

BAB II

LANDASAN TEORI

2.1. Sistem Persediaan Barang

Sistem persediaan barang pada suatu perusahaan bertujuan untuk mencatat mutasi tiap jenis persediaan yang disimpan di gudang. Sistem ini berkaitan erat dengan sistem penjualan, sistem pembelian, mutasi antar gudang, dan lain-lain.

Dalam perusahaan manufaktur, persediaan ini terdiri dari persediaan produk jadi, persediaan produk dalam proses, persediaan bahan baku, dan lain-lain. Sedangkan dalam perusahaan dagang, persediaan terdiri dari satu golongan, yaitu persediaan barang dagangan. Transaksi yang mengubah tiap jenis persediaan yang telah disebutkan di atas bersangkutan dengan transaksi intern perusahaan.

Terdapat dua metode pencatatan persediaan, yaitu metode mutasi persediaan (perpetual inventory method) dan metode persediaan fisik (physical inventory). Dalam metode mutasi persediaan, setiap mutasi persediaan dicatat dalam kartu persediaan. Dalam metode persediaan fisik, hanya tambahan persediaan dari pembelian saja yang dicatat, sedangkan mutasi berkurangnya persediaan karena pemakaian tidak tercatat dalam kartu persediaan.

Dalam sistem inventaris manual diselenggarakan dua catatan akuntansi, yaitu fungsi gudang dan fungsi akuntansi. Pencatatan inventaris dalam fungsi gudang menggunakan kartu gudang yang digunakan untuk mencatat kuantitas persediaan dan mutasi tiap jenis barang yang disimpan di gudang. Pada kartu gudang tidak berisi data

harga pokok tiap jenis barang, tapi hanya informasi kuantitas barang. Selain itu juga dibuat kartu barang yang ditempel pada tempat penyimpanan barang. Kartu barang ini digunakan sebagai identitas barang yang memudahkan pencarian dan pencatatan mutasi kuantitas barang. Sedangkan dalam catatan akuntansi fungsi akuntansi, menggunakan kartu persediaan yang berfungsi untuk mencatat kuantitas dan harga pokok barang digudang. Selain itu juga berfungsi untuk mengontrol catatan kuantitas barang di gudang serta memuat rekening kontrol persediaan yang bersangkutan dalam buku besar.

Menurut fungsinya, terdapat tiga jenis persediaan, yaitu:

a. Batch Stock/Lot Size Inventory

Persediaan yang diadakan karena kita membeli atau membuat bahan-bahan atau barang-barang dalam jumlah yang lebih besar dari jumlah yang dibutuhkan saat itu.

b. Fluctuation Stock

Persediaan yang diadakan untuk menghadapi fluktuasi permintaan konsumen yang tidak dapat diramalkan.

c. Anticipation Stock

Persediaan yang diadakan untuk menghadapi fluktuasi permintaan yang dapat diramalkan, berdasarkan pola musiman yang terdapat dalam satu tahun dan untuk menghadapi penggunaan atau penjualan atau permintaan yang meningkat.

Menurut jenis dan posisi barang, jenis persediaan adalah sebagai berikut :

a. Persediaan bahan baku.

b. Persediaan bagian produk/komponen yang dibeli.

- c. Persediaan bahan-bahan pembantu/penolong.
- d. Persediaan barang-barang setengah jadi/barang dalam proses.
- e. Persediaan barang jadi.

Tujuan pengawasan persediaan adalah, sebagai berikut,

- a. Menjaga jangan sampai kehabisan persediaan.
- b. Supaya pembentukan persediaan stabil.
- c. Menghindari pembelian kecil-kecilan.
- d. Pemesanan yang ekonomis.

Dalam tugas akhir ini metode analisis yang digunakan adalah analisis reorder point dan analisis economic order quantity. Metode analisis reorder point membantu manajemen memperhitungkan kapan mengorder ulang inventaris. Dengan mengetahui kapan mengorder ulang inventaris, manajemen dapat mencegah kehabisan stok.

