

BAB II

LANDASAN TEORI

2.1 Pengendalian Persediaan

Pengendalian persediaan merupakan serangkaian kebijakan pengendalian untuk menentukan tingkat persediaan yang harus dijaga, kapan pesanan untuk menambah persediaan harus dilakukan dan berapa besar pesanan harus diadakan (Herjanto, 2008).

2.2 Material

Ruang Lingkup Manajemen Material pertama adalah Perencanaan dan Pengendalian Material. Material yang dibutuhkan akan direncanakan dan dikendalikan berdasarkan *Sales Forecast* atau Perkiraan Penjualan dan Perencanaan Produksi (*Production Planning*). Perencanaan dan Pengendalian Material ini melibatkan perkiraan kebutuhan setiap material, menyiapkan anggaran material, meramalkan tingkat persediaan, menjadwalkan pemesanan material dan melakukan pemantauan kinerjanya yang berhubungan dengan produksi dan penjualan. (Mulyadi, 2000)

2.3 Forecasting

Forecasting adalah suatu seni dan ilmu pengetahuan dalam memprediksi peristiwa pada masa mendatang. Peramalan akan melibatkan pengambilan data historis (penjualan tahun lalu) dan memproyeksi mereka ke masa yang akan datang dengan model matematika. (Heizer dan Render, 2015)

2.4 Single Moving Average

Single Moving Average adalah suatu metode peramalan yang dilakukan dengan mengambil sekelompok nilai pengamatan, mencari nilai rata-rata tersebut sebagai ramalan untuk periode yang akan datang. *Metode Single Moving Average* mempunyai karakteristik khusus yaitu :

- A. Untuk menentukan ramalan pada periode yang akan datang memerlukan data *historis* selama jangka waktu tertentu. Misalnya, dengan 3 bulan *moving average*, maka ramalan bulan ke 5 baru dibuat setelah bulan ke 4 selesai / berakhir. Jika bulan *moving averages* bulan ke 7 baru bisa dibuat setelah bulan ke 6 berakhir.
- B. Semakin panjang jangka waktu *moving average*, efek pelicinan semakin terlihat dalam ramalan atau menghasilkan *moving average* yang semakin halus. Persamaan matematis *single moving averages* adalah sebagai berikut :

$$M_T = F_{t+1} = \frac{X_t + X_{t-1} + X_{t-2} + \dots + X_{t-n+1}}{n}$$

Keterangan :

M_t : *Moving Average* periode t

F_{t+1} : Ramalan periode $t+1$

X_i : Nilai riil periode ke t

n : Jumlah batas dalam *moving average*

(Rachman, 2018)

2.5 Single Exponential Smoothing

Metode ini juga digunakan Digunakan untuk data-data yang bersifat stasioner dan tidak menunjukkan pola atau tren, serta dapat digunakan untuk meramalkan suatu data untuk periode ke depan. Kasus yang paling sederhana dari pemulusan (smoothing) eksponensial tunggal (SES) dapat di kembangkan dari persamaan matematis sebagai berikut:

$$F_{t+1} = F_t + \left(\frac{X_t}{n} - \frac{X_{t+1}}{n} \right) \quad (2.1)$$

Misalkan pengamatan yang lama X_{t-N} tidak tersedia sehingga tempatnya harus digantikan dengan suatu nilai pendekatan. Salah satu pengganti yang mungkin adalah nilai ramalan pada periode yang sebelumnya F_t . Dengan melakukan substitusi ini persamaan (2.1) menjadi persamaan (2.2) dan dapat ditulis kembali sebagai (2.3), berikut persamaannya:

$$F_{t+1} = F_t + \left(\frac{X_t}{n} - \frac{X_{t+1}}{n} \right) \quad (2.2) \quad \text{atau} \quad F_{t+1} = \left(\frac{1}{n} \right) X_t + \left(1 - \frac{1}{n} \right) F_t \quad (2.3)$$

