

# **BAB I**

## **PENDAHULUAN**

### **A. BASIS DATA & PEMAKAINYA**

#### 1. DEFINISI DASAR

- Basis data : sekumpulan data yang satu sama lain saling berhubungan  
Data : Fakta yang dapat disimpan dan mempunyai arti yang implisit
- Sifat-sifat implisit dari basis data :
  - Menyajikan sebagian dari dunia nyata.
  - Sekumpulan data yang koheren secara logika, dengan sejumlah arti yang inheren
  - Basis data perlu dirancang, dibuat dan dipopulasikan dengan data untuk suatu tujuan tertentu
- Basis data dapat dibuat dan dipelihara secara manual atau oleh mesin / computer.

Sekumpulan program yang memungkinkan pengguna basis data untuk membuat dan memelihara suatu basis data disebut : Database Management System (DBMS).

DBMS : Sistem software yang multi guna, yang menyediakan fasilitas untuk mendefinisikan, membangun dan memanipulasi basis data untuk aplikasi-aplikasi yang beraneka ragam :

#### MEMDEFINISIKAN :

melibatkan spesifikasi : tipe data, struktur, kendala (constraint) dari data yang akan disimpan.

#### MEMBANGUN :

berkaitan dengan proses penyimpanan data itu sendiri pada suatu media penyimpan yang dikontrol oleh DBMS

#### MEMANIPULASI :

termasuk di dalamnya fungsi-fungsi sebagai “query” terhadap basis data, misalnya retrieve, update, generate report

- Software yang digunakan untuk memanipulasi data + data itu sendiri disebut :  
“SISTEM BASIS DATA”