Rumus Reorder Point adalah sebagai berikut,

$$ROP = \frac{X \cdot L}{Day/Week} + SS$$

Dimana:

X = rata-rata pengeluaran barang

L = rata-rata Lead Time

Days/Week = jumlah hari kerja selama 1(satu) minggu

SS = Safety Stock

Untuk mendapatkan nilai rata-rata pengeluaran barang dengan rumus

$$X = \frac{\sum_{i=1}^n Xi}{n}$$

Dimana:

Xi = jumlah pengeluaran barang

n = periode pengeluaran barang

Untuk mendapatkan nilai rata-rata Lead Time dengan rumus

$$L = \frac{\sum_{i=1}^n Li}{n}$$

Dimana:

Li = Lead time setiap kali pembelian

n = periode lead time

Metode analisis economic order quantity dapat membantu manajemen menentukan jumlah barang yang akan diorder ulang. Analisis ini membantu manajemen membantu meningkatkan efisiensi penggunaan sumber daya organisasi, yaitu menurunkan jumlah tenaga manual untuk memperhitungkan barang mana yang perlu diorder ulang dan berapa banyak perlu diorder ulang.

Economic Order Quantity (EOQ) dapat dirumuskan sebagai berikut,

$$EOQ = \sqrt{\frac{2.U.Co}{Ch.P}}$$

Dimana:

U = pengeluaran barang dalam unit per periode

Co = biaya pesan per pemesanan

Ch = biaya penyimpanan persediaan

P = harga per unit

2.2. Oracle Power Objects

Oracle Power Objects (OPO) merupakan aplikasi pengembang (Application Programming Tool) Front End/Client yang dikembangkan oleh Oracle Inc. dan berjalan pada sistem operasi Microsoft Windows dan Macintosh. OPO dirancang untuk memberikan kemudahan dalam pemakaian dan kemampuan besar pada pengembang untuk membuat aplikasi dalam lingkungan client/server. OPO memiliki Tools, Methods, dan Controls yang lengkap untuk membuat, mengubah, dan memelihara komponen-komponen aplikasi seperti application, form, dan report. OPO juga dirancang dengan pendekatan Objects yang meliputi Classes, Properties, dan Methods.

Beberapa kemampuan dasar dari Oracle Power Objects antara lain:

1. Desain grafis.
2. Desain berorientasi objek.
3. Mempunyai database lokal.
4. Mengakses berbagai database.

Dalam konsep Object Oriented Programming sebagaimana yang dianut oleh Oracle Power Objects, setiap komponen yang ada baik itu Applications (form, report, class) maupun databases (table, view, index) dianggap sebagai objek yang memiliki properties, event dan method. Setiap objek mempunyai warna dan bentuk (properties), juga memiliki fungsi (methods) tersendiri. Fungsi (methods) dari setiap

komponen tersebut dapat diaktifkan dengan suatu aksi (event). Jika fungsi standar dari komponen-komponen yang tersedia tidak sesuai dengan yang diinginkan, maka dapat dibuat fungsi baru (dengan menggabungkan beberapa objek/fungsi) sesuai kebutuhan dan menjadikan fungsi baru tersebut sebagai standar fungsi baru (classes). Selain itu juga dapat digunakan komponen dari pihak ketiga (ActiveX/OCX).

Terdapat berbagai jenis objek dalam Oracle Power Objects yang dapat digunakan dalam membangun suatu aplikasi, sebagai berikut:

1. Application Objects
2. Library Objects
3. Session Objects
4. Database Objects
5. Recordset Objects
6. Properties
7. Methods
8. Events
9. Class
10. Record Manager

2.3. Sistem Basis Data

Sistem Manajemen Basis Data / Database Management System (DBMS) terdiri dari koleksi data yang saling berhubungan dan kumpulan program untuk mengakses data tersebut. Koleksi data tersebut biasanya dikenal sebagai basis data.

Tujuan utama Sistem Manajemen Basis Data adalah untuk menyediakan lingkungan untuk mencari dan menyimpan informasi.

Sistem basis data dirancang untuk mengatur informasi yang besar. Pengaturan data ini mencakup definisi struktur tempat informasi dan mekanisme memanipulasi informasi. Jadi sistem basis data harus mendukung penyimpanan informasi, mencegah kegagalan sistem, atau mencegah pengaksesan ilegal.

Penyusunan suatu sistem basis data digunakan untuk mengatasi masalah-masalah pada penyusunan data, yaitu:

- Redudansi dan inkonsistensi data
- Kesulitan pengaksesan data
- Isolasi data untuk standarisasi
- Multiuser
- Masalah keamanan
- Masalah integrasi
- Masalah data independence

Pada perancangan basis data model konseptual ditekankan pada struktur data dan relasi antar file. Pendekatan yang dilakukan pada perancangan model konseptual adalah menggunakan model data relational. Terdapat dua buah teknik perancangan basis data, yaitu:

1. Teknik Normalisasi

Yaitu proses pengelompokan data menjadi tabel-tabel yang menunjukkan entitas dan relasinya untuk mengurangi penyimpangan informasi dalam proses

perubahan (update), penyisipan (insert), penghapusan (delete), membaca (retrieve) dan optimasi penggunaan tempat penyimpanan.

