

| | | |
|---|---|---|
| Mata Kuliah | : | Sistem Basis Data |
| Bobot Sks | : | 3 sks |
| Dosen Pengembang | : | Cian Ramadhona Hassolthine, S.Kom., M.Kom |
| Tutor | : | Anita Ratnasari, S.Kom, M.Kom |
| Capaian Pembelajaran Mata Kuliah | : | Perancangan basis data fisikal |
| Kompetensi Akhir Di Setiap Tahap (Sub-Cpmk) | : | Mahasiswa dapat mengetahui tahapan dari metode Perancangan basis data fisikal |
| Minggu Perkuliahan Online Ke- | : | Sesi 15 |

Perancangan Basis Data Fisikal

Tahap Implementasi

- 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 sebuah **basis data** secara fisik. Sedangkan komponen-komponen ER yang berupa himpunan entitas dan himpunan relasi akan diwujudkan menjadi **tabel-tabel**. Selanjutnya, atribut-atribut yang melekat pada masing-masing himpunan entitas dan himpunan relasi akan dinyatakan sebagai **field-field** dari tabel yang sesuai
- Performansi basis data ditentukan oleh :
 - Kualitas dan bentuk perancangan basis data
 - Kualitas mesin / komputer
 - Platform yang dipilih
 - Sistem operasi
 - DBMS yang digunakan

Definisi

Tahap ini merupakan proses untuk menghasilkan sebuah deskripsi dari implementasi basis data pada media penyimpanan kedua, dimana mendeskripsikan base relation, organization file, dan pengindeksan yang digunakan untuk mendukung efisiensi dari akses data, dan integritas constraint beserta security measure. Bahwa internal skema dapat diubah oleh DBA tanpa mengganggu konseptual skema. Dengan kata lain perancangan basis data fisik menunjukkan kekebalan konseptual skema terhadap perubahan internal skema.

Contoh :

Sebuah file Pegawai yang akan didefinisikan

- Tingkat Konseptual :

PEGAWAI

| Coloumn | Type | Length |
|---------------|-----------|--------|
| NOMOR_PEGAWAI | CHARACTER | 6 |
| NOMOR_DEPT | CHARACTER | 4 |
| GAJI | NUMERIC | 6 |

- Tingkat Internal

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 | |

Tahapan

- Menerjemahkan model data logikal untuk DBMS yang digunakan
- Merancang organisasi file dan index
- Merancang view pengguna
- Merancang mekanisme keamanan

Studi Kasus

Menerjemahkan Model Data Logika untuk DBMS yang Digunakan

Tujuan langkah ini adalah untuk menghasilkan skema basisdata relasional dari model data logikal yang dapat diterapkan pada DBMS yang digunakan.

- Merancang relasi dasar

Langkah ini guna memutuskan bagaimana merepresentasikan base relation yang diidentifikasi pada model data logikal kedalam sasaran DBMS.

MasterSupplier

Domain KodeS : Integer
Domain NamaS : Variable length character string, length 75
Domain AlamatS : Variable length character string, length 100
Domain KdPosS : Variable length character string, length 5
Domain TelpS : Variable length character string, length 40
Domain FaxS : Variable length character string, length 40

MasterSupplier(

| | | |
|---------|---------|-----------|
| KodeS | KodeS | NOT NULL, |
| NamaS | NamaS | NOT NULL, |
| AlamatS | AlamatS | NOT NULL, |
| KdPosS | KdPosS | NOT NULL, |
| TelpS | TelpS | NOT NULL, |
| FaxS | FaxS | NOT NULL, |

Primary Key(KodeS),

Foreign Key KdPos references MasterKdPos(KdPos) ON UPDATE CASCADE ON DELETE NO ACTION);

- Merancang representasi representasi derived data

Tujuan langkah ini adalah untuk memutuskan bagaimana representasi derived data yang terdapat pada model data logika di dalam DBMS yang digunakan.

- Merancang general constraint

Langkah ini bertujuan untuk merancang constraints perusahaan untuk diterapkan pada DBMS yang akan digunakan.

