SISTEM DATABASE
BAB I
PENDAHULUAN

1. Data dan Informasi

- Data merupakan nilai (*value*) yang turut merepresentasikan deskripsi dari suatu objek atau kejadian (*event*)
- Informasi merupakan hasil dari pengolahan data dalam suatu bentuk yang lebih berguna dan lebih berarti bagi penerimanya yang menggambarkan suatu kejadian-kejadian yang nyata (*fact*) yang digunakan untuk pengambilan keputusan
- Data lebih bersifat historis, sedangkan informasi mempunyai tingkatan yang lebih tinggi, lebih dinamis, serta mempunyai nilai yang sangat penting

2. Sistem Informasi

- SI adalah suatu suatu sistem dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur dan pengendalian untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan
- SIM adalah sekumpulan elemen yang saling berhubungan, saling berinteraksi
    - dan bekerjasama antara berbagai bagian dengan cara-cara tertentu untuk melakukan fungsi pengolahan data, pemasukan data, dan menghasilkan keluaran berupa informasi yang berguna dan mempunyai nilai nyata, sebagai dasar pengambilan keputusan, mendukung kegiatan manajemen dan operasional dengan memanfaatkan berbagai sumberdaya yang ada bagi proses tersebut guna mencapai tujuan organisasi

3. Komponen sistem informasi

SI terdiri dari beberapa komponen, antara lain :

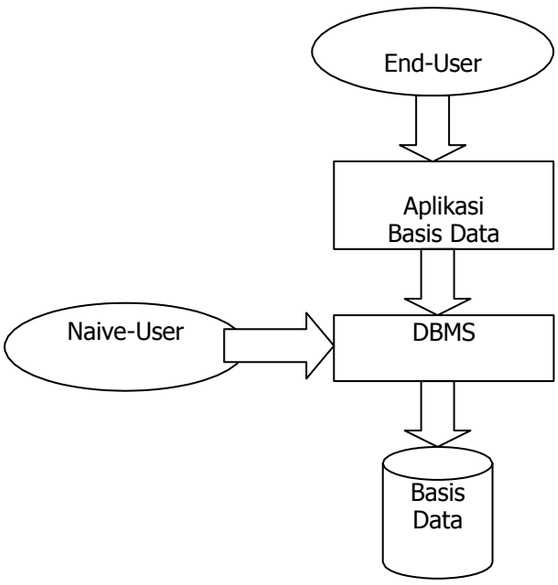
- Hardware : CPU, Disk, Terminal, Printer
- Software : Sistem operasi, sistem basis data, program aplikasi
- Personil : Operator sistem, Penyedia masukan, Pengguna keluaran
- Data : data yang tersimpan dalam jangka waktu tertentu
- Prosedur : instruksi dan kebijaksanaan untuk mengoperasikan sistem

4. **Basis data**
   a. BD adalah suatu kumpulan data terhubung yang disimpan secara bersama-sama pada suatu media, yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan dengan software untuk melakukan manipulasi untuk kegunaan tertentu
   b. **Operasi dasar basis data :**
      - Create database
      - Drop database
      - Create table
      - Drop table
      - Insert
      - Retrieve / Search
      - Update
      - Delete
   c. **Pemanfaatan basis data :**
      - Salahsatu komponen penting dalam sistem informasi, kerana merupakan dasar dalam menyediakan informasi
      - Menentukan kualitas informasi : akurat, tepat waktu dan relevan.
      - Mengurangi duplikasi data (*data redundancy*)
      - Hubungan data dapat ditingkatkan
      - Manipulasi terhadap data dengan cepat dan mudah
      - Efisiensi penggunaan ruang penyimpanan

   d. **Penerapan basis data**
      - Tidak ada sistem informasi yang yang bisa dibangun tanpa adanya basis data

   e. **Kriteria basis data :**
      - Bersifat data oriented dan bukan program oriented
      - Dapat digunakan oleh beberapa program aplikasi tanpa mengubah basis datanya
      - Dapat berkembang dengan mudah, baik volume maupun strukturnya
      - Dapat digunakan dengan cara berbeda-beda
      - Kerangkapan data minimal

5. **Sistem Manajemen Basis Data (DBMS)**
   a. Merupakan perangkat lunak yang didisain untuk melakukan penyimpanan dan pengaturan basis data
   b. DBMS juga menerapkan mekanisme pengamanan data, pemakaian data secara bersama, pemaksaan keakuratan data, dll.



6. Mengapa menggunakan DBMS

DBMS diperlukan untuk :

a. Independensi data dan akses yang efisien
b. Mereduksi waktu pengembangan aplikasi
c. Integritas dan keamanan data
d. Administrasi keseragaman data
e. Akses bersamaan dan perbaikan dari terjadinya *crash*

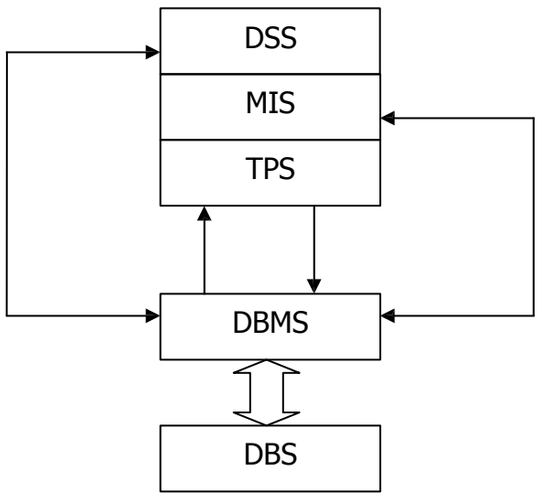
7. Peranan basis data dalam pengembangan SIM

a SIM berperan sebagai sistem karena mempunyai ruang lingkup yang relatif lebih luas dan lebih kompleks. Sedangkan sistem basis data merupakan subsistem karena menjadi bagian dan berada di dalam SIM

b Sistem basis data adalah sistem informasi yang mengintegrasikan kumpulan dari data yang saling berhubungan satu dengan yang lain dan membuatnya tersedia untuk beberapa aplikasi yang bermacam-macam di dalam suatu organisasi

c Keberadaan sistem basis data di dalam SIM adalah mutlak. SIM tidak akan terwujud tanpa melibatkan basis data

Sistem basis data sebagai infrastruktur SIM



Keterangan :

DSS : Decission Support Systems
MIS : Management Information Systems
TPS : Transaction Processing Systems
DBMS : Database Management Systems
DBS : Database Systems

BAB II
SISTEM BASIS DATA

1. Pengertian sistem basis data

SBD merupakan sekumpulan basis data dengan para pemakai yang menggunakan basis data secara bersama-sama, personil yang merancang dan mengelola basis data, teknik-teknik untuk merancang dan mengelola basis data, serta sistem komputer yang mendukungnya

2. Komponen sistem basis data

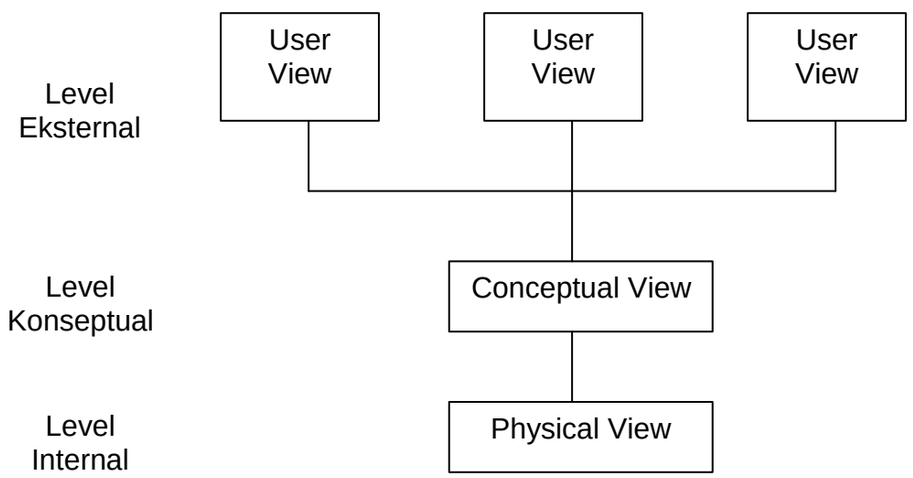
Komponen-komponen utama penyusun sistem basis data adalah :

a. Perangkat keras
b. Sistem operasi
c. Basis data
d. Sistem pengelola basis data (DBMS)
e. Pemakai (Programmer, User mahir, user umum, user khusus)

3. Abstraksi data

a Sistem basis data biasanya menyembunyikan detil tentang bagaimana data disimpan dan diperlihara. Oleh karena itu, seringkali data yang terlihat oleh pemakai sebenarnya berbeda dengan yang tersimpan secara fisik

b Abstraksi data merupakan level dalam bagaimana melihat data dalam sebuah sistem basis data

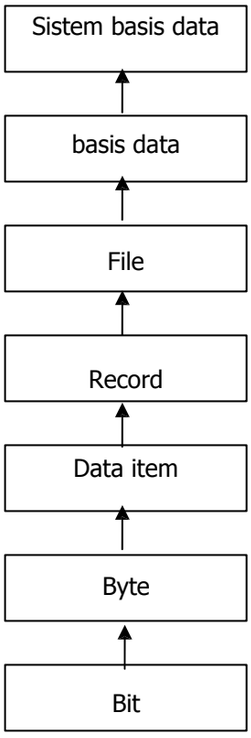


Penjelasan :

- **Conceptual view** merupakan pandangan yang berkaitan dengan permasalahan data-data apa saja yang diperlukan untuk disimpan dalam basis data dan penjelasan mengenai hubungan antar data yang satu dengan lainnya. Conceptual view dapat disetarakan dengan schema, dilakukan database administrator
- **Physical view** merupakan bentuk implementasi dari conceptual view, yaitu pandangan tentang bagaimana data disimpan dalam media penyimpan data
- **User view** dapat disejajarkan dengan sub-schema

4. Penyusun sistem basis data

Sistem basis data merupakan lingkup terbesar dalam organisasi data. Sistem basis data mencakup semua bentuk komponen data yang ada dalam suatu sistem. Sedangkan basis data merupakan komponen utama yang menyusun sistem basis data



Contoh : Data bilangan bulat (integer), Byte (1 byte), Small-Integer (2 byte), Long Integer (4 byte), Data bilangan nyata, Single (4 byte), Double (8 byte).

Keterangan :

a **Bit**, merupakan sistem angka biner yang terdiri atas angka 0 dan 1

b **Byte**, merupakan bagian terkecil, dapat berupa karakter numerik, huruf, ataupun karakter khusus yang membentuk suatu item data / field. 1 Byte digunakan untuk mengkodekan 1 karakter

c **Data item (field),** merepresentasikan suatu atribut dari suatu record yang menunjukkan suatu item dari data, misalnya nama, alamat. Kumpulan dari field membentuk suatu record

d **Record**, menggambarkan suatu unit data individu yang tertentu. Kumpulan dari record membentuk suatu file.

e **File**, terdiri dari record-record yang menggambarkan satu kesatuan data yang sejenis

f **Basis data**, sekumpulan dari berbagai macam tipe record yang mempunyai hubungan terhadap suatu objek tertentu

g **Sistem basis data**, merupakan sekumpulan basis data, yang tersusun dari beberapa file

5. Tipe File

Tipe file yang digunakan dalam sistem basis data :

a. File induk (master file)
   Ada 2 file induk :
   - File induk acuan (reference master file)
     - Recordnya relatif statis, jarang berubah nilainya
     - Contoh : file daftar gaji, matakuliah
   - File induk dinamik (dynamic master file)
     - Nilai dari recordnya sering berubah atau diupdate sebagai hasil suatu transaksi
     - Contoh : file stok barang

b. File transaksi (Transaction file)
   1. Disebut juga file input. Digunakan untuk merekam data hasil transaksi
   2. Contoh file penjualan barang

c. File laporan (report file)
   Disebut juga file output. Berisi informasi sementara yang akan ditampilkan sebagai laporan d.
   File sejarah (history file)
   1. Disebut juga file arsip (archieval file).
   2. Merupakan file yang berisi data masa lalu yang sudah tidk aktif lagi, tapi masih disimpan sebagai arsip

e. File pelindung (bacup file)
   1. Merupakan salinan dari file-file yang masih aktf di dalam basis data pada saat tertentu
   2. Digunakan sebagai cadangan apabila file basis data yang aktf mengalami kerusakan atau hilang

6. Bahasa basis data

a Bahasa basis data merupakan perantara bagi pemakai dengan basis data dalam berinteraksi, yang telah ditetapkan oleh pembuat DBMS

b Dapat dibedakan menjadi 2, yaitu :

- **Data Definition Language (DDL)**
  - Dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur tabel, dll.

- Hasil dari kompilasi perintah DDL menjadi Kamus Data, yaitu data yang menjelaskan data sesungguhnya
- Contoh : Create, Modify report, Modify structure

- **Data Manipulation Language (DML)**
  - Berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data, yang berupa insert, update, delete, dll.
  - Ada 2 jenis, yaitu prosedural (ditentukan data yang diinginkan dan cara mendapatkannya) dan non-prosedural (tanpa menyebutkan cara mendapatkannya)
  - Contoh : dbase 3+, foxbase, SQL, QBE

7. Pengguna basis data

a Secara umum dapat dikelompokkan menjadi 2, yaitu :

- **Database administrator**
  - Orang yang memiliki kewenangan untuk melakukan pengawasan baik data maupun program
  - Fungsi DBA adalah :
    - Mendefinisikan pola struktur basis data
    - Mendefinisikan struktur penyimpanan dan metode akses
    - Memodifikasi pola dan organisasi fisik
    - Memberikan kewenangan pada user untuk mengakses data
    - Menspesifikasikan keharusan integritas data

- **Database user**

  Ada 4 pemakai basis data, yaitu :

  - Programmer aplikasi
    - Merupakan pembuat program aplikasi
  - Casual user / Naïve User
    - Pemakai yang sudah mahir, berinteraksi dengan sistem tanpa menulis program, tapi menggunakan query
  - End user
    - Pemakai yang belum mahir, tinggal menjalankan aplikasi yang sudah dibuat oleh programmer aplikasi
  - Specialized user
    - Pemakai khusus yang menuliskan aplikasi database tidak dalam kerangka pemrosesan data, namun untuk keperluan khusus seperti CAD, AI, ES, dll

BAB III
LINGKUNGAN BASIS DATA

1. Kekangan dalam basis data

Penyusunan basis data digunakan untuk mengatasi masalah-masalah paa penyusunan data, yaitu :

a. Redundansi data

- Yaitu munculnya data-data yang sama secara berulang-ulang pada beberapa file basis data yang semestinya tidak diperlukan
- Akan mengakibatkan proses updating lebih lama dan memungkinkan terjadinya *inconsistency data*

  Contoh :

  File Mahasiswa Nama text(20), Nomhs text(10), alamat text(40)
  File KRS Nama text(20), Nomhs text(10), Jml_Mtk integer
  File Dosen NIK text(10), Nama text(30), Gol text(4), Gapok double

b. Inkonsistensi data

- Yaitu munculnya data yang tidak konsisten pada field yang sama untuk beberapa file dengan kunci yang sama
- Terjadi akibat kesalahan dalam pemasukan data atau update data.

