Sistem Basis Data

ORGANISASI FILE

Alif Finandhita, S.Kom

ORGANISASI FILE

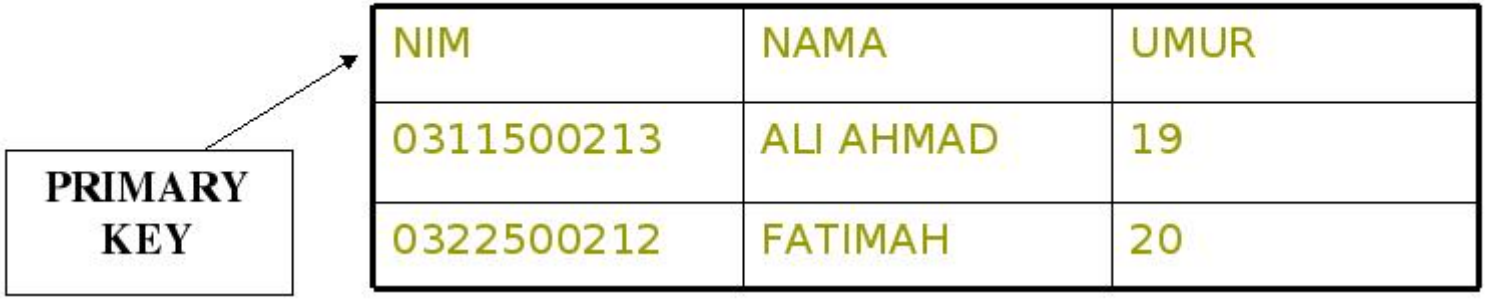
- Penyimpanan ataupun penulisan character demi character yang ada di dalam external memory, harus diatur sedemikian rupa sehingga komputer bisa dengan mudah menemukan kembali data-data yang tersimpan didalamnya. Aturan inilah yang kemudian dikenal sebagai organisasi file
- File diorganisasi (disusun) berdasarkan urutan-urutan record-record
- Record-record dipetakan ke dalam blok-blok dalam harddisk
- 1 Blok berisi lebih dari 1 record

Konsep Dasar Organisasi File

- **Field** : Satuan informasi terkecil yang menyusun record
- **Record** : Kumpulan dari field yang berhubungan satu sama lain
- **File** : Kumpulan dari record-record
- **Basis data** : Kumpulan file yang digunakan oleh program aplikasi serta membentuk hubungan tertentu di antara record-record di file-file tersebut
- **Key** : Elemen record yang dipakai untuk menemukan record tersebut pada waktu akses

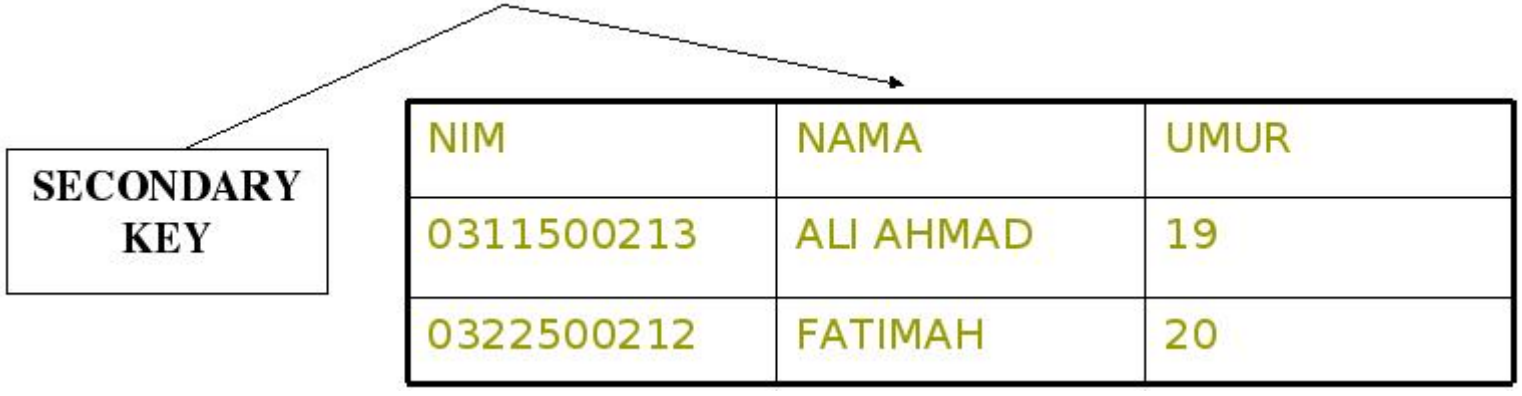
Jenis – jenis Key

- **Primary Key** : Field yang mengidentifikasikan sebuah record dalam file. Bersifat unik.