Format nomor KodePeg pada MasterPegawai berupa 'PGXXX' dimana X mewakili karakter berupa digit 0-9. Batasan ini diterapkan dengan menambah constraint di dalam tabel MasterPegawai.

CONSTRAINT formatKodePeg

CHECK(KodePeg like 'PG[0-9] [0-9] [0-9]')

Merancang Organisasi File dan Index

Langkah ini bertujuan untuk menentukan pilihan organisasi file yang optimal untuk menyimpan relasi – relasi dasar dan index yang diperlukan untuk mencapai kinerja yang memuaskan serta bagaimana relasi dan record data disimpan didalam secondary storage.

- Menganalisa transaksi

Langkah ini bertujuan untuk memahami fungsionalitas transaksi yang akan berjalan didalam basis data dan menganalisa transaksi – transaksi yang penting.

- Data Entry
- Data Update
- Data Queries

- Memilih organisasi file

Langkah ini bertujuan untuk menentukan organisasi file yang efisien untuk tiap – tiap relasi dasar jika diperbolehkan oleh DBMS yang digunakan.

- Memilih index

Langkah ini bertujuan untuk memutuskan apakah penambahan index akan meningkatkan kinerja sistem basis data.

| No | Relasi | Clustered Index | NonClustered Index |
|----|------------------|-----------------|--------------------|
| 1 | MasterSupplier | KodeS | - |
| 2 | MasterPerusahaan | KodeP | - |
| 3 | MasterCabang | KodeC | KodeP |
| 4 | MasterPegawai | KodePeg | KdJabatan |

| | | | |
|---|---------------|-----------|-------|
| | | | KodeC |
| 5 | MasterJabatan | KdJabatan | - |

- Memperkirakan kebutuhan disk space

Langkah ini bertujuan untuk menghitung kapasitas penyimpanan yang dibutuhkan oleh basis data.

| Atribut | Tipe Data | Length |
|---------|-------------|--------|
| KodeS | Integer | 4 |
| NamaS | Varchar(75) | 75 |
| AlamatS | Varchar(75) | 100 |
| KdPosS | Varchar(75) | 5 |
| TelpS | Varchar(75) | 40 |
| FaxS | Varchar(75) | 40 |

| Atribut | Tipe Data | Length |
|--|-----------|--------|
| Kapasitas dari table MasterSupplier 264 Byte Diperkirakan dalam 1 bulan terjadi 1 penambahan MaterSupplier baru Dalam satu tahun pertumbuhan dari tabel ini adalah $1*1*12*264 = 3168$ Byte atau 3,168 KByte | | |

Merancang View Pengguna

Langkah ini bertujuan untuk merancang userview yang telah ditentukan dalam tahap analisis dan pengumpulan persyaratan.

```
CREATE VIEW View_Pembelian AS
```

```
Select dbo.Pembelian.NoRegPmb, FROM dbo.Pembelian
```

Merancang Mekanisme Keamanan

Langkah ini bertujuan untuk merancang mekanisme keamanan pada basis data seperti yang dispesifikasikan oleh pengguna.

| Tabel | User | | |
|---------------------|-------|----------|-------|
| | Admin | Direktur | Staff |
| MasterSupplier | ALL | I, S, U | I, S |
| MasterPerusahaan | ALL | I, S, U | I, S |
| MasterCabang | ALL | I, S, U | I, S |
| MasterPegawai | ALL | I, S, U | I, S |
| DataKonsumen | ALL | I, S, U | I, S |
| MasterUnitKendaraan | ALL | I, S, U | I, S |
| Pembelian | ALL | I, S | I, S |
| StokUnit | ALL | I, S | I, S |
| Penjualan | ALL | I, S | I, S |
| Piutang | ALL | I, S | I, S |

