



PENGANTAR TEKNOLOGI SISTEM INFORMASI MANAJEMEN 2

PERTEMUAN 2

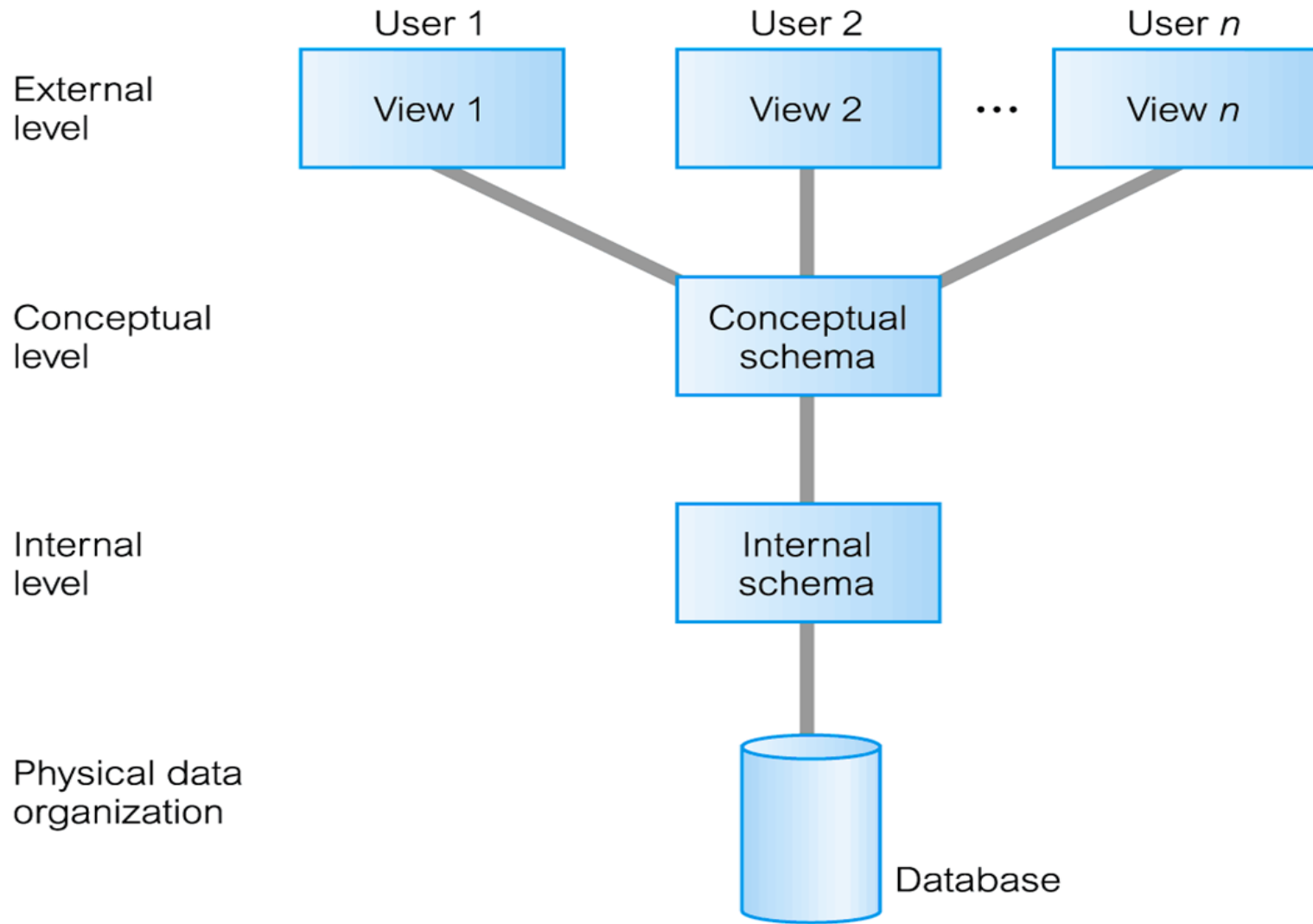
Lingkungan Basis Data

SAP PERTEMUAN 2

- Arsitektur Basis Data
- Data Independence
- Konsep DBMS, Komponen DBMS, Fungsi DBMS, dan Bahasa yang digunakan di dalam DBMS
- Model data : berbasis objek, berbasis record, konseptual dan fisik (overview model data berbasis record : model data relasional, jaringan, hirarki)
- Data Dictionary
- Arsitektur DBMS multiuser : file server, teleprocessing, client - server



ARSITEKTUR BASIS DATA



ARSITEKTUR BASIS DATA LANJ.

- Terdiri dari 3 lapis :

1. Lapis Internal

- berkenaan dengan penyimpanan secara fisik. -
- level terendah untuk merepresentasikan basis data.
- Record disimpan dalam media penyimpanan dalam format byte.

Ex:

FILE_PEGAWAI	LENGTH = 22
PREFIX	TYPE = BYTE (6), OFFSET = 0
EMP#	TYPE = BYTE (6), OFFSET = 6, INDEX = EMPX
DEPT#	TYPE = BYTE (4), OFFSET = 12
PAY	TYPE = FULLWORD, OFFSET = 16

ARSITEKTUR BASIS DATA LANJ.

2. Lapis Konseptual / Logical Level

- menghubungkan antara level internal & external level.
- menjelaskan simpanan data dalam database, dan relasi antar data.

Ex : Entity, relationship.

PEGAWAI

NOMOR_PEGAWAI CHARACTER 6

NOMOR_DEPT CHARACTER 4

GAJI NUMERIC 6



ARSITEKTUR BASIS DATA LANJ.

3. Lapis External

- lapis yang berkenaan dengan apa yang kelihatan bagi para pemakai akhir (*end users*).

