

SISTEM DATABASE



DAFTAR ISI

BAB 1.....	6
PENGANTAR DATABASE.....	6
2. Sistem Informasi	6
3. Komponen sistem informasi	6
4. Pengertian Database.....	6
LATIHAN SOAL.....	12
BAB 2.....	13
SISTEM BASIS DATA.....	13
▪ Sistem basis Data;	13
▪ Komponen sistem basis Data;.....	13
▪ Abstraksi Database;	13
▪ Penyusunan Sistem basis data	14
▪ Tipe File	14
▪ DATA INDEPENDENCE	15
▪ Bahasa Basis Data	15
▪ Komponen DBMS	16
▪ Komponen Data Manager.....	17
▪ Penggunaan Basis Data	17
▪ Database User	17
▪ Multi User DBMS Arsitektur.....	18
LATIHAN SOAL.....	20
BAB 3.....	21
LINGKUNGAN BASIS DATA	21
▪ Batasan dalam database	21
▪ Organisasi file basis data.....	22
▪ Metode susunan file :	23
▪ Schema dan Subschema	23
▪ Arsitektur Sistem Basis data	23
▪ Konsep DBMS.....	24
▪ Kamus Data	25
▪ Model Data	25
LATIHAN SOAL.....	29
BAB 4.....	30
RELATIONAL DATABASE MODEL	30
▪ Relasi Tabel	30
▪ Komponen relasi	30
▪ Kunci Relasi	31
▪ Relasi antar entity	31
▪ Basis data yang baik.....	32
LATIHAN SOAL.....	33
BAB 5.....	34
ENTITY RELATIONSHIP DIAGRAM	34

▪ Pengantar ERD	34
▪ Notasi dan artinya	34
▪ Notasi simbolik dalam bentuk E-R	35
▪ Jenis hubungan / Derajat relasi / Kardinalitas relasi.....	35
▪ Tahap pembuatan diagram E-R.....	36
▪ Langkah – langkah menyusun diagram awal ER:	37
▪ Diagram E-R dengan Kamus Data	38
▪ Derajat Relasi	38
LATIHAN SOAL.....	40
BAB 6.....	41
NORMALISASI.....	41
▪ Normalisasi.....	41
▪ Tujuan normalisasi :	41
▪ Aturan Normalisasi.....	41
▪ Penyimpangan dalam modifikasi	41
▪ Efek – efek normalisasi	43
▪ Jenis – Jenis Key Field / attribute tabel	43
▪ Domain dan tipe data	44
▪ Tahapan Normalisasi.....	44
LATIHAN SOAL.....	50
BAB 7	51
FRAGMENTASI DATA.....	51
▪ Fragmentasi	51
▪ Fungsi Fragmentasi :	51
▪ Database Terdistribusi	54
BENTUK-BENTUK TOPOLOGI DISTRIBUSI DATA :	54
LATIHAN SOAL.....	57
BAB 8.....	58
IMPLEMENTASI BASIS DATA	58
▪ Implementasi DB.....	58
▪ Pengkodean / Abstrasi data	58
▪ Transformasi Model data ke Basis data fisik.....	59
▪ DBMS dan Struktur tabel	61
▪ Indeks dan Struktur penyimpanan.....	62
LATIHAN SOAL.....	64
BAB 9	65
PENGEMBANGAN SISTEM BASIS DATA.....	65
▪ Tujuan pengembangan sistem basis data.....	65
▪ Proses Pengembangan basisdata.....	66
▪ Langkah-langkah pengembangan sistem basis data.....	67
▪ Langkah-langkah mendisain basis data untuk SIM	67
▪ Alat bantu dan metode dalam pengembangan sistem basis data	68
LATIHAN SOAL.....	69
BAB 10.....	70

BAB 1

PENGANTAR DATABASE

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 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. Pengertian Database

Hampir semua aplikasi komputer didukung fasilitas database. Database merupakan komponen terpenting dalam pembangunan SI, sebagai tempat menampung dan mengorganisasi seluruh data yang ada dalam sistem. Database merupakan himpunan sekelompok data yang saling berkaitan, yang diolah dan diorganisasikan sehingga menjadi sebuah informasi yang berguna. Basis data (database) dapat dibayangkan sebagai sebuah lemari arsip dan berwenang /bertugas untuk :

1. Memberi sampul atau map pada kumpulan/bundel arsip yang akan disimpan.

2. Menentukan kelompok/jenis arsip yang akan disimpan
3. Memberikan penomoran dengan pola tertentu yang nilainya unik pada setiap sampul/map.
4. Menempatkan arsip tsb dengan cara/urutan tertentu didalam lemari.

Basis: markas atau gudang, tempat bersarang/berkumpul.

Data: representasi fakta dunia nyata yang mewakili suatu object seperti manusia, barang, hewan dll yang direkam dalam bentuk angka, huruf, teks, gambar, bunyi atau kombinasinya

▪ Basis Data:

- Himpunan kelompok data (arsip) yang saling berhubungan, yang diorganisasi sedemikian rupa, sehingga kelak dapat dimanfaatkan kembali dengan cepat.
- Kumpulan data yang saling berhubungan yang disimpan secara bersama tanpa adanya pengulangan (redudansi) data.
- Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.
- Basis data dan lemari arsip memiliki prinsip kerja dan tujuan yang sama : Prinsip utama adalah pengaturan data/arsip. Dan tujuan utamanya adalah kemudahan dan kecepatan dalam pengambilan kembali data / arsip. Perbedaannya hanya terletak pada media penyimpanan yang digunakan yaitu lemari dan disk.
- Yang sangat ditonjolkan dalam basis data adalah pengaturan / pemilahan / pengelompokan / pengorganisasian data yang akan kita simpan sesuai fungsi / jenisnya.
- Pemilahan / pengelompokan / pengorganisasian ini dapat berbentuk sejumlah file/tabel terpisah atau dalam bentuk pendefinisian kolom-kolom/field-field data dalam setiap file / tabel.

▪ Tujuan dari Basis Data :

1. Kecepatan dan Kemudahan (speed)

Memungkinkan penyimpanan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan cepat dan mudah.

2. Efisiensi Ruang Penyimpanan (space)

Dengan basis data, efisiensi/optimalisasi penggunaan ruang penyimpanan dapat dilakukan, karena dapat melakukan penekakan jumlah redudancy data.

3. Keakuratan (Accuracy)

Pemanfaatan pengkodean atau pembentukan relasi antar data bersama dapat menekan ketidakakuratan pemasukan/penyimpanan data.

4. Ketersediaan (Availability)

Data yang sudah jarang atau bahkan tidak pernah digunakan dapat diatur untuk dilepaskan atau disimpan di lokasi tertentu.

5. Kelengkapan (Completeness)

Dapat menambah record-record data, tetapi juga dapat melakukan perubahan struktur dalam basis data.

6. Keamanan (Security)

Dapat menentukan siapa-siapa (pemakai) yang boleh menggunakan basis data dan menentukan jenis-jenis operasi apa saja yang boleh dilakukan.

7. Kebersamaan Pemakaian (Sharability)

Data dapat digunakan oleh banyak pemakai.

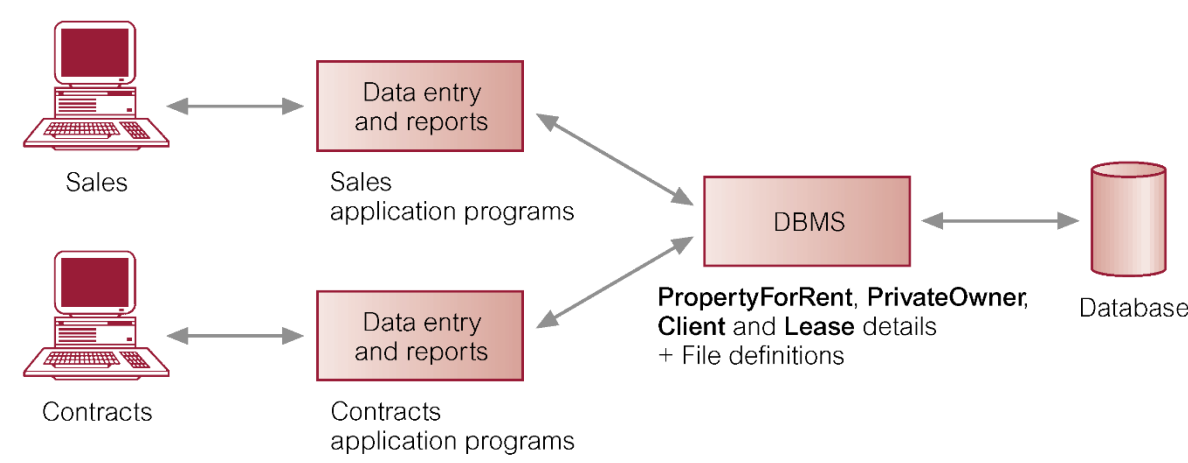
- Pemanfaatan basis data demi efisiensi, akurasi dan kecepatan operasi antara lain:
 1. Kepegawaian
 2. Pergudangan (inventory)
 3. Akutansi
 4. Reservasi
 5. Layanan Pelanggan (Customer Care)
- Komponen Sistem Basis Data:
 1. Perangkat Keras (Hardware)
Komputer, memori, storage (Harddisk), peripheral, dll.
 2. Sistem Operasi (Operating System)
Program yang menjalankan sistem komputer, mengendalikan resource
 3. komputer dan melakukan berbagai operasi dasar sistem komputer.
Basis Data (Database)
 4. Menyimpan berbagai obyek database (struktur tabel, indeks,dll)
DBMS (Database Management System)
 5. Perangkat lunak yang maintain data dalam jumlah besar.
Pemakai (User)
 6. Para pemakai database.
Aplikasi (perangkat lunak) lain.
 7. Program lain dalam DBMS.
- Operasi dasar yang berkenaan dengan basis data :
 1. Pembuatan basis data baru (create database)
 2. Penghapusan basis data (drop database)
 3. Pembuatan file/tabel baru (create table)
 4. Penghapusan file/table (drop table)
 5. Penambahan/pengisian data baru (insert)
 6. Pengambilan data dari sebuah table/file (retrieve/search)
 7. Pengubahan data dari sebuah file/table (update)
 8. Penghapusan data (delete)
- DBMS (Data Base Management System);
Adalah suatu program komputer yang digunakan untuk memasukkan, mengubah, menghapus, memanipulasi dan memperoleh data / informasi secara praktis dan efisien.
- DBMS berguna untuk :
 - Mendefinisikan struktur data
 - Memanipulasi nilai data yang ada dalam database.
 - Pengendalian akses terhadap suatu data :
 - untuk pengamanan sistem
 - untuk keterpaduan sistem

- sistem pengendalian persetujuan akses
- sistem pengendalian pemulihan.

▪ Contoh Produk DBMS :

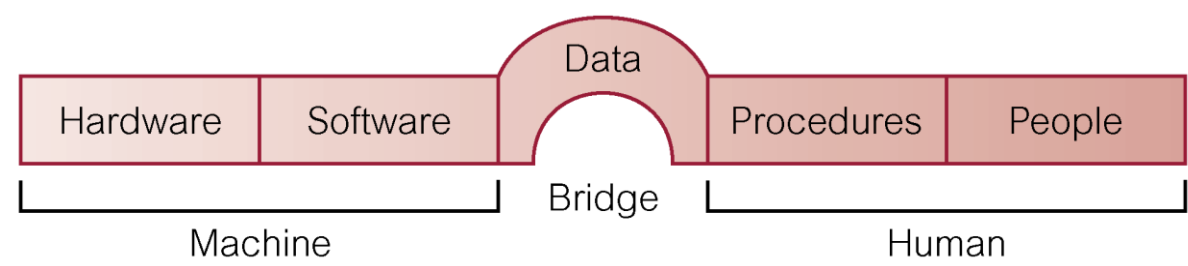
- Oracle (Oracle Corp.)
- Informix
- Sybase
- Dbase
- MS Sql Server
- MySQL

Sasaran utama DBMS menyediakan lingkungan yang nyaman dan efisien dalam menyimpan dan mengambil informasi ke dan dari basis data.



PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

▪ Komponen DBMS;



- Hardware
- Software
- Data
- Procedures
- People

▪ Pemakai / User :

- Programmer Aplikasi
 Cara berinteraksi dengan basis data melalui program yang ditulis
- User Mahir (Casual User)

Cara berinteraksi dengan basis data melalui query yang telah disediakan oleh DBMS

- User Umum (End User/Naïve User)

Cara berinteraksi dengan basis data melalui pemanggilan program aplikasi (executable program)

- User Khusus (Specialized User)

User yang dapat mengakses Basis data tanpa / dengan dbms, misalnya untuk keperluan Artificial Intelligence, Expert System dll

- Database Administrator (DBA);

Adalah orang yang bertanggung jawab terhadap sebuah database:

- Perencanaan
- Pengembangan dan perawatan standar
- Prosedur dan kebijakan penggunaan database
- Merancang database secara konsep maupun
- secara logika (database designers).