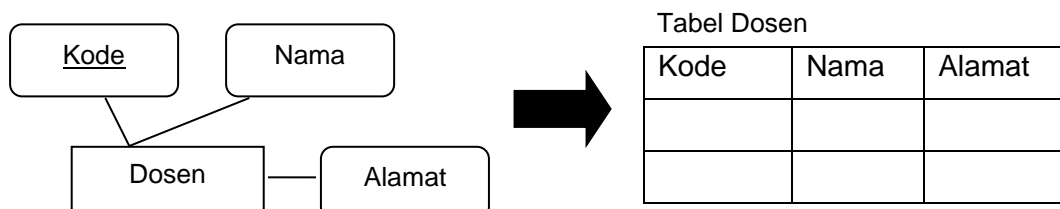
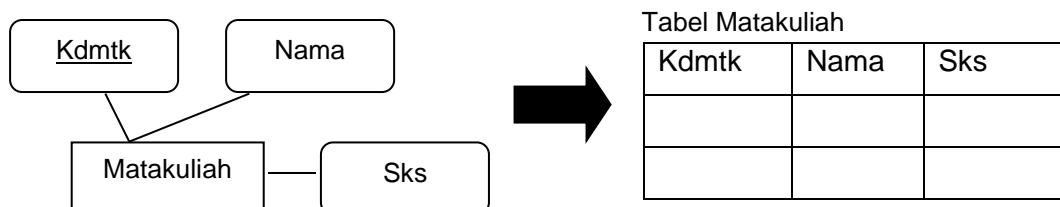
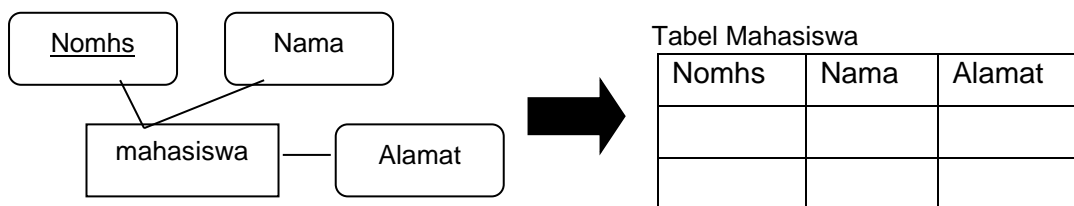
2. Pengkodean / Abstraksi data

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

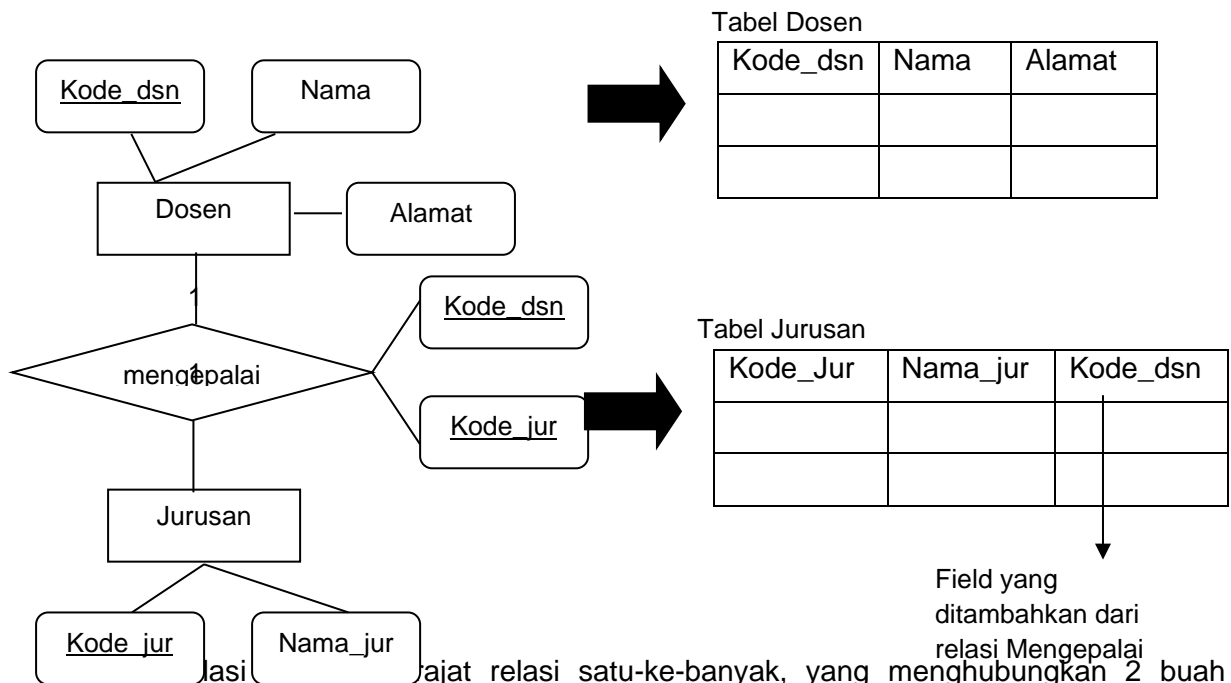
- Sekuensial
 - Pengkodean dilakukan dengan mengasosiasikan data dengan kode yangurut
 - Contoh : predikat kelulusan "Sangat Memuaskan", "Cukup Memuaskan", "Memuaskan" → dikodekan dengan huruf "A", "B", "C"
- Mnemonic
 - Pengkodean dilakukan dengan membentuk suatu singkatan dari data yang hendak dikodekan.
 - Contoh : "Laki-laki" → dikodekan 'L'; "Perempuan" → dikodekan "P"
- Blok
 - Pengkodean dinyatakan dalam format tertentu
 - Contoh : Nomor mahasiswa dengan format XX.YY.ZZZZ → terdiri atas XX = 2 digit tahun masuk, YY = 2 digit kode jurusan, ZZZZ = 4 digit nomor urut