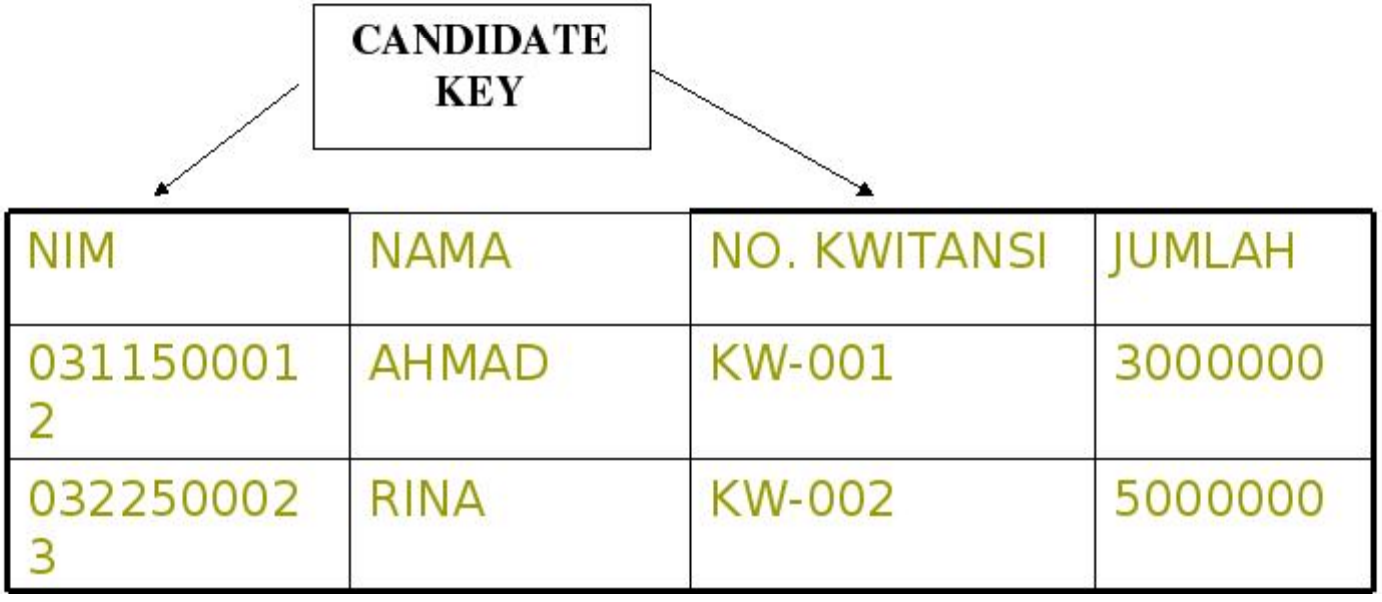
Jenis – jenis Key (2)

- **Secondary Key** : Field yang mengidentifikasikan sebuah record dalam file. Tidak bersifat unik.