- Sejarah Database Sistem;

- First-generation
 - Hierarchical and Network
- Second generation
 - Relational
- Third generation
 - Object Relational
 - Object-Oriented

- Keuntungan DBMS

- Control of data redundancy
- Data consistency
- More information from the same amount of data
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale
- Balanced conflicting requirements
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services

- Kerugian DBMS;

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- Cost of conversion
- Performance
- Higher impact of a failure

LATIHAN SOAL

- 1.** Tuliskan 5 contoh manfaat dari database
- 2.** Jelaskan yang dimaksud dengan:
 - Data
 - Database
 - DBMS
 - Database program aplikasi
 - Data Independen
 - Keamanan data
 - Integrasi data
 - View / tampilan data
- 3.** Jelaskan pendekatan yang dilakukan terhadap penanganan data pada sistem berbasis file pada tahap awal. Diskusikan kelemahan dari pendekatan ini
- 4.** Jelaskan karakteristik utama dari pendekatan database dan kontras dengan pendekatan berbasis file
- 5.** Jelaskan lima komponen DBMS lingkungan dan diskusikan bagaimana kaitannya dengan masing-masing lain.
- 6.** Jelaskan peran personil berikut di lingkungan database:
 - (a) administrator data
 - (b) administrator database
 - (c) perancang database logis
 - (d) perancang basis data fisik
 - (e) pengembang aplikasi
 - (f) pengguna akhir
- 7.** Jelaskan kelebihan dan kekurangan menggunakan DBMS

BAB 2

SISTEM BASIS DATA

- **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

- **Komponen sistem basis Data;**

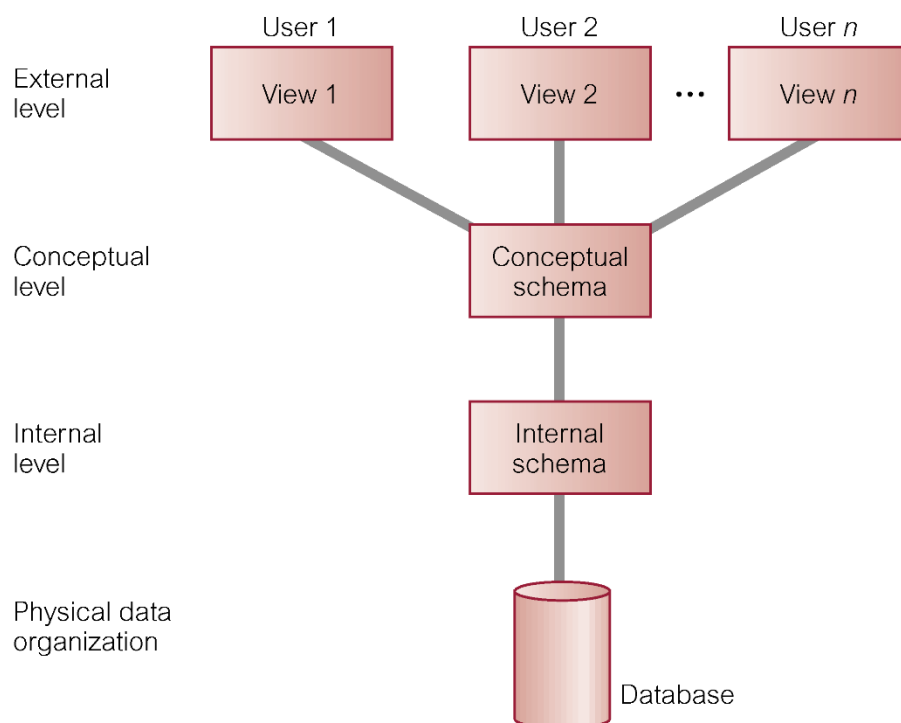
Komponen-komponen utama penyusun sistem basis data adalah :

- Perangkat keras
- Sistem operasi
- Basis data
- Sistem pengelola basis data (DBMS)
- Pemakai (Programmer, User mahir, user umum, user khusus)

- **Abstraksi Database;**

Dalam database, data disimpan dan dipelihara dengan baik dan terstruktur oleh DBMS. Sistem menyembunyikan detail tentang bagaimana data disimpan dan dipelihara. Sehingga seringkali data yang terlihat oleh user, berbeda dengan data yang tersimpan secara fisik.

Abstraksi data merupakan tingkatan/level dalam melihat bagaimana menampilkan data dalam sebuah sistem database.



- **Level Konseptual (Conceptual Level)**

Level terendah dalam abstraksi data, yang menunjukkan bagaimana sesungguhnya suatu data disimpan.

User melihat data sebagai gabungan dari struktur dan datanya sendiri.

Tingkatan ini berurusan dengan:

- Alokasi ruang penyimpanan untuk data dan indeks
- Deskripsi record untuk penyimpanan
- Penempatan record data
- Teknik kompresi dan enkripsi data

- **Level Fisik (Physical Level)**

Menggambarkan data apa yang sebenarnya disimpan dalam database, serta hubungannya (relationship) dengan data lainnya.

Berisi struktur logik database yang hanya dapat dilihat oleh DBA.

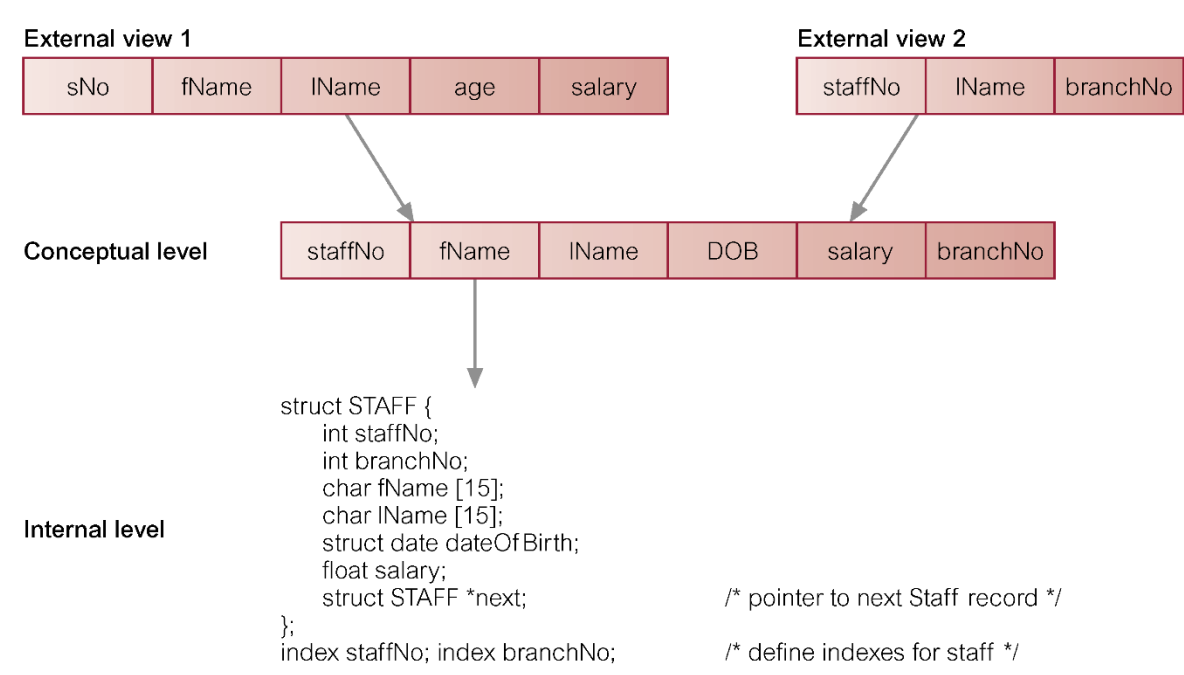
Tingkat konseptual ini menyatakan:

- Entitas, atribut dan relasinya
- Konstrain-konstrain terhadap data
- Informasi semantiks data
- Informasi keamanan dan integritas data

- **Level Pandangan (View Level)**

Level tertinggi dari abstraksi data, yang hanya menampilkan hanya sebagian dari database.

Tidak semua user membutuhkan semua data dalam database.



- **Penyusunan 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

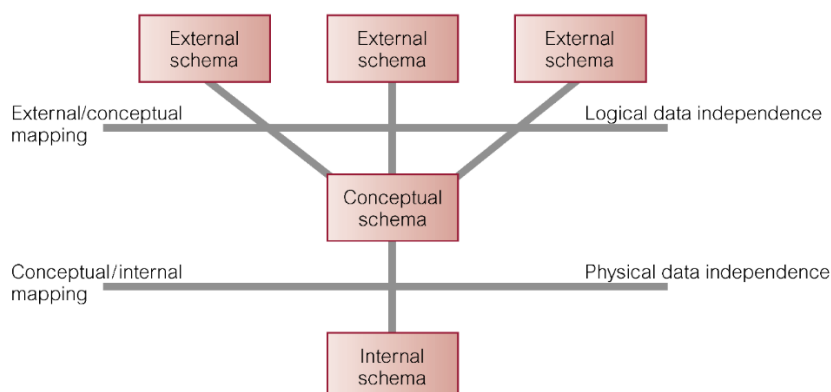
Bit → Byte → Data Item → Recoird → File → Basis Data → Sistem Basis Data

- **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)
 - o Recordnya relatif statis, jarang berubah nilainya
 - o Contoh : file daftar gaji, matakuliah
 - File induk dinamik (dynamic master file)
 - o Nilai dari recordnya sering berubah atau diupdate sebagai hasil suatu transaksi
 - o Contoh : file stok barang
- b. File transaksi (Transaction file)
 - Disebut juga file input. Digunakan untuk merekam data hasil transaksi
 - Contoh file penjualan barang
- c. File laporan (report file)
 - Disebut juga file output. Berisi informasi sementara yang akan ditampilkan sebagai laporan File laporan (report file)
- d. File sejarah (history file)
 - Disebut juga file arsip (archieval file).
 - Merupakan file yang berisi data masa lalu yang sudah tidk aktif lagi, tapi masih disimpan sebagai arsip
- e. File pelindung (bacup file)
 - Merupakan salinan dari file-file yang masih aktf di dalam basis data pada saat tertentu
 - Digunakan sebagai cadangan apabila file basis data yang aktf mengalami kerusakan atau hilang

▪ DATA INDEPENDENCE



▪ Bahasa Basis Data

DBMS merupakan perantara antara user dengan database. Cara komunikasi diatur dalam suatu bahasa khusus yang telah ditetapkan oleh DBMS.

Contoh: SQL, dBase, QUEL, dsb.

Bahasa database, dibagi dalam 2 bentuk:

- Data Definition Language (**DDL**)

- Data Manipulation Language (DML)

- **Data Definition Language (DDL)**

Digunakan dalam membuat tabel baru, indeks, mengubah tabel, menentukan struktur tabel, dsb. Hasil dari kompilasi perintah DDL berupa kumpulan tabel yang disimpan dalam file khusus: **Kamus Data (Data Dictionary)**.

Data Dictionary: merupakan metadata (superdata), yaitu data yang mendeskripsikan data sesungguhnya. Data dictionary ini akan selalu diakses dalam suatu operasi database sebelum suatu file data yang sesungguhnya diakses.

- **Data Manipulation Language (DML)**

Digunakan dalam memanipulasi dan pengambilan data pada database. Manipulasi data, dapat mencakup:

- Pemanggilan data yang tersimpan dalam database (query)
- Penyisipan/penambahan data baru ke database
- Penghapusan data dari database
- Pengubahan data pada database

- **Terdapat dua (2) jenis DML:**

- **Prosedural**

Menghendaki user untuk menspesifikasikan data apa yang diperlukan dan bagaimana cara mendapatkan data itu.

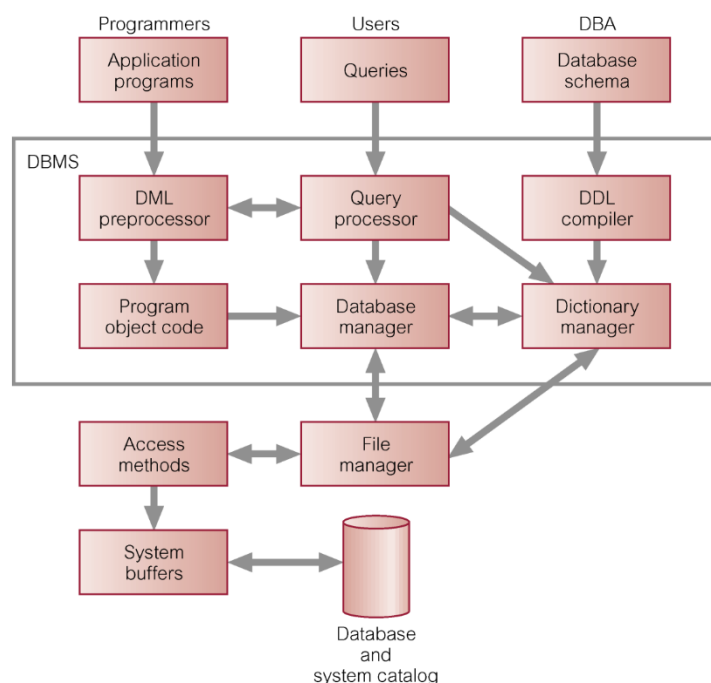
Contoh: bahasa C/C++, PL/SQL, dsb.

- **Nonprosedural**

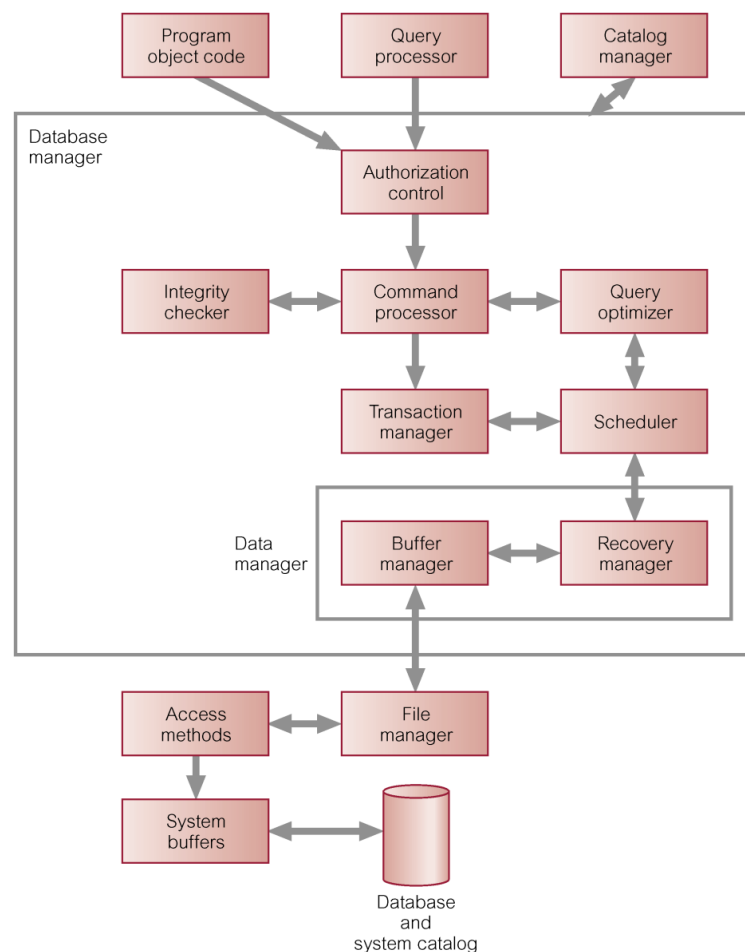
Menghendaki user untuk menspesifikasikan data apa yang dibutuhkan, tanpa harus menspesifikasikan bagaimana cara mendapatkan data tersebut.