  Akan mengakibatkan kesalahan pada hasil pengolahan basis data yang tidak sesuai dengan fakta
- Contoh : pada file mahasiswa dan krs diatas

c. Isolasi data untuk standarisasi

- Disebabkan oleh pemakaian beberapa file basis data yang tersebar dalam beberapa file, hal ini menyulitkan programmer untuk mengambil dan menyimpan data
- Contoh : akan sulit apabila data tersimpan dalam format text, BASIC, dll

d. Banyak pemakai (multi user)

Basis data dapat diakses oleh beberapa pemakai secara simultan, karena data yang diolah tidak bergantung dan menyatu dalam program tapi terlepas dalam satu kelompok data

e. Masalah keamanan (security)

- Pada prinsipnya file basis data hanya boleh diakses oleh pemakai tertentu yang mempunyai wewenang.

- Pembatasan dapat dilakukan melalui DBMS atau program aplikasi

f. Masalah integritas (integrity)

- Untuk menjaga agar unjuk kerja sistem tetap dalam pengendalian penuh.
- Secara teknis maka ada kunci primer yang menghubungkan beberapa file yang saling berkaitan

g. Masalah kebebasan data (independence)

- Basis data yang dirancang hendaknya tidak bergantung pada program aplikasi yang dibangun
- Sehingga apabila ada perubahan thd field, tidak perlu merubah programnya

2. Organisasi file basis data

a. Tujuan organisasi file dalam sistem basis data:

- Menyediakan sarana pencarian record bagi pengolahan, seleksi, atau penyaringan
- Memudahkan pembuatan atau pemeliharaan file

b. Ada 2 jenis media penyimpan file :

a) SASD (Sequential Access Storage Device)
   a. Proses pembacaan record harus berurutan
   b. Tidak ada pengalamatan
   c. Data disimpan dalam bentuk blok
   d. Proses penulisan hanya bisa dilakukan sekali
      Contoh : magnetic tape

b) DASD (Direct Access Storage Device)
   a. Pembacaan record tidk harus urut
   b. Mempunyai alamat
   c. Data dapat disimpan dalam karakter atau blok
   d. Proses penulisan dapat dilakukan beberapa kali
   e. Contoh : harddisk, floppy disk

c. Metode susunan file :

a). Sequential (urut)
- Record disimpan berdasarkan suatu kunci
- Pencarian record tertentu dilakukan record demi record berdasarkan kuncinya

b). Random (Acak)
- Kunci record ditransformasikan ke alamat penyimpanan dalam media fisik secara acak

c). Indexed Sequential
- Merupakan gabungan antara metode urut dan acak
- Record disimpan secara berurutan dengan menggunakan kunci
- Masing-masing record memiliki indeks
- Pengalamatan dilakukan secara acak

d). Indexed Random
- Record disimpan secara acak
- Masing-masing record memiliki indeks

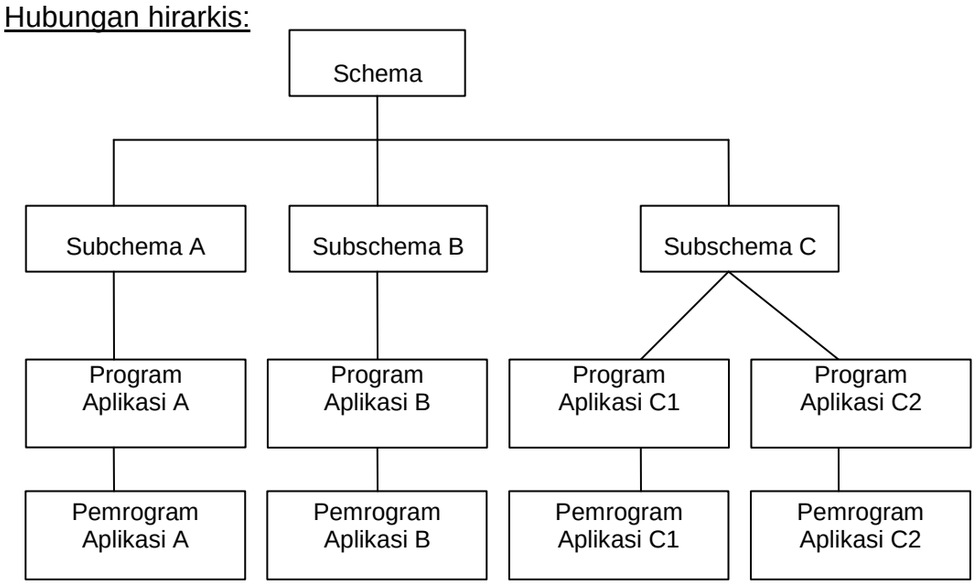
3. Schema dan Subschema

Schema dan Subschema diperlukan untuk menggambarkan hubungan logik antara data dalam basis data

a. Schema, memberikan deskripsi hubungan logik secara lengkap dari basis data, yang meliputi rinci data, record, set, dan area untuk aplikasi yang menggunakan basis data tersebut

b Subschema, merupakan deskripsi terpisah dari rinci data, record, set dan area yang digunakan oleh program aplikasi

Hubungan hirarkis:



4. Arsitektur sistem basis data

a. Pertimbangan dalam memilih arsitektur sistem basis data :

- Keunggulan teknologi
- Biaya pengembangan
- Sesuai dengan kebutuhan pengguna

b. Jenis arsitektur sistem basis data:

- **Sistem tunggal (Standalone)**
  - DBMS,basis data,dan aplikasi basis data ditempatkan pada komputer yang sama.
  - Hanya bisa dipakai oleh satu pemakai pada saat yang bersamaan
- **Sistem Terpusat (Centralized system)**

  Terdiri dari sebuah server dan sejumlah terminal

  Yang terpusat adalah basis data, DBMS, dan aplikasi basis data

  Ada dua macam :

  Aplikasi dan basis data terpusat; diakses oleh dumb terminal

  Basis data terpusat; aplikasi ada pada terminal
- **Sistem Client-server**

  Ditujukan untuk mengatasi kelemahan yang terdapt pada sistem terpusat

  Terdiri dari 2 komponen utama yaitu client dan server. Client berisi aplikasi basis data; server berisi DBMS dan basis data

  Ada dua macam :

  Arsitektur 2 lapis (2-tier)

  Arsitektur 3 lapis (3-tier)

5. Konsep DBMS

DBMS(Data Base Management System) adalah perangkat lunak yang memberikan fasilitas untuk melakukan fungsi pengaturan, pengawasan, pengendalian, pengolahan, dan koordinasi terhadap semua proses yang terjadi pada sistem basis data

Komponen-komponen utama DBMS :

Query language

- Digunakan oleh bagian lain dengan sedikit perintah sederhana
- Contoh : SQL (Structure Query Language), QBE (Query By Example)

Report generator

- Dirancang untuk membuat cetakan, yang memiliki perintah-perintah untuk membuat header, judul, kolom, summary, dll.

DML (Data Manipulation Language)

- Terdiri dari perintah-perintah yang disediakan dalam program aplikasi untuk melakukan manipulasi data seperti append, list, atau update

DDL (Data Definition Language)

- Dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, menentukan struktur tabel, dll.
- Hasil dari kompilasi perintah DDL menjadi Kamus Data, yaitu data yang menjelaskan data sesungguhnya
- Contoh : Create, Modify report, Modify structure

Recovery

- Merupakan kemampuan untuk mengembalikan data yang rusak atau hilang akibat operasi basis data (insert, update, delete, dll.)

Data dictionary

- Digunakan untuk memelihara definisi-definisi standar seluruh rinci data dalam lingkup kecil pada sistem basis data

Database

- Merupakan bagian dari DBMS yang menyediakan data dalam berbagai tipe dan format untuk memenuhi kebutuhan pemakai

Access routine

- Suatu rutin yang dapat dipanggil dan dipergunakan oleh program lain untuk mengakses basis data

6. Kamus data

DBMS memberikan fasilitas data dictionary (kamus data) untuk mendefinisikan nama-nama rinci data dan format penyimpanannya

Kamus data digunakan untuk :

a. Pada tahap analisis, sebagai alat komunikasi antara analis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem
b. Pada tahap perancangan sistem, digunakan untuk merancang input, laporan-laporan dan database

Kamus data berisi : Nama arus data, alias, bentuk data, arus data, penjelasan atau keterangan-keterangan, periode terjadinya transaksi, volume arus data yang mengalir dalam periode tertentu, struktur data

7. Model data

Model data merupakan suatu cara untuk menjelaskan bagaimana pemakai dapat melihat data secara logik

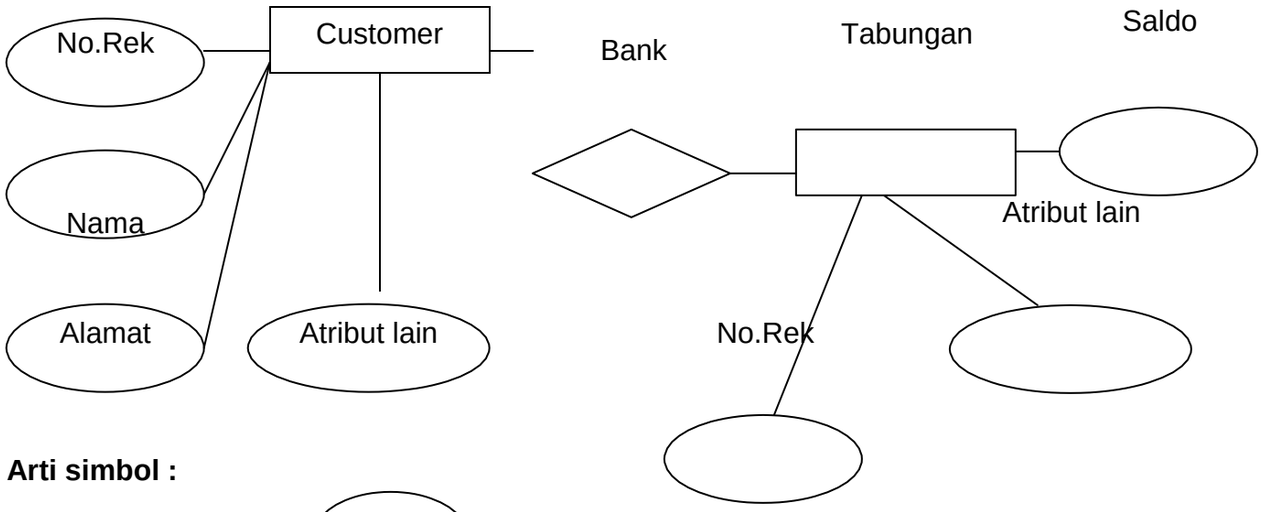
Ada 3 jenis model data :

a. Model data berbasis objek

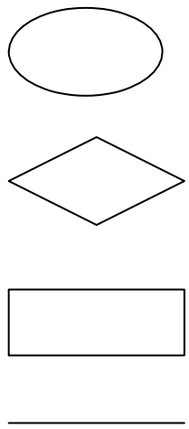
- Merupakan himpunan data dan relasi yang menjelaskan hubungan logik antar data dalam suatu basis data berdasarkan objek datanya
- Terdiri dari 2 jenis :
  - Entity Relationship model

    Merupakan model untuk menjelaskan hubungan antar data dalam basis data berdasarkan persepsi bahwa real world (dunia nyata) terdiri dari objek-objek dasar yang mempunyai hubungan / relasi antara objek tersebut

    Contoh :



Arti simbol :



Objek dasar

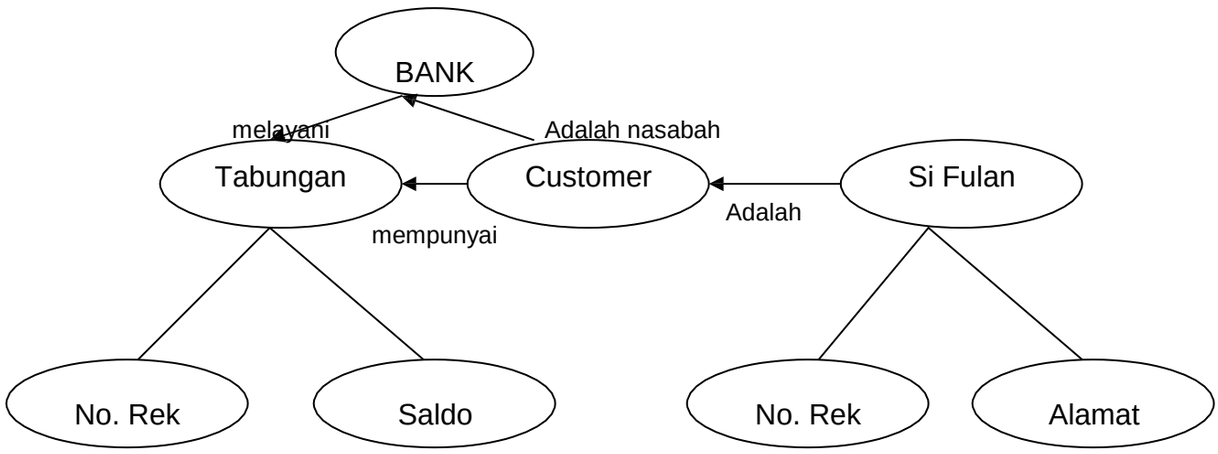
Relasi

Atribut dair objek dasar

Adanya hubungan

- Semantic model

  Relasi antar objek dinyatakan dengan kata-kata (semantic)
  Contoh



Arti tanda :

Menunjukkan adanya relasi

Menunjukkan atribut

b. Model data berbasis record

- Model ini mendasarkan pada record untuk menjelaskan kepada user tentang hubungan logik antar data dalam basis data
- Ada 3 jenis :

  - Relational Model

    Menjelaskan tentang hubungan logik antar data dalam basis data dengan memvisualisasikan ke dalam bentuk tabel-tabel yang terdiri dari sejumlah baris dan kolom yang menunjukkan atribut tertentu