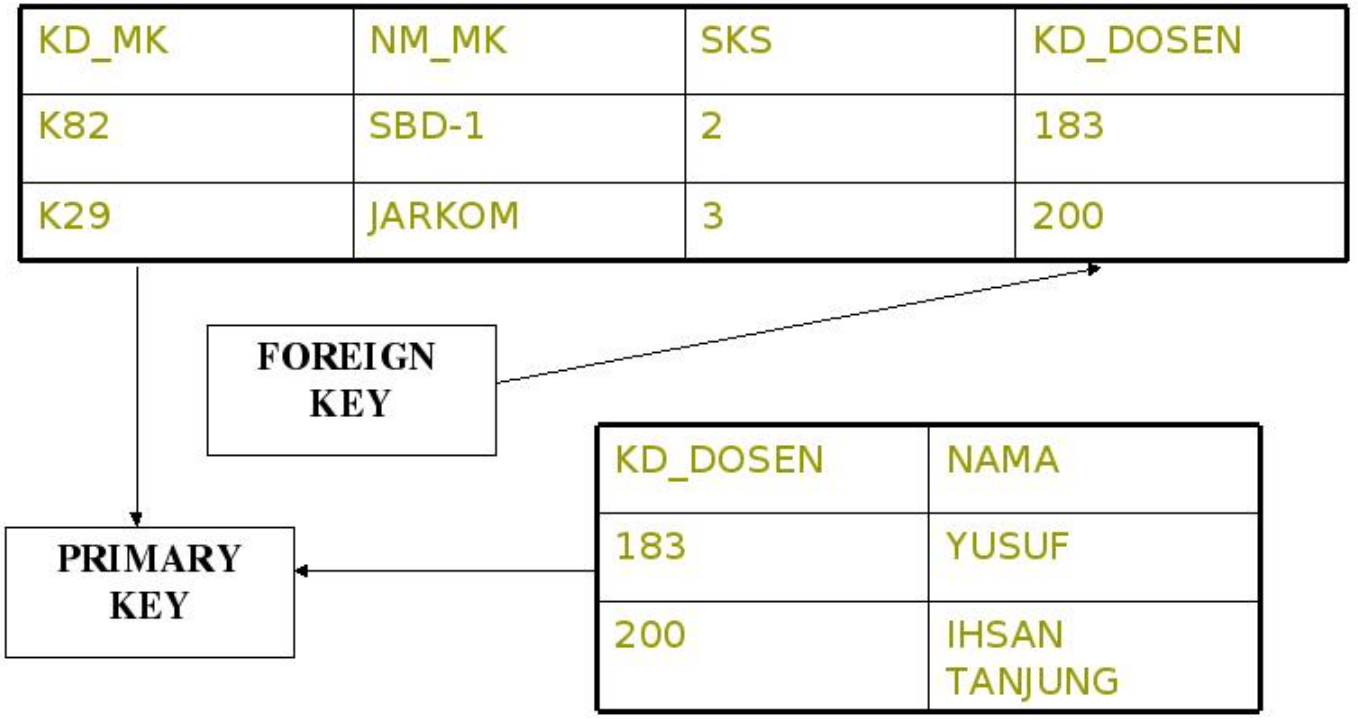
Jenis – jenis Key (3)

- **Candidate Key** : Field-field yang bisa dipilih (dipakai) menjadi primary key.



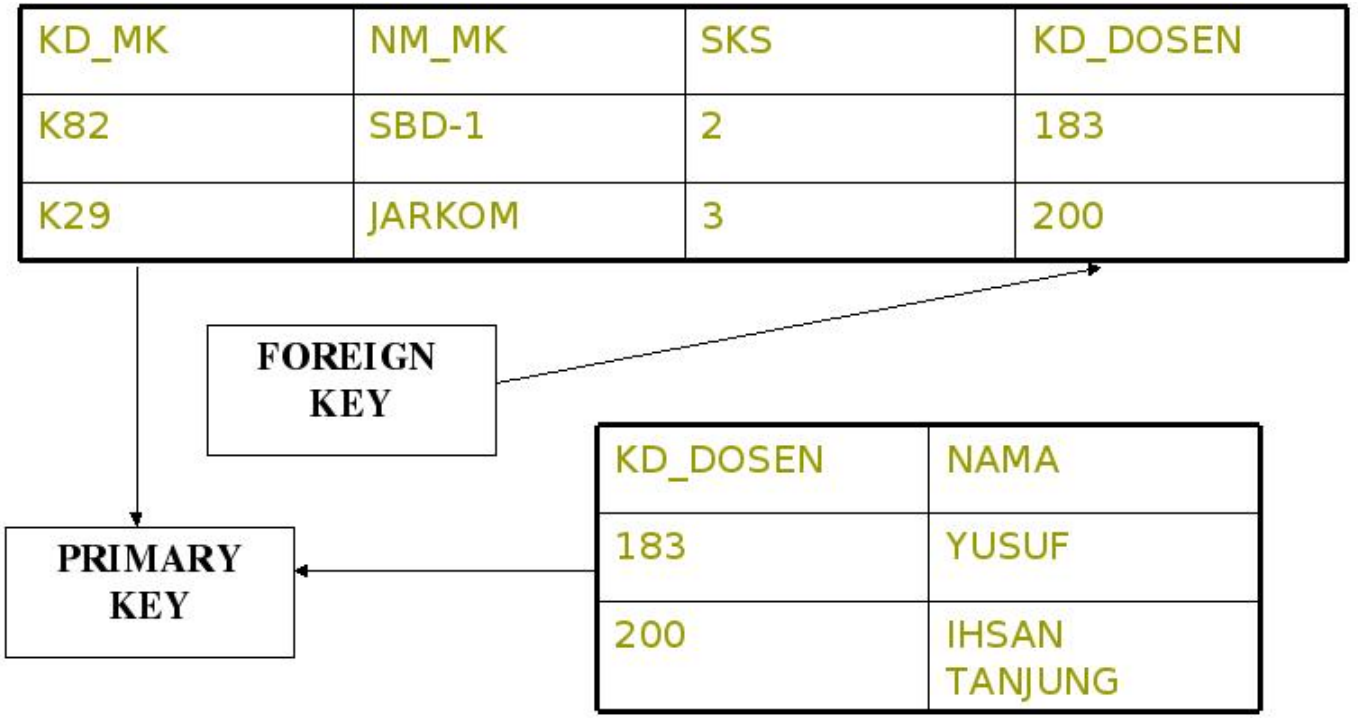
Jenis – jenis Key (4)

- **Foreign Key** : Field yang bukan key, tetapi adalah primary key pada file yang lain.



