

II. TINJAUAN PUSTAKA

1 PERAMALAN

Peramalan merupakan suatu kegiatan memprediksikan kondisi-kondisi di masa mendatang. Peramalan permintaan dilakukan sebagai tahap awal dalam perencanaan produksi dan persediaan untuk mengetahui besarnya permintaan di masa mendatang.

Peramalan kuantitatif adalah salah satu metode peramalan permintaan yang digunakan dalam perencanaan produksi. Metode peramalan kuantitatif merupakan sebuah metode yang menggunakan model matematis dalam mengolah data masa lalu untuk memprediksi permintaan di masa mendatang. Pemilihan metode peramalan kuantitatif yang akan digunakan berdasarkan pola data permintaan masa lalu. Berikut ini adalah beberapa metode peramalan kuantitatif yang dapat digunakan :

1.1. Metode Peramalan Kuantitatif

1.1.1. Metode rerata sederhana (*simpleaverage*)

Metode ini menarik rata-rata secara sederhana dari semua data masa lalu yang ada untuk meramalkan permintaan pada periode yang akan datang. Metode ini cocok untuk data dengan komponen *time series* acak. Rumus yang digunakan :

$$F_t = \frac{\sum_{i=1}^n Y_i}{n} \quad (2.1)$$

dimana F_t = ramalan untuk periode t

Y_i = data pada periode i

n = jumlah data masa lalu

1.1.2. Metode pemulusan eksponensial tunggal (*single exponential smoothing*)

Metode peramalan ini memberikan bobot lebih besar pada data masa lalu yang lebih akhir terjadi daripada data masa lalu yang lebih awal terjadi. Besarnya bobot yang diberikan adalah antara 0 dan 1. Metode ini cocok untuk data yang mempunyai komponen *time series* acak yang menunjukkan adanya trend.

Rumus umum yang digunakan :

$$F_{t+1} = \alpha Y_t + (1 - \alpha) \cdot F_t \quad (2.2)$$

dimana F_{t+1} = ramalan untuk periode $t + 1$

α =bobot

Y_t = data pada periode t

F_t = ramalan untuk periode t

1.2. Parameter Kesalahan

Kegiatan peramalan terhadap kondisi-kondisi di masa mendatang pada umumnya tidak dapat persis sama dengan kenyataan sesungguhnya yang terjadi di masa mendatang. Dalam setiap peramalan yang dibuat selalu dihasilkan bias.

Kesalahan peramalan didefinisikan sebagai perbedaan nilai antara hasil ramalan dengan keadaan sesungguhnya. Besarnya kesalahan peramalan permintaan dalam suatu periode merupakan besarnya permintaan sesungguhnya dikurangi peramalan permintaan untuk periode tersebut. Nilai kesalahan peramalan dapat menunjukkan apakah

peramalan yang dilakukan sudah cukup baik Peramalan terbaik adalah yang menghasilkan nilai kesalahan peramalan yang terkecil. Beberapa cara yang digunakan untuk mengukur besarnya kesalahan peramalan adalah :

1.2.1. *Mean absolute deviation (MAD)*

MAD menunjukkan rata-rata dari nilai absolut kesalahan yang terjadi selama periode peramalan. MAD cocok digunakan jika data mempunyai rata-rata besar dan deviasi kecil. Persamaan yang digunakan:

$$\text{MAD} = \frac{\sum_{t=1}^n |e_t|}{n} = \frac{\sum_{t=1}^n |Y_t - \hat{Y}_t|}{n} \quad (2.3)$$

1.2.2. *Mean square error (MSE)*

MSE menunjukkan rata-rata dari kuadrat kesalahan yang terjadi selama periode peramalan. MSE cocok digunakan jika data mempunyai rata-rata kecil dan deviasi besar. Persamaan yang digunakan =

$$\text{MSE} = \frac{\sum_{t=1}^n (e_t)^2}{n} = \frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{n} \quad (2.4)$$

2. *MATERIAL REQUIREMENT PLANNING (MRP)*

Material Requirement Planning (MRP) merupakan suatu teknik perencanaan dan pengendalian produksi yang mengatur kegiatan produksi dan pemesanan material untuk produk yang mempunyai permintaan yang bersifat

dependen. Pengaturan yang yang dilakukan meliputi material apa saja yang **harus** diproduksi atau dipesan, berapa jumlah yang dibutuhkan, kapan material tersebut diperlukan dan kapan material tersebut **harus** dipesan.