    Lebih mudah dipahami dibandingkan model-model lainnya

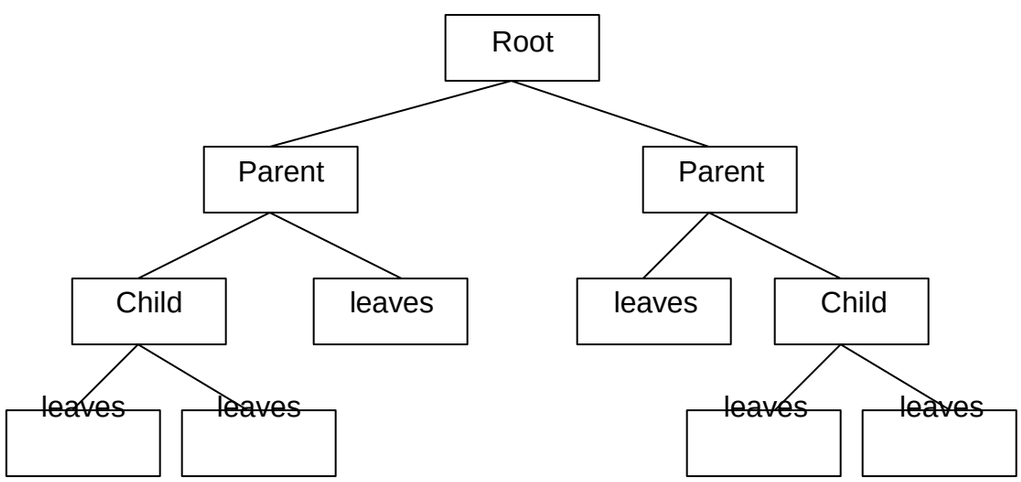
    Contoh :
    MAHASISWA

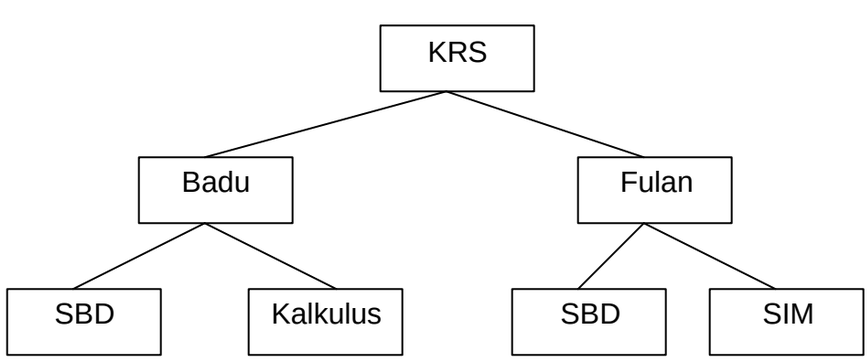
Nomhs	Nama
00351234	Fulan
01351346	Badu
02351370	Ayu

Keterangan :
- Jumlah kolom disebut *degree*, ada 2
- Baris disebut *atribut*, ada 3
- Tiap baris disebut record / tuple, ada 3 record
- Banyaknya baris dalam satu tabel disebut *cardinality*

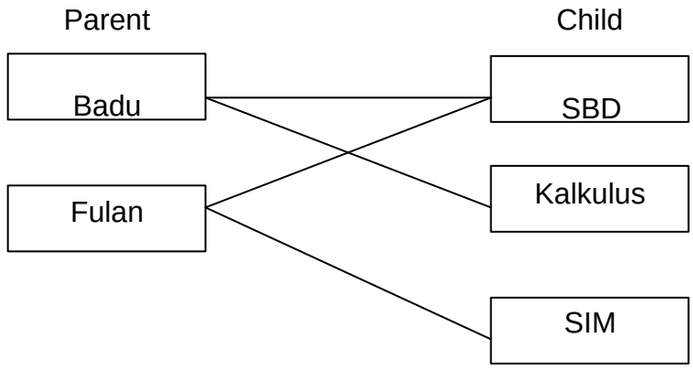
- Hirarchycal Model (Tree structure)
  - Menjelaskan tentang hubungan logik abtar data dalam basis data dalam bentuk hubungan bertingkat (hirarki)
  - Elemen penyusunnya disebut node, yang berupa rinci data, agregat data, atau record
  - Contoh :

Model hirarki





- Network Model (Plex structure)
  - Hampir sama dengan model hirarki, dan digambarkan sedemikian rupa sehingga child pasti berada pada level yang lebih rendah daripada parent
  - Sebuah child dapat mempunyai lebih dari satu parent
  - Contoh :



c. Model data berbasis fisik

- Digunakan untuk menjelaskan kepada pemakai bagaimana data-data dalam basis data disimpan dalam media penyimpanan secara fisik, yang lebih berorientasi pada mesin
- Ada 2 model :
  - Unifying model
  - Frame memory

BAB IV
RELATIONAL DATABASE MODEL

1. Terminologi

Model ini menjelaskan tentang hubungan logik antar data dalam basis data dengan cara memvisualisasikan ke dalam bentuk tabel dua dimensi yang terdiri dari sejumlah baris dan kolom yang menunjukan atribut-atribut

Istilah-istilah dalam model basis data relasional :

a. Record : sebuah baris dalam suatu relasi. Disebut juga tuple
b. Cardinality : banyaknya record dalam sebuah relasi
c. Atribut : suatu kolom dalam sebuah relasi
d. Domain : batasan nilai dalam atribut dan tipe datanya
e. Derajat / degree : banyaknya kolom dalam relasi
f. Candidate Key : atribut atau sekumpulan atribut yang unik yang dapat digunakan untuk membedakan suatu record
g. Primary Key : salah satu dari CK yang dipilih dan dipakai untuk membedakan suatu record
h. Alternate key : CK yang tidak dipilih menjadi PK
i. Unary relation : suatu relasi yang hanya mempunyai satu kolom
j. Binary relation : suatu relasi yang hanya mempunyai dua kolom
k. Ternary relation : suatu relasi yang mempunyai tiga kolom

2. Karakteristik model basis data relasional

Relasi dalam model basis data relasional memiliki karakteristik :

a. Semua entry / elemen data pada suatu baris dan kolom tertentu harus mempunyai nilai tunggal (single value), atau suatu nilai yang tidak dapat dibagi lagi (atomic value), bukan suatu kelompok pengulangan
b. Semua entry / elemen data pada suatu kolom tertentu dalam rlasi yang sama harus mempunyai jenis yang sama
c. Masing-masing kolom dalam sutu relasi mempunyai nama yang unik
d. Pada suatu relasi / tabel yang sama tidak ada dua baris yang identik

3. Komponen relasi

Tabel relasional mempunyai 2 komponen :

a. Intention

   Terdiri dari dua bagian yaitu struktur penamaan (naming structure) dan batasan integritas (integrity constraint)

Struktur penamaan menunjukkan nama tabel dan nama atribut yang ada lengkap dengan dengan batasan nilai dan tipe datanya
Batasan integritas dipengaruhi oleh integritas referential yang meliputi key constraint dan referensial constraint.
Key constraint tidak mengijinkan adanya nilai null pada atribut yang digunakan sebagai PK
Referentil constraint memberikan aturan bahwa nilai-nilai dalam atribut kunci yang digunakan untuk menghubungkan ke basis data lain tidak diijinkan memiliki nilai null

b. Extention

Menunjukkan isi dari tabel-tabel pada suatu waktu, cenderung berubah sewaktu-waktu

4. Kunci relasi

Dasar penentuan PK adalah bahwa nilai-nilai rinci data dari atribut yang digunakan sebagai PK harus unik, tidak mungkin ada nilai rinci data yang sama pada semua record dalam basis data

Aturan-aturan lainnya :

Integritas entity

- Nilai atribut yang dipilih sebagai PK tidakboleh null untuk setiap record yang ada dalam relasi
- Aturan ini menjamin bahwa semua record yang ada dalam basis data akan dapat diakses karena semua record dapat diidentidikasi berdasarkan kunci yang unik
- Contoh :

Nomhs *)	**Nama**	**Sex**
123456	Ali baba	L
123457	Pipiyot	P
123467	Nirmala	P

*) Primary key

Integritas referensial

- Jika dua buah tabel direlasikan maka PK harus menjamin bahwa untuk setiap nilai PK tertentu dalam tabel A, harus ada pula record dengan nilai PK yang sama pada tabel B
- Contoh :

Tabel mahasiswa

Nomhs *)	**Nama**	**Sex**
123456	Ali baba	L
123457	Pipiyot	P
123467	Nirmala	P

Tabel KRS

Nomhs *)	**JMTK**	**JSKS**
123456	7	21
123457	6	18
123467	6	19
123455	4	16

→ Tidak ada dalam tabel mahasiswa

5. **Relasi antar entity**

Ada dua jenis :

Relasi antar entity dalam satu tabel

- Berupa relasi antar entity yang berupa record untuk menyediakan data atau informasi dari atribut-atribut dalam satu tabel
- Contoh : dalam tabel mahasiswa dapat diperoleh informasi bahwa nomhs 12346 bernama Ali baba dengan jenis kelamin laki-laki

Relasi antar entity dalam banyak tabel

- Tipe ini mempunyai kerelasian yang lebih rumit
- Ada 3 jenis : Tree, Simple network, Complex network
- Contoh : Sistem basis data, Edhy, hal. 60 - 66

o Yang harus diperhatikan adalah bagaimana agar relasi-relasi yang ada dalam sistem basis dta dapat dihubungkan satu sama lain

6. **Basis data yang baik**

Pembentukan basis data yang baik akan memberikan sejumlah keuntungan :

Tabel-tabel dan relasi lebih kompak
Struktur masing-masing tabel lebih efisien dan sistematik
Kebutuhan ruang penyimpanan data lebih efisien
Redundansi data yang optimal akan meningkatkan integritas data
Tidak ada ambiguitas data disemua tabel

BAB V
NORMALISASI

1. Pengertian

Normalisasi adalah suatu teknik yang menstrukturkan data dalam cara-cara tertentu untuk membantu mengurangi atau mencegah timbulnya masalah yang berhubungan dengan pengolahan data dalam basis data

Kriteria yang mendefinisikan level-level pada normalisasi adalah bentuk normal (*norm form*)

2. Tujuan normalisasi

Normalisasi perlu dilakukan agar kerelasian dalam basis data menjadi mudah dimengerti, mudah dipelihara, mudah memprosesnya, dan mudah untuk dikembangkan sesuai kebutuhan baru

3. Penyimpangan dalam modifikasi

Penyimpangan dalam proses modifikasi data disebut anomalies

Ada 3 bentuk penyimpangan :

a. Delete anomalies

Adalah proses penghapusan suatu entity logik yang mengakibatkan hilangnya informasi tentang entity yang tidak direlasikan secara logik

Contoh :

Tabel Kuliah

Nomhs	Nama	Kode Mtk	SKS
123456	Ali baba	INA 101	3
123457	Pipiyot	TFD 234	2
123467	Nirmala	INA 201	3
123445	Lala	INA 101	3

Apabila “Ali baba” membatalkan mengambil matakuliah “INA 101”, maka apabila record tersebut dihapus akan menyebabkan seluruh informasi tentang ‘Ali baba” akan ikut terhapus

b. Insert anomalies

Adalah proses penyisipan entity logik yang memerlukan penyisipan entity logik yang lain

c. Update anomalies

Adalah proses mengupdate data pada suatu entity logik yang mengakibatkan perubahan pada lebih dari satu tempat dalam suatu relasi

Contoh : Perubahan SKS pada “INA 101” tidak hanya dilakukan pada satu record saja, tetapi pada record dan relasi lain yang memuat data tersebut

4. Keharusan menghilangkan masalah-masalah akibat ketergantungan

Yang harus dilakukan adalah jika struktur data dalam relasi dirancang sedemikian rupa sehingga atribut-atribut bukan kunci hanya tergantung pada atribut kunci dan tidak pada atribut lain

Ada 3 ketergantungan :

a. Functional Dependence (FD)

- FD akan muncul diantara dua rinci data dalam suatu struktur data jika nilai salah satu rinci data mengimplikasikan nilai pada rinci data kedua
- Atau rinci data pertama menentukan (determines) rinci data kedua
- Contoh :

  Matakuliah (Kode, Nama, SKS, Semester)
  FD = Matakuliah.Kode Æ (Matakuliah.Nama, Matakuliah.Semester)
  Matakuliah.nama Æ (Matakuliah.Kode, Matakuliah.Semester)

b. Full Functional Dependence (FFD)

- Suatu rinci data dikatakan FFD pada suatu kombinasi rinci data jika

c. Transitive Dependence (TD)

- Muncul jika suatu nilai pada rinci data pertama menentukan nilai pada rinci data kedua yang bukan CK, dan nilai pada rinci data kedua menentukan nilai pada rinci data ketiga
- Jadi TD terjadi jika suatu nilai rinci data mempunyai ketergantungan dengan pada dua nilai rinci data

5. Efek-efek normalisasi

Akibat yang muncul dalam proses normalisasi :

a. Masalah kekangan dalam basis data

- Duplikasi rinci data
- Adanya Integritas referensial yang harus terjaga dan nilai-nilai pada AK tidak boleh null maka proses dekomposisi akan menghasilkan suatu set yang yang inheren pada batasan integritas referensial

b. Ketidakefisienan dalam menampilkan kembali data tersebut

6. Atribut tabel

Atribut adalah karakteristik atau sifat yang melekat pada sebuah tabel, atau disebut juga kolom data

Pengelompokan atribut :

a. Atribut Key

Adalah satu atau gabungan dari beberapa atribut yang dapat membedakan semua baris data dalam tabel secara unik (tidak boleh ada dua atau lebih baris data dengan nilai yang sama untuk atribut tetentu)

Ada 3 key :

Superkey

- Merupakan satu atau kumpulan atribut yang dapat membedakan setiap baris data dalam sebuah tabel secara unik
- Contoh : superkey di tabel mahasiswa
  - (nomhs, nama, alamat, tgllahir)
  - (nomhs, nama, tgllahir)
  - (nomhs, nama)
  - (nomhs)

Candidate key

- Merupakan kumpulan atribut minimal yang dapat membedakan setiap baris data dalam sebuah tabel secara unik
- Sebuah CK pasti superkey, tapi belum tentu sebaliknya
- Contoh : pada tabel mahasiswa
  - (nomhs)
  - (nama)

Primary key

- Dari beberapa CK dapat dipilih satu untuk dijadikan PK, yang memiliki keunikan paling baik
- Contoh : dari tabel mahasiswa, yang layak dijadikan PK adalah nomhs

b. Atribut deskriptif
Merupakan atribut yang bukan merupakan anggota dari PK

c. Atribut sederhana
Adalah atribut atomik yang tidak dapat dipilah lagi
Contoh : Nomhs, Nama d.