Jenis – jenis Key (5)

- **Foreign Key** : Field yang bukan key, tetapi adalah primary key pada file yang lain.



Jenis Record (Berdasarkan Panjang)

- **Fixed Length Record**

  Record yang panjangnya tetap

- **Variable Length Record**

  Record yang panjangnya tidak tetap (sesuai dengan jumlah karakter pada record)

Fixed Length Record

- **Misal : Untuk Membuat Record Mahasiswa**

RECORD 1	0411500005	Ahmad Zaki	Cipondoh
RECORD 2	0422500025	Sinta	Kebayoran Lama
RECORD 3	0422500035	Indra Gunawan	Cipulir
RECORD 4	0433500058	Bekti Sularso	Cidodol
RECORD 5	0444500057	Tini Lestari	Cileduk

Fixed Length Record (2)

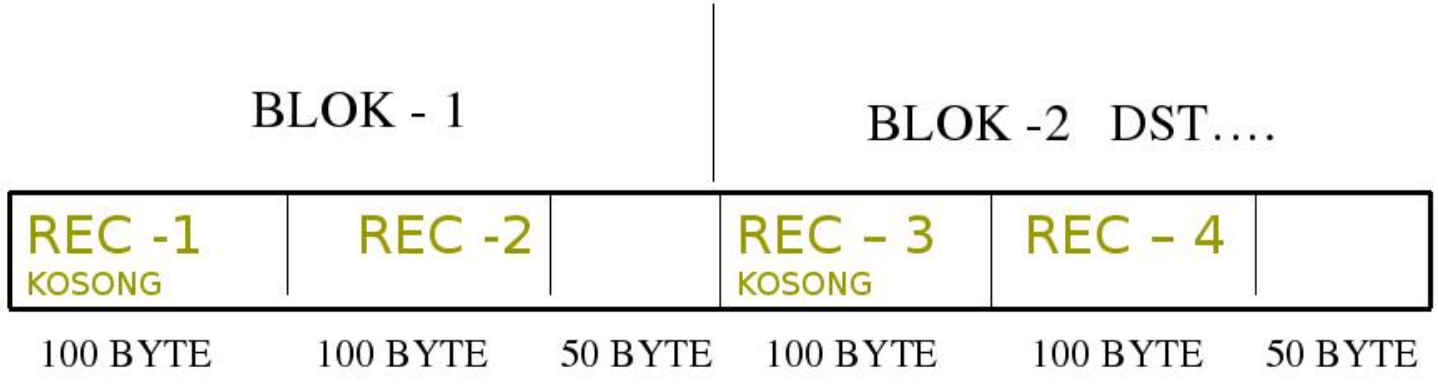
"TYPE MAHASISWA = RECORD
NIM : CHAR(10);
NAMA : CHAR(40);
ALAMAT : CHAR(50);
END"

- Tiap karakter menyimpan 1 byte, maka record ke 1 untuk data mahasiswa di atas akan menyimpan 100 byte, kemudian 100 byte untuk record yang kedua dan seterusnya.
- Penempatan record pada blok disebut blocking.
- Metode blocking untuk record berukuran tetap adalah fixed length blocking.

Fixed Length Record (3)

Contoh Metode Fixed Length Blocking :

- 1 block dapat menyimpan 250 byte, jika 1 record panjangnya 100 byte maka blocking adalah sebagai berikut :



Fixed Length Record (4)

Kelebihan Fixed Length Record :

- Mudah dalam pemrograman, karena untuk menyisipkan atau menghapus record mudah karena panjang recordnya sama.

Kekurangan Fixed Length Record :

- Tempat penyimpanan record boros dan tidak efektif.