Penyusunan *Material Rrequirement Planning* membutuhkan informasi dari *Master Production Schedule (MPS)* dan *Bill of Material (BOM)*. *MPS* merupakan perencanaan kuantitas produksi dari satu atau sejumlah produk tertentu yang didasarkan pada peramalan permintaan yang telah dibuat. Sedangkan *BOM* merupakan daftar komponen-komponen yang menyusun suatu produk, yang disertai dengan jumlah kebutuhan dari tiap-tiap komponen tersebut. *BOM* juga dapat menggambarkan urutan proses manufaktur dari **suatu** produk dan hubungan ketergantungan antara komponen yang **satu** dengan komponen yang lain.

Hasil yang diberikan oleh *MRP* antara lain adalah memberikan catatan tentang penjadualan pesanan yang harus dilakukan, memberikan indikasi untuk pembatalan pemesanan, dan memberikan **informasi** mengenai keadaan persediaan.

MRP memiliki beberapa komponen antara lain :

2.1. *Gross Requirement (GR)*

Gross Requirement atau kebutuhan kotor suatu komponen adalah jumlah komponen yang dibutuhkan dalam tiap periode untuk memproduksi produk **akhir** sesuai dengan *MPS* yang telah dibuat. Kebutuhan kotor untuk komponen pada level selanjutnya ditentukan oleh *MRP* yang telah dibuat pada level yang lebih tinggi.

2.2. *Schedule Receipt (SR)*

Schedule receipt menyatakan jumlah material yang sudah atau sedang dipesan dari supplier dan diharapkan datang pada awal periode yang telah dijadualkan.

2.3. **On Hand Inventory (OHI)**

On hand inventory menyatakan jumlah persediaan **suatu** item pada tiap periode. OHI dapat dirumuskan sebagai berikut:

$$(OHI)_t = \text{maks} \{ (OHI)_{t-1} - (GR)_{t-1} + (SR)_{t-1}, 0 \} \quad (2.5)$$

2.4. **Net Requirement (NR)**

Net requirement menyatakan jumlah kebutuhan bersih yang **harus** disediakan **untuk** memenuhi kebutuhan kotor dari komponen. **Net** requirement dapat **dirumuskan** sebagai berikut:

$$(NR)_t = \text{maks} \{ (GR)_t - (SR)_t - (OHI)_t, 0 \} \quad (2.6)$$

2.5. **Planned Order Receipt (POR)**

Planned Order Receipt menunjukkan jumlah pesanan komponen yang harus diterima pada awal periode tertentu yang belum mulai dikirimkan. Besarnya POR bisa Sama dengan besarnya NR pada periode yang sama jika digunakan teknik **lotfor lot**.

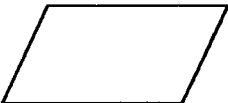
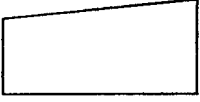
2.6. **Planned Order Release (PORE)**

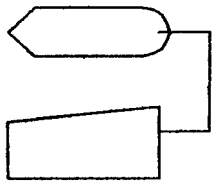
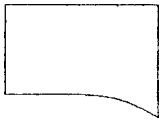
Planned Order Release menunjukkan jumlah pesanan komponen yang dilepaskan dan kapan harus dilepaskan supaya tersedia tepat pada saat dibutuhkan.

3. SISTEM *FLOWCHART*

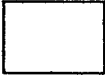
Sebuah *flowchart* adalah sebuah teknik analitis untuk menjelaskan beberapa aspek dari sebuah sistem informasi dengan cara yang jelas, tepat dan **logis**. *Flowchart* pada dasarnya menggunakan satu kumpulan simbol-simbol standar untuk menjelaskan prosedur sebuah proses dalam sebuah organisasi atau perusahaan serta untuk menjelaskan aliran data dalam sebuah sistem tertentu dalam organisasi tersebut.