Contoh: SQL

- **Komponen DBMS**



- **Komponen Data Manager**



- **Penggunaan Basis Data**

- 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 struktu
 - 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 database, yaitu:

- Programmer aplikasi

- Membuat program aplikasi

- Casual user / Naïve User

- Pemakai yang sudah mahir, berinteraksi dengan sistem tanpa menulis program, tapi menggunakan query

- End User

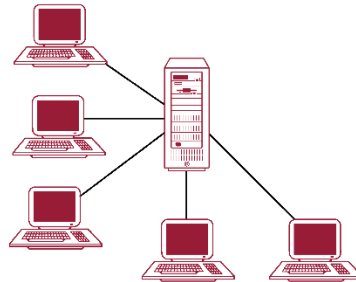
- Pemakai yang belum mahir, tinggal menjalankan aplikasi yang sudah dibuat oleh programmer aplikasi

- Specialized User

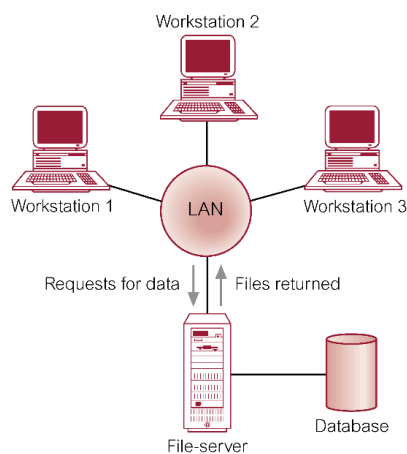
- Pemakai khusus yang menuliskan aplikasi databaset tidak dalam kerangka pemrosesan data, namun untuk keperluan khusus seperti CAD, AI, EIS, dll

▪ Multi User DBMS Arsitektur

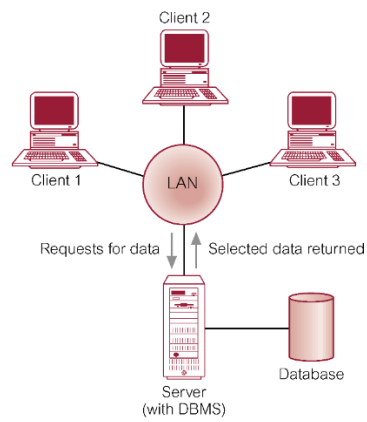
- Teleprocessing
 - Traditional Architecture
 - Single mainframe dengan beberapa terminal
 - Trend is now towards downsizing



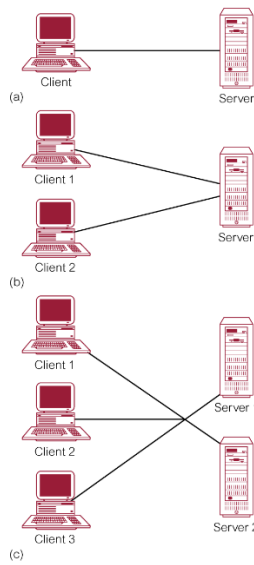
- File Server
 - File-server is connected to several workstations across a network.
 - Database resides on file-server.
 - DBMS and applications run on each workstation.
 - Disadvantages include:
 - Significant network traffic.
 - Copy of DBMS on each workstation.
 - Concurrency, recovery and integrity control more complex.



- Client Server
 - Server holds the database and the DBMS.
 - Client manages user interface and runs applications.
 - Advantages include:
 - wider access to existing databases;
 - increased performance;
 - possible reduction in hardware costs;
 - reduction in communication costs;
 - increased consistency.

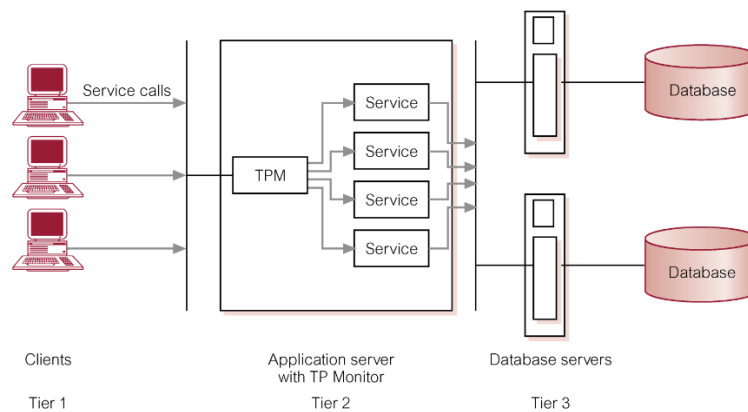


- Alternatif Clie server Topologies



- Transaction Processing Monitors

Program that controls data transfer between clients and servers in order to provide a consistent environment, particularly for Online Transaction Processing (OLTP).



LATIHAN SOAL

1. Jelaskan yang dimaksud dengan sistem basis data
2. Sebutkan komponen sistem basis data
3. Sebutkan dan Jelaskan setiap level / tingkatan dalam melihat tampilan sistem data base
4. Sebutkan dan jelaskan Tipe - tipe file
5. Sebutkan dan jelaskan setiap user database
6. Jelaskan perbedaan antara DDL dan DML berikan contoh
7. Sebutkan dan jelaskan setiap jenis arsitektur DBMS

BAB 3

LINGKUNGAN BASIS DATA

- **Batasan dalam database**

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

- a. Redudansi Data**

- Yaitu muncul data-data yang sama secara berulang pada beberapa file 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), alamattext(40)

- File Krs → NIK text(10), Nama text(30), Gol text(4), Gapok Double

- b. Inkonsistensi Data**

- Yaitu muncul data yang inkonsisten pada field yang sama dengan 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
 - 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. Multi user**

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

- e. Masalah Keamanan**

- 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 terhadap field, tidak perlu merubah programnya

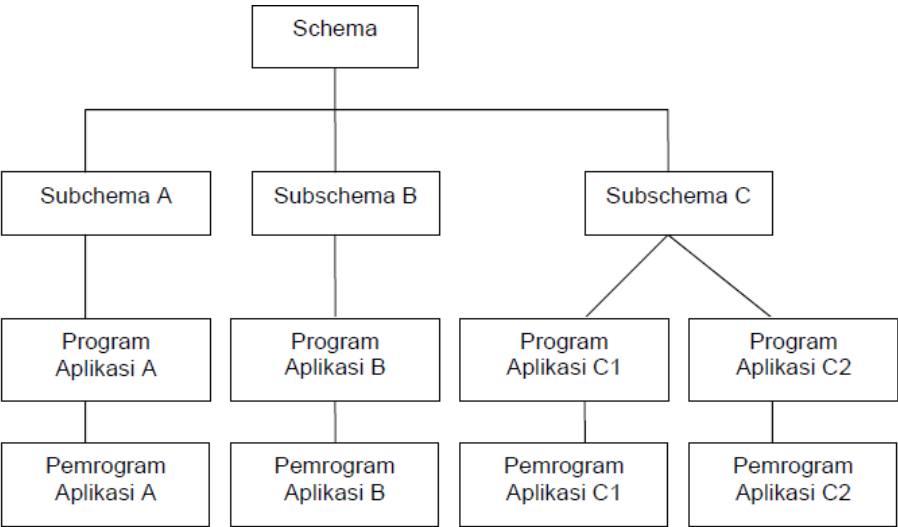
▪ **Organisasi file basis data**

- Tujuan organisasi file dalam sistem basis data:
 - Menyediakan sarana pencarian record bagi pengolahan, seleksi ,atau penyaringan
 - Memudahkan pembuatan atau pemeliharaan file
- Ada 2 jenis media penyimpan file :
 - SASD (Sequential Access Storage Device)
 - Proses pembacaan record harus berurutan
 - Tidak ada pengalamatan
 - Data disimpan dalam bentuk blok
 - Proses penulisan hanya bisa dilakukan sekali
 - Contoh : magnetic tape
 - DASD (Direct Access Storage Device)
 - Pembacaan record tidak harus urut
 - Mempunyai alamat
 - Data dapat disimpan dalam karakter atau blok
 - Proses penulisan dapat dilakukan beberapa kali
 - Contoh : harddisk, floppy disk

- **Metode susunan file :**
 - Sequential (urut)
 - Record disimpan berdasarkan suatu kunci
 - Pencarian record tertentu dilakukan record demi record berdasarkan kuncinya
 - Random (Acak)
 - Kunci record ditransformasikan ke alamat penyimpanan dalam media fisik secara acak
 - 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.
 - Indexed Random
 - Record disimpan secara acak
 - Masing-masing record memiliki indeks

- **Schema dan Subschema**
 - Schema dan Subschema diperlukan untuk menggambarkan hubungan logik antara data dalam basis data
 - 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.
 - Subschema, merupakan deskripsi terpisah dari rinci data, record, set, dan area yang digunakan oleh program aplikasi.

Hubungan hirarkis:



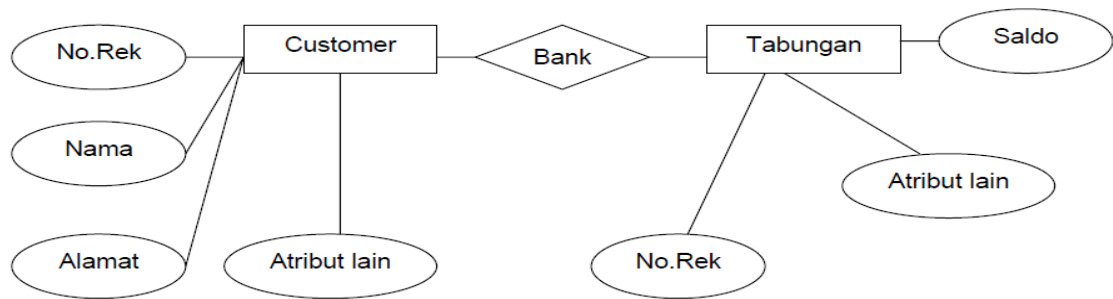
- **Arsitektur Sistem Basis data**
 - Pertimbangan dalam memilih arsitektur sistem basis data:

- Keunggulan teknologi
 - Biaya pengembangan
 - Sesuai dengan kebutuhan pengguna
- 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 basisdata, DBMS, dan aplikasi basis data
 - **Jenisnya:**
 - Aplikasi dan basisdata terpusat; diakses oleh dumb terminal
 - Basisdata terpusat; aplikasi ada pada terminal
 - ✓ **Sistem Client server**
 - Ditunjukkan 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.
 - **Jenisnya;**
 - Arsitektur 2 lapis (2-tier)
 - Arsitektur 3 lapis (3-tier)
- **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 table baru, membuat indeks, mengubah tabel, menentukan struktur tabel, dll

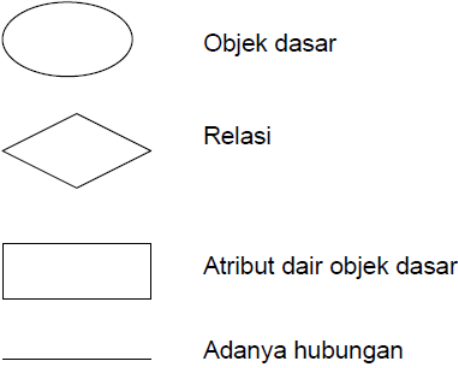
- Hasil dari kompilasi perintah DDL menjadi kamus data, yaitu data yang menjelaskan data sesungguhnya
 - Contoh : Create, Modify, report, Modify Structure
 - Recovery
 - Merupakan kemampuan untuk mengembalikan data yang rusak atau hilang akibat operasi basis data (insert, update, delete.dll)
 - Data 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.
- **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.
 - **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.
 - Jenisnya:
 - i. Entity Relationship model
 - ✓ Merupakan model untuk menjelaskan hubungan antar data dalam basis data berdasarkan persepsi bahwa real world(dunia nyata) terdiri

dari objek-objek yang mempunyai hubungan / relasi antar objek tersebut.

✓ Contoh:



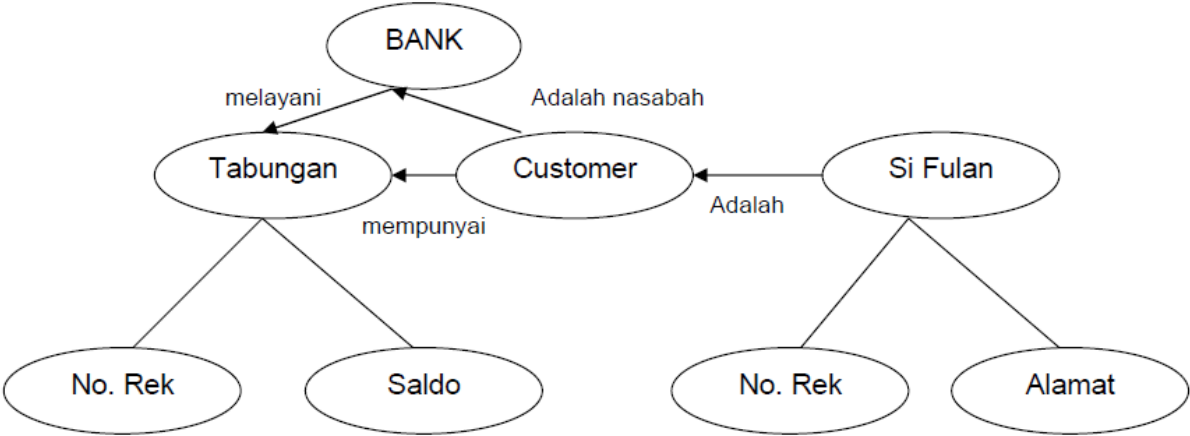
✓ Arti simbol :



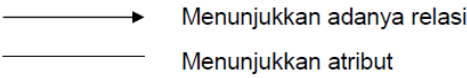
ii. Semantic model

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

✓ Contoh:



Arti tanda :



b. Model Data berbasis Record

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

i. 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

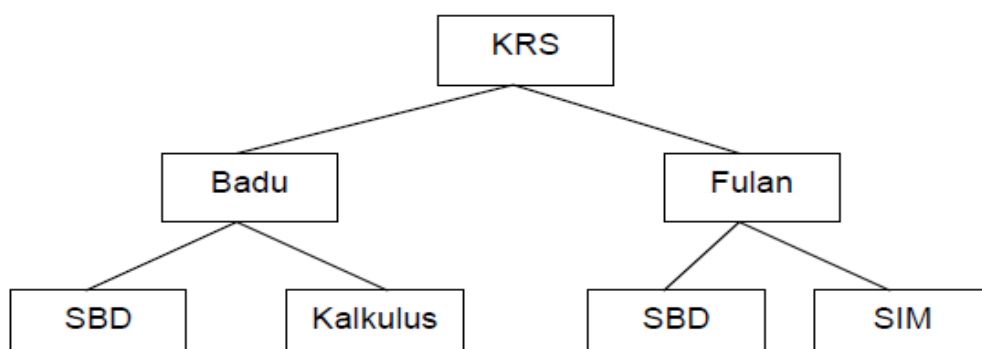
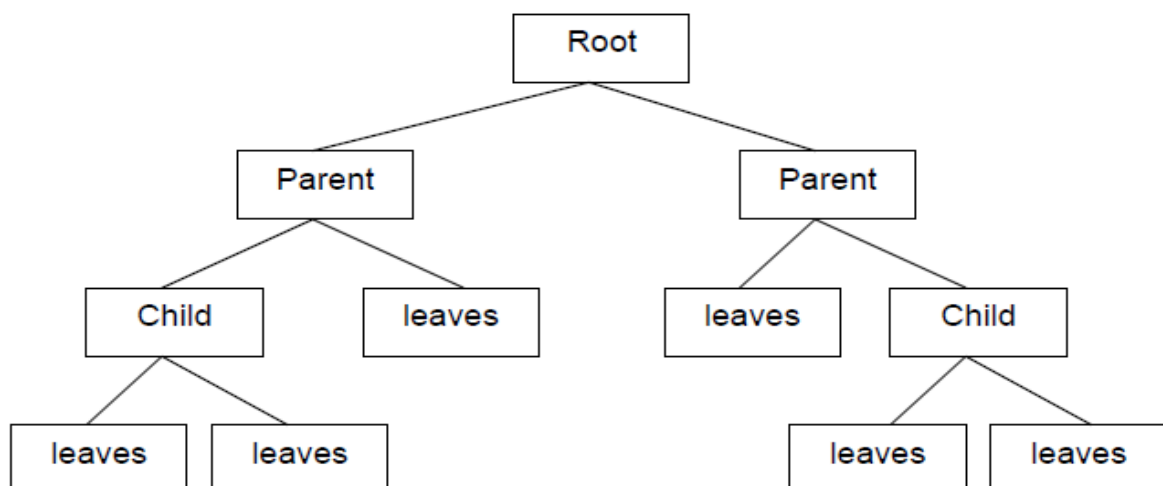
Nomhs	Nama
00351234	Fulan
01351346	Badu
02351370	Ayu