Dari persamaan (2.3) dapat dilihat bahwa ramalan ini (F_{t+1}) di dasarkan atas pembobotan observasi yang terakhir dengan suatu nilai bobot ($1/N$) dan pembobotan ramalan yang terakhir sebelumnya (F_t) dengan suatu bobot [$1 - (1/N)$], karena N merupakan suatu bilangan positif, $1/N$ akan menjadi suatu konstanta antara nol (jika N tak terhingga) dan 1(jika $N=1$) dengan mengganti $1/N$ dengan α , sehingga persamaan (2.3) akan menjadi:

$$F_{t+1} = \alpha X_t + (1 - \alpha) F_t \quad (2.4)$$

(Mardhiyah, 2016)

2.6 Algoritma

Dalam suatu program aplikasi tentunya pembuat harus menentukan terlebih dahulu mengenai *input* dan *output* yang diterima oleh aplikasi tersebut nantinya. Dari *input* tersebut nantinya akan diolah menjadi *output* yang diinginkan dan dapat mempermudah pengguna aplikasi dalam berinteraksi dengan aplikasi tersebut. Alur proses yang sistematis tentang jalanya suatu aplikasi dari awal hingga akhir disebut dengan algoritma. (Winda Swastika, 2018)

2.7 Internet Dan Aplikasi Web

2.7.1 Internet

Pada dasarnya *internet* merupakan suatu jaringan dunia yang dapat menghubungkan setiap komputer diseluruh dunia. Dengan adanya *internet*, sebuah komputer dapat memperoleh data dari komputer lain, baik itu dalam jarak yang cukup dekat ataupun jarak yang mencangkup antar negara bahkan benua. (Priyanto Hidayatulloh dan Jauhari Khairul Kawistara, 2014)

2.7.2 Website

Website merupakan suatu sekumpulan halaman – halaman yang didalamnya terdapat informasi berupa *text* ataupun media baik itu animasi, gambar, suara, video ataupun gabungan dari semuanya yang berbasis data digital yang dapat diakses melalui koneksi *internet*. Sehingga, halaman tersebut dapat dilihat oleh orang banyak diseluruh dunia. Halaman *website* tersebut dibuat dengan menggunakan bahasa pemrograman standar seperti HTML, CSS, JAVA SCRIPT

dan PHP yang nantinya akan diterjemahkan oleh *web browser* untuk menampilkan informasi.

(Rohi Andulloh, 2018)

2.8 Tools Aplikasi Web

2.8.1 HTML

Hypertext Markup Language atau yang sering kita kenal dengan singkatan HTML merupakan bahasa pemrograman standar berupa tag – tag penyusun setiap elemen dari sebuah *web* yang dikelola penggunaannya oleh *World Wide Web Consortium* (W3C). HTML dapat diletakan sesuai keinginan dalam mengatur setiap elemen *web*. Aplikasi *text* editor yang paling standar untuk *skrip* HTML adalah Notepad yang merupakan bawaan dari komputer. Selain Notepad kita bisa menggunakan *text* editor lainnya seperti Visual Studio Code, Sublime, Notepad ++ dan masih banyak yang lainnya. *File* HTML, biasanya disimpan dengan format *.html*. (Rohi Andulloh, 2018)

2.8.2 CSS

Cascading Style Sheet atau yang sering kita kenal dengan singkatan CSS merupakan bahasa pemrograman yang bertugas untuk mengatur *style* atau gaya tampilan elemen pada HTML seperti yang kita inginkan. Banyak orang yang beranggapan bahwa CSS bukan termasuk kedalam bahasa pemrograman, karena bentuk struktur yang terlalu sederhana. CSS berkerja dengan cara memilih element HTML yang disiapkan untuk kemudian nantinya akan diberikan properti *style* atau gaya tampilan sesuai yang kita inginkan. Didalam css terbagi menjadi 3

bagian yaitu pertama untuk memilih elemen untuk menentukan aturan, kedua *property* ialah aturan yang diberikan dan yang terakhir adalah sebagai nilai dari suatu aturan. (Rohi Andulloh, 2018)

2.8.3 PHP

PHP : Hypertext Preprocessor atau PHP merupakan bahasa pemrograman yang disisipkan kedalam *skrip* HTML dan berkerja bersama *server*. Tujuan utama dari bahasa pemrograman ini adalah untuk membantu perancang aplikasi dalam mengembangkan *web* supaya menjadi lebih cepat dan dinamis saat digunakan.