- users ->

a. programmer (ex: C, Cobol, PL/SQL)

b. end user (ex : bahasa query, fasilitas yang sudah tersedia.

c. DBA. Ex: view dari mahasiswa, view dari mata kuliah

Cobol

01 PEG_REC.

02 PEG_NO PIC X(6).

02 DEPT_NO PIC X(4).

02 GAJI PIC 9(6).



DATA INDEPENDENCE

- Tujuan utama dari 3 tingkat arsitektur : memelihara kemandirian data (*data independence*).

Terdapat 2 lapis data independence :

- a. Physical Data Independence
internal schema dapat diubah oleh DBA tanpa mengganggu conceptual schema.
- b. Logical Data Independence
Bahwa conceptual schema dapat diubah oleh DBA tanpa mengganggu external schema.



PRINSIP DATA INDEPENDENCE

- a. DBA dapat mengubah isi, lokasi, perwujudan dalam organisasi basis data tanpa mengganggu program-program aplikasi yang sudah ada.
- b. Pabrik/agen peralatan/software pengolahan data dapat memperkenalkan produk-produk baru tanpa mengganggu program-program aplikasi yang sudah ada.
- c. Untuk memindahkan perkembangan program-program aplikasi.
- d. Memberikan fasilitas pengontrolan terpusat oleh DBA demi keamanan dan integritas data dengan memperhatikan perubahan-perubahan kebutuhan penggunaan.



KONSEP DBMS

TERMINOLOGI :

- **Enterprise**
- **Entity**
- **Attribute**
- **Data Value (nilai/isi data)**
- **Record/Tuple**
- **File**



ENTERPRISE

Suatu bentuk organisasi,
seperti: bank, universitas, rumah sakit, dan
pabrik.

Entity :

Sekumpulan objek yang mempunyai
karakteristik yang sama dan dapat dibedakan
dari lainnya yang dapat diwujudkan dalam basis
data.

Contoh : Nasabah, Mahasiswa, Pegawai, Mobil

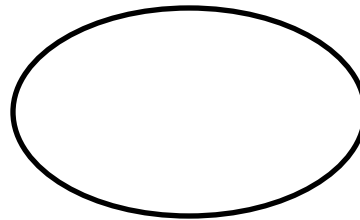
Simbol ->



ATTRIBUTE

- Setiap *entity* mempunyai *attribute*.
- *Attribute* → sebutan untuk mewakili suatu *entity*.
- Contoh:
 - Seorang siswa dapat dilihat dari *attribute*-nya, misalnya nama, nomor siswa, alamat, nama orang tua, hobby.
- *Attribute* → data elemen, data *field*, data *item*.

Simbol :



DATA VALUE (NILAI/ISI DATA)

- *Data value* → data aktual atau informasi yang disimpan pada tiap data elemen atau *attribute*.
- Contoh:
 - *Attribute* Nama_Mhs menunjukkan tempat dimana informasi nama mahasiswa disimpan.
 - *Data value* adalah Adi, Arif, Budi, merupakan isi data dari *attribute* Nama_Mhs tersebut.



ATTRIBUTE

NIM	Nama_Mhs	Tgl_Lahir	Alamat_Mhs	IPK
672010001	Adi	01-Jan-95	Solo	3,01
672010002	Arif	02-Feb-95	Salatiga	3,25
672010003	Budi	05-Mei-95	Semarang	2,9

Data Value



RECORD / TUPLE

- ▶ *Record/Tuple* → Kumpulan elemen elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap.

NIM	Nama_Mhs	Tgl_Lahir	Alamat_Mhs	IPK	
672010001	Adi	01-Jan-95	Solo	3,01	→ Record 1
672010002	Arif	02-Feb-95	Salatiga	3,25	→ Record 2
672010003	Budi	05-Mei-95	Semarang	2,9	→ Record 3



FILE

- *File* → Kumpulan *record-record* sejenis yang mempunyai panjang elemen yang sama, *attribute* yang sama, namun berbeda beda *data valuenya*.
- Dalam satu *file* terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan *entity* yang seragam.



KOMPONEN DBMS

- Perangkat keras
- Perangkat lunak
- Data

Bagi user komponen paling utama DBMS adalah data. Data bertindak sebagai suatu jembatan antara komponen mesin dan komponen manusia. Database berisi kedua-duanya : data yang operasional dan meta-data.

- Prosedur

Prosedur memuat aturan-aturan untuk mendisain dan penggunaan database. Para pemakai sistem database memerlukan dokumentasi prosedur yang berisi cara menggunakan atau menjalankan sistem itu.

- User



KOMPONEN-KOMPONEN DBMS (HOWE, 1991)

Terdiri dari:

- Interface, yang didalamnya terdapat bahasa manipulasi data (data manipulation language)
- Bahasa definisi data (data definition language) untuk skema eksternal, skema konseptual dan skema internal.
- Sistem kontrol basis data (Database Control System) yang mengakses basis data karena adanya perintah dari bahasa manipulasi data.



FUNGSI DBMS

- Mendefinisikan data dan hubungannya
- Memanipulasi data
- Security dan integritas data
- Recovery/perbaikan dan concurency data
- Data dictionary
- Unjuk kerja / performance



PAKET BAHASA

- basis data biasanya terdapat bahasa-bahasa tertentu yang disebut **Data Sub language**.
- Data sub language adalah subset bahasa yang dipakai untuk operasi manajemen basis data. Dalam penggunaan biasanya dapat ditempelkan (embedded) pada bahasa tuan rumah (Cobol, PL/1, dsb).
- Contoh bahasa menggunakan komponen-komponen tersebut adalah SQL (Structured Query Language). SQL merupakan bahasa standar yang digunakan oleh kebanyakan aplikasi-aplikasi DBMS.