Dalam proses normalisasi ini terdapat beberapa tahap normalisasi, yaitu:

- Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang disimpan apa adanya sesuai kedatangannya dan tidak mengikuti suatu format tertentu.

- Bentuk normal kesatu (1NF/First Normal Form)

Dalam bentuk normal kesatu ini, data disimpan dalam rekord dan kolom.

- Bentuk normal kedua (2NF/Second Normal Form)

Bentuk normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk normal kesatu. Atribut bukan kunci bergantung pada kunci utama (primary key). Dalam normal kedua ini harus sudah ditentukan field-field kunci. Field kunci harus bersifat unik.

- Bentuk normal ketiga (3NF/Third Normal Form)

Untuk menjadi normal ketiga maka relasi haruslah dalam bentuk normal kedua. Semua atribut bukan priper tidak punya hubungan transitif. Tiap atribut bukan kunci harus bergantung hanya pada primary key.

- Boyce-Cadd Normal Form (BCNF)

Dalam BCNF, relasi harus dalam bentuk normal kesatu dan tiap atribut harus bergantung pada atribut super key.

2. Teknik Entity Relationship

Basis data adalah kumpulan file yang saling berkaitan. Pada model data relational, hubungan antar file direlasikan dengan kunci relasi (relation key) yang merupakan kunci utama dari masing-masing file.

Relasi antara dua file atau dua tabel dapat dikategorikan menjadi tiga macam, yaitu:

- One to one relationship

Hubungan antara file pertama dengan file kedua adalah satu berbanding satu.

Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah tunggal.

- One to many relationship

Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau banyak berbanding satu. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda.

- Many to many relationship

Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tanda panah ganda.

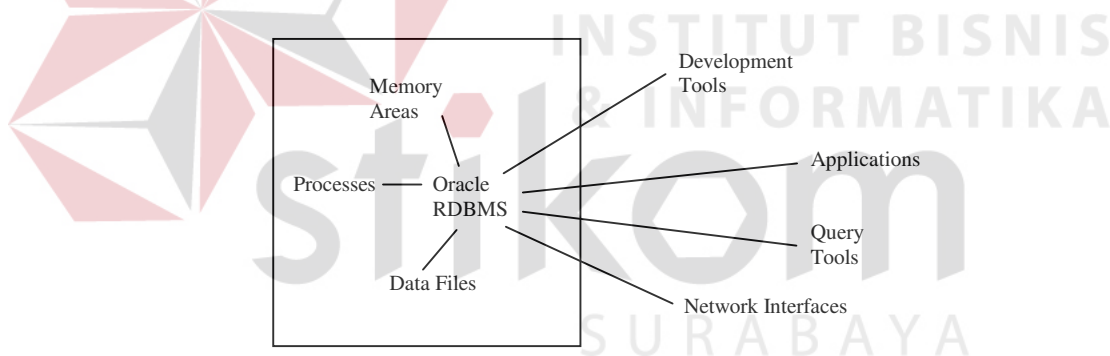
2.4. Oracle Database

2.4.1. Konsep RDBMS Oracle

Konsep dasar RDBMS Oracle adalah perbedaan antara database dan instance. Database adalah kumpulan data terstruktur yang disimpan dalam file-file data. Sedangkan instance adalah kumpulan proses dan memori yang menyediakan kemampuan mengakses data.

Aritektur RDBMS Oracle terbagi atas tiga bagian, yaitu:

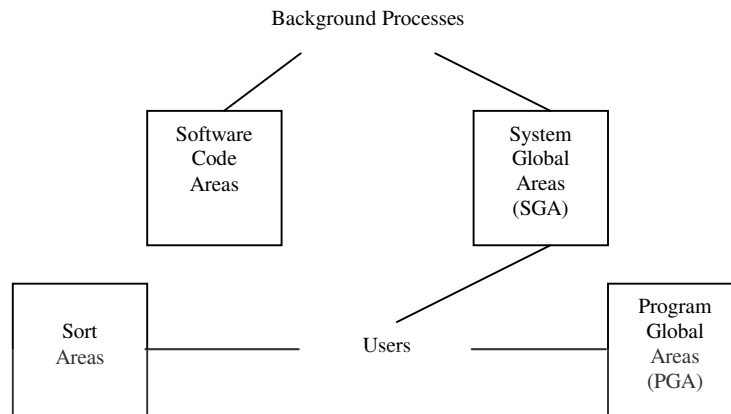
1. Background processes.
2. Shared memory areas.
3. Disk data storage areas.