Atribut komposit
Adalah atribut yang masih bisa diuraikan lagi menjadi sub-atribut yang masing-masing memiliki makna
Contoh : Alamat Æ Alamat, Kota, Propinsi, Kode Pos

e. Atribut bernilai tunggal
Ditujukan pada atribut-atribut yang memiliki paling banyak satu nilai untuk setiap baris data
Contoh : Nomhs, Nama, Tanggal lahir hanya dapat berisi satu nilai untuk seorang mahasiswa

f. Atribut bernilai banyak
Ditujukan pada atribut-atribut yang dapat diisi dengan lebih dari satu nilai, tapi jenisnya sama
Contoh : pada tabel mahasiswa dapat ditambah atribut HOBBY, karena seorang mahasiswa dapat memiliki beberapa hobby

g. Atribut harus bernilai (mandatory)
Adalah atribut yang nilainya tidak boleh kosong, atau harus ada nilainya. Misalnya data Nomhs dan Nama mahasiswa
Nilai NULL digunakan untuk mengisi atribut yang demikian yang nilainya belum siap atau tidak ada
NULL (karakter ke 0) tidaksama dengan SPASI (karakter ke 32)

7. Domain dan tipe data

Domain, memiliki pengertian yang hampir sama dengan tipe data, namun domain lebih ditekankan pada batas-batas nilai yang diperbolehkan pada suatu atribut
Contoh : data SKS bertipe integer. Namun dalam kenyataan tidak ada sks yang bernilai negatif. Berarti domain nilai sks adalah integer > 0

Tipe data merujuk pada kemampuan penimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat kelayakan data tersebut bila dilihat dari kenyataan pemakaiannya

8. Bentuk-bentuk normal

Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data, teknik pengelompokan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi)

Bentuk-bentuk normal :

a. Normal pertama (1st normal form)
   - Aturan :
     - Mendefinisikan atribut kunci
     - Tidak adanya grup berulang
     - Semua atribut bukan kunci tergantung pada atribut kunci

b. Normal kedua (2nd normal form)
   - Aturan :
     - Sudah memenuhi bentuk normal pertama
     - Tidak ada ketergantungan parsial (dimana seluruh field hanya tergantung pada sebagian field kunci)

c. Normal ketiga (3rd normal form)
   - Aturan :
     - Sudah berada dalam bentuk normal kedua
     - Tidak ada ketergantungan transitif (dimana field bukan kunci tergantung pada field bukan kunci lainnya)

d. Normal Boyce-Codd (Boyce Codd Norm Form)
   - Aturan :
     - Sudah berada dalam bentuk normal ketiga
     - Semua determinannya merupakan candidate key

Catatan :

- Bentuk normal seharusnya berada dalam bentuk normal tertinggi dan bergerak dari bentuk normal pertama dan seterusnya untuk setiap kali membatasi hanya satu jenis redundansi
- Keseluruhan ada 5 bentuk normal. Tiga bentuk normal pertama menekankan redundansi muncul dari *Functional Dependencies*, sedangkan bentuk keempat dan kelima menekankan redundansi yang muncul dari kasus *Multi Valued Dependencies*

9. Contoh kasus

BAB VI
ENTITY RELATIONSHIP DIAGRAM

1. Pengantar

ERD merupakan notasi grafis dalam pemodelan data konseptual yang mendeskripsikan hubungan antara penyimpanan

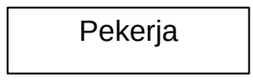
ERD digunakan untuk memodelkan struktur data dan hubungan antar data, karena hal ini relatif kompleks.

Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilakukan. Dengan ERD kita mencoba menjawab pertanyaan seperti :

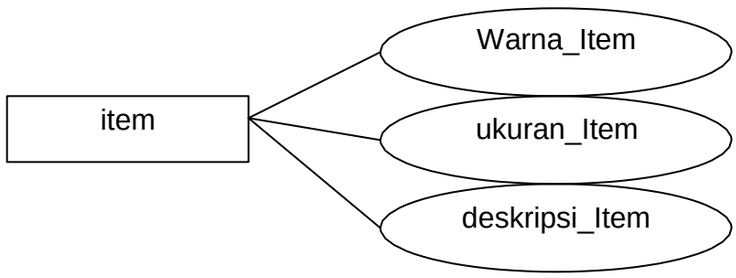
- Data apa yang diperlukan ?
- Bagaimana data yang satu berhubungan dengan yang lain ?

2. Notasi dan artinya

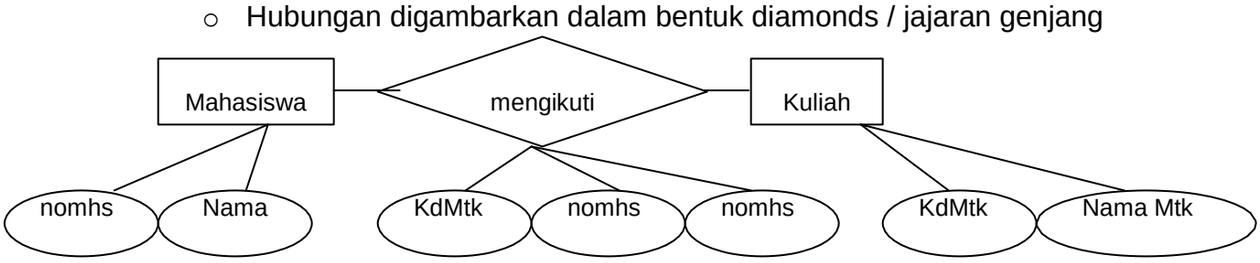
a. **Entiti** : adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai, sesuatu yang penting bagi pemakai dalam konteks sistem yang akan dibuat. Sebagai contoh pelanggan, pekerja, mahasiswa, dll.
   - Contoh : Seandainya A adalah seorang pekerja maka A adalah isi dari pekerja, sedangkan jika B adalah seorang pelanggan maka B adalah isi dari pelanggan.
   - Karena itu harus dibedakan antara entiti sebagai bentuk umum dari deskripsi tertentu dan isi entiti seperti A dan B dalam contoh diatas.
   - Himpunan entitas : merupakan sekelompok entitas sejenis dan berada dalam lingkup yang sama. Misalnya Mobil merupakan himpunan entitas; sedangkan suzuki, toyota, honda merupakan entitas
   - Entiti digambarkan dalam bentuk persegi panjang



b. **Atribut** : Entiti mempunyai elemen yang disebut atribut, dan berfungsi mendeskripsikan karakter entiti. Misalnya atribut nama pekerja dari entiti pekerja.
   - Setiap ERD bisa terdapat lebih dari satu atribut
   - Entiti digambarkan dalam bentuk elips



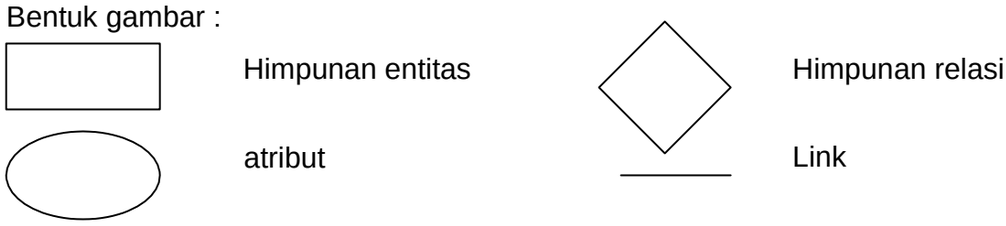
c. **Hubungan** : menunjukkan adanya hubungan / relasi diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda
   - sebagaimana halnya entiti maka hubunganpun harus dibedakan antara hubungan atau bentuk hubungan antar entiti dengan isi dari hubungan itu sendiri.
   - Misalnya dalam kasus hubungan antara entiti siswa dan entiti mata_kuliah adalah mengikuti, sedangkan isi hubungannya dapat berupa nilai_ujian.
   - Hubungan digambarkan dalam bentuk diamonds / jajaran genjang



3. Notasi simbolik dalam diagram E-R

- Persegi panjang Æ himpunan entitas
- Elips Æ atribut (atribut yang sebagai kunci digarisbawahi)
- Belah ketupat Æ himpunan relasi
- Garis Æ penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya
- Kardinalitas relasi dinyatakan dengan banyaknya garis cabang atau dengan pemakaian angka (1 dan 1 untuk relasi satu-ke-satu, 1 dan M untuk relasi satu-ke-banyak, M dan M untuk relasi banyak-ke-banyak)

- Bentuk gambar :



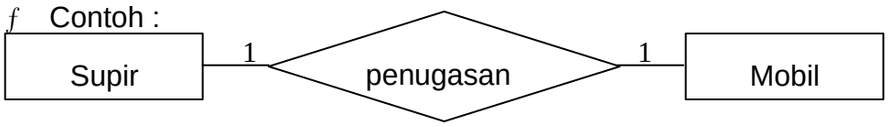
4. Jenis-jenis hubungan / Derajat Relasi / Kardinalitas relasi

- Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berrelasi dengan entitas pada himpunan entitas yang lain
- Contoh : entitas-entitas pada himpunan entitas Mahasiswa dapat berelasi dengan satu entitas, banyak entitas, atau bahkan tidak satupun entitas dari himpunan entitas Kuliah

- Jenis-jenis hubungan :

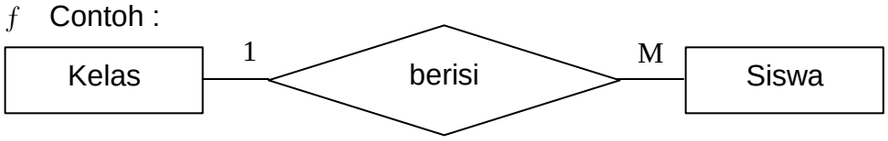
a. satu ke satu (one to one)

- setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan begitu sebaliknya setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A
- Contoh :



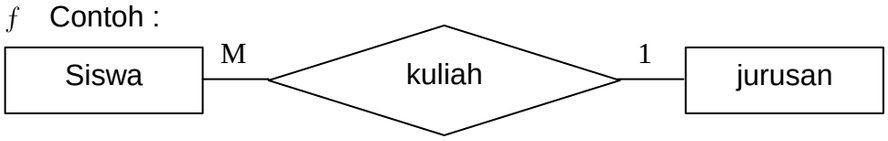
b. satu ke banyak (one to many)

- setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B berhubungan dengan paling banyak satu entitas pada himpunan entitas A
- Contoh :



c. banyak ke satu (many to one)

- setiap entitas pada himpunan entitas A berhubungan dengan paling banyak satu entitas pada himpunan entitas B, dan tidak sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A
- Contoh :



d. banyak ke banyak (many to many)

- setiap entitas pada himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunan entitas B, dan sebaliknya dimana setiap entitas pada himpunan entitas B dapat berhubungan dengan banyak entitas pada himpunan entitas A
- Contoh :

5. Tahapan pembuatan Diagaram E-R

- Ada dua kelompok pentahapan yang biasa ditempuh dalam pembuatan Diagram E-R :

a. Tahap awal pembuatan *(preliminary design)*

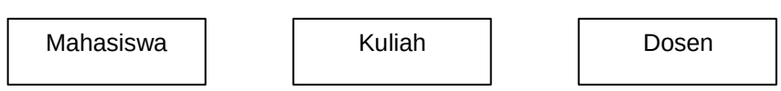
- Untuk mendapatkan rancangan basis data minimal yang dapat mengakomodasi kebtuuhan penyimpanan data terhadap sistem yang akan dibangun
- Pada umumnya mengabaikan adanya penyimpangan-penyimpangan

b. Tahap optimasi *(final design)*

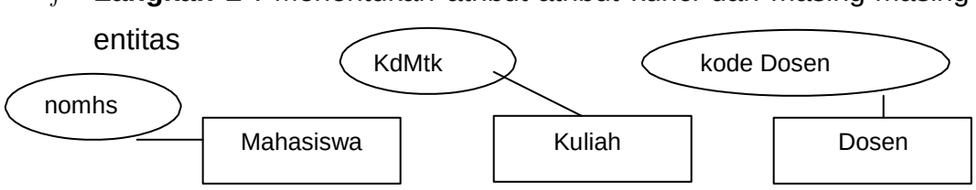
- Dilakukan koreksi terhadap hasil tahap awal, dengan memperhatikan aspek efisiensi, performansi, dan fleksibilitas
- Bentuk-bentuk koreksi yang dilakukan :
  - Dekomposisi himpunan entitas
  - Penggabungan himpunan entitas
  - Pengubahan derajat relasi
  - Penambahan relasi baru
  - Penambahan dan pengurangan atribut untuk masing-masing entitas dan relasi

- Langkah-langkah menyusuan diagram awal ER :
  a. Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
  b. Menentukan atribut-atribut kunci dari masing-masing himpunan entitas
  c. Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key-nya
  d. Menentukan derajat relasi (cardinality) untuk setiap himpunan relasi
  e. Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci)

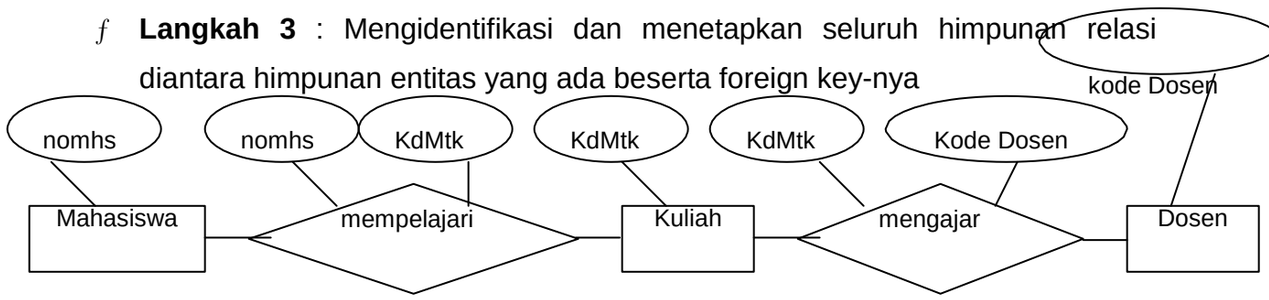
- **Contoh : kasus pada perkuliahan**
  - **Langkah 1** : Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat



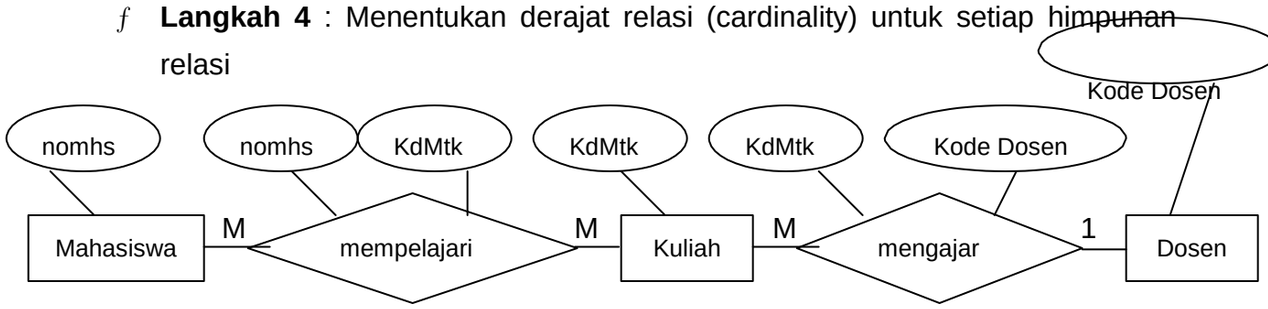
  - **Langkah 2** : Menentukan atribut-atribut kunci dari masing-masing himpunan entitas



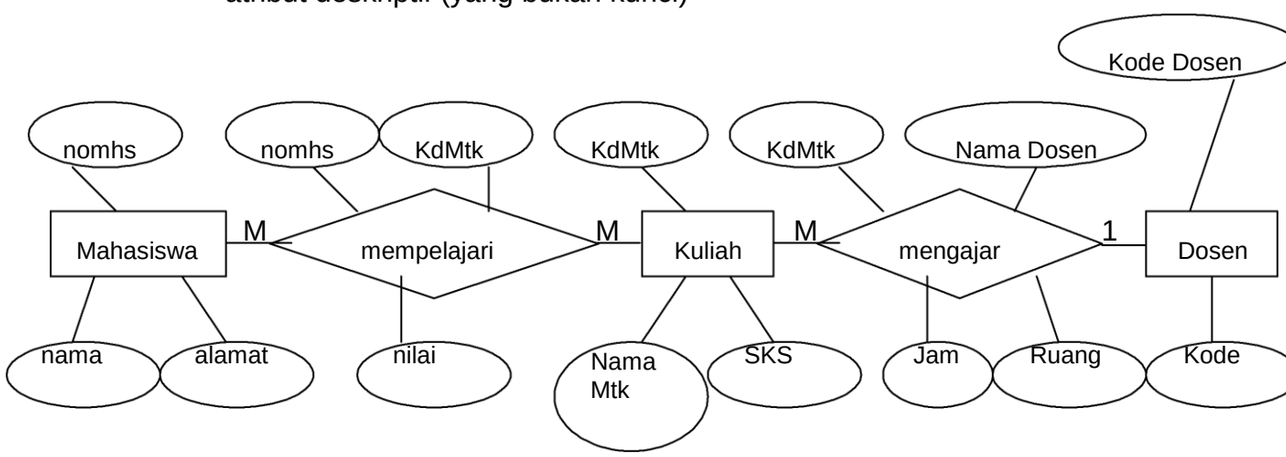
- **Langkah 3** : Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key-nya



- **Langkah 4** : Menentukan derajat relasi (cardinality) untuk setiap himpunan relasi

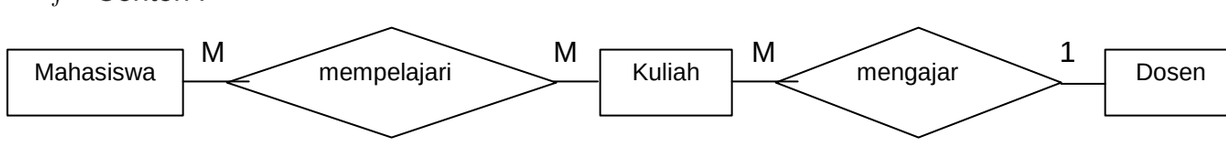


- **Langkah 5 :** Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci)



6. Diagram E-R dengan Kamus Data

- Pada sistem yang ruang lingkupnya lebar dan kompleks, penggambaran atribut-atribut dalam ERD seringkali malah mengganggu tujuan yang ingin dicapai. Oleh karena itu dapat dinyatakan dalam **Kamus Data**
- Kamus data berisi daftar atribut yang diapit tanda ‘{‘ dan ‘}’.
  - Atribut yang merupakan kunci digarisbawahi
- Contoh :



Kamus data :

Mahasiswa = {nomhs, nama, alamat}

Kuliah = {kdmtk, nama mtk, sks}

Dosen = {kode dosen, nama}

Mempelajari = {nomhs, kdmtk, nilai}

Mengajar = {kdmtk, kode dosen, jam, ruang}

BAB VII
IMPLEMENTASI BASIS DATA

1. Pengantar

a. Tahap implementasi basis data merupakan upaya untuk membangun basis data fisik yang ditempatkan dalam media penyimpan (disk) dengan bantuan DBMS

b. Tahap ini diawali dengan melakukan transformasi dari model data yang telah selesai dibuat struktur basis data sesuai DBMS yang dipilih

c. Secara umum, sebuah ERD akan diwujudkan menjadi sebuah **basis data** secara fisik. Sedangkan komponen-komponen ER yang berupa himpunan entitas dan himpunan relasi akan diwujudkan menjadi **tabel-tabel**. Selanjutnya, atribut-atribut yang melekat pada masing-masing himpunan entitas dan himpunan relasi akan dinyatakan sebagai **field-field** dari tabel yang sesuai

d. Performansi basis data ditentukan oleh :
   - Kualitas dan bentuk perancangan basis data
   - Kualitas mesin / komputer
   - Platform yang dipilih
   - Sistem operasi
   - DBMS yang digunakan

2. Pengkodean / Abstraksi data

a. Data yang dilihat oleh pemakai awam (end-user) bisa berbeda dengan bagaimana data / informasi itu disimpan. Apa yang dilihat oleh end-user bisa jadi merupakan hasil pengolahan yang tidak disimpan sama sekali dalam basis data, atau bisa dinyatkan dalam bentuk lain

b. Alasan untuk membuat suatu pengkodean adalah untuk efisiensi ruang penyimpanan

c. Dari pemakaiannya, ada dua bentuk pengkodean :
   - **Eksternal (user-defined coding)**
     - Mewakili pengkodean yang telah digunakan secara terbuka dan dikenal dengan baik oleh pemakai awam

       Contoh : Nomor mahasiswa dan Kode matakuliah Æ sudah dikenal baik oleh pemakai awam
   - **Internal (system coding)**

     Menggambarkan bagaimana data disimpan dalam kondisi sebenarnya, sehingga lebih berorientasi pada mesin

- Ada tiga bentuk pengkodean :
  - Sekuensial

    Pengkodean dilakukan dengan mengasosiasikan data dengan kode yang urut

    Contoh : predikat kelulusan “Sangat Memuaskan”, ”Cukup Memuaskan”, “Memuaskan” Æ dikodekan dengan huruf “A”, “B”, “C”
  - Mnemonic

    Pengkodean dilakukan dengan membentuk suatu singkatan dari data yang hendak dikodekan.

    Contoh : “Laki-laki” Æ dikodekan ‘L”; “Perempuan” Æ dikodekan “P”
  - Blok

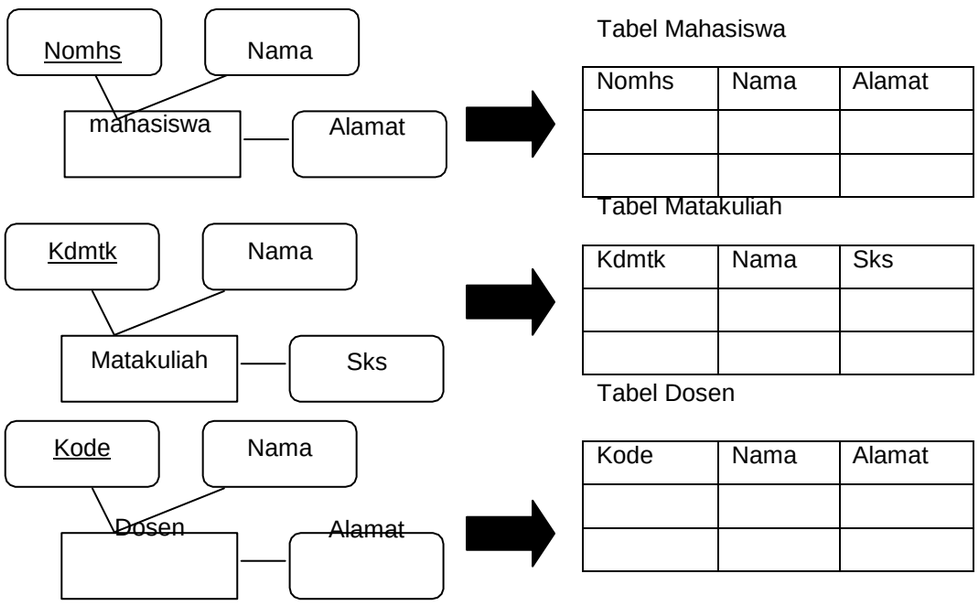
    Pengkodean dinyatakan dalam format tertentu

    Contoh : Nomor mahasiswa dengan format XX.YY.ZZZZ Æ terdiri atas XX = 2 digit tahun masuk, YY = 2 digit kode jurusan, ZZZZ = 4 digit nomor urut

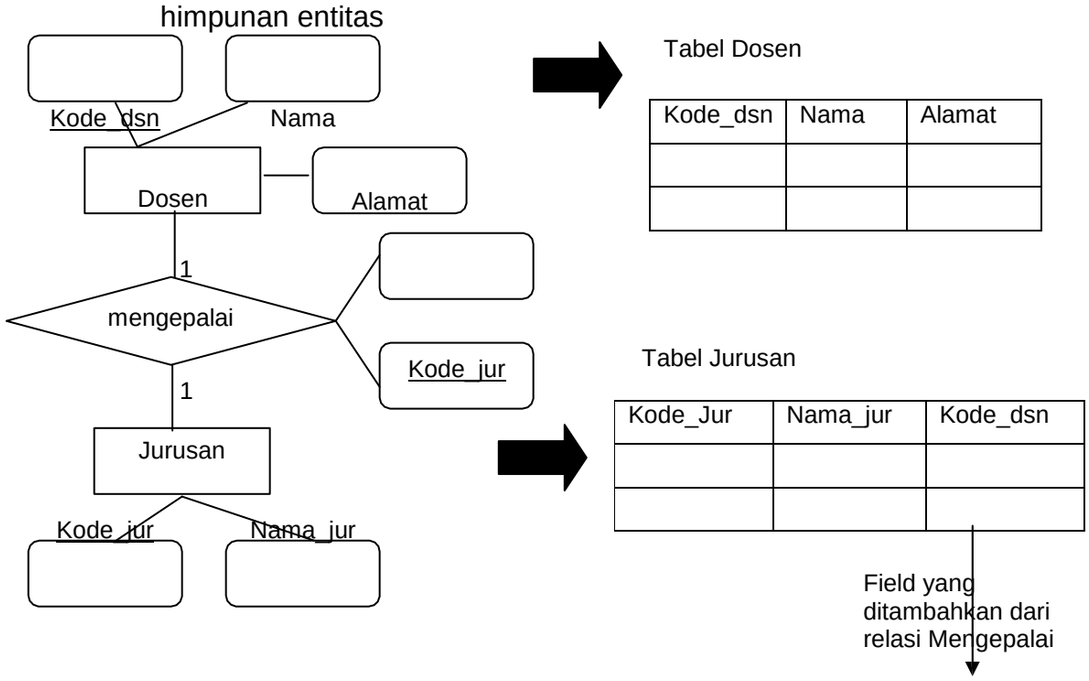
3. Transformasi Model data ke Basis data fisik

Aturan umum dalam pemetaan model data yang digambarkan dalam ERD (level konseptual) menjadi Basis data fisik (level fisik) adalah :

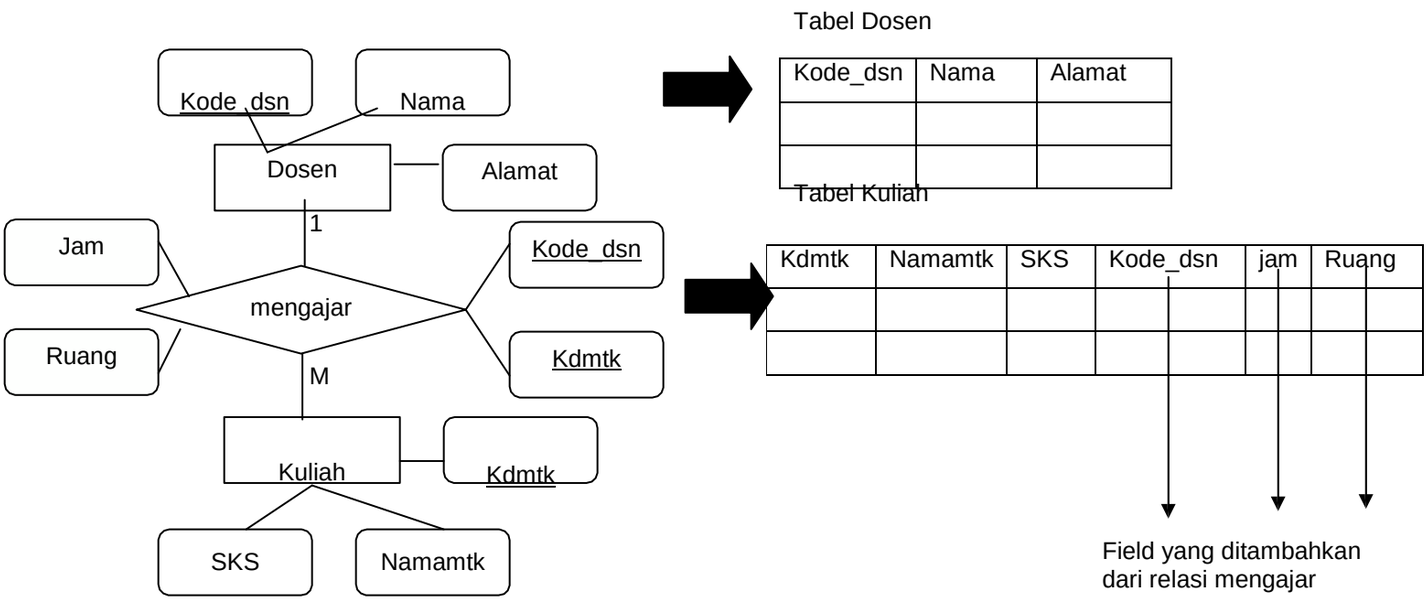
a. Setiap himpunan entitas akan diimplementasikan sebagai sebuah tabel (file data)