Sebelum menggunakan PHP, terlebih dahulu harus mempersiapkan *Software* sebagai berikut :

1. *Server Web* : Apache, IIS, Personal *Web Server* dan lain – lain.
2. *Server PHP*.
3. *Server Database* : MYSQL, Interbase, MS SQL dan lain – lain.

Tidak perlu menginstal satu persatu, karena didalam aplikasi Xampp ataupun Appserv sudah tersedia menjadi satu paket aplikasi yang siap digunakan. (Rohi Andulloh, 2018)

2.8.4 JavaScript

JavaScript ialah bahasa pemrograman *web* yang berkerja disamping klien. Oleh karena itu JavaScript bisa dijalankan walaupun hanya dengan menggunakan *browser web* saja. Ketika JavaScript dijalankan maka akan beraneka ragam perintah tertentu yang terjadi dihalaman *web*. Baik itu perintah yang dilakukan klien ataupun secara otomatis dilakukan *web*. (Rohi Andulloh, 2018)

2.8.5 JQuery

JQuery adalah salah satu perpusataan dari bahasa pemrograman JavaScript. JQuery biasanya herisikan kumpulan fungsi JavaScript yang sudah siap digunakan, sehingga memudahkan klien dalam membuat *script* JavaScript. JQuery dapat mengubah bahasa pemrograman JavaScript yang panjang menjadi hanya satu baris kode saja. Sebelum menggunakan JQuery, alangkah lebih baiknya jika memiliki pengetahuan mengenai HTML, JavaScript dan CSS terlebih dahulu. (Rohi Andulloh, 2018)

2.8.6 Bootstrap

Bootstrap merupakan *framework* dari bahasa pemrograman CSS yang paling populer. Tampilan *design web* dengan menggunakan bootstrap dapat menjadi lebih responsif dan mudah dibuka diberbagai jenis ukuran dengan *design* yang menarik. Selain itu dengan bootstrap dapat membuat *script* CSS menjadi lebih mudah dan cepat. Bootstrap sudah *Support* diberbagai jenis *browser* baik itu *desktop* atau pun *mobile*. (Rohi Andulloh, 2018)

2.8.7 XAMPP

XAMPP ("X (*Windows / Linux*) Apache MySQL PHP dan Perl") ialah suatu aplikasi yang terdiri dari paket *server web* dan *database*, dikalangan klien *web* yang paling populer adalah menggunakan *web server* PHP dan *database* MySQL. Xampp bisa digunakan pada operating sistem windows, linux dan macos. XAMPP sangat mudah digunakan untuk membantu pengembang dalam membuat aplikasi

web. Pembaruan aplikasi yang rutin membuat XAMPP sangat bagus untuk dipilih.
(Betha Sidik, 2014)

2.9 Database Dan MySQL

2.9.1 Database

Berikut ini adalah beberapa pengertian basis data atau *database* :

1. Suatu sekumpulan data yang saling terhubung sedemikian rupa, sehingga dapat digunakan dengan mudah.
2. Suatu sekumpulan data yang terhubung dan disimpan secara bersamaan tanpa melalui pengulangan.
3. Suatu sekumpulan *file*, tabel, atau arsip yang saling terhubung dan disimpan didalam suatu penyimpanan eletronik. (Uus Rusmawan, 2019)

2.9.2 MySQL

MYSQL merupakan suatu *Software DBMS* yang *multiuser*, *multithread* dengan instalasi aplikasi 6 jutaan diseluruh dunia. MySQL AB selaku *Software company* membuat MYSQL untuk *Support* diberbagai jenis perangkat dan dibagikan secara gratis, akan tetapi walaupun terdapat MYSQL yang gratis MySQL AB tetap membuat MYSQL untuk dijual secara lisensi komersial untuk kasus – kasus tertentu apabila klien tidak suka dengan menggunakan GPL yang terdapat pada *Software* gratisnya. (Uus Rusmawan, 2019)




2.10 UML

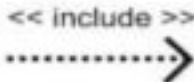
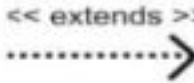
2.10.1 Use Case Diagram