### LINGKUNGAN SISTEM BASIS DATA (SIMPLIFIED)

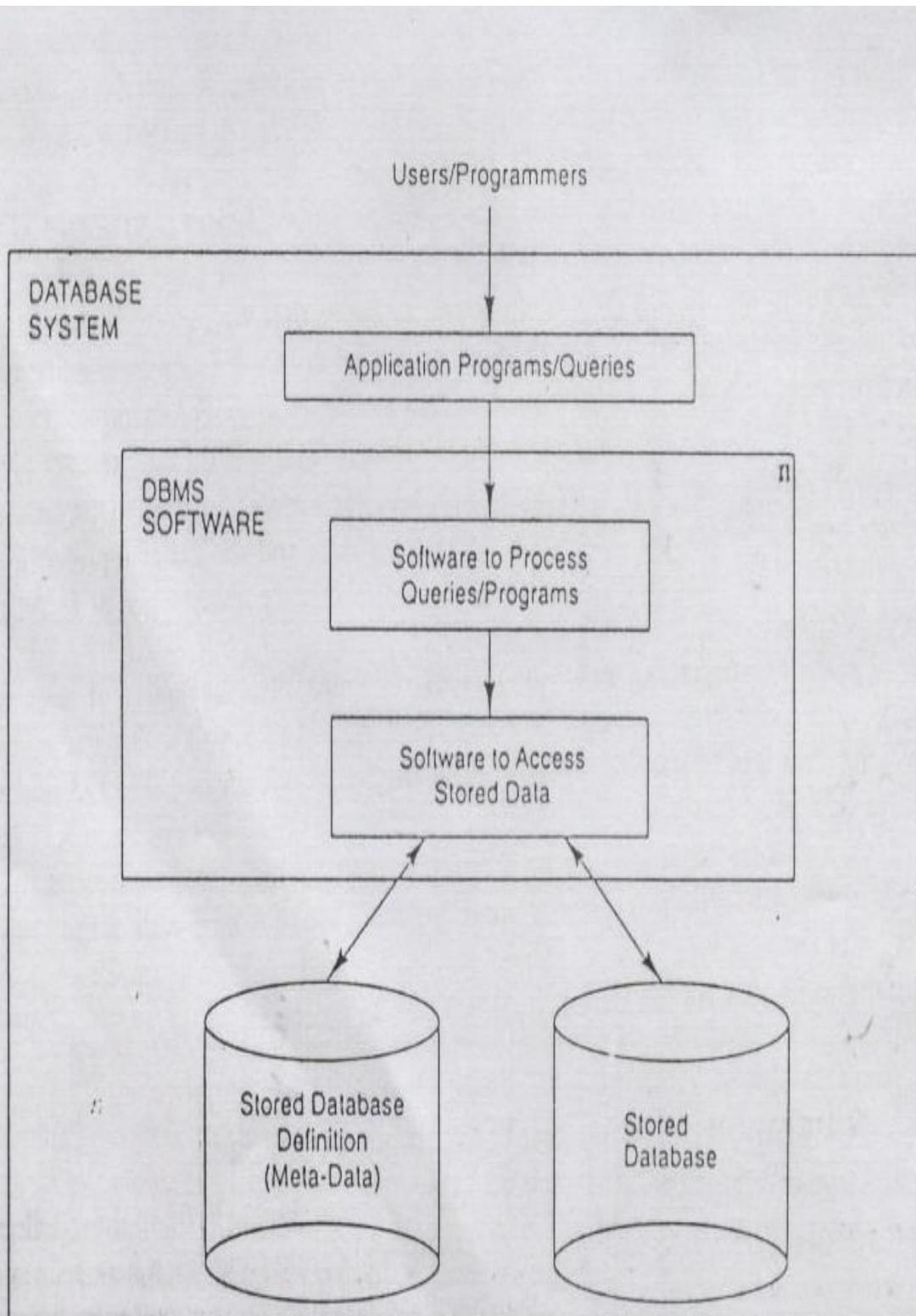


Figure 1.1 A simplified database system environment.

## 2. KARAKTERISTIK BASIS DATA

(DIBANDING DENGAN PEMROSESAN FILE TRADISIONAL)

### a. SIFAT 'SELF DESCRIBING' DARI SISTEM BASIS DATA

BASIS DATA :

- Sifat yang fundamental bahwa basis data tidak hanya berisi data saja , tapi lengkap dengan definisi dari data itu sendiri
  
- Definisi data disimpan dalam katalog system(meta data) yang berisi :
  - struktur setiap file
  - tipe dan format penyimpanan dari setiap item data
  - constraint dari data
  
- DBMS s/w dapat mengakses basis data tertentu dengan cara mengekstraksi data dari katalog dan menggunakannya

PEMROSESAN FILE

- definisi data merupakan bagian dari aplikasi program
- program hanya dapat digunakan secara spesifik untuk suatu basis data tertentu (yang strukturnya telah dideklarasikan dalam program)
- Software pada pemrosesan file hanya dapat mengakses basis data secara spesifik.

### b. "ISOLASI" ANTARA PROGRAM DAN DATA DENGAN ABSTRAKSI DATA.

BASIS DATA :

- "DBMS access program" ditulis secara terpisah dari file-file yang bersifat spesifik.
- Struktur data disimpan dalam katalog DBMS yang terpisah dari program (Program data independence)

PEMROSESAN FILE :

- Struktur data dijadikan satu dengan program (embedded), sehingga adanya perubahan data menyebabkan perubahan semua program yang mengakses data tersebut.

c. MULTI - VIEW DARI DATA

- memungkinkan user yang berbeda untuk mendapatkan perspektif (view) basis data yang berbeda
- satu view dapat berupa sub set dari basis data atau berisi basis data semu (virtual) yang diturunkan dari basis data yang ada (tetapi tidak disimpan secara eksplisit)

d. SHARING DATA DAN PEMROSESAN TRANSAKSI MULTI USER

- memungkinkan sejumlah user mengakses data secara bersamaan. Untuk ini DBMS harus menyediakan "Concurrency Control Software" sehingga data yang diakses valid.

3. ORANG-ORANG YANG BERPERAN LANGSUNG

a. Database Administrator (DBA)

- Orang yang bertanggung jawab terhadap administrasi penggunaan sumber daya basis data (basis data dan DBMS)

Bertugas :

- mengatur otorisasi akses terhadap basis data
- memonitor penggunaan basis data
- melayani permintaan s/w dan h/w

b. Database Designer

- orang yang bertanggung jawab dalam perancangan basis data (dalam memenuhi permintaan user) :
- mengidentifikasi data yang akan disimpan dalam basis data
- memilih struktur yang sesuai dalam menyajikan dalam basis data.

c. End User

- orang yang pekerjaannya memerlukan akses terhadap basis data untuk keperluan :
- query
- update
- generate report

- End users dibagi dalam 4 kategori :

1. Casual end users :

- mengakses basis data secara kadan-kadang, tetapi mungkin memerlukan informasi yang berbeda untuk setiap kalinya.
- menggunakan bahasa query yang rumit dalam memspesifikasikan query

2. Naïve/Parametric end users :

- Biasanya secara berkala melakukan query dan update basis data dengan menggunakan jenis query dan update yang standar (transaksi yang telah diprogram dan dites)

3. Sophisticated end users :

- meliputi engineers, scientists & business analysts – yang telah mengenal dengan baik dan menyeluruh mengenai fasilitas-fasilitas DBMS untuk memenuhi kebutuhan-kebuuhan yang kompleks.