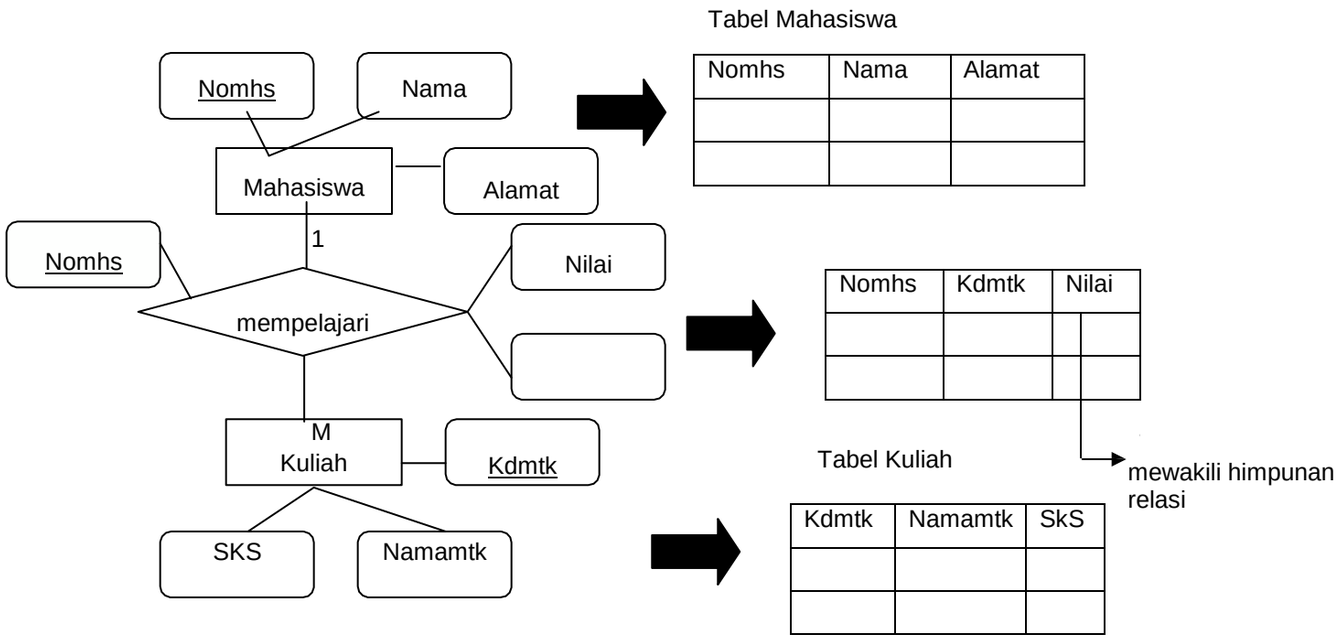
b. Relasi dengan derajat relasi satu-ke-satu, yang menghubungkan 2 buah himpunan entitas akan direpresentasikan dalam bentuk penambahan / penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas



c. Relasi dengan derajat relasi satu-ke-banyak, yang menghubungkan 2 buah himpunan entitas, juga akan direpresentasikan dalam bentuk pemberian / pencantuman atribut kunci dari himpunan entitas pertama (yang berderajat 1) ke tabel yang mewakili himpunan entitas kedua (yang berderajat M)



d. Relasi dengan derajat relasi banyak-ke-banyak, yang menghubungkan 2 buah himpunan entitas akan diwujudkan dalam bentuk tabel khusus, yang memiliki field (atau foreign key) yang berasal dari kunci-kunci dari himpunan entitas yang dihubungkannya



4. DBMS dan Struktur tabel

- Dalam menentukan struktur dari tabel, paling tidak setiap struktur tabel berisikan nama field, tipe field dan ukurannya
- Tatacara penamaan field, pilihan tipe field serta fasilitas tambahan lainnya untuk struktur tabel sangat tergantung pada DBMS yang digunakan
- Tipe data yang bersifat umum adalah :
  - **Data Alphanumerik**, isinya berupa angka tapi tidak menunjukkan jumlah, sehingga dianggap sebagai teks. Misalnya : Nomhs, NIP
  - **Data Numerik**, isinya berupa angka yang menunjukkan jumlah. Misalnya : SKS, Gaji pokok
  - **Data bilangan bulat (integer),** Byte (1 byte), Small-Integer (2 byte), Long Integer (4 byte)
  - **Data bilangan nyata,** Single (4 byte), Double (8 byte). Tipe data single dapat menampung hingga 7 digit pecahan, sedangkan double hingga 15 digit pecahan
    - Dalam komputasi, data integer akan membutuhkan waktu lebih cepat dalam pengolahan data dibandingkan real. Begitu juga,

karena ruang penyimpanan yang dibutuhkan lebih kecil, maka data single akan lebih cepat dalam pengolahan dibandingkan double

- **Data uang (currency)**, pemakaian tipe ini sangat membantu dalam mengatur tampilan data yang berkaitan dengan nilai uang, misalnya dengan adanya pemisahan ribuan/jutaan dan adanya tanda mata uang
- **Data teks,** ada dua jenis yaitu ukuran tetap (fixed character) dan ukuran dinamis (variable character). Misalnya field nomhs lebih tepat bertipe fixed character karena ukurannya pasti dan pendek. Sedangkan nama mahasiswa sebaiknya bertipe variable character karena panjang dan bervariasi

- Pertimbangan dalam menentukan tipe data bagi setiap field adalah :
  - Kecukupan domain
    - Harus dapat menjamin bahwa tipe data yang dipilih pada tiap field akan dapat menampung semua nilai yang akan diisikan ke dalam field tersebut
  - Efisiensi ruang penyimpanan
    - Apabila pemilihan tipe data tidak tepat (berlebihan), akibatnya akan memperbesar ukuran tabel secara keseluruhan
  - Kecepatan pengolahan data
    - Pada akhirnya, pemilihan tipe yang tidak tepat juga mengakibatkan pengaksesan data menjadi lebih lambat

5. Indeks dan Struktur penyimpanan

- Pada tahap implementasi, atribut-atribut entitas / relasi yang ditetapkan sebagai kunci (key) akan diwujudkan sebagai Indeks Primer (*primary index*). Dan dapat juga ditambahkan Secondary index
- Ada 2 indeks :

a. Indeks Primer (primary index)
   - IP pada setiap tabel hanya ada satu dan hampir selalu berasal (ditentukan) dari kunci primer yang telah ditetapkan dalam sebuah entitas / relasi
   - IP yang baik terdiri atas field-field dengan kriteria sbb :
     - Field yang menjadi komponen IP harus bersifat mandatory (datanya tidak boleh kosong atau berisi nilai null)
     - Keseluruhan nilai IP bersifat unik
     - Nilai-nilainya lebih permanen (idealnya tidak pernah berubah)
     - Berukuran kecil (pendek) dengan jumlah field minimal (sedikit)

b. Indeks Sekunder (secondary index)

Digunakan untuk mendukung keberadaan IP yang dibuat untuk suatu tabel dengan alasan untuk mempermudah berbagai cara pengaksesan ke suatu tabel

Misalnya : field Nama_Mahasiswa Æ untuk memudahkan pencarian data berdasar nama mahasiswa; disamping pencarian berdasar NOMHS

Catatan :

- Jumlah IS dalam sebuah tabel boleh lebih dari Satur
- Nilai-nilai field yang menjadi pembentuk IS tidak harus bersifat unik

6. Struktur penyimpanan

Ada 7 pilihan struktur penyimpanan dasar yang dapat diterapkan pada suatu tabel (bergantung pada DBMS yang dipakai) yaitu : Pile, Heap, hash, Sekuensial Berindeks, File berindeks, Multiring

a. Heap

- Merupakan struktur penyimpanan yang paling sederhana dan paling hemat dalam kebutuhan ruang penyimpanan
- Setiap baris data disusun berdasar kronologis penyimpanannya. Record yang pertama disimpan akan ditempatkan di posisi awal ruang penyimpanan, dan begitu seterusnya
- Pengubahan data tidak akan mengubah urutan record tersebut. Jika terjadi penghapusan, maka record-record dibawahnya akan dimampatkan untuk mengisi tempat yang kosong akibat penghapusan
- Pencarian data berjalan dengan lambat, karena dilakukan secara sekuensial baris demi baris
- Struktur ini cocok untuk tabel berukuran kecil dan jarang berubah

b. Hash

- Baris-baris data ditempatkan berdasar nilai alamat fisik yang diperoleh dari hasil perhitungan (fungsi hashing) terhadap nilai key-nya. Karena itu penempatan record dalam tabel tidak tersusun berdasarkan kedatangannya. Bisa jadi record yang terakhir dimasukkan justru menempati urutan pertama
- Memiliki performansi yang paling baik dalam hal pencarian data tunggal berdasar kunci indeks
- Struktur ini cocok untuk tabel-tabel yang sering menjadi acuan bagi tabel lain

- Kelemahannya membutuhkan ruang penyimpanan awal yang besar, untuk menjamin agar record-record yang disimpan tidak menempati alamat yang sama Æ dibutuhkan alokasi ruang penyimpanan

c. Sekuensial berindeks

- Menempatkan data engan urutan tertentu berdasarnilai indeks primernya
- Record yang memiliki nilai IP paling kecil dibandingkan record yang lain akan ditempatkan di awal ruang penyimpanan tabel meskipun dimasukkan belakangan
- Performansi turun pada saat terjadi penambahan atau perubahan data yang menyangkut nilai indeks primernya, karena perlu dilakukan penataan ulang
- Struktur ini cocok untuk tabel yang sifatnya statis, dan untuk pencarian data kelompok dalam suatu tabel (lebih baik daripada hash)

d. File berindeks

- Dikembangkan dari struktur heap. Record-record disusun berdasar kronologis penyimpanannya (seperti heap). Namun disediakan pula file indeks yang disusun berdasar nilai key setiap record yang berguna untuk membantu proses pencarian data ke suatu tabel
- Terdapat 2 komponen yaitu komponen data dan komponen indeks. Komponen data disusun dengan struktur heap, dan komponen indeks disusun dengan struktur sekuensial berindeks
- Struktur ini cocok untuk tabel yang dinamis dan berukuran besar

BAB VIII
PENGEMBANGAN SISTEM BASIS DATA

1. Pengantar

Pengembangan basisdata selalu membutuhkan kerjasama dari beberapa orang dengan keahlian yang berbeda-beda. Proses ini melibatkan pemakai, analis data, ahli komputer, database administrator, serta wakil dari pihak manajemen yang akan memakai sistem.

2. Tujuan pengembangan sistem basis data

Tujuan pengembangan sistem basis data adalah :

a. **Akses data yang fleksibel (data flexibility)**
   - Untuk memberikan kemudahan dalam menampilkan kembali data-data yang diperlukan dan menampilkannya dalam format yang berbeda

b. **Pemeliharaan Integritas data (data integrity)**
   - Untuk selalu meyakinkan bahwa nilai-nilai data dalam SBD adalah benar, konsisten, dan selalu tersedia

c. **Proteksi data dari kerusakan dan akses ilegal (data security)**
   - Keamanan data diperlukan untuk melindungi data dari kerusakan yang terjadi karena alam (kebakaran, banjir, dll) atau akses yang ilegal
   - Recovery merupakan proses untuk menyusun kembali basis data yang mengalami kerusakan

d. **Menghilangkan ketergantungan data pada program aplikasi (data independence)**
   - Ada 2 bentuk ketergantungan, yaitu logik dan fisik
   - Ketergantungan logik, bahwa perubahan kebutuhan user terhadap data dapat berubah, tapi hal tsb tidak mengakibatkan perubahan pada pandangan user terhadap basis data
   - Ketergantungan fisik (schema), bahwa diskripsi logik data tidak mengalami ketergantungan pada perubahan-perubahan yang terjadi dalam teknik penyimpanan secara fisik

e. **Minimalisasi kerangkapan data (reduced data redundancy)**
   - Kerangkapan data menyebabkan media penyimpan tidak efisien, waktu akses yang lama, dan menimbulkan masalah integritas data

f. **Penggunaan data secara bersama-sama (data shareability)**
   - SBD yang dikembangkan harus dapat digunakan oleh pemakai yang berbeda-beda

g. **Keterhubungan data (data relatability)**
   - Adalah kemampuan untuk menetapkan hubungan logik antara tipe-tipe record yang berbeda

h. **Standarisasi definisi rinci data (data item)**
   - Menunjukkan definisi rinci data dalam batas presisi yang digunakan pada definisi nama rinci data dan format pentimpanan dalambasia data

i. **Meningkatkan produktivitas personal (personal productivity)**
   - SBD diharapkan mampu meningkatkan produktifitas kerja setiap personal, yang mampu memenuhi kebutuhan data sederhana hingga bentuk laporan yang lebih rumit

3. Proses Pengembangan basisdata

Secara garis besar, proses pengembangan basis data adalah :

a. **Penentuan tujuan**

   Tujuan ditetapkan berdasar parameter pemakai dan data. Pemakai menentukan tujuan dari aplikasi yang akan dipakai. Sedangkan data menentukan bagaimana tujuan tersebut dapat dicapai.

   Tujuan dinyatakan tanpa adanya kekangan, misalnya respon yang seketika, dapat dipercaya, dan perlindungan terhadap kebebasan pribadi.

b. **Ikatan (bindings)**

   Bindings merupakan ukuran tingkat fleksibilitas yang dilakukan untuk mencapai efisiensi dalam perancangan basisdata.

   Ukuran-ukuran tersebut misalnya : struktur file, model basisdata, skema / relasi, pemanggilan informasi, serta perawatan data dan integritas basisdata.

   Faktor fleksibilitas seringkali bertentangan dengan unjuk kerja. Jika mementingkan fleksibilitas maka struktur record menjadi sangat bermacam-macam. Jika mementingkan unjuk kerja maka akan terjadi pemaksaan pada hal-hal tertentu.

c. **Dokumentasi**

   Dokumentasi yang penting adalah model basisdata. Model basisdata akan menentukan proses yang diperlukan untuk pembentukan file, perawatan file, dan pemanggilan informasi.