PAKET BAHASA

SELECT
INSERT
UPDATE
DELETE
MERGE

Data manipulation language (DML)

CREATE
ALTER
DROP
RENAME
TRUNCATE
COMMENT

Data definition language (DDL)

GRANT
REVOKE

Data control language (DCL)

COMMIT
ROLLBACK
SAVEPOINT

Transaction control



MODEL DATA

PENGERTIAN MODEL DATA :

Sekumpulan konsep-konsep untuk menerangkan data, hubungan-hubungan antara data dan batasan-batasan data yang terintegrasi di dalam suatu organisasi

JENIS-JENIS MODEL DATA

- A. Model data berbasis objek
- B. Model data berbasis record
- C. Model data fisik
- D. Model data konseptual



A. MODEL DATA BERBASIS OBJEK

Model data berbasis objek menggunakan konsep entitas, atribut dan hubungan antar entitas.

Terdiri dari :

1. Entity Relationship model
2. Semantik data model

1. *ENTITY RELATIONSHIP MODEL*

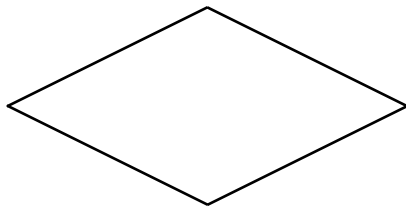
Model untuk menjelaskan hubungan antar data dalam basis data berdasarkan suatu persepsi bahwa real word terdiri dari objek-objekt dasar yang mempunyai hubungan atau relasi antara objek-objekt tersebut



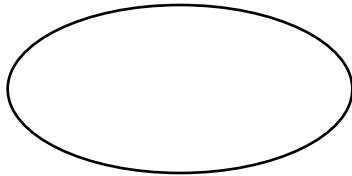
SIMBOL YANG DIGUNAKAN :