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

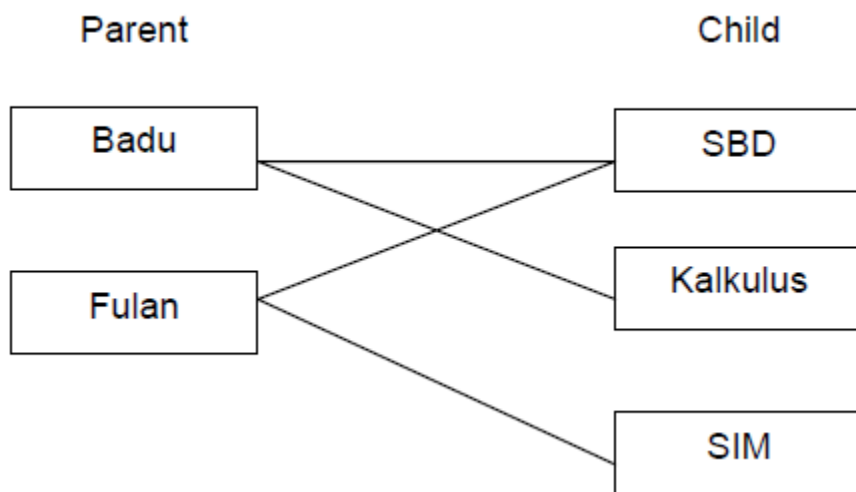
ii. Hirarchycal Model

- ✓ Menjelaskan tentang hubungan logic antar data dalam basis data dalam bentuk hubungan bertingkat (hirarki)
- ✓ Elemen penyusunannya disebut node, yang berupa rincian data, agregat data, atau record
- ✓ Contoh; Model Hirarki



iii. Network Model

- ✓ Hampis sama dengan model hirarki, dan digambarkan sedemikian rupa sehingga child pasti berada pada level yang lebih rendah dari pada 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:
 - i. Unifying model
 - ✓ Model ini menggabungkan memori dan transaksi database dalam satu kesatuan model
 - ii. Frame memory
 - ✓ Sebuah virtual model dari tempat penyimpanan sekunder yang digunakan untuk mendukung penyimpanan record database

LATIHAN SOAL

1. Sebutkan dan jelaskan masalah / batasan dalam menyusun database
2. Sebutkan dan jelaskan metode dalam penyusunan file
3. Jelaskan yang dimaksud dengan Schema dan Subschema
4. Sebutkan dan jelaskan jenis-jenis model data
5. Jelaskan perbedaan antara SASD dan DASD

BAB 4

RELATIONAL DATABASE MODEL

▪ Relasi Tabel

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 menunjukkan atribut-atribut

- Istilah-istilah dalam model basis data relasional:
 - Record : sebuah baris dalam suatu relasi. Disebut juga tuple
 - Cardinality : banyaknya record dalam sebuah relasi
 - Atribut : suatu kolom dalam sebuah relasi
 - Domain : batasan nilai dalam atribut dan tipe datanya
 - Derajat / degree : banyaknya kolom dalam relasi
 - Candidate Key : atribut atau sekumpulan atribut yang unik yang dapat digunakan untuk membedakan suatu record
 - Primary Key : salah satu dari CK yang dipilih dan dipakai untuk membedakan suatu record
 - Alternate Key : CK yang tidak dipilih menjadi PK
 - Unary relation : suatu relasi yang hanya mempunyai satu kolom
 - Binary relation : suatu relasi yang hanya mempunyai dua kolom
 - Ternary relation : suatu relasi yang mempunyai tiga kolom
- 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 kolom pengulangan.
 - b. Semua entry / elemen data pada suatu kolom tertentu dalam relasi yang sama harus mempunyai jenis yang sama.
 - c. Masing-masing kolom dalam suatu relasi mempunyai nama yang unik
 - d. Pada suatu relasi / tabel yang sama tidak ada dua baris yang identik
- **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 batasan nilai dan tipe datanya
 - Batasan integritas dipengaruhi oleh integritas referensial yang meliputi key constraint dan referensial constraint.
 - Key constraint tidak mengijinkan adanya nilai null pada atribut yang digunakan sebagai PK
 - Referential 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

▪ **Kunci Relasi**

- Dasar penentuan PK adalah bahwa nilai- nilai rinci data dari atribut yang digunakan sebagai PK harus unik, tidak mungkin ada nilai rinci yang sama pada semua record dalam basis data
- Aturan-aturan lainnya:
 - Integrity entity
 - Nilai atribut yang dipilih sebagai PK tidak boleh null untuk setiap record yang ada didalam relasi
 - Aturan ini menjamin bahwa semua record yang ada dalam basis data akan dapat diakses karena semua recend dapat diidentifikasi berdasarkan kunci yang unik
 - Contoh:

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

*) Primary key

▪ Integritas referensial

- Jika dua buat 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

▪ **Relasi antar entity**

- Ada dua jenis:
 - Relasi antar entity dalam satu tabel
 - Berupa relasi antar rentity 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
- Yang harus diperhatikan adalah bagaimana agar relasi-relasi yang ada dalam sistem basis data dapat dihubungkan satu sama lain.
- **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 sistematis
 - Kebutuhan ruang penyimpanan data lebih efisien
 - Redundansi data yang optimal akan meningkatkan integritas data
 - Tidak ada ambiguitas data di semua tabel.

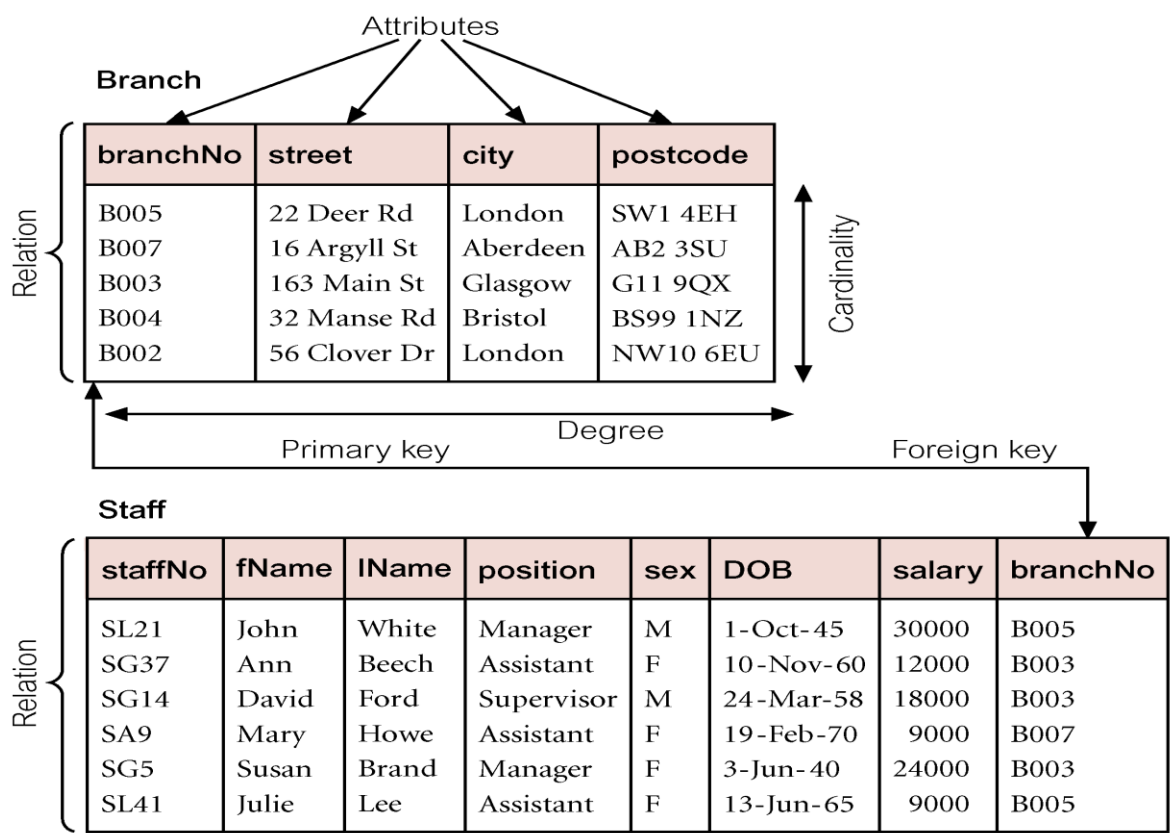
LATIHAN SOAL

1. Jelaskan yang dimaksud dengan relasi tabel
2. Jelaskan kegunaan dari relasi tabel
3. Sebutkan karakteristik model data relasional
4. Sebutkan dan jelaskan komponen-komponen relasional
5. Jelaskan istilah-istilah berikut ini:
 - Record
 - Cardinality
 - Atribut
 - Domain
 - Degree
 - Primary Key
 - Alternate Key
 - Candidate key
 - Unary relation
 - Ternary relation

BAB 5

ENTITY RELATIONSHIP DIAGRAM

- **Pengantar ERD**
 - 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?



- **Notasi dan artinya**
 - a. **Entiti** : adalah suatu objek yang dapat didefinisikan 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

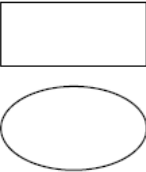
c. Hubungan / Relasi : menunjukkan adanya hubungan / relasi diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda

- 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 / jajar genjang

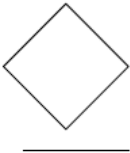
▪ **Notasi simbolik dalam bentuk E-R**

- Persegi panjang → himpunan entitas
- Elips → atribut (atribut yang sebagai kunci digarisbawahi)
- Belah ketupat → himpunan relasi
- Garis → Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atributnya.
- Kardinalitas relasi dinyatakan dengan banyak 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 :



Himpunan entitas
atribut



Himpunan relasi
Link

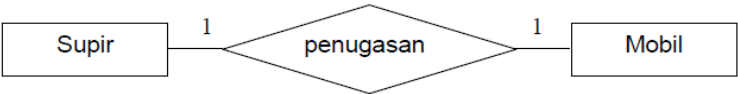
▪ **Jenis hubungan / Derajat relasi / Kardinalitas relasi**

- Kardinalitas relasi menunjukkan jumlah maksimum entitas yang dapat berelasi 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 hubungan:

- 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:

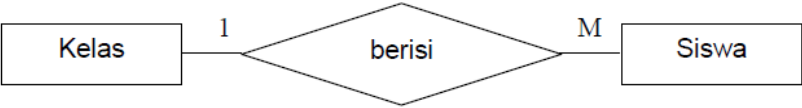


- Satu ke banyak (one to many)

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 A

Contoh:



- 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

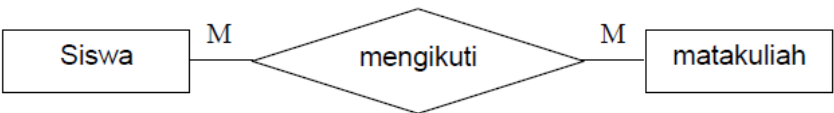
Contoh:



- 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 B dapat berhubungan dengan banyak entitas pada himpunan entitas A.

Contoh:



▪ **Tahap pembuatan diagram E-R**

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

1. Tahap awal pembuatan (preliminary design)

- Untuk mendapatkan rancangan basis data minima yang dapat mengakomodasi kebutuhan penyimpanan data terhadap sistem yang akan dibangun
- Pada umumnya mengabaikan adanya penyimpangan – penyimpangan

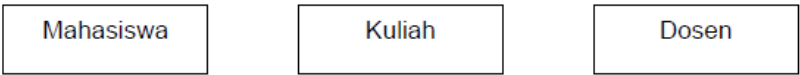
2. Tahap optimallisasi (final design)

- Dilakukan koreksi terhadap hasil tahap awal, dengan memperhatikan aspek efisiensi, performasi, 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

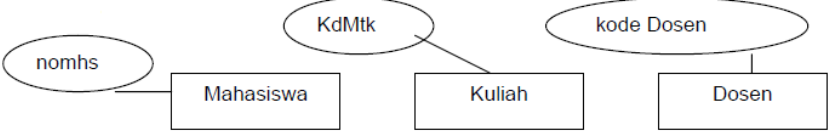
- **Langkah – langkah menyusun diagram awal ER:**
 - Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat
 - Menentukan atribut-atribut kunci dari masing-masing himpunan entitas
 - Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign key nya
 - Menentukan derajat relasi (cardinality) untuk setiap himpunan relasi
 - Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci)

Contoh : kasus pada perkuliahan

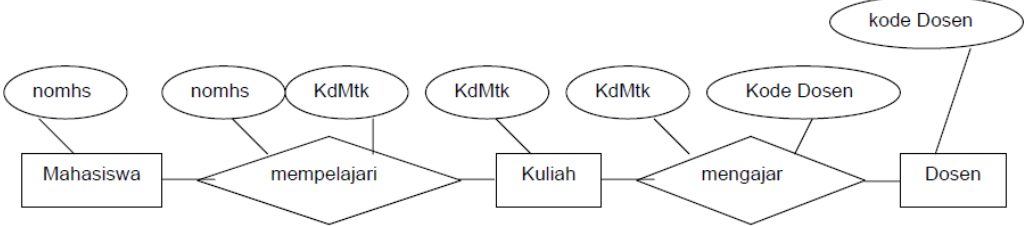
- Langkah 1 : Mengidentifikasi dan menetapkan seluruh himpunan entitas yang akan terlibat



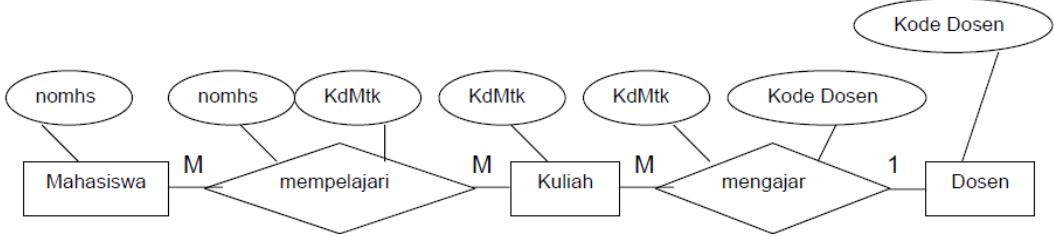
- Langkah 2 : Menentukan atribut-atribut kunci dari masing-masing himpunan entitas