   Bentuk yang harus didokumentasikan adalah skema basis data, relasional basisdata, dan definisi variabel yang dipakai

d. **Pemrograman**

   Implementasi akhir setelah proses perancangan basisdata selesai adalah dengan melakukan pemrograman

4. Langkah-langkah pengembangan sistem basis data

Komponen yang terlibat dalam pengembangan SBD : File Basis data, Software, Hardware, Personil yang terlibat

Langkah-langkah dalam pengembangan SBD :

a. Spesifikasi kebutuhan
    - Definisi masalah dan studi kelayakan
    - Rinci spesifikasi

b. Evaluasi alternatif
    - Indikasi alternatif
    - Seleksi alternatif

c. Desain
    - Spesifikasi dan order perangkat keras
    - Desain logik program
    - Desain struktur data
    - Desain prosedur untuk pemakai dan operator
    - Definisi struktur organisasi pemakai

d. Implementasi
    - Instalasi dan pengujian perangkat keras
    - Coding dan pengujian unit-unit program
    - Konversi data
    - Pembuatan dokumen prosedur
    - Pelatihan pemakai
    - Pengujian menyeluruh

5. Langkah-langkah mendisain basis data untuk SIM

1. Menetapkan disain / model SIM yang digambarkan dalam diagram arus data (DAD)
2. Menentukan kebutuhan file basis data
3. Menentukan parameter dari file basis data, meliputi :
    a. Tipe file : file induk, file transaksi, dll.
    b. Media file : harddisk, disket, dll
    c. Organisasi file :
        i. file tradisional (file urut, urut berindeks, atau file akses langsung)
        ii. organisasi database (struktur berjenjang, jaringan atau hubungan)
    d. Field kunci dari file

4. Alat bantu dan metode dalam pengembangan sistem basis data

Alat bantu merupakan teknik yang digunakan untuk mempermudah atau mendukung kelancaran pelaksanaan kegiatan proyek

Beberapa metode :

- Studi kelayakan (feasibility study)
- Analisis biaya manfaat (cost benefit analysis)

Beberapa tools :

- PERT (Program Evaluation and Review Technique)

  Digunakan untuk penjadwalan dan pengawasan pekerjaan yang kompleks dan mempunyai sifat peka waktu, dan belum diketahui waktunya secara pasti
- CPM (Critical Path Method)

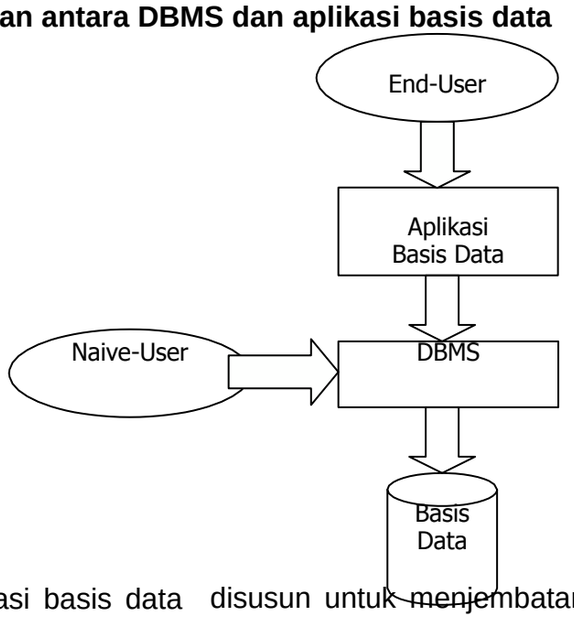
  Digunakan untuk mengawasi dan mengendalikan tugas-tugas dalam proyek yang telah ditentukan waktunya, dengan cara menambah atau mengurangi sumber-sumber yang diperlukan dan tersedia untuk menyelesaikan proyek
- EasyCase

  Digunakan sebagai alat bantu pada tahap perancangan basis data
- S-Designor

  Digunakan sebagai alat bantu pada tahap perancangan basis data

BAB IX APLIKASI BASIS DATA

1. Hubungan antara DBMS dan aplikasi basis data

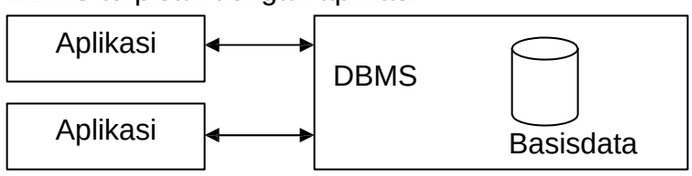


Aplikasi basis data disusun untuk menjembatani perbedaan pandangan antara *end-user* dan *naïve user*, yang dibuat khusus untuk dapat digunakan oleh para pemakai akhir (*end-user*)

Aplikasi ini berisi sejumlah operasi (menu) yang sesuai dengan aktifitas nyata yang dilakukan oleh end-user. Selanjutnya operasi ini akan diterjemahkan oleh aplikasi tersebut menjadi sejumlah operasi basis data yang dapat dikenali oleh DBMS

Terdapat 2 model hubungan DBMS dan aplikasi basis data :

a. DBMS terpisah dengan aplikasi

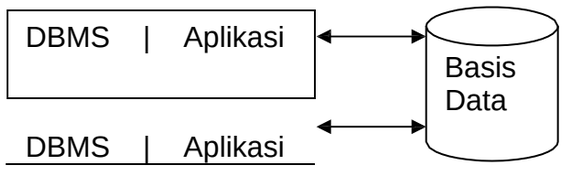


Aplikasi tidak berinteraksi langsung dengan basis data, tapi melalui DBMS sebagai perantara. Bahkan DBMS bisa melakukan aktifitas sendiri yang bisa ditangkap oleh aplikasi

Contoh DBMS : MS SQL Server, Oracle, CA-OpenIngres, Sybase, Informix, IBM DB2

Cocok untuk aplikasi yang *single-user* atau *standalone*, dengan beban kerja yang ringan

b. DBMS menyatu dengan aplikasi



- Aplikasi basis data yang dibuat menyatu dengan DBMS pada saat pemakaiannya
- Dalam model ini, aplikasi basis data berada ‘dibawah’ DBMS (menjadi sub-ordinate), sehingga DBMS harus diaktifkan lebih dulu sebelum menjalankan aplikasi
- Contoh DBMS : dBase III+, FoxBase, FoxPlus, CA-Clipper, MS-Access
- Cocok untuk aplikasi yang multi-user, dengan beban kerja yang berat

2. Pemilihan arsitektur sistem

- Yang menjadi pertimbangan dalam memilih arsitektur sistem :
  - Keunggulan teknologi
  - Faktor biaya
  - Sesuai dengan kebutuhan pemakai

- Jenis-jenis arsitektur sistem :

Sistem tunggal (Standalone)

DBMS, basis data, dan aplikasi basis data ditempatkan pada komputer yang sama.

Hanya bisa dipakai oleh satu pemakai pada saat yang bersamaan

Sistem Terpusat (Centralized system)

Terdiri dari sebuah server dan sejumlah terminal

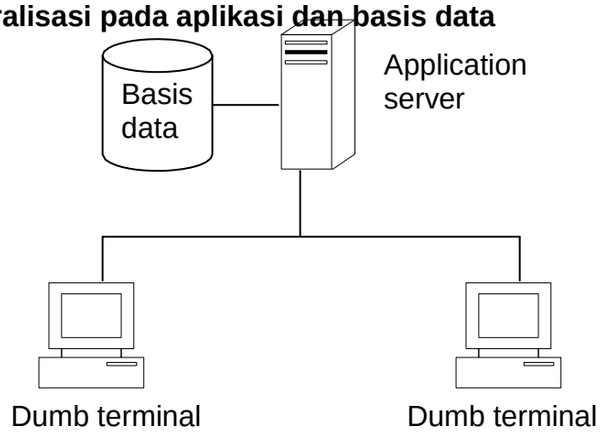
Yang terpusat adalah basis data, DBMS, dan aplikasi basis data atau basis data saja

Ada dua macam :

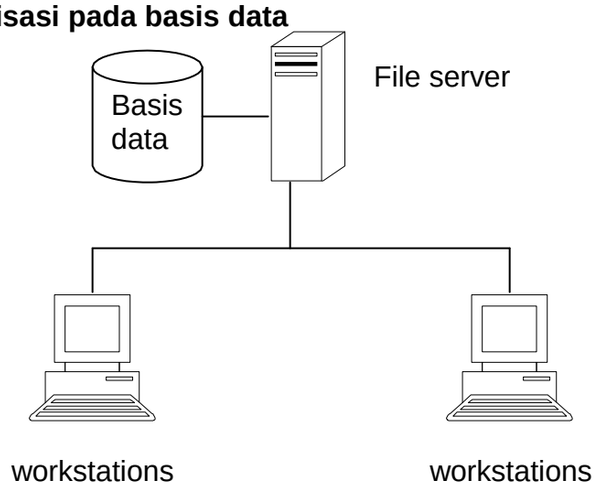
Aplikasi dan basis data terpusat; diakses oleh dumb terminal

Basis data terpusat; aplikasi ada pada terminal

Sentralisasi pada aplikasi dan basis data



Sentralisasi pada basis data



Sistem Client-server

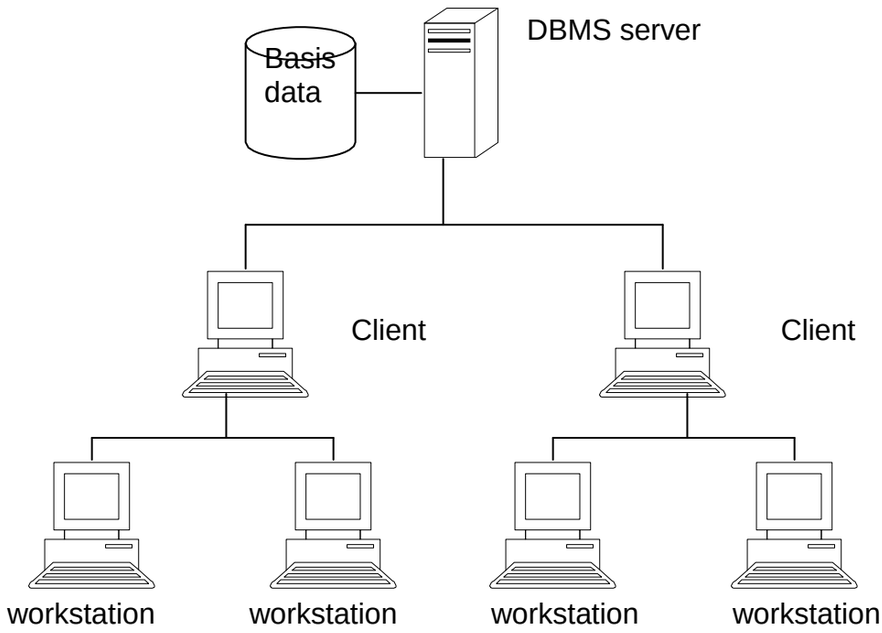
Ditujukan untuk mengatasi kelemahan yang terdapat pada sistem terpusat

Terdiri dari 2 komponen utama yaitu client dan server. Client berisi aplikasi basis data; server berisi DBMS dan basis data

Ada dua macam :

- Arsitektur 2 lapis (2-tier)
- Arsitektur 3 lapis (3-tier)

Arsitektur 3-tier



3. Pemilihan perangkat lunak pembangun aplikasi basis data

Pertimbangan dalam menentukan perangkat lunak pembangun aplikasi basisdata

a. Kecocokan antara DBMS dan *development tools*

Perangkat lunak yang dipilih harus dapat menjamin tersedianya fasilitas yang dapat digunakan untuk berinteraksi dengan DBMS secara penuh

Contoh :

DBMS	**Development tools**
MS-SQL server	MS Visual Basic
Borland Interbase	Borland Delphi
CA-OpenIngres	CA-OpenRoad
Oracle	Developer 2000

b. Dukungan *development tools* terhadap arsitektur sistem

Tidak semua development tools memberi dukungan yang baik terhadap arsitektur client-server

c. Independensi *development tools* dan DBMS

Idealnya hanya ada satu macam DBMS yang dipilih untuk mengelola berbagai basis data.

Sebagai kompromi terhadap banyaknya DBMS yang digunakan, maka harus dipilih development tools yang bisa cocok untuk semua DBMS

d. Kemudahan pengembangan dan migrasi aplikasi

Development tools yang dipilih harus mendukung pengembangan ke masa depan (misalnya berbasis web) dan kemudahan migrasi, misalnya dari berbasis form (form-base) menjadi berbasis web (web-base)

4. Pertimbangan performansi dalam aplikasi

Performansi / kecepatan operasi ke basis data ditentukan oleh :

- DBMS yang digunakan
- Arsitektur perangkat keras yang menjadi platform
- Jumlah pemakai yang terlibat
- Volume data
- Tingkat kompleksitas operasi basis data
- Cara penulisan aplikasi

Hal-hal yang perlu dipertimbangkan pada saat melaksanakan pemrograman :

a. Sedapat mungkin memanfaatkan indeks primer / sekunder dalam setiap proses query ke basis data, contoh :

Select …. From pegawai where idpegawai = vidpegawai
Update pegawai set …. Where idpegawai = vidpegawai
Delete from pegawai where idpegawai = vidpegawai

- Menghindari pemakaian fungsi atau perhitungan pada perintah query, terlebih lagi untuk kriteria query. Contoh :

Select ‘01’+left(nomhs,2) as vthn from mahasiswa where …

Diganti menjadi :

Select nomhs as vnim from mahasiswa where
thn_masuk = ‘01’+left(vnim,2)

Select … from kuliah where left(kdmtk,3)=’TFD’

Diganti menjadi

Select … from kuliah where kdmtk like ‘TFD%’

Kenapa beda ? Pada arsitektur cilent-server :

- pada perintah yang asal (belum diperbaiki) maka pengerjaan pencarian data dan penerapan fungsi dilakukan diserver

- pada perintah perbaikan, karena dibuat fungsi dan perhitungan bukan merupakan bagian dari perintah query, maka server hanya akan melakukan pencarian data, sedangkan pengerjaan mengenai fungsi dilakukan di client
- Pada contoh kedua, pada perintah asal tidak memanfaatkan kunci primer. Sedangkan pada perintah perbaikan menggunakan kunci indeks primer

b. Operasi join pada beberapa tabel dapat digunakan untuk mengefisienkan perintah dan sekaligus banyaknya data yang harus ditangani. Contoh :

Ada dua perintah :

- Select kdmtk as vkdmtk, nilai as vnilai from nilai where nomhs=vnomhs
- Select sks as vsks from kuliah where kdmtk=vkdmtk

Dapat digabungkan menjadi :