: Menunjukkan object dasar



: Menunjukkan relasi



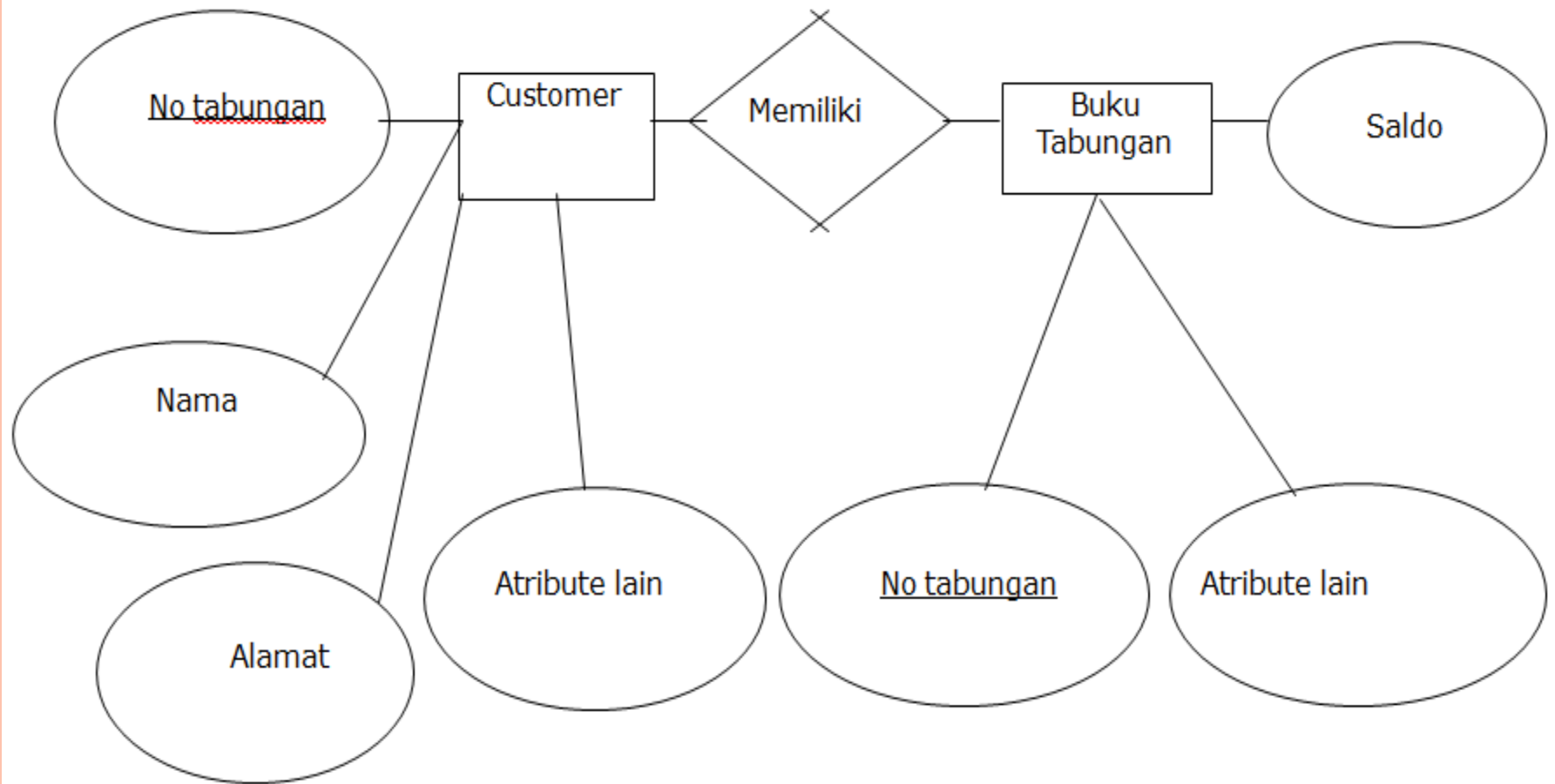
:Menunjukkan atribut dari objek dasar



: Menunjukkan adanya relasi



Contoh kasus ER-model



2. SEMANTIC MODEL

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

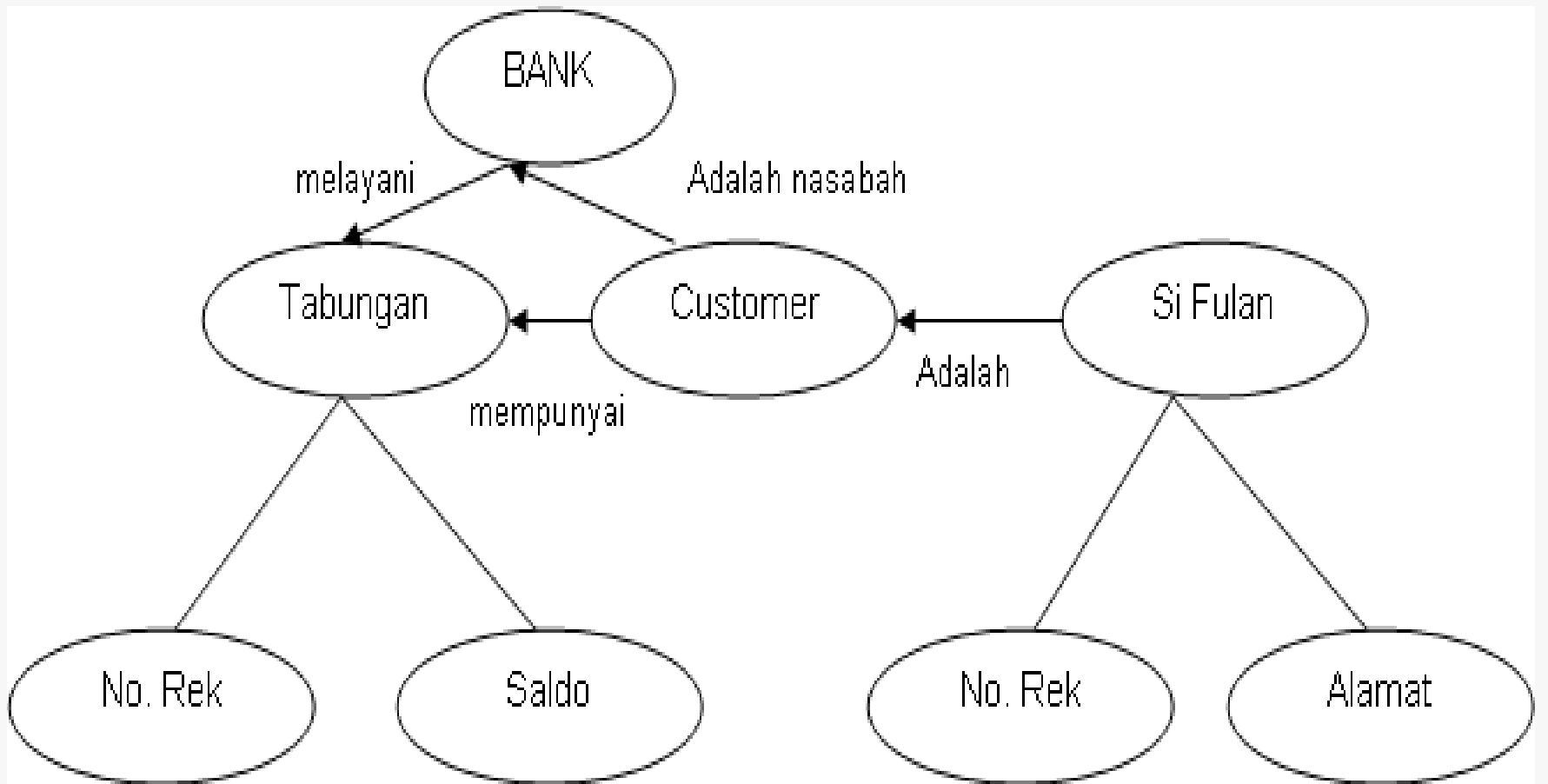
Tanda-tanda yang menggunakan dalam semantic model adalah sebagai berikut :

—————→ : Menunjukkan adanya relasi

————— : menunjukkan atribut



CONTOH SEMANTIC MODEL



MODEL DATA BERBASIS RECORD

- Model ini mendasarkan pada record untuk menjelaskan kepada user tentang hubungan logik antar data dalam basis data.

Ada 3 jenis :

- Model Data Relasional
- Model Data Hirarki
- Model Data Jaringan



1. 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



SUPPLIER

No_supl	Nama_pen	Status	KOTA
S01	PT. OGAH-RUGI	03	MEDAN
s02	PT. SANTAI-DULU	03	SURABAYA
s03	PT. MALU-MALU	02	BANDUNG
s04	CV. ASAL JADI	01	MEDAN