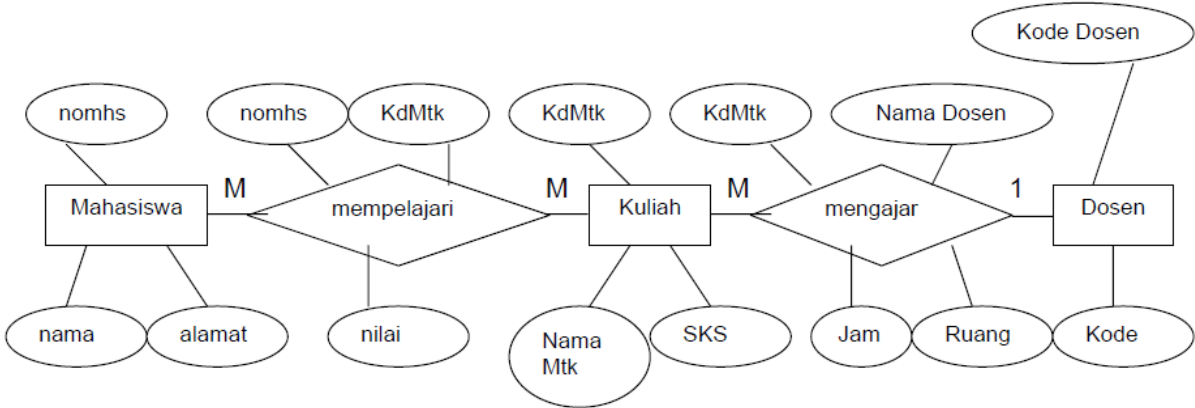
- Langkah 3 : Mengidentifikasi dan menetapkan seluruh himpunan relasi diantara himpunan entitas yang ada beserta foreign keynya



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



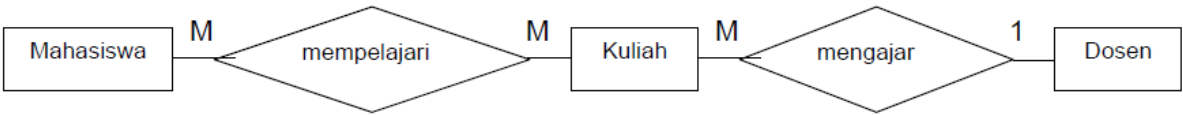
- Langkah 5 : Melengkapi himpunan entitas dan himpunan relasi dengan atribut deskriptif (yang bukan kunci)



▪ **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 digaris bawah

Contoh:



Kamus data :

- Mahasiswa = {nomhs, nama, alamat}
- Kuliah = {kdmk, nama, mk, sks}
- Dosen = {kodedosen, nama }
- Mempelajari = {nomhs, kdmk, nilai }
- Mengajar = {kdmk, kodedosen, jam, ruang }

▪ **Derajat Relasi**

adalah jumlah entitas yang berpartisipasi dalam suatu relasi.

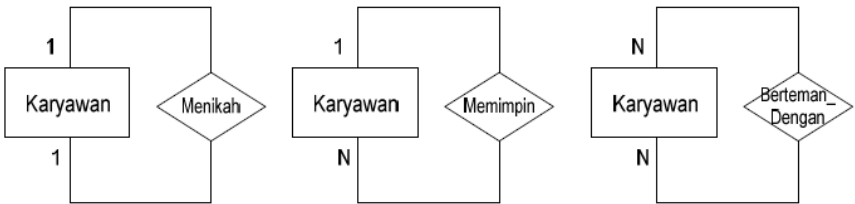
Derajat Relasi dapat berupa:

- Unary Relationship (Relasi Berderajat 1)
- Binary Relationship (Relasi Berderajat 2)
- Ternary Relationship (Relasi Berderajat 3)

• **Unary Relationship (Relasi Berderajat 1)**

Adalah relasi dimana entitas yang terlibat hanya 1, dapat juga disebut relasi rekursif (recursive relationship)

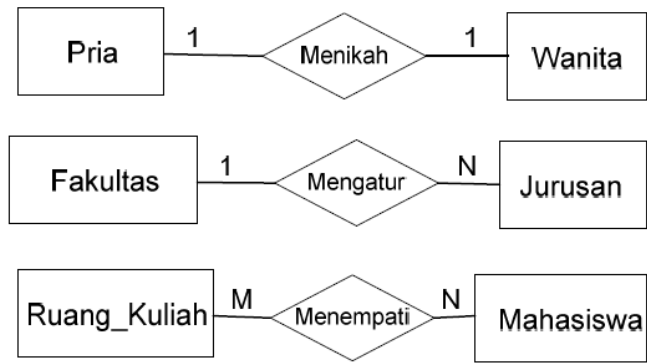
Contoh:



• **Binary Relationship (Relasi Berderajat 2)**

Relasi yang melibatkan 2 entitas

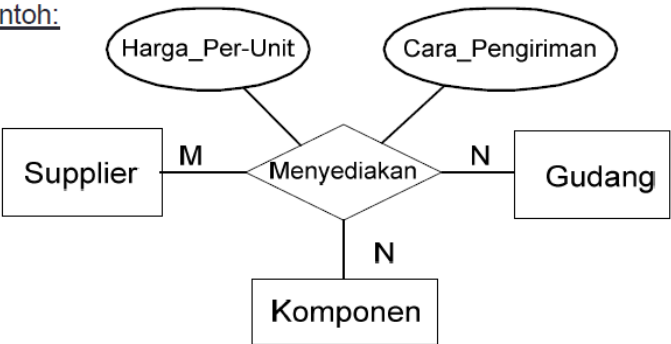
Contoh:



- **Ternary Relationship (Relasi Berderajat 3)**

Relasi tunggal yang menghubungkan 3 entitas yang berbeda

Contoh:



Penentuan derajat relasi tergantung dari aturan bisnis (business rule) yang ada di perusahaan .instansi. Walau notasi derajat relasi tidak sedominan notasi yang lain, tapi derajat relasi sangat berperan dalam tahap implementasi.

LATIHAN SOAL

1. Jelaskan yang dimaksud dengan ERD
2. Jelaskan yang dimaksud dengan :
 - Relasi
 - Atribut
 - Tuple
 - Cardinality
 - Primary key
 - Foreign key
 - Degree
 - Entiti
3. Sebutkan jenis-jenis hubungan atau relasi dalam ER diagram
4. Sebutkan tahap-tahap pembuatan diagram ER
5. Sebutkan dan beri contoh jenis-jenis derajat relasi

BAB 6

NORMALISASI

▪ **Normalisasi**

adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basisdata relasional”. Pada dasarnya, normalisasi adalah proses dua langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel relasional.

▪ **Tujuan normalisasi :**

1. Mengatur data dalam kelompok-kelompok sehingga masing-masing kelompok hanya menangani bagian kecil sistem.
2. Meminimalkan jumlah data yang berulang dalam basisdata.
3. Membuat basisdata yang datanya diakses dan dimanipulasi secara cepat dan efisien tanpa melupakan integrasi data.
4. Mengatur data sedemikian rupa sehingga ketika memodifikasi data, anda hanya mengubah pada suatu tempat.

▪ **Aturan Normalisasi**

1. Hilangkan kelompok berulang, buat tabel terpisah untuk setiap himpunan atribut yang berhubungan dan tentukan kunci utama pada masing-masing tabel.
2. Hilangkan data berulang, jika sebuah atribut hanya tergantung pada sebagian kunci utama gabungan, pindahkan atribut ke tabel lain.
3. Hilangkan kolom yang tidak tergantung pada kunci, jika atribut tidak tergantung pada kunci, pindahkan atribut ke tabel lain.
4. Pisahkan relasi majemuk, tidak ada tabel yang bias mengandung dua atau lebih relasi 1:n atau n:m yang tidak berhubungan langsung.
5. Pisahkan relasi majemuk yang berhubungan secara semantik, ada batasan pada informasi yang memperbolehkan pemisahan relasi many-to-many yang berhubungan secara logis.
6. Bentuk normal optimal, sebuah model hanya dibatasi oleh fakta sederhana.
7. Bentuk normal domain-key, sebuah model harus terbebas dari semua anomaly

▪ **Penyimpangan dalam modifikasi**

- Penyimpangan dalam proses modifikasi data disebut anomalies
- Ada 3 bentuk penyimpangan:
 - a. Delete anomalies
 - Adalah proses penghapusan suatu entry logik yang mengakibatkan hilangnya informasi tentang entiti 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
 - o Adalah proses penyisipan entity logik yang memerlukan penyisipan entity logik yang lain
- c. Update anomalies
 - o Adalah proses mengupdate data pada suatu entiti logik yang mengakibatkan perubahan pada lebih dari satu tempat dalam suatu relasi
 - o Contoh : Perubahan SKS pada “INA 101” tidak hanya dilakukan pada satu record saja, tetapi pada record dan relasi lain yang memuat data tersebut.

- **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 :
 - o **Functional Dependence (FD)**
 - FD akan muncul diantara dua rinci data dalam suatu strutktur data jika nilai salah satu rinci data mengimplikasikan nilai pada rinci data kedua
 - Rinci data pertama menentukan (determines) rinci data keduaContoh:
MataKuliah (Kode, Nama, SKS, Semester)
FD = Matakuliah.kode → (MataKuliah.Nama, MataKuliah.Semester)
MataKuliah.nama → (MataKuliah.kode, MataKuliah.Semester)
 - o **Full Functional Depedence (FFD)**
 - Suatu rinci data dikatakan FFD pada suatu kombinasi rinci data jika FD pada kombinasi rinci data dan tidak FD pada bagian lain dari kombinasi rinci data
 - Contoh: SKS pada tabel matakuliah hanya bergantung pada kode matakuliah, dan tidak ditentukan oleh siapa yang mengambil matakuliah tersebut.
 - o **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 data ketiga.

- Jadi TD terjadi jika suatu nilai rinci data mempunyai ketergantungan dengan pada dua nilai rinci data.
- **Efek – efek normalisasi**
 - Akibat yang muncul dalam proses normalisasi:
 - 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 inheren pada batasan integritas referensial
 - Ketidakefisienan dalam menampilkan kembali data tersebut
- **Jenis – Jenis Key Field / attribute tabel**
 - Atribut adalah karakteristik atau sifat yang melekat pada sebuah tabel, atau disebut juga kolom data
 - Pengelompokan atribut:
 1. *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 tertentu)
 - Ada 3 Key:
 - a. SuperKey

Yaitu himpunan dari satu atau lebih entitas yang digunakan untuk mengidentifikasi secara unik sebuah entitas dalam entitas set.

Contoh : superkey di tabel mahasiswa

 - (nomhs, nama, alamat, tgllahir)
 - (nomhs, nama, tgllahir)
 - (nomhs, nama)
 - (nomhs)
 - b. Candidate key

Yaitu satu attribute atau satu set minimal attribute yang mengidentifikasi secara unik suatu kejadian yang spesifik dari entity.

Sebuah CK pasti superkey, tapi belum tentu sebaliknya

Contoh : pada tabel mahasiswa

 - (nomhs)
 - (nama)
 - c. Primary key

Yaitu satu attribute atau satu set minimal attribute yang tidak hanya mengidentifikasi secara unik suatu kejadian yang spesifik tapi juga dapat mewakili setiap kejadian dari suatu entity

Dari beberapa CK dapat dipilih satu untuk dijadikan PK, yang memiliki keunikan paling baik

Contoh : dari tabel mahasiswa, yang layak dijadikan PK adalah nomhs

2. *Alternate key*

Yaitu kunci kandidat yang tidak dipakai sebagai primary key

3. *Foreign key*

yaitu satu atribut (atau satu set atribut) yang melengkapi satu relationship (hubungan yang menunjukkan ke induknya)

4. *Atribut deskriptif*

Merupakan atribut yang bukan merupakan anggota dari PK

5. *Atribut sederhana*

Adalah atribut atomik yang tidak dapat dipilah lagi

Contoh: Nomhs, nama

6. Atribut komposit

Adalah atribut yang masih bisa diuraikan lagi menjadi sub-atribut yang masing-masing memiliki makna

7. Atribut bernilai tunggal

Ditunjukkan 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

8. Atribut bernilai banyak

Ditunjukkan pada atribut-atribut yang dapat diisi dengan lebih dari satu nilai, tapi jenis sama

Contoh: Pada tabel mahasiswa dapat ditambah atribut Hobby, karena seorang mahasiswa dapat memiliki beberapa hobby

9. 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 0) tidak sama dengan SPASI (karakter 32)

▪ **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 penyimpanan data yang mungkin bagi suatu atribut secara fisik, tanpa melihat kelayakan data tersebut bila dilihat dari kenyataan pemakaiannya.

▪ **Tahapan Normalisasi**

1. Bentuk Normal Kesatu (1 NF / First Normal Form)

Bentuk Bentuk Normal Kesatu mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa “atomic 12 value”. Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (multi value). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu arti saja dan juga bukanlah pecahan kata kata sehingga artinya lain.

Atom adalah zat terkecil yang masih memiliki sifat induknya, bila dipecah lagi maka ia tidak memiliki sifat induknya.

2. Bentuk Normal Kedua (2NF)

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya. Bentuk Normal Ketiga (3NF)

3. Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Artinya setiap atribut bukan kunci harus bergantung hanya pada kunci primer secara menyeluruh. Contoh pada bentuk Normal kedua di atas termasuk juga bentuk Normal Ketiga karena seluruh atribut yang ada di situ bergantung penuh pada kunci primernya.

4. Boyce-Codd Normal Form (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut super key.

Contoh

UNNORMAL FORM

SISWA(No_siswa, Nama, Wali_studi, Kelas1, Kelas2, Kelas3)

- Siswa yang punya nomor siswa, nama dan wali studi mengikuti 3 mata pelajaran/ kelas. Disini ada perulangan kelas 3 kali ini bukan bentuk 1 NF.

Contoh Data:

No_siswa	Nama	Wali_studi	Kelas-1	Kelas2	Kelas3
22890100	Tanzania	Zaman	1234	1543	
22890101	Nia	Rizki	1234	1775	1543

KELAS(Kode_kelas,nama_kelas,instruktur)

- merupakan bentuk 1NF karena tidak ada yang berganda dan tiap atribut satu pengertian yang tunggal.

Contoh Data:

Kode_kelas	Nama_kelas	Instruktur
1234	FisikaDasar	Suroso
1543	Matematika	Paulus
1775	Teknik Program	Bagus

No_siswa	Nama	Wali_studi	Kelas-1	Kelas2	Kelas3
22890100	Tanzania	Zaman	1234	1543	
22890101	Nia	Rizki	1234	1775	1543

FIRST NORMAL FORM

Bentuk normal kesatu dari bentuk diatas menjadi:

No_siswa	Nama	Wali_studi	Kode_kelas
22890100	Tanzania	Zaman	1234
22890100	Tanzania	Zaman	1543
22890101	Nia	Rizki	1234
22890101	Nia	Rizki	1775
22890101	Nia	Rizki	1543

PT SANTA PURI				FAKTUR PEMBELIAN	
BARANG					
Jalan Senopati 11					
Yogyakarta					
Kode	Nama Barang	Qty	Harea	Jumlah	
R02	RICE COOKER CC3	10.0	150.000, 00	1. 500.000,00	
Total Faktur				1. 500.000,00	
Jatuh Tempo Faktur: 09/03/90					

PT SANTA PURI				FAKTUR PEMBELIAN	
BARANG					
Jalan Senopati 11					
Yogyakarta					
Kode	Nama Barang	Qty	Harea	Jumlah	
A01	AC SPLIT 1/2 PK	10.0	1,350,000	13,500,000	
A02	AC SPLIT 1 PK	10.0	2,000,000	20,000,000	
Total Faktur				33,500,000	
Jatuh Tempo Faktur: 09/03/90					

1. Step I bentuk unnormalized

Bentuklah menjadi tabel unnormalized, dengan mencantumkan semua field data yang ada. Menuliskan semua data yang akan direkam Terlihat record record yang tidak lengkap, sulit untuk membayangkan bagaimana bentuk record yang harus dibentuk untuk merekam data tersebut.