- Select a.nilai as vnilai, b.sks as vsks from nilai a, kuliah b where a.nomhs=vnomhs and a.kdmtk=b.kdmtk

c. Pada sistem multi-user dengan tingkat konkurensi tinggi (pemakai yang aktif banyak), sesegara mungkin melepaskan penguncian tabel di akhir setiap query. Karena proses dilakukan dalam dua tahap, yaitu menyimpan secara sementara di buffer memory lalu menuliskan ke dalam disk. Untuk membatalkan ada perintah rollback. Contoh :

```
Insert into nilai (nomhs, kdmtk) values (vnomhs, vkdmtk)
  Commit Æ untuk merekam ke disk
```

d. Manfaatkan sebanyak mungkin fungsi-fungsi yang telah disediakan DMS ataupun development tools yang terkait dengan operasi basis data.

```
Select count (*) as vjumlah from mahasiswa where nomhs=vnomhs
  If vjumlah=0 then
    Echo “Tidak ketemu..”
  Else
    Select nama as vnama from mahasiswa where nomhs=vnomhs
  Endif
```

Akan lebih baik, jika diganti menjadi :

```
Select nama as vnama from mahasiswa where nomhs=vnomhs
Inquire_sql (jumlah=rowcount)
If vjumlah=0 then
  Echo “Tidak ketemu…”
```

```
Else
  Echo “Nama : “;&vnama
Endif
```

f. Jika ada perintah perulangan (looping) dengan penelusuran seluruh basis data pada sebuah tabel, sebisa mungkin menempatkan berbagai perintah yang tidak relevan di luar perulangan. Contoh :

```
I=1
Buka tabel X
While (row belum habis) do
        Tampilkan pesan “Sedang diproses…”
        Total=total+y
        Rata=total/i
        I=i+1
        Ke row berikutnya
Endwhile
Tampilkan total dan rata
```

Algoritma tersebut dapat diperbaiki menjadi :

```
I=0
Tampilkan pesan “Sedang diproses…”
Buka tabel X
While (row belum habis) do
        Total=total+y
        I=i+1
        Ke row berikutnya
Endwhile
Rata=total/i
Tampilkan total dan rata
```

5. Pemeliharaan integritas basis data dalam aplikasi

Sebagai sarana untuk meyakinkan bahwa nilai-nilai data dalam sistem basis data selalu benar, konsisten, selalu tersedia. Dapat dilakukan dengan cara :

- Pastikan bahwa nilai-nilai data adalah benar sejak dimasukkan pertama kali
- Membuat program untuk mengecek keabsahan data pada saat dimasukkan ke komputer
  - Penolakan / pembatalan aksi (cancelation)
  - Pengisian nilai kosong pada field tertentu (nullify)
  - Penjalaran perubahan (cascade)

Integritas yang harus dijaga :

a. **Integritas keunikan data**, dilakukan melalui :
   i. pendefinisian struktur tabel dengan membuat indeks primer yang bersifat unik

ii. pengkodean di dalam aplikasi pada saat pemasukan / penambahan data Æ lebih *user-friendly*

iii. kedua cara (i) dan (ii) diterapkan bersama-sama

b. **Integritas domain data**, dilakukan melalui :

i. Penetapan tipe data pada setiap field di dalam tabel

ii. Pengisian *validation rule* dari DBMS

c. **Integritas referensial** (relasi antar tabel)

i. Harus selalu dijaga, karena kesalahan referensial dapat menimbulkan kesalahan baru dalam basis data

ii. Dilakukan pengecekan pada proses penambahan, pengubahan, dan penghapusan data

d. Integritas aturan nyata

i. Sifatnya sangat kasuistis, tidak berlaku umum. Pada kasus yang berbeda, aturannya bisa berbeda pula

ii. Untuk mengakomodasi adanya *business role* ini, dengan menyiapkan tabel khusus yang menampung nilai-nilai konstanta yang dibutuhkan aplikasi pada saat dijalankan yang mudah diubah tanpa mengakibatkan perubahan aplikasi maupun struktur basis data

BAB X
ADMINISTRASI DAN MANAJEMEN BASIS DATA

A. Pengembangan basisdata

Pengembangan basisdata selalu membutuhkan kerjasama dari beberapa orang dengan keahlian yang berbeda-beda. Proses ini melibatkan pemakai, analis data, ahli komputer, database administrator, serta wakil dari pihak manajemen yang akan memakai sistem.

Secara garis besar, proses pengembangan basis data adalah :

1. Penentuan tujuan

   Tujuan ditetapkan berdasar parameter pemakai dan data. Pemakai menentukan tujuan dari aplikasi yang akan dipakai. Sedangkan data menentukan bagaimana tujuan tersebut dapat dicapai.

   Tujuan dinyatakan tanpa adanya kekangan, misalnya respon yang seketika, dapat dipercaya, dan perlindungan terhadap kebebasan pribadi.

2. Ikatan (bindings)

   Bindings merupakan ukuran tingkat fleksibilitas yang dilakukan untuk mencapai efisiensi dalam perancangan basisdata.

   Ukuran-ukuran tersebut misalnya : struktur file, model basisdata, skema / relasi, pemanggilan informasi, serta perawatan data dan integritas basisdata.

   Faktor fleksibilitas seringkali bertentangan dengan unjuk kerja. Jika mementingkan fleksibilitas maka struktur record menjadi sangat bermacam-macam. Jika mementingkan unjuk kerja maka akan terjadi pemaksaan pada hal-hal tertentu.

3. Dokumentasi

   Dokumentasi yang penting adalah model basisdata. Model basisdata akan menentukan proses yang diperlukan untuk pembentukan file, perawatan file, dan pemanggilan informasi.

   Bentuk yang harus didokumentasikan adalah skema basis data, relasional basisdata, dan definisi variabel yang dipakai

4. Pemrograman

   Implementasi akhir setelah proses perancangan basisdata selesai adalah dengan melakukan pemrograman

B. Manajemen aktifitas data

Manfaat basisdata bagi pemakai bukan pada sistem basisdatanya melainkan pada isinya, serta hasil-hasil dari query yang dihasilkan oleh program.

Apabila selama dalam proses penerapan sistem ada perubahan basisdata, maka perubahan dan ujicoba dilakukan pada basisdata cadangan, agar tidak mengganggu sistem yang berjalan

Manajemen aktifitas data merupakan tugas dari DBA. Disamping itu, DBA juga bertugas untuk :

a. Menentukan standard, panduan, pengawasan prosedur, dan membuat dokumentasi untuk memastikan tidak terjadi tumpang tindih dalam mengatur data.
b. Mengatur kepemilikan data, hak akses, dan hak merubah data – terutama apabila beberapa pemakai mengakses data yang sama. Who can do what to which data.
c. Mengembangkan teknik dan prosedur *recovery* – DBA harus dapat mengantisipasi terjadinya kegagalan yang diakibatkan oleh machine failure, media failure, communications failure, dan data user failure. DBA dapat memanfaatkan fasilitas yang dimiliki oleh DBMS secara maksimal
d. Menyampaikan informasi tentang prosedur operasi dan melakukan pelatihan pada user
e. Menerapkan kebijakan yang berkaitan dengan aktifitas data. Apabila ada user yang melanggar, maka DBA berhak memberikan hukuman
f. Bertanggung jawab untuk menyusun dan merawat seluruh dokumentasi sistem, misalnya :
    - Aktifitas data
    - Database standards
    - Data ownership
    - Retrieval and access rights
    - Recovery procedures
    - Policy enforcement

Standard yang dimaksud misalnya :

- Tiap field harus mempunyai nama dan format baku
- Tiap record harus mempunyai standard nama, format, dan metode akses
- Tiap file basisdata harus mempunyai standard nama dan relasi dengan file lain

Hal-hal yang perlu dipantau adalah :

a. Statistik penggunaan perangkat keras

   Merupakan persentase waktu aktifitas yang diperlukan untuk mengakses prosesor, channel, controller, dan disk. Digunakan untuk menentukan tingkat kesibukan kerja sistem. Biasanya dilakukan oleh sistem operasi

b. Statistik penggunaan file

   Merupakan rasio penggunaan akses ke file seperti *fetch*, *get next*, dan *update* Dapat disimpan dalam file *log*.

c. Statistik penggunaan record

   Frekuensi pengaksesan record untuk dibaca atau di-update dapat menjadi bahan pertimbangan dalam optimalisasi basisdata dan pembuatan cadangan (backup)

   Selain itu, tanggal dan waktu pengaksesan dapat digunakan untuk menjaga integritas basisdata

d. Statistik penggunaan atribut

   Merupakan frekuensi penggunaan atribut, baik pada proses update, atau sebagai kunci pada pencarian data.

   Dapat dilihat dari skema / relasi antar tabel

C. Manajemen struktur basisdata

Tanggung jawab DBA dalam menangani struktur basisdata adalah :

a. Merancang skema

   DBA biasanya tidak terlibat dalam perancangan basisdata mulai dari awal. Oleh karena itu, setiap terjadi perubahan struktur basisdata yang berpengaruh pada skema / relasi antar tabel harus selalu dicatat

b. Mengawasi terjadinya redundancy

   Redundancy dapat terjadi pada dua hal, yaitu performance dan data integrity. DBA harus menetapkan prosedur tertentu untuk melakukan rekonsiliasi data untuk menghindari terjadinya redundancy

c. Melakukan pengawasan konfigurasi permintaan atas perubahan struktur basisdata

   DBA bertugas menyusun laporan secara berkala mengenai pemakai yang aktif, serta file dan data yang dipakai, dan metode akses yang digunakan. Disamping itu juga dicatat terjadinya kesalahan. Hal tersebut diperlukan untuk menentukan apakah diperlukan adanya perubahan struktur basisdata demi peningkatan performance

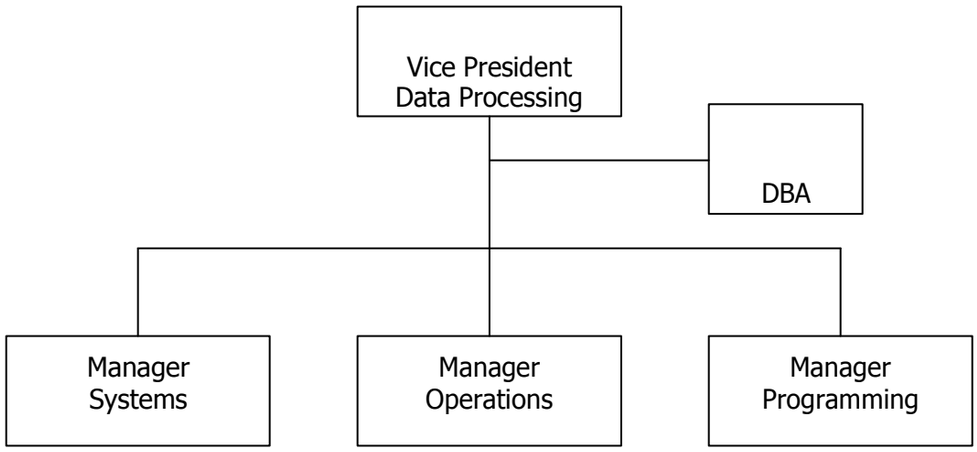
d. Menjadwalkan dan mengadakan pertemuan apabila terjadi perubahan struktur basisdata
e. Menerapkan perubahan skema
   Perubahan harus dilakukan pada basisdata ujicoba, agar pemakai dapat mengujinya sebelum diterapkan pada sistem yang sesungguhnya
f. Merawat dokumentasi pemakai
g. Merawat dokumentasi DBA – untuk memperoleh informasi tentang perubahan yang telah dilakukan, bagaimana dan kapan dilakukan

D. Manajemen DBMS

Tugas DBA berkaitan dengan manajemen DBMS adalah :

a. Menyusun laporan tentang unjuk kerja sistem basis data
   Unjuk kerja sistem dapat diuji dengan dua metode, yaitu dengan menjalankan contoh program dan dengan mencatat waktu proses pada kegiatan nyata.
   Pengujian dapat dilakukan melalui rutin program atau melalui fasilitas dalam DBMS
b. Melakukan investigasi atas keluhan pemakai
c. Melakukan analisa atas laporan dan keluhan
d. Melakukan “tuning” atau “optimizing” sistem basisdata
   Beberapa hal yang bisa di tune-up misalnya :
   - buffers size
   - size of a transaction
   - numbers of shared file
e. Jika memungkinkan, melakukan “tuning” pada perangkat lunak komunikasi dan sistem operasi dengan basisdata – misalnya dengan mengatur agar program tersimpan resident di memori, dengan mengatur alokasi sumberdaya perangkat keras dan saluran komunikasi
f. Mengevaluasi dan menerapkan fasilitas yang baru

E. Personil DBA



Personil	**Tugas**
Database Administrator	1. Mengatur staf untuk memastikan pengembangan basisdata berjalan lancar 2. Merencanakan kebutuhan basisdata di masa mendatang
Documentation and standards manager	1. Menciptakan dan merawat dokumentasi basisdata dan standard 2. Menyebarluaskan informasi tentang standard 3. Mengadakan pelatihan
User representatives	Mewakili user dalam menentukan kebutuhan basisdata dan menyampaikannya pada DBA
Operations representatives	1. Mewakili bagian operasional yang berkaitan dengan komputer. 2. Menetapkan kebutuhan basisdata masa depan yang diperlukan dalam kegiatan operasional 3. Memantau unjuk kerja basisdata 4. Melakukan “tuning” pada sistem operasi
DBMS configuration manager	1. Memahami sistem basisdata dan merawat konfigurasi pengawasan 2. Melakukan pemantauan dan “tuning” pada sistem basisdata 3. Menguji fasilitas baru pada DBMS
Performance monitor	Menyusun dan menganalisa unjuk kerja sistem Melakukan investigasi atas keluhan pemakai

\

DAFTAR PUSTAKA

1. An Introduction To Database Systems
    - C.J Date
    - Addison Wesley Publishing Co., Inc, 1995
2. Database Systems Concepts
    - Korth and Silberschatz
    - Mc. Graw-Hill International Co., 1986
3. Database Design
    - G. Wiedelhold
    - Mc. Graw-Hill International Co., 1988
4. Database Processing : Fundamental, Design, Implementations
    - D.M Kroenke
    - Sciences Research Associates, Inc, 1983
5. Sistem Basis Data
    - Edhy Sutanta
    - Penerbit ANDY, Yogyakarta, 1996
6. Konsep dan Perancangan Database
    - Harianto Kristanto
    - Penerbit ANDY, Yogyakarta, 1993
7. Basis Data
    - Fathansyah
    - Penerbit Informatika, Bandung, 1999