**A**

**>**Sistem Basis Data -> sistem yang terdiri atas kumpulan file

> (tabel) yang saling berhubungan (dalam sebuah basis data

di sebuah sistem komputer) dan sekumpulan program

(DBMS) yang memungkinkan beberapa pemakai dan/atau

program lain untuk mengakses dan memanipulasi file-file

(tabel-tabel) tersebut

**2.** 1. Level fisik *(Physical Level)*

- Level terendah, menunjukkan bagaimana sesungguhnya data disimpan

- Pemakai melihat data sebagai gabungan dari struktur dan datanya.

- Pemakai mengetahui bagaimana representasi fisik dari

simpanan/pengorganisasian data sebagai teks,angka bahkan bit data.

2. Level lojik/konseptual *(Conceptual Level)*

- Level yang menggambarkan data apa yang sebenarnya (secara

fungsional) disimpan dalam basis data.

- Pemakai mengetahui bahwa data pegawai disimpan dalam beberapa tabel

seperti file/tabel identitas, file/tabel pendidikan, file/tabel keluarga dsb.

3. Level penampakan *(View Level)*

- Level tertinggi yang menunjukkan sebagian data dari basis data.

- Kemunculan data/tampilan data dimata pemakai diatur oleh aplikasi *end*

*user* sehingga data pada level penampakan sudah berbentuk data siap saji.

3. *Characters* -> bagian data yang terkecil, dapat berupa

database karakter numerik (angka 0 - 9), huruf ( A - Z, a - z) ataupun

karakter-karakter khusus, seperti \*, &. %, # dan lain-lain.

• *Field* -> merepresentasikan suatu atribut dari record yang

menunjukkan suatu item dari data, seperti misalnya nama,

alamat, dsb.

Setiap field harus mempunyai :

- field name : harus diberi nama yg unik

- field representation : tipe field (karakter, teks, tanggal,

angka, dsb), lebar field (ruang maksimum yang dapat

diisi dengan data)

file/tabel

record

field

data).

- field value: isi dari field

• *Record /* Baris Data -> Kumpulan dari *field* membentuk

suatu *record*. Sebuah *record* menggambarkan suatu unit

data individu yang tertentu.

• *File/*Tabel -> *File* terdiri dari *record-record* yang

menggambarkan satu kesatuan data yang sejenis.

4. Dalam pendekatan **Normalisasi** -> perancang basis data bertitik tolak dari

situasi yang nyata dimana ia telah memiliki item-item data yang siap

ditempatkan dalam baris dan kolom pada tabel-tabel relasional dan ia telah

mengetahui sejumlah aturan tentang keterhubungan antara item-item data

tersebut.

Dalam pendekatan **Model E-R** -> perancang basis data dengan langsung

membuat model data jika yang telah diketahui baru prinsip-prinsip sistem

hati-hati dari Model E-ke bentuk fisik akan menghasilkan hasil yang mirip.

prinsip secara keseluruhan karena adanya kelangkaan data/fakta yang dimiliki.

Faktanya di lapangan, kedua pendekatan ini dilakukan bersama-sama,

berganti-ganti, dan dapat saling memperkuat satu sama lain, dari contoh

data (fakta)yang telah dimiliki dilakukan Normalisasi kemudian hasilnya

diwujudkan dalam Model E-R, setelah itu diimplementasikan dalam bentuk

sejumlah struktur tabel dalam sebuah basis data, struktur tabel ini dapat diuji

kembali dengan menerapkan aturan-aturan Normalisasi, hingga akhirnya

diperoleh sebuah struktur basis data yang benar-benar **efektif dan efisien**.

5. Tahap 1, Pengumpulan data dan analisis

* Tahap 2, Perancangan database secara konseptual
* Tahap 3, Pemilihan DBMS
* Tahap 4, Perancangan *database* secara logika (*data model mapping*)
* Tahap 5, Perancangan *database* secara fisik
* Tahap 6, Implementasi Sistem *database*

6. pembahasan 5 macam atribut :

* Attribut kunci dan attribute desktiptif

Yaitu gabugungan dari beberapa atribut yang dapat memebedakan semua baris data dalam tabel secara unik.

* Attribut sedehana

Merupakan attribute yang tidak bisa di pecah menjadi attribute lain

* Attribute bernilai tunggal (Siggle- valued attribute ) dan bernilai banyak (multi-valued attribute)

Merupakan attribute yang hanya memiliki satu nilai pada setiap barisnya.

* Attribute harus bernilai (mandatory attribute )dan attribute nilai null (nuk value attribute)

Merupakan nilai data untuk setiap barisnya

* Attribute turunan(devided attribute)

Nilai di peroleh dari pengolahan atau dapat di turunkan dari attribute lain yang berkaitan

7. Ketergantungan fungsional ( KF) dan contoh:

Ialah konsep dasar normalisasi yang menjelasskan hubungan antara atribut , atau lebih khusus menjelaskan nilai suatu atribut lainya.

Contoh : A B (A secara fungsional menentukan nilai B)

8. Definisi dari model data berdasarkan objek dan record dan jenis- jenis dari kedua model data

Tersebu.

* Model data objek yaitu menggunakan konsep enttitas, attribute dan hubungan antara entitas .
* Sedangkan model data record menjelasskan pada our tentang hubungan logic antar data dalam basis data,dan digunakan juga untuk mengurai implementasi dari DB( Higher level description of implementasio) .

3 jeis model data,paa data objek dan data record :

1. Modle Relational, Hubungan anatara data di rperesentasikan oleh sejumlah tabel masing-masing ,tabel terdiri dari beberaapa kolom yang Namanya unique.
2. Model adta hirarki, Dimana data serta hubungan antara data di implementasikan dengan record dan link / pomiter), dimana record-record tersebut di susun dalam bentuk tree(pohon) dan masing-masimg node pada tree trb merupakan record / group data elemen dan memeliki hubungan koordinalitas.
3. Model jaringan, dimana data dan hubungan anatra data di representasikan dengan record dan link.

9. 4 macam derajat kardinalitas dalam rancngan database:

Yaitu menunjukan jumlah maksimum entitas yang dapat berelasi dengan entitas pada himpunan entitas yang lain, kardinalitas relasi yang terjadi di antara dua himpuanan dapat berupa:

1. Satu ke satu (one to one) berarti setiap entitas pada himpunan entitas A berhubungan paling banyak dengan entitas pada entitas himpuanan B,begitu juga sebaliknya.

Contoh MHS dan KM

1. Satu ke banyak( one to many)Himpunan entitas A dapat berhubungan dengan banyak entitas pada himpunana entitas B tetapi tidak sebaliknya

Contoh: MHS dan MTK.

1. Banyak ke satu ( many to one),himpunan entitas A behubungan berhubungan paling banyak satu entitas pada himpuanan entitas B ,tetapi tidak sebaliknya

Contoh : MTK dengan DOSEN

1. Banyak ke banyak ( many to many) , himpunan entitas A dpat berhubungan banyak entitas pada himpuanan entitas B dan dekian juga sebaliknya

Contoh : produk dan konsumen.

10. 3 syarat yang harus terpenuhi dalam mebuat tabel yang baik

1. Dekomposisi(pengurutan), tabel maka dekomposisinya harus di jamin aman (lossless- join decomposition)

2. Terpeliharanya ketergantungan fungsional pada saat perubahan data (dependency presorvtion)

3. Tidak melanggar Boyce- Code Normal form (BCNF)

**B**

CDM



PDM