No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	TGL	Jatuh Tempo	Qty	Harga	Jumlah	Total
779	S02	Hitachi	R02	Rice cooker	02/02/2009	09/03/2009	10	150.000		
998	G01	Global	A01	Ac Split ½ pk	7/02/2010	09/03/2010	10	1.250.000		
			A02	AC SPLIT 1 PK			10	1.2450.000		

2. Step II bentuk normal kesatu

Bentuklah menjadi bentuk normal kesatu dengan memisah misahkan data pada field field yang tepat dan bernilai atomic, juga seluruh record harus lengkap adanya. Bentuk file adalah flat file.

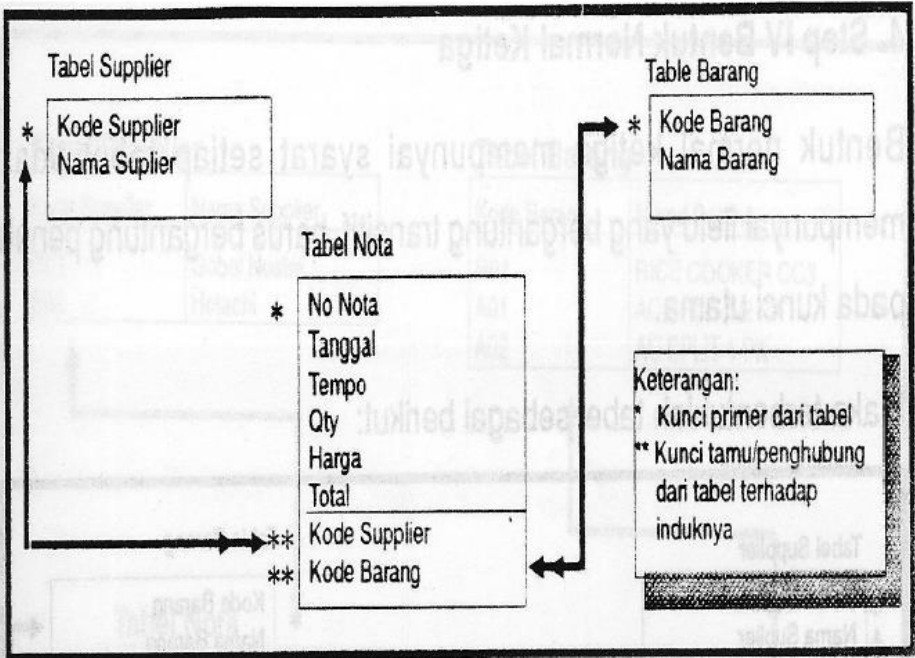
No Fac	Kode Supp	Nama Supp	Kode Brg	Nama Barang	TGL	Jatuh Tempo	Qty	Harga	Jumlah	Total

3. Step III bentuk normal kedua

Pembentukan bentuk normal kedua dengan mencari kunci kunci field yang dapat dipakai sebagai patokan dalam pencarian dan sifatnya unik. Melihat kondisi dari permasalahan factur di atas dapat diambil kunci kandidat yaitu:

- ✓ No factur (no fac)
- ✓ Kode supplier
- ✓ Kode barang

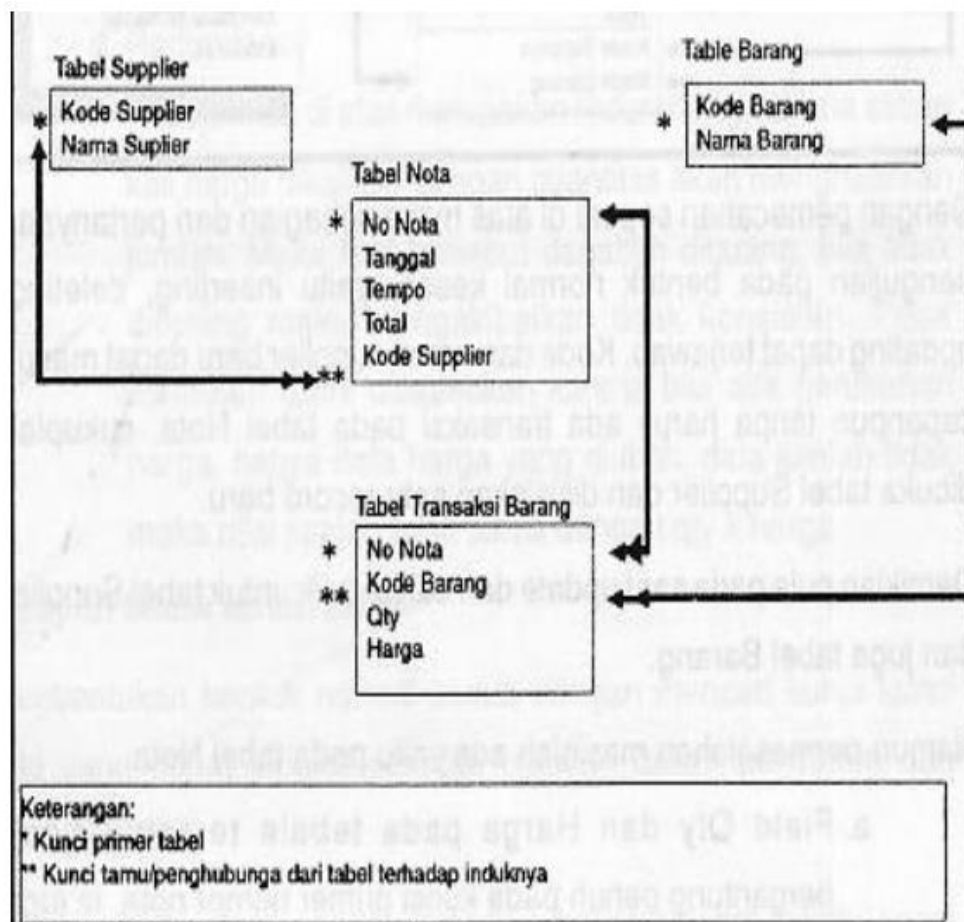
Bentuklah tiga tabel dengan kunci tersebut, lihatlah ketergantungan fungsional field field lain terhadap field kunci, maka didapatkan tabel sebagai berikut:



- Field Qty dan Harga pada tabel tersebut tidak bergantung penuh pada kunci primer nomor nota, ia juga bergantung fungsi pada kode barang. Hal ini disebut sebagai ketergantungan yang transitif dan haruslah dipisahkan dalam dua tabel.
- Masih terdapat redundancy yaitu setiap kali satu nota yang terdiri dari 5 macam barang yang dibeli maka 5 kali pula dituliskan no nota, tanggal nota, tempo, dan total. Ini harus pula dipisahkan bila terjadi pengandaan tulisan berulang ulang.

4. Step IV Bentuk Normal Ketiga

Bentuk normal ketiga mempunyai syarat setiap tabel tidak mempunyai field yang bergantung transitif, harus bergantung penuh pada kunci utama. Maka terbentuklah tabel sebagai berikut:



Step V Pengujian dengan data contoh

- Pengujian disini untuk memastikan kebenaran isi tabel dan hubungan antara tabel tersebut. Ujilah bahwa setiap tabel haruslah punya hubungan dengan tabel yang lainnya. Bila tidak ada penghubung antar tabel maka dapat dikatakan perancangan untuk membuat satu database adalah gagal.

LATIHAN SOAL

- 1. Jelaskan yang dimaksud dengan normalisasi database
- 2. Sebutkan dan jelaskan tujuan normalisasi database
- 3. Sebutkan setiap aturan normalisasi
- 4. Sebutkan setiap tahapan dalam normalisasi
- 5. Sebutkan jenis-jenis atribut tabel
- 6. Normalisasikan tabel dibawah ini:

Nama_Mahasiswa	NIM	Tgl_Lahir	KD-MKul	Kuliah	SKS	Nilai	Bobot
Tim	32980	17/05/84	TE100	Fisika	3	A	4
Tim	32980	17/05/84	TEL101	Matematika	3	B	3
Joni	32895	21/11/84	SAE100	Bahasa	2	A	4
Joanna	32784	20/09/84	SAE100	Bahasa	2	B	3
Joanna	32784	20/09/84	TEL101	Matematika	3	B	3

BAB 7

FRAGMENTASI DATA

Merupakan sebuah proses pembagian atau pemetaan database dimana database dipecah-pecah berdasarkan kolom dan baris yang kemudian disimpan didalam site atau unit komputer yang berbeda dalam suatu jaringan data, sehingga memungkinkan untuk pengambilan keputusan terhadap data yang telah terbagi

▪ **Fragmentasi**

Secara umum proses fragmentasi ini dibagi kedalam tiga jenis, yaitu:

1. Fragmentasi horizontal

Fragmentasi horizontal ini terjadi ketika suatu data yang berada pada suatu table dipisah ke dalam beberapa table yang lain. Tabel-tabel yang lain tersebut mempunyai struktur yang sama persis dengan table yang asli. Fragmentasi horizontal dibagi menjadi dua yaitu:

a. Primary horizontal fragmentation

Merupakan horizontal fragmentation dimana suatu data dalam table secara horizontal dipecah berdasarkan pada suatu nilai dalam satu atau lebih kolom tertentu

b. Derived horizontal fragmentation

Merupakan suatu teknik horizontal fragmentation dimana data yang berada pada suatu table secara horizontal dipecah ke dalam beberapa table yang lain, dengan kriteria data yang dimasukan tergantung apakah data tersebut berelasi yang dengan data yang berada di dalam table yang dipecah secara primary horizontal fragmentation atau tidak

2. Fragmentasi vertical

Merupakan suatu teknik fragmentasi data dimana data dipecah kedalam beberapa table yang merupakan hasil pemisahan beberapa kolom yang berada pada table utama. Tabel yang difragmentasi secara vertical harus dapat dikembalikan lagi seperti semula (reconstruct)

3. Fragmentasi hybrid

Merupakan gabungan antara fragmentasi horizontal dan fragmentasi vertikal

▪ **Fungsi Fragmentasi :**

1. Penggunaan
2. Efisiensi
3. Parallelsisme
4. Keamanan

Kerugian fragmentasi yaitu :

Kinerja yang dapat turun karena data tersebar dan butuh proses untuk penggabungan kembali Integritas yang dapat terganggu dikarenakan kegagalan pada salah satu site database server.

CONTOH KASUS JENIS-JENIS FRAGMENTASI

Ujian (NIM,Nama_Mhs,Kode_MK,Mt_Kuliah,Nil_Akhir,Grade)

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade
123	Fathi	101	Sistem Basis Data	78	B
124	Farah	102	Peranc. Sistem	60	C
125	Sarah	101	Sistem Basis Data	40	D
126	Salsabila	101	Sistem Basis Data	90	A
127	Azizah	103	Visual Basic	70	B
128	Farhan	103	Visual Basic	40	D
129	Faiz	102	Peranc. Sistem	80	A

Fragmentasi Horisontal terbagi menjadi 3 fragment yang berbeda berdasarkan Mt_Kuliah 1.
Relasi Mt_Kuliah=“Sistem Basis Data” Mt_Kuliah=“Sistem Basis Data” (Ujian)

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade
123	Fathi	101	Sistem Basis Data	78	B
125	Sarah	101	Sistem Basis Data	40	D
126	Salsabila	101	Sistem Basis Data	90	A

2. Relasi Mt_Kuliah=“Peranc. Sistem”
Mt_Kuliah=“Peranc. Sistem” (Ujian)

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade
124	Farah	102	Peranc. Sistem	60	C
129	Faiz	102	Peranc. Sistem	80	A

3. Relasi Mt_Kuliah=“Visual Basic”
Mt_Kuliah=“Visual Basic” (Ujian)

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade
127	Azizah	103	Visual Basic	70	B
128	Farhan	103	Visual Basic	40	D

Fragment di atas memenuhi kondisi jika Nama_Mhs dan Mt_Kuliah adalah hal-hal yang memenuhi syarat

Fragmentasi vertical:berdasarkan dekomposisi-nya dengan menambahkan tupel id

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade	Tuple_ID
123	Fathi	101	Sistem Basis Data	78	B	1
124	Farah	102	Peranc. Sistem	60	C	2
125	Sarah	101	Sistem Basis Data	40	D	3
126	Salsabila	101	Sistem Basis Data	90	A	4
127	Azizah	103	Visual Basic	70	B	5
128	Farhan	103	Visual Basic	40	D	6
129	Faiz	102	Peranc. Sistem	80	A	7

Relasi 1 = NIM, Nama_Mhs, Mt,Kuliah, Nil_Akhir, Grade, Tuple_ID

p NIM,Nama_Mhs,Mt,Kuliah,Nil_Akhir,Grade,Tuple_ID (Ujian)

NIM	Nama_Mhs	Mt_Kuliah	Nil_Akhir	Grade	Tuple_ID
123	Fathi	Sistem Basis Data	78	B	1
124	Farah	Peranc. Sistem	60	C	2
125	Sarah	Sistem Basis Data	40	D	3
126	Salsabila	Sistem Basis Data	90	A	4
127	Azizah	Visual Basic	70	B	5
128	Farhan	Visual Basic	40	D	6
129	Faiz	Peranc. Sistem	80	A	7

Relasi 2 = NIM,Kode_MK,Nil_Akhir,Grade,Tuple_ID

p NIM,Kode_MK,Nil_Akhir,Grade,Tuple_ID (Ujian)

NIM	Kode_MK	Nil_Akhir	Grade	Tuple_ID
123	101	78	B	1
124	102	60	C	2
125	101	40	D	3
126	101	90	A	4
127	103	70	B	5
128	103	40	D	6
129	102	80	A	7

Terdapat relasi berdasarkan Mata Kuliah yang sama Relasi 1a.