3.1. Simbol Masukan/Keluaran

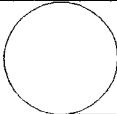
Simbol	Nama	Keterangan
	Dokumen	Sebuah dokumen/ laporan yang disiapkan secara manual ataupun menggunakan komputer.
	Beberapa salinan dari sebuah dokumen	Dilambangkan dengan simbol dokumen yang berlapis-lapis.
	Masukan/Keluaran ; Jurnal/Pembukuan	Menunjukkan sebuah masukan/ keluaran untuk suatu proses yang digambarkan dalam <i>flowchart</i> . Juga dapat melambangkan jurnal akuntan dan pembukuan perusahaan.
	Display (layar tambahan)	Informasi yang ditampilkan dengan sebuah alat elektronik yang terhubung secara online, contohnya komputer PC atau terminal CRT.
	Pemasukan data secara online	Proses memasukkan data dengan menggunakan alat elektronik yang terhubung secara online.

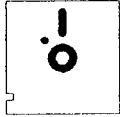
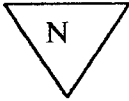
	Terminal CRT, komputer PC	Simbol layar tampilan dan simbol input data secara online digunakan bersama untuk menunjukkan sebuah terminal CRT/komputer PC.
	Transmittal Tape	Biasanya dipersiapkan secara manual sebagai pengontrol dan pembanding dengan hasil dari pengontrolan komputer.

3.2. Simbol Pemrosesan

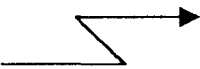
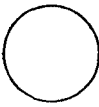
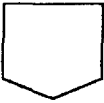
Simbol	Nama	Keterangan
	Proses (secara komputerisasi)	Fungsi pemrosesan menggunakan komputer, biasanya ditandai dengan adanya perubahan data/informasi.
	Operasi manual	Fungsi pemrosesan secara manual (tidak menggunakan alat).
	Operasi tambahan	Fungsi pemrosesan oleh suatu alat selain komputer.
	Pemasukan data secara offline/manual	Misalnya menggunakan mesin kasir.

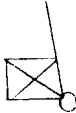
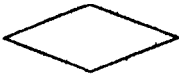
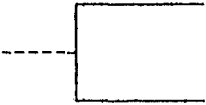
3.3. Simbol Penyimpanan

Simbol	Nama	Keterangan
	Piringan magnetis	Data disimpan secara permanen pada sebuah piringan magnetis sebagai data utama.
	Kaset magnetis	Data disimpan pada kaset magnetis.

	Disket magnetis	Data disimpan pada sebuah disket magnetis.
	Penyimpanan online	Data disimpan pada file penyimpanan sementara, seperti disket.
	File	Dokumen dari file yang dapat disimpan dan dibuka secara manual; huruf di tengahnya menandakan : N = secara numerik A = secara alfabet D = urut tanggal

3.4. Simbol Aliran dan Simbol-simbol Lain

Simbol	Nama	Keterangan
	Aliran dokumen/ pemrosesan	Aliran yang normal arahnya ke kanan atau ke bawah.
	Aliran data/ informasi	Arah data/aliran informasi; biasanya digunakan untuk menunjukkan data yang disalin dari satu dokumen ke dokumen yang lain.
	Penghubung komunikasi	Transmisi data dari satu lokasi ke lokasi lain.
	Penghubung <i>flowchart</i> yang putus	Menghubungkan aliran proses pada halaman yang sama.
	Penghubung <i>flowchart</i> yang putus	Masukan dari halaman sebelumnya, atau keluar ke-, ke halaman lain.

	Aliran barang	Menunjukkan aliran barang .
	Terminal	Suatu awal, akhir, atau titik interupsi dalam suatu proses atau program.
	Keputusan	Sebuah langkah pengambilan keputusan.
	Catatan/keterangan (<i>annotation</i>)	Penambahan deskripsi perintah atau penjelasan untuk klarifikasi.

4. DATABASE

Sebuah database mempunyai empat elemen utama :

1) User data

Yaitu data yang disimpan dalam database. Sekumpulan data yang terhubung disebut relasi. Apabila relasi belum dideskripsikan dalam struktur yang jelas dan paling singkat, **maka** relasi ini butuh dinormalisasi terlebih dahulu.