3. Transformasi Model data ke Basis data fisik

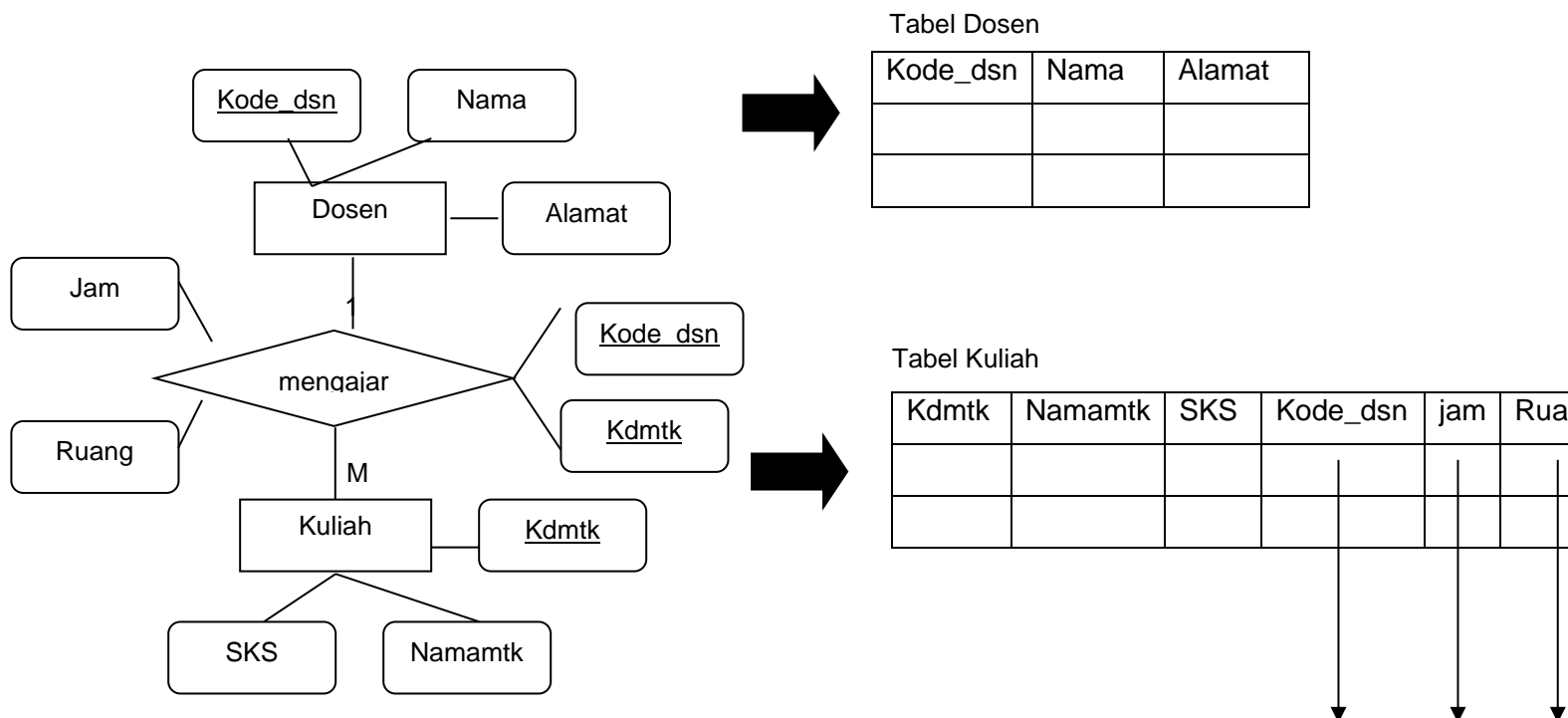
- Aturan umum dalam pemetaan model data yang digambarkan dalam ERD (level konseptual) menjadi Basis data fisik (level fisik) adalah :
 - a. Setiap himpunan entitas akan diimplementasikan sebagai sebuah tabel (file data)