**p NIM,Nama_Mhs,Mt_Kuliah,Nil_Akhir,Grade,Tuple_ID(s Mt_Kuliah=“Sistem Basis Data”
(Ujian))**

NIM	Nama_Mhs	Mt_Kuliah	Nil_Akhir	Grade	Tuple_ID
123	Fathi	Sistem Basis Data	78	B	1
125	Sarah	Sistem Basis Data	40	D	3
126	Salsabila	Sistem Basis Data	90	A	4

Relasi 1b.

p NIM, Nama_Mhs, Mt_Kuliah, Nil_Akhir, Grade, Tuple_ID(s Mt_Kuliah=“Peranc.Sistem” (Ujian))

NIM	Nama_Mhs	Mt_Kuliah	Nil_Akhir	Grade	Tuple_ID
124	Farah	Peranc. Sistem	60	C	2
129	Faiz	Peranc. Sistem	80	A	7

Relasi 1c

p NIM,Nama_Mhs,Mt_Kuliah,Nil_Akhir,Grade,Tuple_ID(s Mt_Kuliah=“Visual Basic” (Ujian))

NIM	Nama_Mhs	Kode_MK	Mt_Kuliah	Nil_Akhir	Grade
127	Azizah	103	Visual Basic	70	B
128	Farhan	103	Visual Basic	40	D

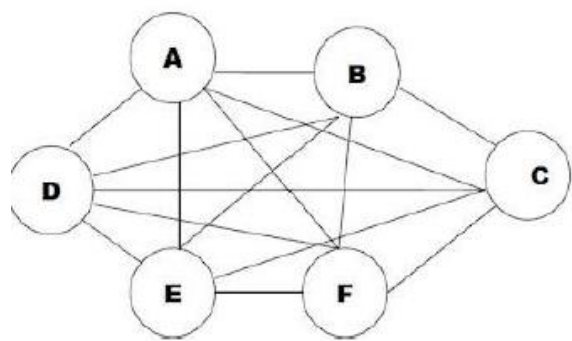
Bagaimana bentuk database untuk data yang telah mengalami proses fragmentasi?

Fragmentasi data merupakan langkah yang diambil untuk menyebarkan data dalam database terdistribusi . Selanjutnya akan dibahas apa yang dimaksud Database terdistribusi.

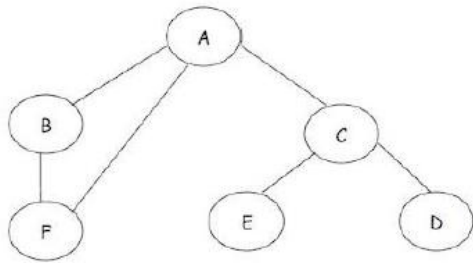
- **Database Terdistribusi**
Yaitu kumpulan data yang digunakan bersama yang saling terhubung secara logik tetapi tersebar secara fisik pada suatu jaringan komputer.Karakteristik Database terdistribusi, yaitu :
 1. Kumpulan data yang digunakan bersama secara logik tersebar pada sejumlah komputer yang berbeda
 2. Komputer yang dihubungkan menggunakan jaringan komunikasi
 3. Data pada masing-masing situs dapat menangani aplikasiaplikasi lokal secara otonom
 4. Data pada masing situs dibawah kendali satu DBMS
 5. Masing-masing DBMS berpartisipasi dalam sedikitnya satu aplikasi global

BENTUK-BENTUK TOPOLOGI DISTRIBUSI DATA :

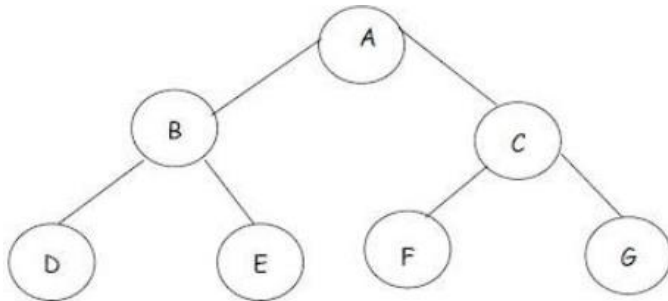
- a. Fully Connected network



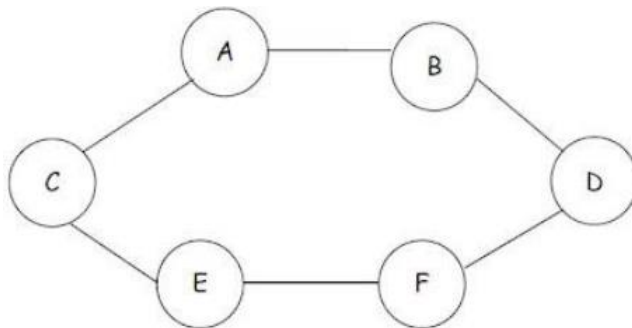
- b. Partially conneted network



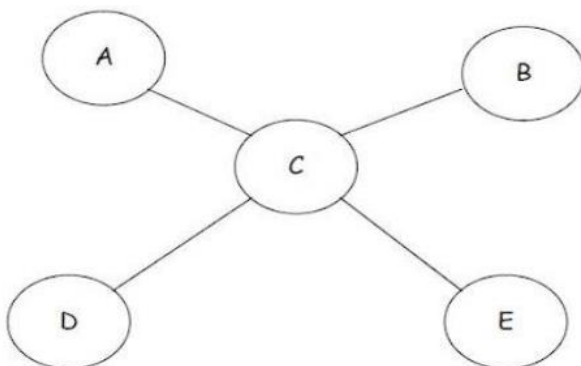
c. Tree Strutured Network



d. Ring network



e. Star network



Keuntungan Dan Kerugian Database Terdistribusi

KEUNTUNGAN :

1. Secara alami mengikuti struktur organisasi
2. Adanya otonomi lokal
3. Sifatnya dapat dipakai secara bersama
4. Peningkatan ketersediaan
5. Peningkatan kehandalan
6. Peningkatan kinerja
7. Ekonomis

8. Pertumbuhan yang modular

KERUGIAN :

1. Harga software mahal (Biaya)
2. Kompleksitas
3. Kelemahan dalam keamanan
4. Sulitnya menjaga keutuhan data
5. Kurangnya standar
6. Kurangnya pengalaman
7. Perancangan basisdata lebih kompleks

LATIHAN SOAL

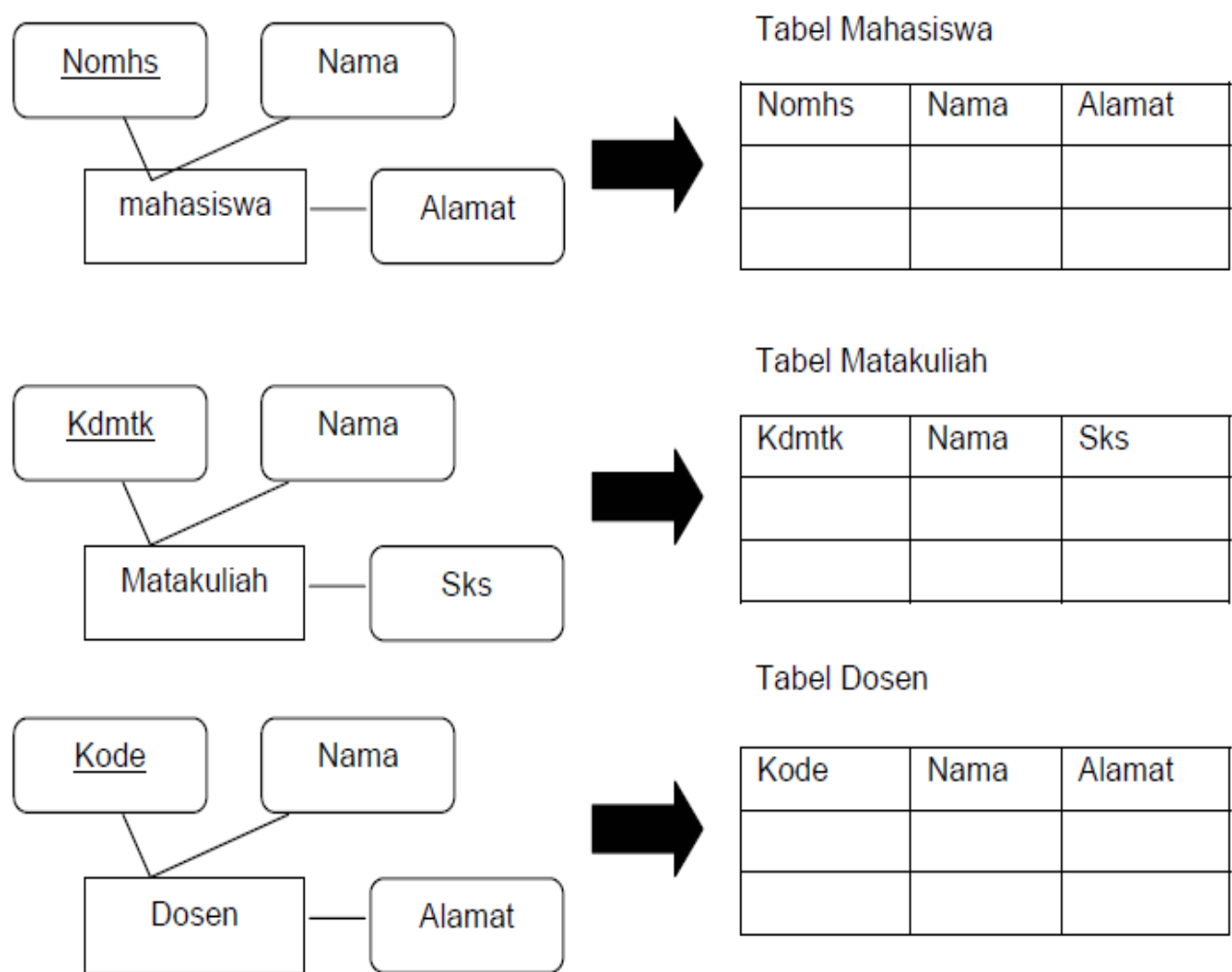
1. Jelaskan yang dimaksud dengan fragmentasi
2. Jelaskan proses-proses fragmentasi
3. Jelaskan setiap fungsi dari fragmentasi
4. Sebutkan kelebihan dan kekurangan fragmentasi
5. Jelaskan yang dimaksud dengan database terdistribusi
6. Sebutkan karakteristik database terdistribusi
7. Gambarkan topologi distribusi data

BAB 8

IMPLEMENTASI BASIS DATA

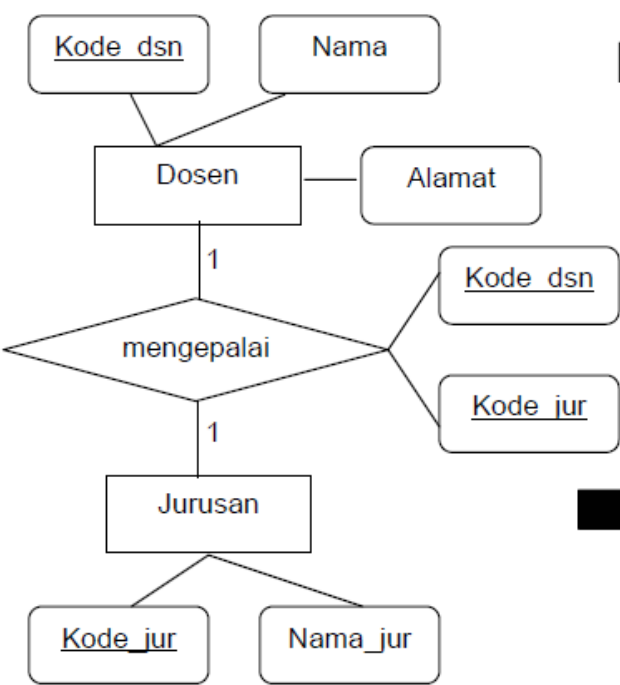
- **Implementasi DB**
 - Tahap implementasi basis data merupakan upaya untuk membangun basis data fisik yang ditempatkan dalam media penyimpanan (disk) dengan bantuan DBMS
 - Tahap ini diawali dengan melakukan transformasi dari model data yang telah selesai dibuat struktur basis data sesuai DBMS yang dipilih.
 - Secara umum, sebuah ERD akan diwujudkan menjadi 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 himpuna relasi akan dinyatakan sebagai field-field dari tabel yang sesuai
 - Performansi basis data ditentukan oleh :
 - Kualitas dan bentuk perancangan basis data
 - Kualitas mesin / komputer
 - Platform yang dipilih
 - Sistem operasi
 - DBMS yang digunakan
- **Pengkodean / Abstrasi data**
 - Data yang dilihat oleh pemkasi 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 dinyatakan dalam bentuk lain.
 - Alasan untuk membuat suatu pengkodean adalah untuk efisiensi ruang penyimpanan
 - 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 yangurut
 - 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” → dikodekan ‘L’; “Perempuan” → 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
- **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

himpunan entitas



Tabel Dosen

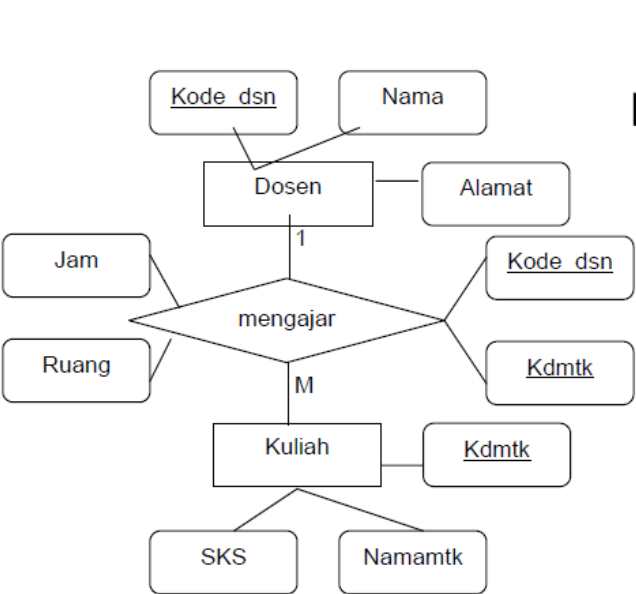
Kode_dsn	Nama	Alamat

Tabel Jurusan

Kode_Jur	Nama_jur	Kode_dsn

Field yang
ditambahkan dari
relasi Mengepalai

- 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)