NO_SUPL	NO_PART	JUML
S01	P01	200
S01	P02	300
S01	P04	250
S02	P01	300
S02	P02	400
S03	P03	400
S04	P04	300

PENGIRIMAN

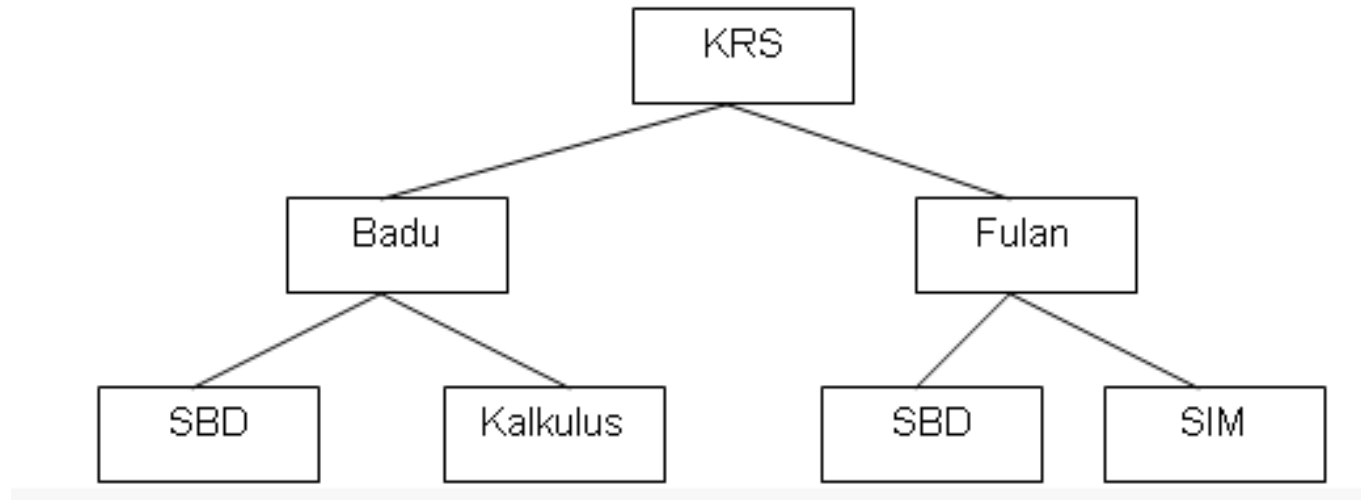
SUKU CADANG

NO_PA RT	NAMA_PART	BAHAN BAKU	BERA T	KOTA
P01	BAUT-3 cm	BESI	10	JAKARTA
P02	MUR - 3 cm	BESI	8	JAKARTA
P03	BAUT -10 cm	ALUMUNIUM	45	SURABAYA
P04	PACKING	KARET	6	MEDAN
P05	RING-MM	ALUMUNIUM	2	JAKARTA

2. HIRARCHICAL MODEL (TREE STRUCTURE)

- Menjelaskan tentang hubungan logik antar data dalam basis data dalam bentuk hubungan bertingkat (hirarki).
- Elemen penyusunnya disebut node, yang berupa rinci data, agregat data, atau record.
- memiliki hubungan cardinalitas 1:1 dan 1:M

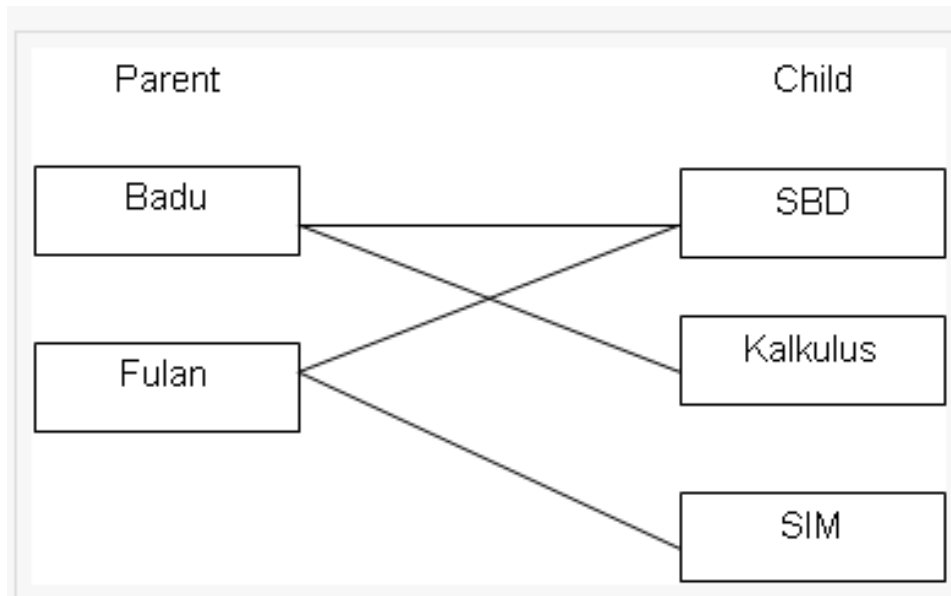
Contoh :



3. NETWORK MODEL

- Hampir sama dengan model hirarki, dan digambarkan sedemikian rupa sehingga child pasti berada pada level yang lebih rendah daripada parent.
- Sebuah child dapat mempunyai lebih dari satu parent.
- menyatakan hubungan cardinalitas 1:1, 1:M dan N:M

Contoh :



C. MODEL DATA FISIK

Digunakan untuk menguraikan data pada internal level

Beberapa model yang umum digunakan :

- Unifying model

Model ini menggabungkan memori dan transaksi database dalam satu kesatuan model.

- Frame memory

Frame Memory adalah sebuah virtual view dari tempat penyimpanan sekunder yang digunakan untuk mendukung penyimpanan record database



D. MODEL DATA KONSEPTUAL

- Model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas (entity) serta hubungan (relationship) antara entitas-entitas itu. Biasanya direpresentasikan dalam bentuk *Entity Relationship Diagram*.
- Manfaat Penggunaan CDM dalam perancangan database :
 - Memberikan gambaran yang lengkap dari struktur basis data yaitu arti, hubungan, dan batasan-batasan
 - Alat komunikasi antar pemakai basis data, designer, dan analis.
-



ARSITEKTUR DBMS MULTIUSER

Arsitektur yang digunakan akan sangat memengaruhi dalam segi keefisienan, kecepatan, dan keamanan untuk user dan sistem basis data itu sendiri.

3 jenis Arsitektur DBMS :

