mengandung tanda-tanda tertentu yang digunakan untuk menentukan tampilan suatu teks dan tingkat kepentingan dari teks tersebut dalam suatu dokumen. Kita dapat menentukan baris-baris mana yang merupakan judul menentukan gambar yang harus tampil, pengaturan format teks dan lain sebagainya.

2.6.3 CSS

Menurut (Yeni Kustiyahningsih, 2011) menyatakan bahwa *CCS* (*Cascading Style Sheet*) adalah kumpulan kode-kode yang berurutan

dan saling berhubungan untuk mengatur format atau tampilan suatu halaman HTML.

Cascading Style Sheet (CSS) merupakan salah satu bahasa pemrograman web untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Sama halnya styles dalam aplikasi pengolahan kata seperti Microsoft Word yang dapat mengatur beberapa style, misalnya heading, subbab, bodytext, footer, images, dan style lainnya untuk dapat digunakan bersama-sama dalam beberapa berkas (file). Pada umumnya CSS dipakai untuk memformat tampilan web yang dibuat dengan bahasa HTML dan XHTML.

CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

Ada dua sifat CSS yaitu internal dan eksternal. Jika internal yang dipilih, maka skrip itu dimasukkan secara langsung ke halaman website yang akan didesain. Kalau halaman web yang lain akan didesain dengan model yang sama, maka skrip CSS itu harus dimasukkan lagi ke dalam halaman web yang lain itu. Sifat yang kedua adalah eksternal dimana skrip CSS dipisahkan dan diletakkan dalam

berkas khusus. Nanti cukup gunakan semacam tautan menuju berkas CSS itu jika halaman web yang didesain akan dibuat seperti model yang ada di skrip tersebut.

2.6.4 PHP

Menurut (Budi Raharjo, 2012) dalam Buku “Modul Pemrograman Web (HTML, PHP, & MYSQL)”. PHP adalah salah satu pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari web browser, program yang ditulis dengan PHP akan di-parsing didalamweb server oleh interpreter PHP dan diterjemahkan dalam dokumen HTML, yang selanjutnya akan ditampilkan kembali ke web browser. Karena pemrosesan program PHP dilakukan dilingkungan web server, PHP dikatakan sebagai bahasa sisi server (server-side) oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “view source” pada web browser yang mereka gunakan. Selain menggunakan PHP, aplikasi web juga dapat dibangun dengan Java (JSP- JavaServer Pages dan Servlet), Perl, maupun ASP (Active Server Pages).

Syntax Program PHP adalah bahasa yang dirancang untuk mudah diletakkan dalam kode HTML. Banyak dijumpai kode PHP yang menyatu dengan kode HTML. Kode PHP diawali dengan tag<?php dan diakhiri dengan tag ?>. apabila kita melakukan konfigurasi terhadap file PHP ini untuk mengizinkan pengguna tag pendek (short tag) dengan mengubah nilai short_open_tag menjadi on,

maka tag tersebut dapat digantikan dengan `<? dan ?>`. dalam PHP, nilai default dari `short_open_tag` adalah `off`. Selain itu, PHP kita juga dapat menggunakan tag gaya ASP `<% dan %>`, dengan mengubah nilai `asp_tags` dalam file PHP ini menjadi `on`. Berikut ini contoh kode PHP :

```
<? Php  
  
echo "Hello World";  
  
?>
```

Perintah `echo` di dalam PHP berguna untuk mencetak nilai, baik teks maupun numerik ke layar web browser.

2.6.5 JavaScript

Menurut (R.H. Sianipar, 2015) javascript adalah bahasa scripting yang populer di internet dan dapat bekerja di sebagian besar browser populer seperti *Internet Explorer*, *Mozilla Firefox*, *Netscape* dan *Opera*. Kode javascript dapat disisipkan dalam halaman web menggunakan tag `script`.

Berikut ini beberapa sifat dari javascript:

1. Menambahkan interaktivitas ke halaman HTML.
2. Merupakan bahasa pemrograman scripting.
3. Bahasa Scripting merupakan bahasa yang ringan.
4. Javascript merupakan bahasa terinterpretasi.

2.6.6 Bootstrap

Menurut (Husein Alatas, 2013) bootstrap merupakan *Framework* ataupun *Tools* untuk membuat aplikasi web ataupun situs web responsive secara cepat, mudah dan gratis.

Bootstrap terdiri dari CSS dan HTML untuk menghasilkan *Grid, Layout, Typography, Table, Form, Navigation*, dan lain-lain. Di dalam Bootstrap juga sudah terdapat jQuery plugins untuk menghasilkan komponen UI yang cantik seperti *Transitions, Modal, Dropdown, Scrollspy, Tooltip, Tab, Popover, Alert, Button, Carousel* dan lain-lain.

Dengan bantuan Bootstrap, kita bisa membuat *responsive website* dengan cepat dan mudah dan dapat berjalan sempurna pada browser-browser populer seperti *Chrome, Firefox, Safari, Opera dan Internet Explorer*

2.6.7 JQuery

Menurut (Lukmanul Hakim, 2010) dalam buku *Bikin Website Super Keren dengan PHP & jQuery*, jQuery adalah Javascript Library berupa kumpulan kode/fungsi javascript siap pakai, sehingga mempermudah dan mempercepat kita dalam membuat kode javascript. Secara standar, apabila kita membuat kode javascript, maka diperlukan kode yang cukup panjang, bahkan terkadang sangat sulit dipahami, jQuery digunakan untuk menyederhanakan kode javascript. Hal ini sesuai dengan slogannya “*Write less, do more*”, cukup tulis sedikit tapi bisa melakukan banyak hal.

2.6.8 MySQL

Menurut (R.H. Sianipar, 2015) dalam Buku “ Membangun Web dengan PHP & MYSQL untuk Pemula & Programmer ”. MySQL bukan termasuk bahasa pemrograman. MySQL merupakan salah satu database populer dan mendunia. MySQL bekerja menggunakan *SQL Language (Structure Query Language)*. Pada umumnya, perintah yang paling sering digunakan dalam MySQL adalah *SELECT* (mengambil), *INSERT* (menambah), *UPDATE* (mengubah), dan *DELETE* (menghapus). Selain itu, SQL juga menyediakan perintah untuk membuat *database*, *field*, ataupun *index* untuk menambah atau menghapus data.