Variable Length Record

ORGANISASI FILE

RECORD 1	0411500005	Ahmad Zaki	Cipondoh
RECORD 2	0422500025	Sinta	Kebayoran Lama
RECORD 3	0422500035	Indra Gunawan	Cipulir
RECORD 4	0433500058	Bekti Sularso	Cidodol
RECORD 5	0444500057	Tini Lestari	Cileduk

Panjang record 1 = 28 byte

Panjang record 2 = 29 byte

Panjang record 3 = 30 byte dst…

Variable Length Record (2)

"TYPE MAHASISWA = RECORD
NIM : CHAR(10);
NAMA : CHAR(40);
ALAMAT : CHAR(50);
END"

- Panjang tiap record berbeda-beda tergantung dari isi dari masing-masing record.
- Penempatan record dalam blok tergantung dari panjang record.
- Metode blocking untuk record berukuran tidak tetap adalah variable length spanned blocking dan variable length unspanned blocking.

Variable Length Record (3)

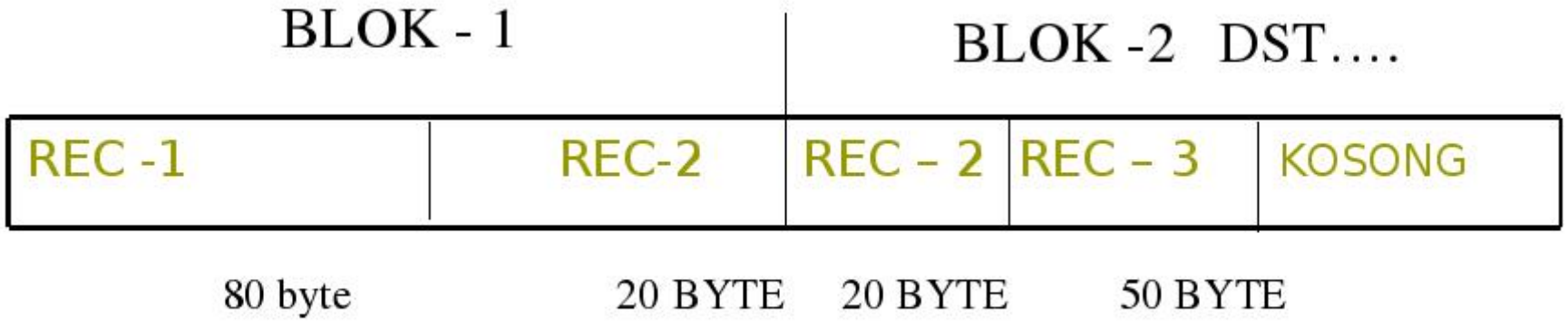
Variable Length Spanned Blocking

- Record ditempatkan dalam blok sesuai dengan ukurannya
- 1 record dapat dipotong / dibagi ke dalam blok terpisah.

Variable Length Record (4)

Misal : 1 blok dapat memuat 100 byte

- Panjang Record 1 = 80 byte, Panjang Record 2 = 40 byte, Panjang Record 3 = 50 byte



Variable Length Record (5)

Variable Length Unspanned Blocking

- Record ditempatkan dalam blok sesuai dengan ukurannya
- 1 record tidak boleh dipotong (1 record harus berada di dalam 1 blok).

Variable Length Record (6)

Misal : 1 blok dapat memuat 100 byte

- Panjang Record 1 = 80 byte, Panjang Record 2 = 40 byte, Panjang Record 3 = 50 byte

Variable Length Record (7)

Kelebihan Variable Length Record :

- Tempat penyimpanan record lebih hemat dan efektif.

Kekurangan Variable Length Record :

- Relatif sulit digunakan dalam pemrograman, karena panjang record berbeda maka tiap akhir record digunakan symbol end of record yang menandakan record sudah berakhir.

Cara Pengorganisasian Record

Record tersusun dalam sebuah file. Beberapa cara pengorganisasian record :

- **Organisasi File Heap**
  - Tiap record ditempatkan di mana saja di dalam file selama masih terdapat tempat untuk record tersebut
  - Tidak ada pengurutan dalam record
- **Organisasi File Sekuensial**
- **Organisasi File Index**
- **Organisasi File Hashing**

Organisasi File Sequential

Konsep Dasar

- Penempatan dan pembacaan record secara serial berdasarkan sebuah key
- File sequential didesign untuk efisiensi pemrosesan record pada saat pengurutan berdasarkan beberapa key.
- File dengan data yang tersusun dalam suatu urutan tertentu.
- Tiap record mempunyai field yang sama & dengan susunan yang sama.
- Untuk memungkinkan record tersusun secara urut perlu ditentukan key dari tiap record.

Organisasi File Sequential (2)

Insert Sebuah Record

- Menambahkan sebuah data baru ke dalam file
- Insert pada ujung akhir sebuah file, hanyalah menambah banyaknya data waktu yang dibutuhkan kecil

1	2	3	4	5	6	7	8	9	...
A	B	C	D	E	F	...	...	...	...