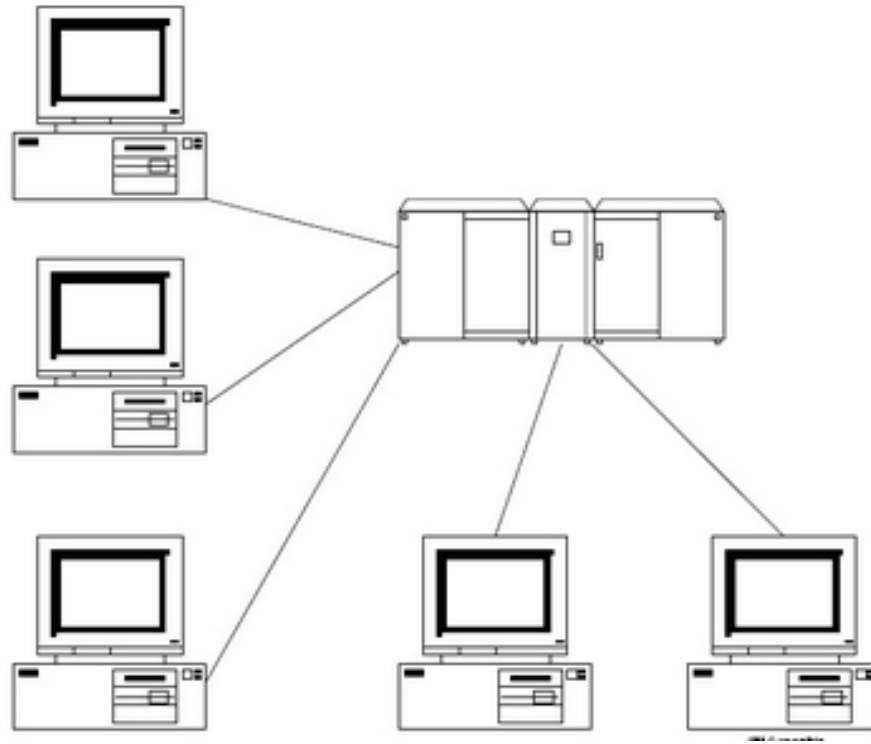
1. TELEPROCESSING,
2. FILE-SERVER,
3. CLIENT-SERVER,



1. ARSITEKTUR TELEPROSESING

- Suatu arsitektur yang memiliki metode dimana perangkat I/O berada pada lokasi yang berbeda dan jarak yang jauh dari CPU pusat Basis Data tersebut.
- Media yang digunakan untuk mengirimkan data bisa menggunakan jaringan telepon, seperti telepon kabel, jaringan gprs, maupun 3g, kemudian gelombang microwave, dan satelit. Data akan dikirim atau diterima CPU dari terminal, atau pun sebaliknya.





- CPU menggunakan sistem yang disebut time sharing system,
- CPU akan membagi waktu untuk melayani perintah tiap-tiap terminal secara bergantian,
- didukung dengan murahnya biaya penyediaan jasa telekomunikasi seperti jaringan telepon, turut mewujudkannya sebuah real time system, yang memungkinkan untuk tiap terminal dengan waktu yang telah diatur.



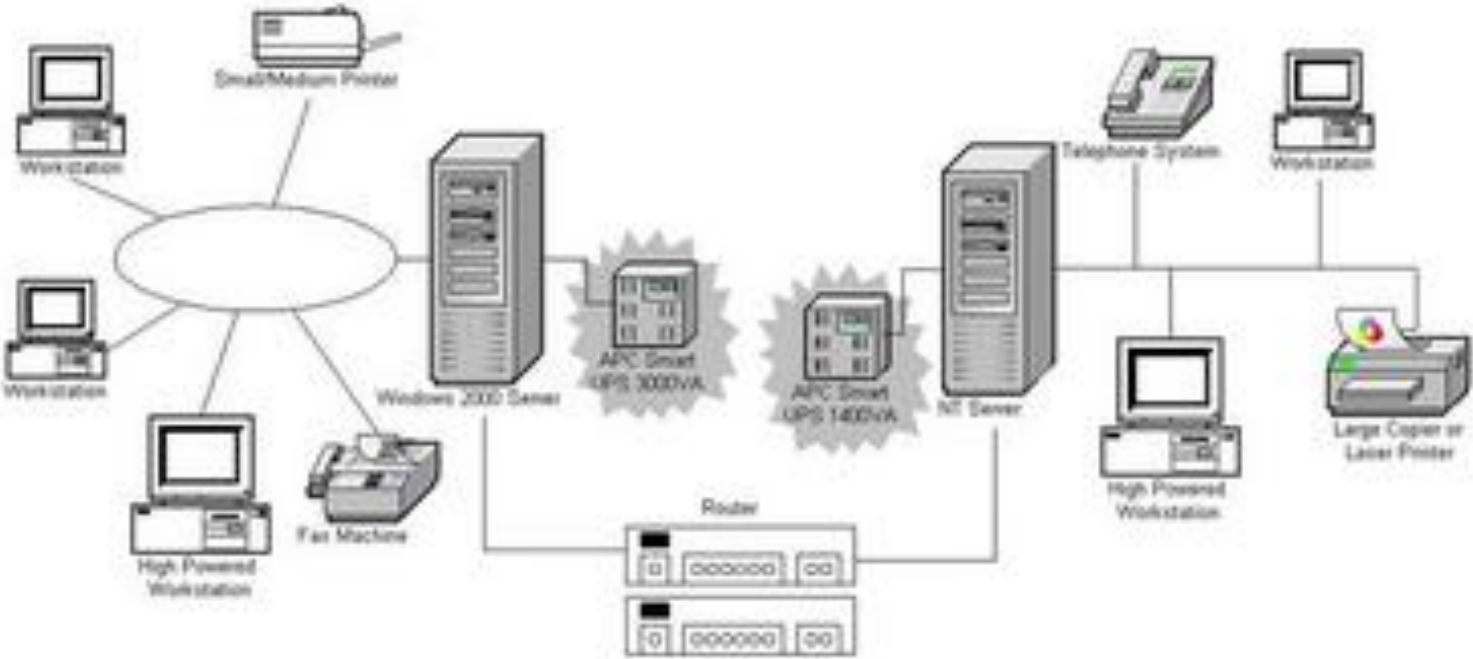
2. ARSITEKTUR FILE-SERVER

- Sebuah arsitektur yang memiliki komputer sentral yang hanya mampu untuk menjadi media penyimpanan data oleh workstation .
- File-server akan mengatur file, yang diperlukan oleh aplikasi dan DBMS
- File-server tidak akan melakukan penghitungan atau komputasi dan tidak mengatur program yang akan atau sedang dijalankan oleh workstation
- dapat kita gambarkan file-server seperti hardisk, namun hard disk ini digunakan oleh banyak workstation yang tersambung pada LAN.