4. Stand – alone users :

- Mereka yang memelihara basis data personal dengan menggunakan paket-paket program yang telah dibuat dan menyediakan menu-menu yang mudah untuk digunakan. Misal : user dari suatu “tax package”.

d. System Analysts & Application Programmers

- System analysts bertugas mendefinisikan kebutuhan-kebutuhan end user (khususnya naïve end user), dan mengembangkan spesifikasi untuk transaksi-transaksi yang memenuhi kebutuhannya.
- Application Programmers berugas mengimplemmentasikan spesifikasi menjadi program (yang telah di test secara intensif)

#### 4. ORANG-ORANG DI BELAKANG LAYAR

a. DBMS Designers & Implementers

Orang-orang yang merancang dan mengimplementasikan modul-modul DBMS dan interfacenya sebagai satu paket software

b. Tool Developers

Orang-orang yang mengembangkan paket-paket software yang memberikan fasilitas dalam perancangan dan penggunaan system basis data (misal : paket-paket untuk performance monitoring, GUI, prototyping, simulation, dlsb)

c. Operators & Maintenance Personal

5. MANFAAT PENGGUNAAN DBMS

- a. Pengendalian redundansi
- b. Pembatasan Akses (Restricting unauthorized access)
- c. Persistent storage untuk obyek-obyek program dan struktur data
- d. Inferensi basis data menggunakan aturan-aturan deduksi
- e. Tersedianya multiple user interface
- f. Dapat menyajikan relasi-relasi yang kompleks antar data yang dilibatkan
- g. Pemaksaan Integrity Constraints
- h. Tersedianya fasilitas backup & recovery

6. IMPLIKASI PENGGUNAAN PENDEKATAN BASIS DATA

- a. Mempunyai potensi untuk memaksakan standarisasi
- b. Mengurangi waktu pengembangan aplikasi
- c. Fleksibilitas
- d. Tersedianya informasi yang up to date (kekinian)
- e. Skala ekonomis

7. KAPAN UNTUK TIDAK MENGGUNAKAN DBMS

Beberapa situasi yang dapat menimbulkan biaya overhead dibandingkan dengan pemrosesan file tradisional, dalam penggunaan basis data (DBMS) antara lain disebabkan oleh:

- Investasi awal yang tinggi dalam pengadaan hardware, software & pelatihan
- Overhead untuk menyediakan security, concurrency control, backup & recovery, dan fungsi-fungsi integrity

Persoalan-persoalan lain dapat muncul jika database designer dan DBA tidak merancang database secara tepat dan benar, atau jika aplikasi-aplikasi database tidak diimplementasikan secara benar.

⇒ Untuk itu, mungkin lebih disarankan untuk menggunakan pemrosesan file biasa dalam keadaan-keadaan sbb. :

- Basis data dan aplikasi sederhana, terdefiniskan dengan baik, dan diharapkan tidak berubah
- Adanya kebutuhan-kebutuhan nyata dari program yang tidak akan terpenuhi akibat dari overhead dari DBMS
- Tidak diperlukan multiple access terhadap data.

## **B. KONSEP DAN ARSITEKTUR SISTEM BASIS DATA**

### **1. MODEL-MODEL DATA, SCHEMA & INSTANCE**

- Model data merupakan alat utama untuk menyediakan abstraksi data.
- Model data merupakan konsep yang dapat digunakan untuk menjelaskan struktur dari basis data (tipe data, relasi dan constraint)
- Model data meliputi sejumlah operasi-operasi dasar untuk menspesifikasikan retrieval & update dari basis data

#### **a. Kategori Model Data**

- Model data tingkat tinggi
  - Menggunakan konsep seperti : entity, attribute and relationship
- Model data representasional/implementasi
  - paling banyak digunakan dalam DBMS komersial
  - termasuk dalam jenis ini adalah model data relasional , jaringan dan hirarki
  - data disajikan dengan menggunakan struktur record (record-based data model)
- Model Data Fisik
  - menjelaskan bagaimana data disimpan dalam komputer (format-format : record, urutan-urutan record dan access path)

#### **b. Schemas & Instances**

- Database Schema (Meta –data)
  - diskripsi dari basis data yang dibuat pada saat perancangan dan diharapkan tidak akan sering berubah

- model data mempunyai konvensi-konvensi tertentu untuk menampilkan skema (schema diagram). Diagram skema hanya menampilkan beberapa aspek dari skema basis data, yaitu : nama tipe record, item-item data dan constraintnya.