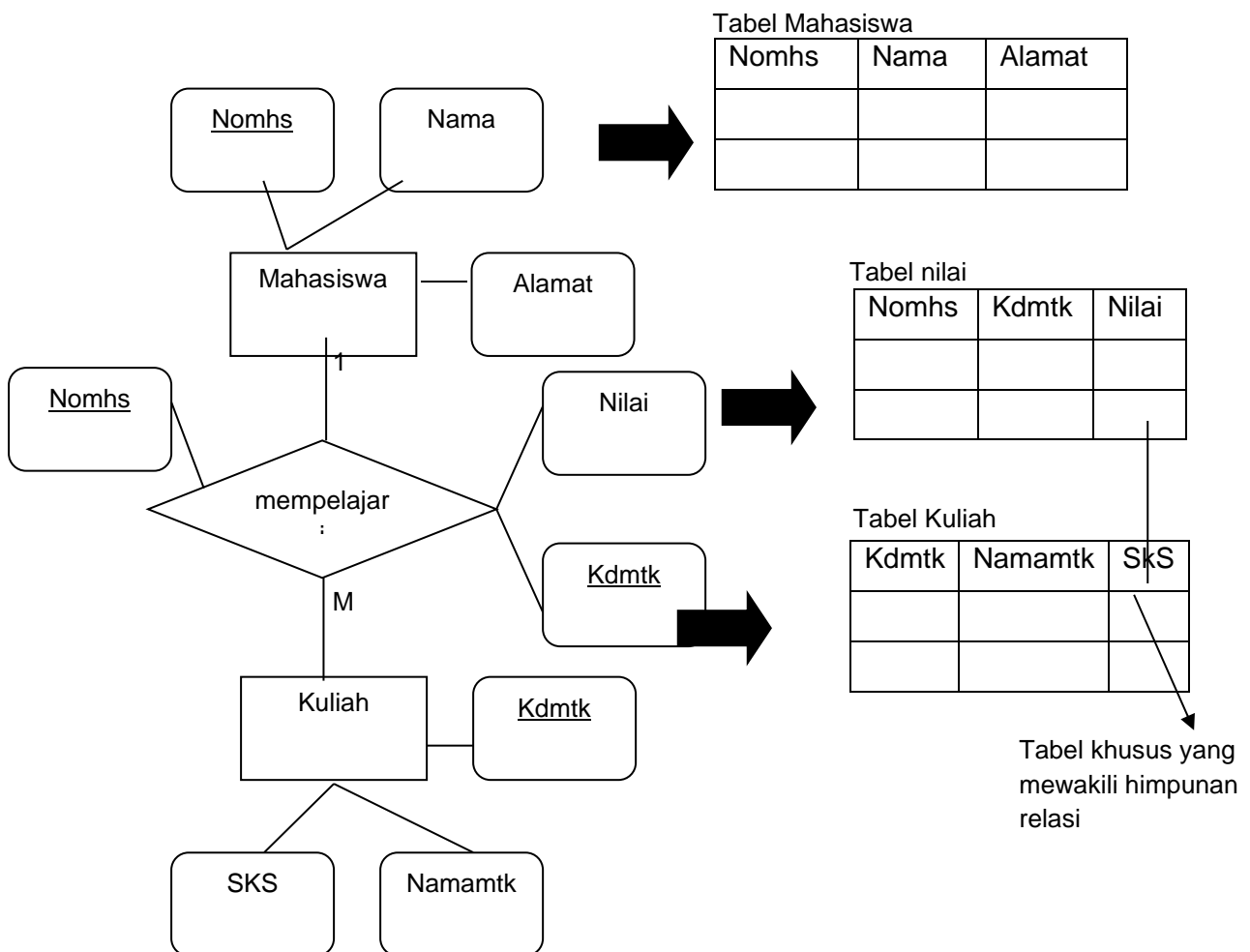
- b. Relasi dengan derajat relasi satu-ke-satu, yang menghubungkan 2 buah himpunan entitas akan direpresentasikan dalam bentuk penambahan / penyertaan atribut-atribut relasi ke tabel yang mewakili salah satu dari kedua himpunan entitas