Gambar 2.1. Arsitektur RDBMS Oracle

Oracle menggunakan beberapa proses untuk melakukan proses penulisan data. Sebagai contoh, ketika seorang user menyimpan perubahan data dalam database, data ini disimpan dalam memori. Pada saat memori sudah siap untuk menulis data ke dalam media penyimpan seperti hard disk, proses penulisan database mengirimkan data ini dari memori ke dalam record-record yang tersedia dalam media penyimpan.

2.4.2. Memori dan Kecepatan



Gambar 2.2. Area memori Oracle

Memori merupakan kunci kecepatan dari RDBMS Oracle seperti pada Gambar 2.3. Berikut adalah beberapa komponen atau area memori yang digunakan Oracle:

1. Software code areas.

Area memori ini digunakan untuk menyimpan software yang menjalankan database.

2. System Global Area (SGA).

Area memori ini digunakan untuk mempercepat proses transaksi untuk proses pengambilan data. Terdapat empat komponen kunci dalam SGA, yaitu :

- a. Database buffer cache.
- b. Redo log buffer.
- c. Shared SQL areas.

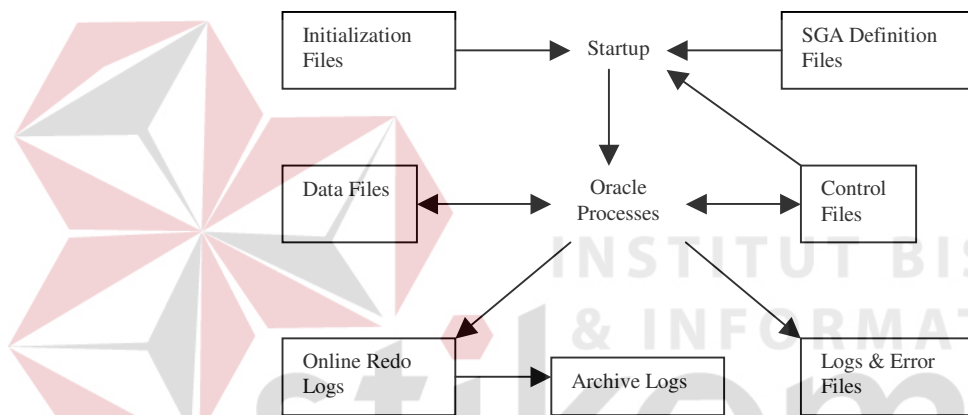
d. Data dictionary cache.

3. Program Global Area (PGA).

PGA berfungsi untuk menyimpan informasi yang digunakan dalam proses single-user. PGA berisi stack space, yaitu informasi tentang user yang *connect* dengan sebuah database instance.

4. Sort Areas.

Sort areas meningkatkan kemampuan proses pengurutan data dalam database.



Gambar 2.3. File data Oracle

2.4.3. Disk Storage

RDBMS Oracle menggunakan tujuh macam file data seperti pada Gambar

2.4., yaitu :

1. Data files.

Data files berisi record-record data yang diatur dalam tabel-tabel dan software yang tersimpan dalam database file. Software ini berupa packages dan procedures.

2. Online redo log files.

Online redo log files berfungsi sebagai backup untuk setiap perubahan yang terjadi dalam database. File ini dapat digunakan untuk memperbaiki data apabila terjadi kerusakan pada data files.

3. Archive log files.

4. Initialization files.

Initialization files berisi inisialisasi parameter dan konfigurasi Oracle yang dibaca pada saat startup Oracle database.

5. Control files.

File ini berfungsi menjaga penempatan data file dan mencatat jumlah transaksi yang paling akhir terjadi dalam database.

6. SGA definition files.

File ini berisi definisi-definisi detail dari Oracle database yang dijalankan pada saat startup database.

7. Oracle processing log dan error files.

File ini mencatat proses-proses apa saja yang terjadi serta mencatat kesalahan proses apabila terjadi suatu masalah.