INSERT X PADA AKHIR RECORD

1	2	3	4	5	6	7	8	9	...
A	B	C	D	E	F	X	...	...	...

Organisasi File Sequential (3)

- Insert ditengah file mengakibatkan pergeseran ataupun perubahan struktur data yang tidak sederhana.

1	2	3	4	5	6	7	8	9	...
A	B	C	D	E	F	...	...	...	...

INSERT X PADA RECORD KE 3

1	2	3	4	5	6	7	8	9	...
A	B	X	C	D	E	F	...	...	...

→ RECORD KE-3 DST BERGESER

Organisasi File Sequential (4)

Delete Sebuah Record

- Mencari lokasi data & menghapus isinya, agar bisa dipakai oleh data yang lain
- Setelah itu dilakukan pergeseran ataupun pengaturan struktur data kembali

1	2	3	4	5	6	7	8	9	...
A	B	C	D	E	F	...	...	...	...

→ HAPUS

BILA RECORD D DIHAPUS, MAKA AKAN TERJADI PEMBACAAN DAN
PENULISAN ULANG RECORD E, F, DST

1	2	3	4	5	6	7	8	9	...
A	B	C	E	F	...	...	...	...	...

Organisasi File Sequential (5)

- Kadangkala delete dilakukan dengan hanya memberi tanda saja (tombstone / flag), tanpa dilakukan penghapusan ataupun pengaturan struktur datanya.

1	2	3	4	5	6	7	8	9	...
A	B	C	D	E	F	...	...	...	...

→ HAPUS

1	2	3	4	5	6	7	8	9	...
A	B	C	*	E	F	...	...	...	...

→ record yang sudah dihapus "Delete"

Organisasi File Index

Konsep Dasar

- Sebuah File Akan Terus Diakses Untuk Mencari Datanya (Fetch Data) Untuk Kemudian Data Tersebut Diambil Dari File (Retrieve Data)
- Untuk mencari data pada sebuah tabel dapat dilakukan secara sekuential. Namun cara pencarian ini akan memakan waktu lama jika file terdiri dari banyak record

SEKUENTIAL SEARCH MULAI DARI RECORD-1 ……. DST

0411500005	Ahmad Zaki	Cipondoh
0422500025	Sinta	Kebayoran Lama
0422500035	Indra Gunawan	Cipulir
0433500058	Bekti Sularso	Cidodol
0444500057	Tini Lestari	Cileduk

Organisasi File Index (2)

Terdapat 2 Macam Pengurutan :

- *Pengurutan secara indeks :*
  Berdasarkan urutan dari sebuah nilai

- *Pengurutan secara hash :*
  Berdasarkan fungsi hash yang digunakan

Organisasi File Index (3)

Tiap Pengurutan Memperhatikan Faktor :

- *TIPE AKSES :*
  Tipe akses dalam mencari record. Yang lebih dipilih tentunya yang lebih efisien
- *WAKTU AKSES :*
  Waktu yang dibutuhkan untuk menemukan sebuah record
- *WAKTU HAPUS :*
  Waktu yang dibutuhkan untuk menghapus sebuah item
- *RUANG SPASI :*
  Ruang tambahan yang diminta oleh stuktur index

Organisasi File Index (4)

Index Terurut

- Untuk mengatasi pencarian record dalam sebuah file secara acak, dapat digunakan struktur index.
- Tiap struktur index dihubungkan sesuai dengan key yang dicari (search key).
- Sebuah file dapat mempunyai beberapa file indeks, dengan search key yang ber beda-beda.
- Jika search key yang dipakai adalah primary key pada sebuah file master maka file index yang dibuat disebut primary indeks.

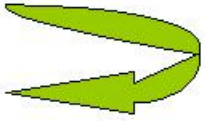
Organisasi File Index (5)

- Jika search key yang dipakai adalah bukan primary key pada sebuah file master maka file index yang dibuat disebut secondary index.
- File index terdiri dari nomor record serta field yang digunakan sebagai search key.
- Sebelum mencari data pada file master, data dicari terlebih dahulu pada file index, jika data tersebut ada, maka file index langsung menunjuk lokasi dari data tersebut pada file master.