2) Metadata

Yaitu deskripsi struktur dari beberapa tabel data. Setiap metadata memiliki sebuah **primary key** yaitu nama identifikasi untuk satu **baris** data yang sifatnya **unik** (tidak ada yang sama antara satu record dengan record yang lain).

3) Indexes

Digunakan untuk mengurutkan data, agar data enak dibaca dan dapat diakses secara cepat.

4) Aplikasi metadata

Digunakan untuk menyimpan struktur dan format **form**, report, **queries**, dan komponen aplikasi lain.

5. DATABASE MANAGEMENT SYSTEM (DBMS)

DBMS **adalah** fitur untuk memproses dan mengakses database. DBMS digunakan baik oleh **user** maupun oleh **developer** (perancang database). DBMS produk **microsoft (microsoft access 2000)** memiliki tiga **buah feature** dalam menjalankan fungsinya :

1. Alat pendesain

Merupakan fasilitas untuk merancang dan membuat database. **Microsoft access 2000** memiliki dua buah **bahasa dalam feature** ini, bahasa makro (yang tidak memerlukan keahlian pemrograman), dan bahasa pemrograman basic (visual basic).

2. Runtime

Merupakan fasilitas untuk memproses komponen aplikasi yang telah dibuat menggunakan alat pendesain, misalnya untuk **menampilkan** data pada **form**, mencatat laporan, **serta** mengevaluasi permintaan.

3. DBMS engine (motor DBMS)

Sebagai fasilitas penghubung antara data **feature** alat pendesain dan **feature runtime**. **Microsoft** menyediakan dua motor DBMS yang berbeda untuk **access 2000**, **the jet engine** dan SQL server. **The jet engine** digunakan untuk database pribadi atau kelompok (**workgroup**) yang kecil. Sedangkan SQL server yang merupakan produk **microsoft** yang terpisah dari **access 2000**, digunakan untuk database departemen yang lebih **besar**.

6. PEMODELAN DATA

Pemodelan data adalah proses perancangan bentuk yang **logis** yang dapat mewakili struktur dari database. Agar dapat berfungsi **secara** tepat, model data ini harus dapat mewakili semua sudut pandang pengguna database (**user**).

6.1. Model *Entity-Relationship* (model E-R)

Elemen-elemen penting dari model E-R adalah :

1. **Entity** adalah representasi barang/sesuatu yang ingin diteliti.
2. **Attributes** biasa disebut juga dengan **properties** yang berfungsi untuk mendeskripsikan karakteristik dari **entity**.
3. **Identifier** dimiliki oleh setiap **entity** merupakan atribut yang digunakan untuk mengidentifikasi suatu **entity**. **Identifiers** bisa berisi satu atau lebih atribut. **Identifiers** bisa bersifat unik maupun tidak.
4. **Relationship** adalah hubungan di antara dua **entity**. E-R model menggambarkan hubungan antar kelas.

Ada 3 tipe relasi, yaitu

- Relasi **one to one**. Menggambarkan hubungan satu entity dengan entity yang lain. Digambarkan dalam notasi 1:1.
- Relasi **one to many**. Dapat dinotasikan sebagai 1:N.
- Relasi **many to many**. Notasinya N:M.

Sedangkan konstrain yang ada dalam relasi disebut **maximum cardinality** untuk angka yang maksimum dan **minimum cardinality**.

6.2. Model *Semantic Object*

Semantic object adalah salah **satu** model, arti dari data pemakai.

Dengan model SO persepsi pemakai lebih tertutup.

Definisi **semantic object**

- SO adalah presentasi dari sesuatu/barang yang dapat diidentifikasi dari lingkungan. Secara umum, SO adalah nama kumpulan atribut yang cukup dapat mendeskripsikan suatu perbedaan identitas.
- SO dikelompokkan ke dalam kelas, ditulis dengan HURUF BESAR.
- Suatu **object** mempunyai kumpulan atribut yang mempresentasikan karakteristik yang diidentifikasi, sehingga tidak perlu melihat ke **object** yang lain (*sufficient description*).