Diagram 1

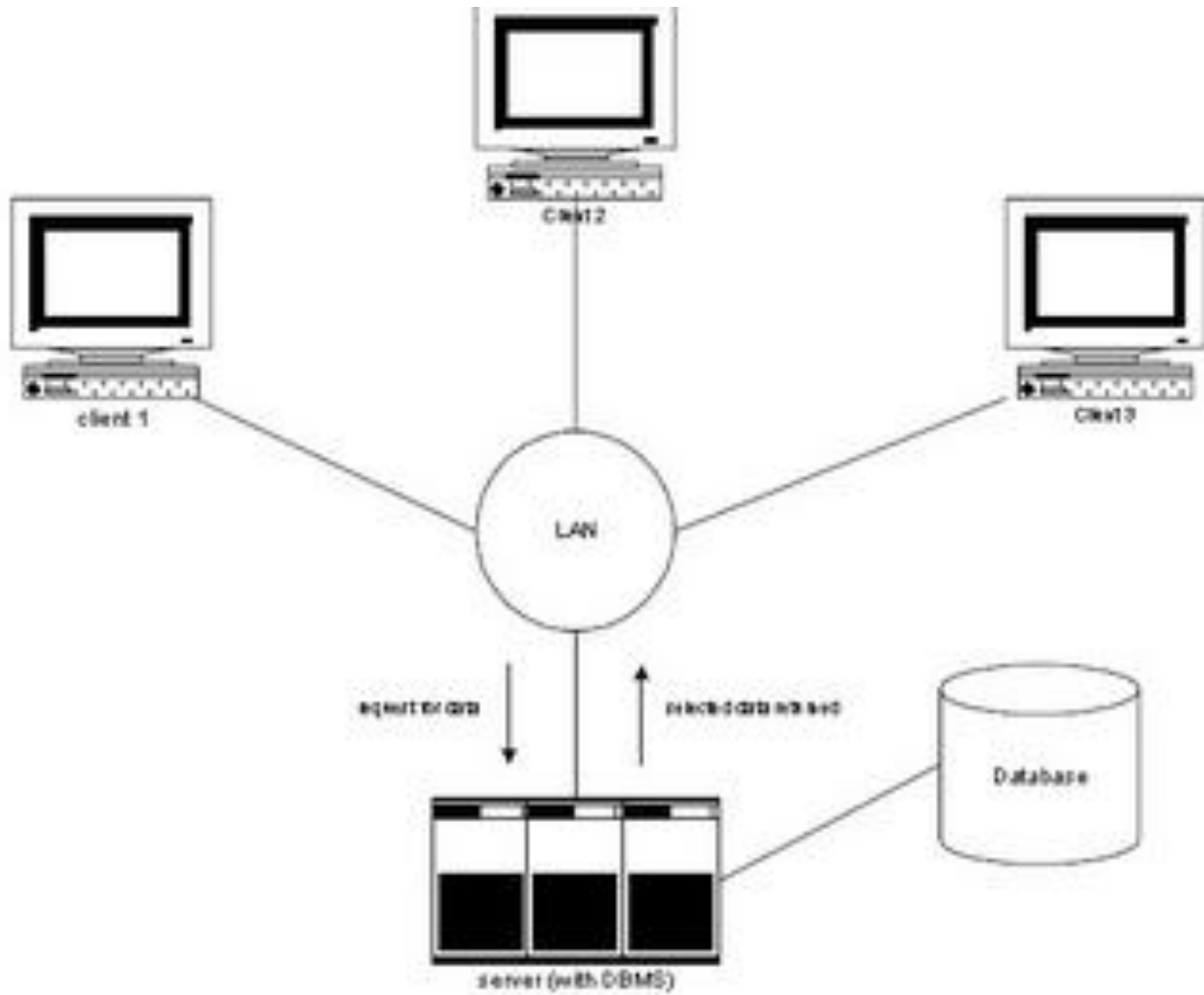
Server Power Protection and Management



3. ARSITEKTUR CLIENT-SERVER

- dimana sebuah komputer dapat menjadi sebuah media pengolah data untuk client.
- Dalam arsitektur ini, client berfungsi sebagai workstation yang dimana program aplikasi dijalankan dan pengatur interface juga sebagai penerima permintaan pemakai
- sedangkan server akan mengolah data yang dikirimkan dan dikembalikan lagi output dari olahan tersebut ke client asal.





DATA DICNIONARY

- *Data dictionary* adalah tempat penyimpanan informasi yang menggambarkan data dalam database.
- *Data dictionary* biasa disebut juga dengan *metadata* atau *data mengenai data*. Modul pengontrol otorisasi menggunakan *data dictionary* untuk memeriksa apakah seorang pemakai perlu mempunyai wewenang.
- Kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga user dan analis sistem mempunyai pengertian yang sama tentang input, output dan komponen data store.
- Pembentukan kamus data didasarkan pada alur data yang terdapat pada DFD



FORM KAMUS DATA

Data Flow Dictionary Entry	Data Store Dictionary Entry
Data Structure Dictionary Entry	
Data Element Dictionary Entry	



DATA FLOW DICTIONARY ENTRY (DFDE)

- Menerangkan setiap data flow pada DFD dan hanya berisi summary data (data ringkasan) serta menerangkan alur yang mengidentifikasikan dari mana alur itu berasal dan kemana alur itu menuju.



DATA FLOW DICTIONARY ENTRY

Use : To describe each data flow in a data flow diagram

DATA FLOW NAME : Sales Orders

DESCRIPTION : -

FROM : 1. Open Mail

TO : 2. Enter Sales Order Data

DATA STRUCTURES : Sales Order Record

COMMENTS : -

Keterangan :

- Data flow name : nama yang digunakan pada DFD.
- Description : Menjelaskan secara singkat aturan flow didalam sistem.
- From : Menunjukkan asal dari data flow (dapat berupa proses, data store dan terminator).
- To : Menunjukkan tujuan dari data (dapat berupa proses, data store dan terminator).
- Comments : Memberikan keterangan - keterangan yang penting saja.