Organisasi File Index (6)

INDEX YANG TERURUT

NIM	NO. REC
0233500058	1
0322500025	2
0411500005	3
0422500035	4
0444500057	5



FILE INDEX YANG BERJENIS PRIMARY INDEKS

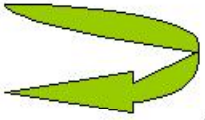
FILE MASTER DENGAN NIM SEBAGAI PK

NO. REC	NIM	NAMA	ALAMAT
1	0233500058	Ahmad Zaki	Cipondoh
2	0322500025	Sinta	Kebayoran Lama
3	0411500005	Indra Gunawan	Cipulir
4	0422500035	Bekti Sularso	Cidodol
5	0444500057	Tini Lestari	Cileduk

Organisasi File Index (7)

INDEX YANG TERURUT

NAMA	NO. REC
Ahmad Zaki	1
Bekti Sularso	4
Indra Gunawan	3
Sinta	2
Tini Lestari	5



FILE INDEX YANG BERJENIS SECONDARY INDEKS

FILE MASTER DENGAN NIM SEBAGAI PK

NO. REC	NIM	NAMA	ALAMAT
1	0411500005	Ahmad Zaki	Cipondoh
2	0322500025	Sinta	Kebayoran Lama
3	0422500035	Indra Gunawan	Cipulir
4	0233500058	Bekti Sularso	Cidodol
5	0444500057	Tini Lestari	Cileduk

Organisasi File Index (8)

Primary Index

- Pada file indeks yang menggunakan primary indeks, semua file master telah diurutkan berdasarkan primary key.
- File indeks juga telah diurutkan berdasarkan primary key semua file yang ada di atas disebut file indeks sekuential.
- Record indeks terdiri dari search key dan pointer yang menunjuk pada satu atau lebih record.
- Pointer terdiri dari identifier dari blok tempat record berada dalam disk.

Organisasi File Index (9)

2 TIPE PENGURUTAN INDEKS YANG DIGUNAKAN

- *Dense index :*
  Semua nilai dari search key muncul pada file index

- *Sparse index :*
  Hanya sebagian dari nilai search key yang muncul pada file index

Organisasi File Index (10)

DENSE INDEKS DAN SPARSE INDEKS

DENSE INDEX

FILE INDEX

CABANG	POINTER
BOGOR	
DAGO	
MALANG	
PADANG	

FILE MASTER

NO. REK	CABANG	JUMLAH
A-217	BOGOR	750
A-099	DAGO	450
A-101	DAGO	500
A-065	MALANG	300
A-135	MALANG	300
A-215	MALANG	700
A-201	PADANG	900
A-218	PADANG	700

Organisasi File Index (11)

DENSE INDEKS DAN SPARSE INDEKS

SPARSE INDEX

FILE INDEX

CABANG	POINTER
BOGOR	
MALANG	
PADANG	

FILE MASTER

NO. REK	CABANG	JUMLAH
A-217	BOGOR	750
A-099	DAGO	450
A-101	DAGO	500
A-065	MALANG	300
A-135	MALANG	300
A-215	MALANG	700
A-201	PADANG	900
A-218	PADANG	700

Organisasi File Index (12)

- **Kelebihan Dense Index :**
  Mencari lokasi record lebih cepat dibanding sparse index

- **Kekurangan Dense Index :**
  - Membutuhkan tempat indeks lebih besar dibanding sparse index
  - Jika file master berubah, maka file index juga harus dirubah (maintenace lebih sulit dibanding dengan sparse index))

Organisasi File Index (13)

- **Kelebihan Sparse Index :**
  - Membutuhkan tempat indeks lebih kecil dibanding dense index.
  - Maintenace relatif lebih mudah dibanding dengan dense index.

- **Kekurangan Sparse Index :**
  Mencari lokasi record lebih lambat dibanding dense index.

Organisasi File Index (14)

Multilevel Index

- Meskipun menggunakan sparse index, file index dapat menjadi besar sehingga proses pencarian tidak efisien.
- Misal, jika file master mempunyai record 100.000, dengan tiap blok menyimpan 10 record. Jika 1 record pada file index menyimpan 1 blok, maka file index mempunyai 10.000 record.

Organisasi File Index (15)

- File index yang terbentuk masih sangat besar untuk disimpan dalam sebuah disk. Jika file index tersebut tidak cukup dimuat di dalam main memory, maka pencarian data akan lambat.

- Untuk mengatasi masalah ini, maka dibuatlah sparse index pada primary index **(multilevel index)**.