Use Case Diagram merupakan suatu teknik yang digunakan untuk merekam fungsional dalam sebuah sistem. *Use case* menggambarkan suatu hubungan antara pengguna dengan sistem, dengan memberikan suatu narasi mengenai bagaimana suatu sistem tersebut akan digunakan nantinya oleh pengguna. Dalam *Use Case Diagram* terdapat dua hal penting yaitu skenario dan aktor. Skenario ialah suatu rangkaian tahapan – tahapan yang menggambarkan suatu hubungan antara pengguna dengan aplikasi tersebut, sedangkan aktor ialah gambaran dari pengguna itu sendiri. (Uus Rusmawan, 2019)

Berikut ini adalah beberapa simbol *Use Case Diagram* beserta keterangannya terdapat :

Tabel 2.1. Tabel Deskripsi Simbol *Use Case Diagram* (Uus Rusmawan, 2019)

Simbol	Nama	Keterangan
	<i>Actor</i>	Seseorang yang berinteraksi dengan suatu sistem yang sedang dibuat atau dijalankan.
	<i>Use Case</i>	Mendefinisikan mengenai seseorang dalam menggunakan sistem tersebut.
	Relasi Asosiasi	Suatu garis penghubung yang digunakan untuk menunjukkan relasi atau interaksi antara dengan <i>use case</i> .



	<p>Relasi <i>Include</i></p>	<p>Memungkinkan <i>use case</i> dalam menggunakan suatu fungsional yang telah disediakan oleh <i>use case</i>.</p>
	<p>Relasi <i>Extend</i></p>	<p>Memungkinkan <i>use case</i> secara optimal dalam menggunakan suatu fungsional yang telah disediakan oleh <i>use case</i>.</p>




2.10.2 Activity Diagram

Activity Diagram merupakan suatu rangkaian aktivitas yang menunjukkan suatu alur kerja dari awal kejadian hingga akhir keputusan dengan merinci didalam perkembangan suatu proses atau peristiwa yang berada didalam suatu aktivitas tersebut. *Activity diagram* memiliki bentuk tertentu didalam komponennya yang dihubungkan dengan suatu tanda panah. Panah mengarah sesuai tahapan – tahapan aktivitas dari awal kejadian hingga akhir keputusan. (Uus Rusmawan, 2019)

Berikut ini adalah beberapa simbol *Activity Diagram* beserta keterangannya terdapat :

Tabel 2.2. Tabel Deskripsi Simbol *Activity Diagram* (Uus Rusmawan, 2019)

Simbol	Nama	Keterangan
	<i>Start State</i>	Merupakan bentuk suatu titik awal dalam suatu aktivitas.
	<i>End State</i>	Merupakan bentuk suatu titik akhir keputusan dari


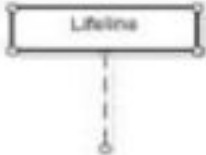
		aktivitas tersebut.
	<i>Activity</i>	Menggambarkan suatu aktivitas yang dilakukan <i>actor</i> tersebut.
	<i>Decision</i>	Menggambarkan untuk menentukan keputusan dalam suatu aktivitas.
	<i>Interaction</i>	Menggambarkan jalur penghubung antar aktivitas yang mengarah.


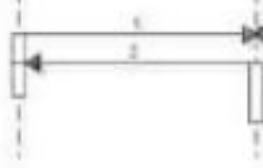

2.10.3 Sequence Diagram

Sequence Diagram merupakan suatu susunan mengenai urutan waktu tertentu. *Sequence Diagram* dapat menggambarkan satu tugas dari beberapa tugas didalam suatu *use case* secara lebih *detail*. (Uus Rusmawan, 2019)

Berikut ini adalah beberapa simbol *Sequence Diagram* beserta keterangannya terdapat :

Tabel 2.3. Tabel Deskripsi Simbol *Sequence Diagram* (Uus Rusmawan, 2019)

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan pengguna yang akan melakukan aktivitas tersebut.
	<i>Lifeline</i>	Menggambarkan suatu objek didalam suatu sistem atau komponennya.

	<p><i>Create Message</i></p>	<p>Membuat suatu pesan sederhana antar elemen dan melakukan interaksi antar suatu objek.</p>
	<p><i>Synchronous Message</i></p>	<p>Pesan yang mengaktifkan suatu proses hingga sampai, baru bisa melakukan pengiriman sebuah pesan baru..</p>
	<p><i>Message to self</i></p>	<p>Menggambarkan sebuah hasil kembalian dari suatu operasi dan berjalan kepada objek itu sendiri..</p>