DATA STORE DICTIONARY ENTRY

- Menerangkan setiap data store yang unik pada DFD . Jika data store yang sama muncul lebih dari sekali, maka bentuk tunggal yang digunakan.



DATA STORE DICTIONARY ENTRY

Use : describe each unik data store in a data flow diagram

DATA STORE NAME : Sales Order Form File

DESCRIPTION : -

DATA STRUCTURES : Sales order record

VOLUME : Approximately 140 per day

ACTIVITY : -

ACCES : Order departement personel

COMMENTS : -

Keterangan :

- Data store name : Nama data store yang digunakan pada DFD.
- Description : Menjelaskan secara singkat jenis data yang terkandung dalam data store.
- Data Structures : Data Struktur yang ada pada data store.
- Volume : Menunjukkan ukuran dari data store.
- Activity : Menunjukkan informasi yang berhubungan dengan record yang aktif di dalam file.
- Access : Batasan-batasan pada data.
- Comments : Memberikan keterangan - keterangan yang penting saja.



DATA STRUCTURE DICTIONARY ENTRY

- Data Structure Dictionary Entry dilengkapi dengan setiap struktur yang ada pada data store dan data flow.
- Tujuan : Untuk menghubungkan summary description dari Data Flow dan Data Store Dictionary Entry ke deskripsi detail dari Data Element Dictionary Entry.



DATA STRUCTURE DICTIONARY ENTRY

Use : To describe each unique data structure that exist in
(1) data flows and (2) data stores

STRUCTURE NAME : Sales Order record

DESCRIPTION : -

DATA ELEMENTS : CUSTOMER.NUMBER
CUSTOMER.ORDER.NUMBER
SALESPERSON.NUMBER
CUSTOMER.ORDER.DATE
*ITEM.NUMBER
*ITEM.DESCRPTION
*ITEM.QUANTITY
*ITEM.UNIT.PRICE
*ITEM.EXTENDED.PRICE

COMMENTS : Elements marked with asterisks occur
for each item record.

Keterangan :

- Structure name : nama yang sama dengan form data store dan data flow.
- Description : Menjelaskan bagaimana struktur digunakan.
- Data Elements : Daftar elemen data yang terkandung dalam struktur.
- Comments : Memberikan keterangan - keterangan yang penting saja.



DATA ELEMENT DICTIONARY ENTRY

- Data Structure Dictionary Entry menyediakan dasar untuk skema database.
- Tujuan : Untuk menstandarkan deskripsi dari suatu elemen sehingga elemen itu direferensikan dengan cara yang sama setiap kali digunakan.



DATA ELEMENT DICTIONARY ENTRY

Use : To describe each unique data element contained in
Data structure

DATA ELEMENT NAME	:	SALESPERSON.NUMBER
DESCRIPTION	:	The number that identifies the salesperson
TYPE	:	Numeric
LENGTH	:	4
NO. DECIMAL POS	:	-
ALIASES	:	Salesman Number, Sales Rep Number
RANGE OF VALUES	:	0001-9999
TYPICAL VALUES	:	
SPECIFIC VALUES	:	
OTHER EDITING DETAILS	:	

Keterangan :

- Data Element name : nama elemen data.
- Description : Menjelaskan elemen data.
- Type : tipe data (Alphabet, numeric, alphanumeric).
- Length : ukuran elemen data
- No. Dec. Pos : dapat diisi jika tipenya numeric
- Aliases : Nama lain
- Range of Values : informasi yang digunakan oleh programmer untuk mendeteksi kesalahan data. Demikian juga dengan field Typical Values dan Specific Values diisi bila diperlukan.
- Others editing details : Keterangan tambahan yang dianggap penting.
- Misal : Employee Age dapat digantikan dengan Date of Birth.



KAMUS DATA MENDIFINISIKAN DATA ELEMEN DENGAN CARA :

- Menguraikan arti dari alur data dan data store dalam DFD
- Menguraikan komposisi paket data pada alur data ke dalam alur yang lebih kecil. Contoh : Alamat langganan yang terdiri dari nama jalan, kota dan kode pos.
- Menguraikan komposisi paket data dalam data store.
- Menspesifikasikan nilai dan unit informasi dalam alur data dan data store.
- Menguraikan hubungan yang terinci antara data store dalam suatu ERD.



NOTASI KAMUS DATA

NOTASI	ARTI
=	Terdiri dari, terbentuk dari, sama dengan
+	Dan
()	Optional
{ }	Iterasi / pengulangan misal : 1 { } 10
[]	Pilih satu dari beberapa alternatif (pilihan) Misal : [A B C D]
**	Komentar
@	Identifier suatu data store
	Pemisah dalam bentuk []
Alias	Nama lain untuk suatu data



Contoh :

- NOTASI "="

Nama=Nama_Depan + Nama_belakang

- NOTASI "("

Nama_Langganan = (title) + Nama_Depan + (Nama_Tengah)
+ Nama_Belakang

Customer_Address = (Shipping_Address) + (Billing_Address)

- NOTASI "{}"

Order=Customer_Name + Shipping_Address + 1 { item} 10

- NOTASI "[]"

Jenis_Kelamin = [Pria | Wanita]

- NOTASI "* *"

Penjualan = *Jumlah penjualan setiap tahun*

Pajak_rate = *Pajak yang berlaku ditentukan oleh pemerintah *

- NOTASI "Alias"

Client Alias Customer



SUMBER

- Devie Rosa Anamisa. 2014. Konsep dasar DBMS
- TSI Perbankan. B1 – Konsep Dasar DBMS dan SQL.
- parno.staff.gunadarma.ac.id/.../SI_10_Kamus_Data.pdf
- Arsitekturbasisdata-130331193607-phpap.pptx