2.5. Structured Query Language (SQL)

Secara umum bahasa aplikasi database berfungsi sebagai pembuat tabel, menentukan field dan isi tabel, menyediakan cara bagi tabel-tabel tersebut untuk saling berhubungan serta memanipulasi data dalam database. Bahasa pertama yang didefinisikan dalam memenuhi tugas tersebut dibuat oleh Dr. E. F. Codd dan IBM

San Jose Laboratory pada tahun 1970 dan dinamakan Structured English Query Language (SEQUEL). Kemudian generasi kedua dinamakan SQL yang oleh American National Standard Institute (ANSI) didefinisikan sebagai SQL-92 dan distandardisasi oleh International Standard Organization sebagai ISO IEC 9057:1992, “Database Language SQL”.

Bahasa SQL menyediakan perintah-perintah untuk berbagai macam tugas, yaitu:

1. Mengumpulkan data.
2. Menyisipkan (Insert), mengubah (Update), dan menghapus (Delete) data dalam sebuah tabel.
3. Menciptakan (Create), mengganti (Replace), mengubah (Alter), dan menghapus (Drop) objek-objek database.
4. Mengontrol akses ke database dan objek-objeknya.
5. Menjamin konsistensi dan integritas database.

2.6. Model Client/Server dan Sistem Terdistribusi

Model Client/server merupakan dasar sistem terdistribusi. Model ini merupakan perkembangan dari keterbatasan model mainframe client/host dimana sebuah mainframe melayani akses data untuk banyak dumb terminal. Model client/server juga merupakan perkembangan dari model local area network (LAN), dimana banyak sistem yang mengakses sebuah file server yang tidak mempunyai kemampuan pemrosesan.

Arsitektur client/server menyediakan data dan servis yang terintegrasi dan membuat client terpisah dari kesulitan-kesulitan yang kompleks seperti protokol komunikasi. Kesederhanaan dari arsitektur client/server ini memungkinkan client untuk dapat mengirimkan permintaan kepada server yang dituju. Permintaan ini dibuat dalam form transaksi. Transaksi- transaksi Client adalah berupa SQL atau prosedur-prosedur PL/SQL and fungsi-fungsi yang mengakses database dan servis.

Model client/server terdiri atas tiga komponen :

1. Client

Client adalah sebuah mesin (workstation atau PC) yang menjalankan aplikasi front end dan berinteraksi dengan user melalui keyboard, display, and mouse.

Client juga berhubungan langsung dengan proses-proses client yang dijalankan dalam mesin client. Pada client tidak secara langsung mengakses data namun mengirimkan permintaan pemrosesan kepada server dan menampilkan data yang sudah diolah oleh server. Jadi, client workstation dapat dimaksimalkan tugasnya.

2. Server

Server adalah mesin yang menjalankan software Oracle dan mengendalikan fungsi-fungsi yang dibutuhkan secara bersamaan. Server dikenal sebagai back end dan juga berhubungan dengan proses server yang berjalan dalam mesin server.

Server menerima dan memroses perintah-perintah SQL dan PL/SQL yang dikirimkan dari aplikasi client. Server juga dapat dimaksimalkan untuk menjalankan tugasnya.

3. Network

Network memungkinkan pengaksesan data jarak jauh (remote data access through) melalui komunikasi client/server dan server-to-server. Oracle memungkinkan database dan aplikasi berdiri sendiri pada mesin-mesin yang berbeda dengan sistem operasi berbeda dan tetap dapat saling berhubungan.

Keuntungan-keuntungan model client/server adalah, sebagai berikut :

1. Aplikasi-aplikasi client dapat berkonsentrasi pada permintaan input dari user, meminta data yang dibutuhkan dari server, dan menampilkan data.
2. Oracle menggunakan fasilitas multitasking and shared memory dari sistem operasi yang digunakan.
3. Multiple server dapat digunakan untuk mendistribusikan pemrosesan database melalui jaringan (horizontal scaling).
4. Penggunaan oracle pada komputer berkekuatan lebih rendah (seperti microcomputer) dengan Oracle berjalan pada sebuah minicomputer atau mainframe untuk mengambil keuntungan dari performa sistem yang lebih besar (vertical scaling).
5. Data disimpan dalam server.
6. Client dapat mengakses remote data server dengan efektif.
7. Kepadatan lalu lintas jaringan lebih rendah karena hanya permintaan-permintaan dan hasil-hasil proses saja yang melalui lalu lintas jaringan.
8. Sistem client/server memberikan kebebasan di antara komponen-komponen aplikasi dan mengurangi biaya pemeliharaan.