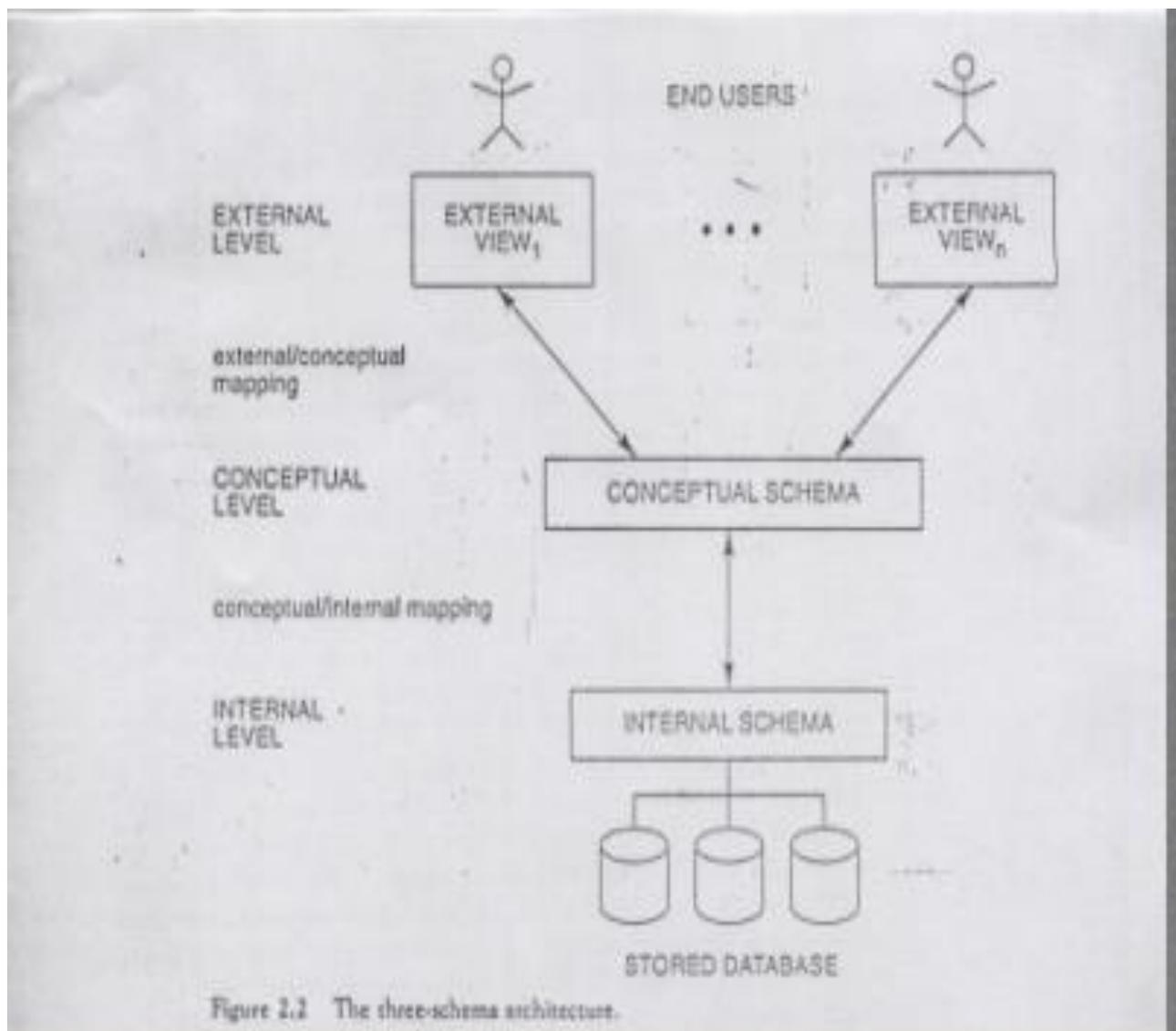
- Data Instance / Occurrence

- menyatakan nilai-nilai sebenarnya dari skema basis data untuk suatu keadaan (state) tertentu