6.3. Perbandingan antara model E-R dan model **semantic object**

- Persamaan di antara keduanya adalah :
 Berfungsi untuk **memahami** dan mendokumentasikan struktur **dari** data dan hubungan di antara data tersebut.
- Perbedaan di antara keduanya adalah :

Model E-R	Model Semantic object
Memandang konsep <i>entity</i> sebagai dasar dari pemodelan data. <i>Entity</i> dan hubungannya adalah seperti atom-atom dari model data, mereka adalah struktur inti pembentuk model tersebut. Atom-atom ini dapat dikombinasikan untuk membentuk apa yang oleh model E-R disebut sudut pandang pengguna (user), yang mana merupakan kombinasi dari <i>entity-entity</i> yang strukturnya sama dengan yang terdapat pada <i>semantic object</i> .	Memandang konsep <i>semantic object</i> sebagai dasar dari pemodelan data. Kumpulan dari semantic object dalam suatu model data adalah struktur yang oleh pengguna dianggap sebagai yang paling penting dan paling mendasar, bagaikan atom untuk model data seperti perumpamaan sebelumnya. Sebaliknya, menurut sudut pandang model <i>semantic object</i> , <i>entity</i> itu dianggap tidak ada.

7. DESAIN DATABASE

Dalam mendesain database, ada dua hal penting yang harus dibentuk yaitu :

7.1. *Relational* model

Functional dependencies adalah hubungan antara atau diantara atribut.

Functional dependencies dapat meliputi grup atribut.

Key adalah grup dari satu atau lebih atribut yang dapat mengidentifikasi baris menjadi unik.

7.2. Normalisasi

Normalisasi adalah proses untuk merubah suatu **relation** yang **bermasalah** menjadi dua atau lebih **relation**. Normalisasi juga digunakan untuk memeriksa kelayakan dan kebenaran relasi. Normalisasi membuat relasi mempunyai **satu** tema, jika mempunyai dua atau lebih tema harus memecah relasi tersebut sehingga hanya mempunyai *satu* tema.

Relasi dapat diklasifikasikan dalam beberapa tipe dari modifikasi **anomaly**. Ada 5 kelas dalam relasi :

1. Bentuk normal pertama disebut **repetitive**.
2. Bentuk normal kedua disebut **parsial dependencies**.
3. Bentuk normal ketiga disebut **transitive dependency**.
4. Bentuk normal keempat disebut **fourth normal form**.
5. Bentuk normal kelima disebut **boyce-codd normal form**.

Setelah tahapan untuk mencapai bentuk normal tersebut dipenuhi maka terbentuklah bentuk normal yang baru yang disebut **domain/key normal form** (DK/NF).

Relation dapat digambarkan dalam bentuk tabel 2 dimensi. Setiap baris dari tabel menyimpan data tentang sesuatu. Kolom dari tabel disebut **attribute** sedangkan barisnya disebut **tuples**.

Ketentuan :

1. Sel pada tabel harus berisi nilai tunggal.
2. Semua isi dalam kolom/atribut harus dalam jenis yang sama.
3. Setiap kolom harus mempunyai nama yang unik, urutan kolom dalam **tabel** tidak terlalu penting.

Untuk mengerti model relasi dan normalisasi diperlukan **functional dependency** dan **key**. **Functional dependency** adalah relasi diantara atribut, sedangkan **key** harus **unik**.

Untuk beberapa relasi, merubah data dapat mengakibatkan **modification anomalies** (diluar konsekuensi). **Anomali** dapat dibatasi dengan mendefinisikan kembali relasi ke dalam 2 atau lebih relasi. Ada beberapa modifikasi **anomali**, antara lain :

1. Menghapus **anomali (deletion anomaly)**. Biasanya dilakukan dengan menghapus satu entity. Menghapus **satu** entity berarti menghapus entity yang lain.
2. **Insertion anomaly**. Tidak dapat menyisipkan satu entity sebelum menambah entity yang lain. Kerugian membagi relasi adalah **referential integrity constraint**. Pada saat akan menambah satu entity dimana entity pada relasi lain tidak ditemukan.