Tabel Dosen

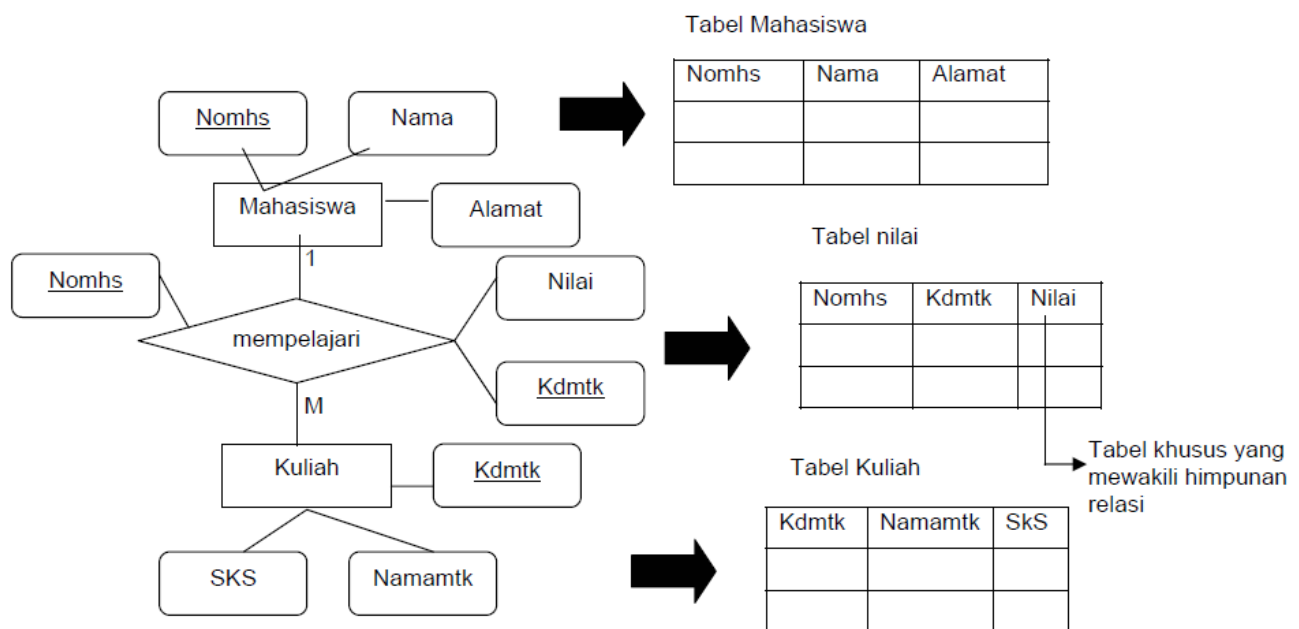
Kode_dsn	Nama	Alamat

Tabel Kuliah

Kdmtk	Namamtk	SKS	Kode_dsn	jam	Ruang

Field yang ditambahkan
dari relasi mengajar

- 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



▪ 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 Alphanumeric**, 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
- **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 Satu
 - Nilai-nilai field yang menjadi pembentuk IS tidak harus bersifat unik

❖ **Struktur penyimpanan**

- Ada 4 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 → dibutuhkan alokasi ruang penyimpanan

c. Sekuensial berindeks

- Menempatkan data engan urutan tertentu berdasar nilai 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

LATIHAN SOAL

1. Jelaskan tahap-tahap implementasi database
2. Sebutkan dan jelaskan faktor yang mempengaruhi performa sebuah database
3. Jelaskan yang dimaksud dengan *user define coding* dan *system coding*
4. Jelaskan perbedaan antara sekuen, mnemonic dan blok dalam bentuk pengkodean
5. Jelaskan tahap-tahap transformasi model data ke basis data fisik
6. Sebutkan dan jelaskan struktur penyimpanan dasar yang diterapkan pada suatu tabel

BAB 9

PENGEMBANGAN SISTEM BASIS DATA

❖ 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.

- **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 penyimpanan dalam basis data

i. Meningkatkan produktivitas personal (personal productivity)

- SBD diharapkan mampu meningkatkan produktivitas kerja setiap personal, yang mampu memenuhi kebutuhan data sederhana hingga bentuk laporan yang lebih rumit

▪ **Proses Pengembangan basis data**

- 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 basis data.

Ukuran-ukuran tersebut misalnya : struktur file, model basis data, skema / relasi, pemanggilan informasi, serta perawatan data dan integritas basis data. 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 basis data. Model basis data akan menentukan proses yang diperlukan untuk pembentukan file, perawatan file, dan pemanggilan informasi.

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

d. Pemrograman

Implementasi akhir setelah proses perancangan basis data selesai adalah dengan melakukan pemrograman

- **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
- **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

- **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

LATIHAN SOAL

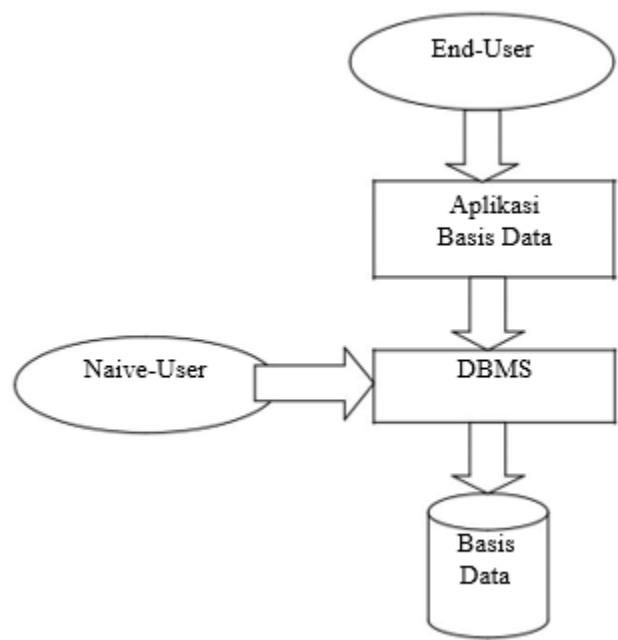
- 1.** Jelaskan setiap tujuan pengembangan sistem basis data
- 2.** Sebutkan proses pengembangan sistem basis data
- 3.** Jelaskan langkah-langkah pengembangan sistem basis data
- 4.** Jelaskan langkah-langkah mendisain basis data untuk Sistem Informasi Manajemen
- 5.** Sebutkan alat bantu dan metode yang digunakan untuk pengembangan sistem basis data

BAB 10

APLIKASI BASIS DATA

Hal yang perlu dipertimbangkan ketika menerapkan aplikasi basis data

1. Hubungan antara DBMS dan aplikasi basis data

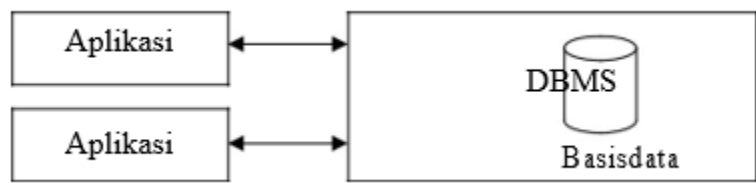


- Aplikasi basis data disusun untuk menjembatani perbedaan pandangan antara *end-user* dan *naïve user*, yang dibuat khusus untuk dapat digunakan oleh para pemakai akhir (*end-user*)
- Aplikasi ini berisi sejumlah operasi (menu) yang sesuai dengan aktifitas nyata yang dilakukan oleh end-user. Selanjutnya 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 :

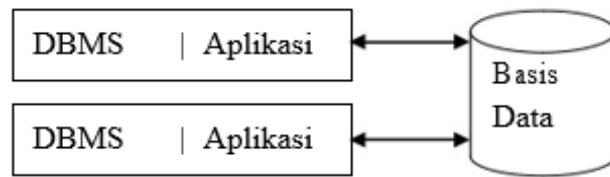
a. DBMS terpisah dengan aplikasi



- f* Aplikasi tidak berinteraksi langsung dengan basis data, tapi melalui DBMS sebagai perantara. Bahkan DBMS bisa melakukan aktifitas sendiri yang bisa ditangkap oleh aplikasi
- f* Contoh DBMS : MS SQL Server, Oracle, CA-OpenIngres, Sybase, Informix, IBM DB2

f 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 subordinate), 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 :
 - o Keunggulan teknologi
 - o Faktor biaya
 - o Sesuai dengan kebutuhan pemakai

f Jenis-jenis arsitektur sistem :

f **Sistem tunggal (Standalone)**

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

f Hanya bisa dipakai oleh satu pemakai pada saat yang bersamaan

f **Sistem Terpusat (Centralized system)**

f Terdiri dari sebuah server dan sejumlah terminal

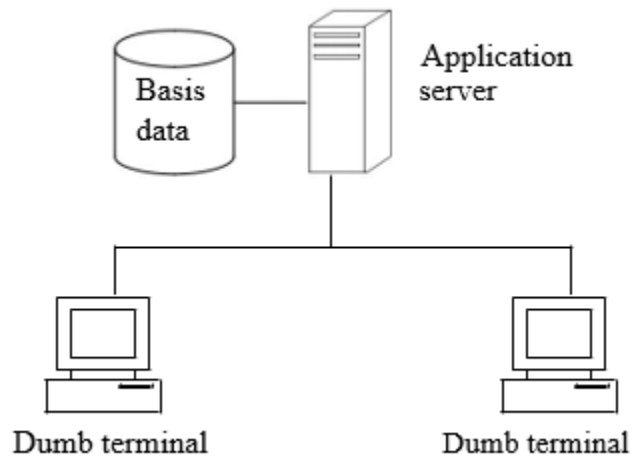
f Yang terpusat adalah basis data, DBMS, dan aplikasi basis data atau basis data saja

f Ada dua macam :

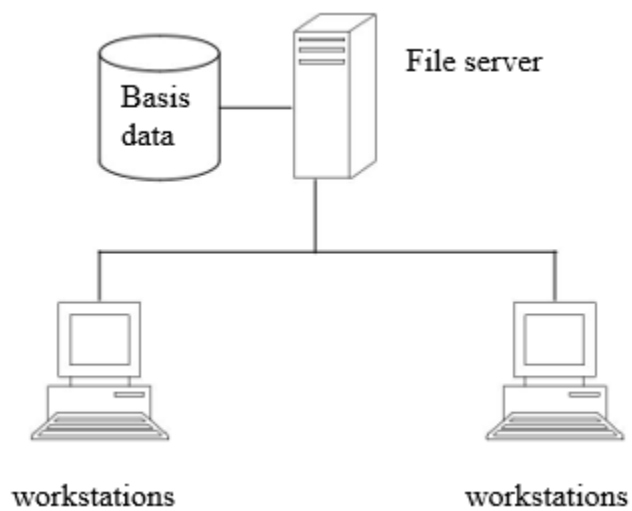
f Aplikasi dan basis data terpusat; diakses oleh dumb terminal

f Basis data terpusat; aplikasi ada pada terminal

Sentralisasi pada aplikasi dan basis data



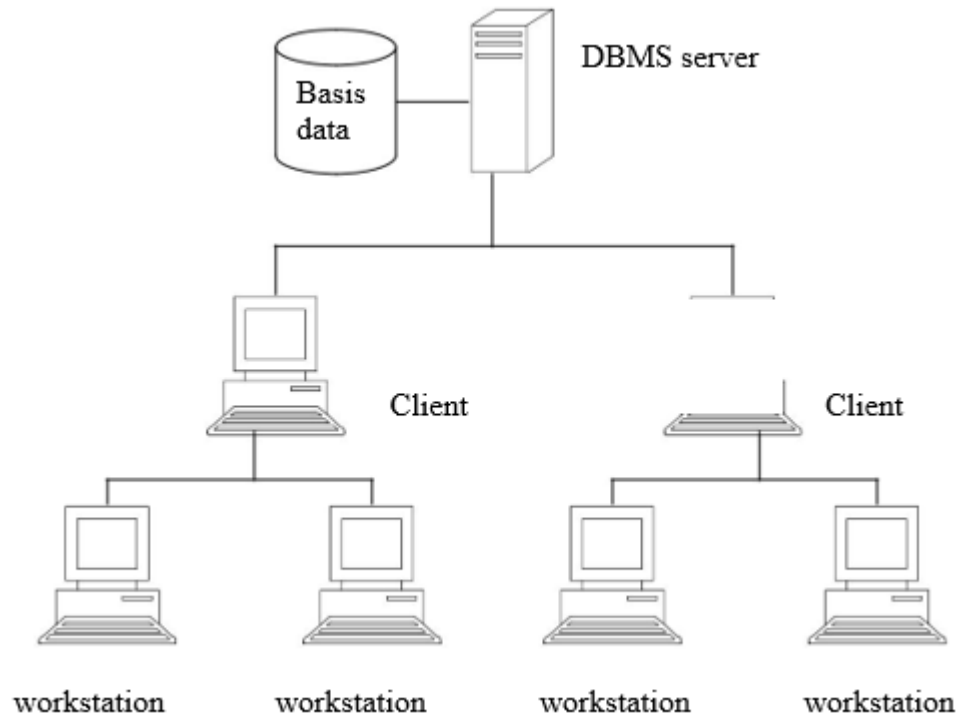
Sentralisasi pada basis data



f Sistem Client-server

- f* Ditujukan untuk mengatasi kelemahan yang terdapat pada sistem terpusat
- f* Terdiri dari 2 komponen utama yaitu client dan server. Client berisi aplikasi basis data; server berisi DBMS dan basis data
- f* Ada dua macam :
 - f* Arsitektur 2 lapis (2-tier)
 - f* Arsitektur 3 lapis (3-tier)

Arsitektur 3-tier



3. Pemilihan perangkat lunak pembangun aplikasi basis data

f Pertimbangan dalam menentukan perangkat lunak pembangun aplikasi basisdata a.
Kecocokan antara DBMS dan *development tools*

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

f 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

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

c. Independensi *development tools* dan DBMS

- f* Idealnya hanya ada satu macam DBMS yang dipilih untuk mengelola berbagai basis data.
- f* Sebagai kompromi terhadap banyaknya DBMS yang digunakan, maka harus dipilih development tools yang bisa cocok untuk semua DBMS

d. Kemudahan pengembangan dan migrasi aplikasi

- f* 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

- f* Performansi / kecepatan operasi ke basis data ditentukan oleh :
 - o DBMS yang digunakan
 - o Arsitektur perangkat keras yang menjadi platform
 - o Jumlah pemakai yang terlibat
 - o Volume data
 - o Tingkat kompleksitas operasi basis data

- o Cara penulisan aplikasi
- f Hal-hal yang perlu dipertimbangkan pada saat melaksanakan pemrograman :
 - a. Sedapat mungkin memanfaatkan indeks primer / sekunder dalam setiap proses query ke basis data, contoh :
 - f `Select From pegawai where idpegawai = vidpegawai`
 - f `Update pegawai set Where idpegawai = vidpegawai`
 - f `Delete from pegawai where idpegawai = vidpegawai`
 - o Menghindari pemakaian fungsi atau perhitungan pada perintah query, terlebih lagi untuk kriteria query. Contoh :
 - f `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)*

f `Select ... from kuliah where left(kdmtk,3)='TFD'`

Diganti menjadi

Select ... from kuliah where kdmtk like 'TFD%'

f Kenapa beda ? Pada arsitektur client-server :

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

f 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 :

f Insert into nilai (nomhs, kdmtk) values (vnomhs,
vkdtmk) 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.

f 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
Endi

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

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

o Pastikan bahwa nilai-nilai data adalah benar sejak dimasukkan pertama kali o Membuat program untuk mengecek keabsahan data pada saat dimasukkan

ke komputer

f Penolakan / pembatalan aksi (cancelation)

f Pengisian nilai kosong pada field tertentu (nullify)

f Penjalaran perubahan (cascade)

f 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 kasuistik, 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

LATIHAN SOAL

1. Jelaskan yang dimaksud dengan aplikasi basis data
2. Jelaskan tujuan dan manfaat aplikasi basis data
3. Sebutkan dan jelaskan model hubungan antara DBMS dan aplikasi basis data
4. Sebutkan dan jelaskan jenis jenis arsitektur sistem
5. Sebutkan dan jelaskan hal yang perlu diperhatikan ketika menerapkan aplikasi database