Organisasi File Index (16)

MULTILEVEL INDEKS

FILE INDEX LEVEL 1

CABANG	POINTER
ACEH	
JAKARTA	
PADANG	

FILE INDEX LEVEL 2

CABANG	CABANG	POINTER
ACEH	BOGOR	
	BALIKPAPAN	
	CIAMIS	
	DAGO	
JAKARTA	JAKARTA	
	MALANG	
	MEDAN	
PADANG	PADANG	

FILE MASTER

NO. REK	CABANG	JUMLAH
000001	BOGOR	750
:	:	:
001000	DAGO	500
:	:	:
002500	JAKARTA	300
:	:	:
005000	MALANG	700
:	:	:
075000	PADANG	900
:	:	:
100000	PADANG	700

Organisasi File Index (17)

INDEX UPDATE

File indeks harus diupdate jika proses insert atau delete record terjadi.

Insert Record :

- *Pada dense indeks :*
  Jika nilai yang diinsert belum ada pada file indeks , maka nilai dari search key diinsert pada file indeks.

- *Pada sparse indeks :*
  Jika pada file index, nilai yang yang diinsert sudah ada, maka file index tidak usah dirubah, sebaliknya jika pada file index nilai yang diinsert tidak ada, maka file index harus dirubah.

Organisasi File Index (18)

Delete record (record harus dicari dulu):

- *Pada dense indeks :*
  Jika nilai yang didelete hanya satu pada file master, maka pada file indeks nilai yang didelete harus dihapus. Jika nilai yang didelete lebih dari satu pada file master, maka pada file indeks, nilai yang didelete tidak perlu dihapus.
- *Pada sparse indeks :*
  Jika pada file index, nilai yang yang didelete ada, maka nilai tersebut pada file index harus dihapus, sebaliknya jika pada file index nilai yang didelete tidak ada, maka file index tidak dirubah.

Organisasi File Index (18)

Delete record (record harus dicari dulu):

- *Pada dense indeks :*
  Jika nilai yang didelete hanya satu pada file master, maka pada file indeks nilai yang didelete harus dihapus. Jika nilai yang didelete lebih dari satu pada file master, maka pada file indeks, nilai yang didelete tidak perlu dihapus.
- *Pada sparse indeks :*
  Jika pada file index, nilai yang yang didelete ada, maka nilai tersebut pada file index harus dihapus, sebaliknya jika pada file index nilai yang didelete tidak ada, maka file index tidak dirubah.

Organisasi File Index (19)

SECONDARY INDEKS

SECONDARY INDEKS HARUS BERUPA DENSE INDEKS

FILE INDEX

JUMLAH	POINTER
300	
450	
500	
700	
750	
900	

FILE MASTER

NO. REK	CABANG	JUMLAH
A-217	BOGOR	750
A-099	DAGO	450
A-101	DAGO	500
A-065	MALANG	300
A-135	MALANG	300
A-215	MALANG	700
A-201	PADANG	900
A-218	PADANG	700

Organisasi File Hashing

Konsep Dasar

- Pada organisasi file hash, untuk mencari alamat dari record secara langsung dengan menghitung fungsi dari nilai search key dari record (memakai perhitungan matematis untuk menemukan alamat dari sebuah record).

- Agar dapat dilakukan direct access, key dari record dipakai sebagai alamat di dalam file.

Organisasi File Hashing (2)

Komponen Hashed File:

- *File Space :*
  Terbagi dalam slot-slot. Tiap slot menyimpan sebuah record.

- *Rumus :*
  Menghasilkan slot address, dihitung berdasarkan key dari sebuah record.

Organisasi File Hashing (3)

Overview Hashed File

- Berbasis kemampuan direct access ke dalam file dengan memanfaatkan relatif address.
- *Relatif Address :* Sebuah record dapat ditemukan hanya dengan memanggilnya lewat nomor urut record di dalam file.
- Harus ada rumus untuk mengubah key dari sebuah record menjadi nomor urut (kat -> key to address transformation).

Organisasi File Hashing (4)

KAT (Key To Address Transformation) :

- Tujuannya untuk menghasilkan slot number yang berbeda bagi tiap record dengan cara mengubah key menjadi relative address.

Hambatan KAT:

- Key umumnya sesuatu yang bersifat natural (nim / no_ktp / no_pegawai / dll)
- Natural key biasanya panjang (nim = 10 digit)

Organisasi File Hashing (5)

Persyaratan KAT:

- Ukuran key harus diperpendek agar sesuai dengan slot address (relative address).
- Slot address yang dihasilkan harus unik.
- Menggunakan algoritma tertentu pada saat menentukan/membuat KAT.

Organisasi File Hashing (6)

NIM	NAMA	SLOT ADDRESS
0011500001	BUDIMAN	1
0011500002	HERMAN	2
-	-	-
-	-	-
0011500105	ACHMAD	105
0011500106	ENDANG	106
0011500107	SEPHIA	107

KAT : 3 DIGIT TERAKHIR DARI NIM