- “Database state” merupakan keadaan-keadaan nilai yang ada pada setiap saat dari database schema yang selalu harus valid (memenuhi constraint yang telah ditentukan)

## 2. ARSITEKTUR DBMS & KEBEBASAN DATA

### a. Arsitektur Three Level (Three Schema)



- Internal Level (internal Scema)
    - menjelaskan struktur penyimpanan fisik dari basis data
    - menggunakan model data fisik
  
  - Conceptual Level (Conceptual Schema)
    - menjelaskan struktur dari keseluruhan basis data untuk dipakai oleh satu komunitas user
    - model data tingkat tinggi atau model data implementasi dapat digunakan pada level ini
  
  - External atau View Level (external schema atau user view)
    - menjelaskan sebagian basis data yang menjadi “interest” dari sekelompok user tertentu
    - model data tingkat tinggi (implementasi) dapat digunakan pada level ini.
- b. Kebebasan Data (Data Independence)
- Arsitektur Three Schema dapat digunakan untuk menjelaskan konsep ‘kebebasan data’ yang dapat didefinisikan sebagai :
- Kemampuan untuk merubah skema pada suatu level dari system basis data tanpa harus menyebabkan perubahan dari skema pada tingkat-tingkat yang lebih tinggi*
- c. Jenis Data Independence
- Logical Data Independence
    - kemampuan untuk merubah skema konseptual tanpa harus merubah skema eksternal
    - hanya definisi dari view dan mapping yang perlu diubah dalam DBMS
    - constraint dari basis data juga dapat diubah dalam skema konseptual tanpa mempengaruhi skema eksternal
  
  - Physical Data Independence
    - Kemampuan untuk merubah skema internal tanpa harus merubah skema konseptual (eksternal)

Perubahan terhadap skema internal mungkin diperlukan karena file-file fisik yang harus diorganisasikan kembali.

*Misal : membuat struktur akses tambahan untuk menambah kinerja dalam retrieval/updating.*

- Lebih mudah untuk dilakukan daripada logical data independence karena adanya isolasi aplikasi dari susunan penyimpanan fisik

### 3. BAHASA DAN INTERFACE DARI BASIS DATA

#### a. Bahasa-bahasa DBMS

- digunakan untuk membuat spesifikasi skema konseptual dan internal, dan mapping antara keduanya.
- DDL (Data Definition Language) dapat digunakan untuk menspesifikasikan kedua skema di atas, jika dalam DBMS *tidak ada* pemisahan yang ketat antara kedua level tersebut.
- DDL hanya digunakan untuk menspesifikasikan skema konseptual, jika DBMS mempunyai pemisahan yang jelas.
- Untuk mapping antara keduanya dapat digunakan VDL (View Definition Language) untuk menspesifikasikan user view dan mappingnya menjadi skema konseptual.
- DML (Data Manipulation Language) digunakan untuk melakukan manipulasi data (setelah dilakukan proses kompilasi skema konseptual)
- SQL (Structured Query Language) merupakan contoh bahasa yang digunakan untuk manipulasi basis data relasional, yang mengintegrasikan DDL, VDL dan DML.

DML terdiri dari 2 jenis.

#### 1. High Level (Non Procedural) DML

- digunakan secara interkatif (interpreter)
- dapat dijadikan satu dengan general purpose programming language (embedded)

#### 2. Low Level (Procedural) DML

- digunakan secara embedded dalam suatu general purpose prog. Language

Bilamana kedua jenis DML di atas digunakan secara “embedded”, maka :

- Bahasa pemrograman yang digunakan disebut sebagai “HOST LANGUAGE”.
- DML nya disebut “SUB LANGUAGE”.

High level DML yang hanya digunakan secara interaktif disebut “QUERY LANGUAGE”.