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



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



4. DBMS dan Struktur tabel

- Dalam menentukan struktur dari tabel, paling tidak setiap struktur tabel berisikan nama field, tipe field dan ukurannya
- Tatacara penamaan field, pilihan tipe field serta fasilitas tambahan lainnya untuk struktur tabel sangat tergantung pada DBMS yang digunakan
- Tipe data yang bersifat umum adalah :
 - **Data 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

5. Indeks dan Struktur penyimpanan

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

- Ada 2 indeks :
 - a. Indeks Primer (primary index)
 - IP pada setiap tabel hanya ada satu dan hampir selalu berasal (ditentukan) dari kunci primer yang telah ditetapkan dalam sebuah entitas / relasi
 - IP yang baik terdiri atas field-field dengan kriteria sbb :
 - Field yang menjadi komponen IP harus bersifat mandatory (datanya tidak boleh kosong atau berisi nilai null)
 - Keseluruhan nilai IP bersifat unik
 - Nilai-nilainya lebih permanen (idealnya tidak pernah berubah)
 - Berukuran kecil (pendek) dengan jumlah field minimal (sedikit)
 - b. Indeks Sekunder (secondary index)
 - Digunakan untuk mendukung keberadaan IP yang dibuat untuk suatu tabel dengan alasan untuk mempermudah berbagai cara pengaksesan ke suatu tabel
 - Misalnya : field Nama_Mahasiswa → 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

6. Struktur penyimpanan

- Ada 7 pilihan struktur penyimpanan dasar yang dapat diterapkan pada suatu tabel (bergantung pada DBMS yang dipakai) yaitu : Pile, Heap, hash, Sekuensial Berindeks, File berindeks, Multiring
 - a. **Heap**
 - Merupakan struktur penyimpanan yang paling sederhana dan paling hemat dalam kebutuhan ruang penyimpanan
 - Setiap baris data disusun berdasar kronologis penyimpanannya. Record yang pertama disimpan akan ditempatkan di posisi awal ruang penyimpanan, dan begitu seterusnya
 - Pengubahan data tidak akan mengubah urutan record tersebut. Jika terjadi penghapusan, maka record-record dibawahnya akan dimampatkan untuk mengisi tempat yang kosong akibat penghapusan

- Pencarian data berjalan dengan lambat, karena dilakukan secara sekuensial baris demi baris
- Struktur ini cocok untuk tabel berukuran kecil dan jarang berubah

b. Hash

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

c. Sekuensial berindeks

- Menempatkan data dengan 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

Daftar Pustaka

Korth, H & Mc Graw Hill. 1991. *Database System Concept, 4th edition*. New York