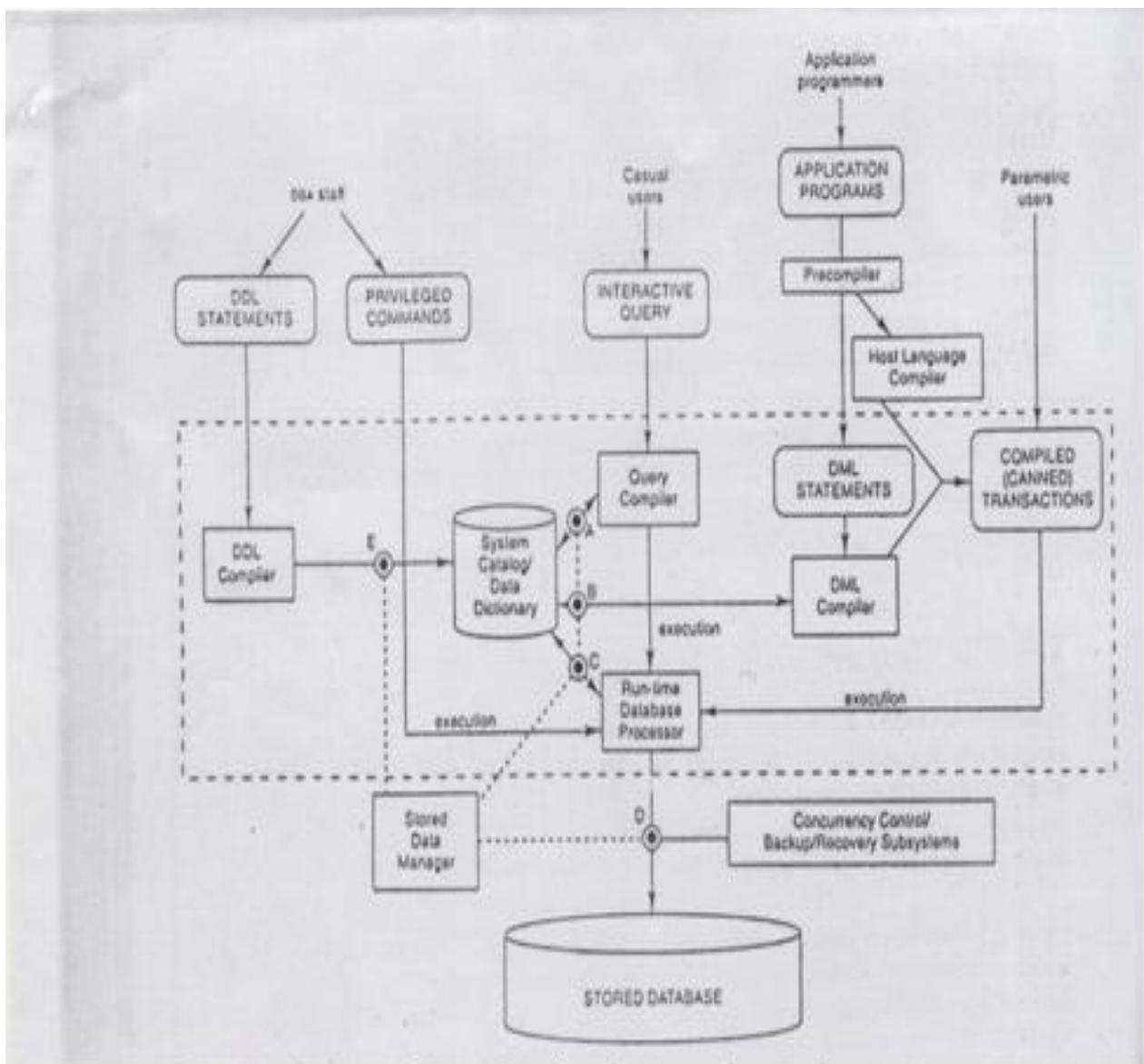
b. DBMS Interfaces :

Dibuat untuk memudahkan user (user friendly), meliputi

- Menu based
- Graphical interfase (GUI)
- Form based
- Natural language
- Interface untuk Parametric user
- Interface untuk DBA

#### 4. LINGKUNGAN SISTEM BASIS DATA

a. Modul-modul komponen DBMS



b. Utilitas-utilitas system Basis Data

Digunakan untuk membantu DBA dalam menjalankan system basis data, meliputi fungsi-fungsi :

- Loading utility
- Back up
- File reorganization
- Performance monitoring
- Utilitas penunjang untuk : sorting file, kompresi data, monitoring akses dlsb.

c. Fasilitas Komunikasi

- Untuk remoter terminal, distributed Database dll.

5. KLASIFIKASI DBMS

- a. Kriteria utama didasarkan pada model data yang disupport (relasional, hirarki, network, atau object oriented)
- b. Kriteria kedua : jumlah user yang dapat ditangani oleh DBMS
- c. Kriteria ketiga : jumlah sites (lokasi) > centralized, distributed, client server.
- d. Kriteria keempat : biaya / harga dari DBMS
- e. Kriteria lain, dapat didasarkan pada :
  - Jenis access path (mis inverted file)
  - General purpose atau special pupose (mis. OLTP : On Line Transaction Processing).