



File Organization

IKI20410 – Basis Data

Aniati Murni

Fakultas Ilmu Komputer

Universitas Indonesia



TOPIK PEMBAHASAN

- ✂ Struktur Index File
 - ✂ Indexed Sequential File
 - ✂ Secondary Index
 - ✂ Multilevel Index
 - ✂ Index B+-Tree
-
- ✂ Elmasri Bab 6



Struktur Index File

- ✦ *Mempelajari suatu access structure yang disebut index*
- ✦ *Merupakan suatu upaya untuk mempercepat proses data retrieval*
- ✦ *Index merupakan nilai suatu atribut data yang digunakan untuk menentukan urutan (ordering) record dalam file*



Ordering Key

- ✦ Index terdiri dari 2 field: index entry dan pointer ke record / block / multilevel index
- ✦ Primary key: merupakan ordering key field, dimana setiap record mempunyai nilai primary key yang unik
- ✦ Clustering index: beberapa record bisa mempunyai nilai ordering key yang sama
- ✦ Secondary key: digunakan untuk menyusun record menurut nilai atribut-atribut tertentu
- ✦ Multilevel index: penggunaan index bertingkat untuk fast search dengan mengurangi jumlah index yang dicari
- ✦ Dense index: mempunyai index entry untuk semua nilai search key
- ✦ Sparse key: hanya tersedia index entry untuk beberapa nilai search key



Indexed Sequential File (1)

Karakteristik:

- ✎ Ordered data file melalui primary index
- ✎ Record dapat di-akses secara sequential atau secara individual menggunakan index
- ✎ Mempunyai primary storage area dengan index overflow area yang terpisah. Contoh: IBM's Indexed Sequential Access Method (ISAM)
- ✎ Setiap struktur file indeks terkait dengan search key tertentu
- ✎ Primary/Clustering Index: search key menentukan urutan data dalam file
- ✎ Secondary Index: search key tidak menentukan urutan data dalam file

Indexed Sequential File (2)

| Index | | sno | sname | scity |
|-------|---|------|-----------|---------|
| S001 | → | S001 | PT Bijak | Jakarta |
| S050 | → | S050 | PT Barata | Bandung |
| S060 | → | S060 | PT Kemala | Jakarta |
| S070 | → | S070 | PT Aman | Jakarta |
| S100 | → | S100 | PT Citra | Jogya |



Indexed Sequential File (3)

Kemampuan:

- ✂ Direct Access
- ✂ Proses Insert dan Delete memerlukan reorganisasi Index File dan File Data
- ✂ Perlu tambahan / overhead space untuk Index File
- ✂ Index File dapat berupa Dense Index atau Sparse Index

Secondary Index (1)

- ✖ Juga merupakan ordered index file seperti primary index file
- ✖ Atribut yang digunakan sebagai index tidak terurut pada file data
- ✖ Indeks tidak perlu terdiri dari nilai yang unik, ada 3 cara menanganinya:
 - Dense index, membolehkan duplicate key values pada index
 - Distinct index, menggunakan tempat yang multivalues
 - Distinct index, menggunakan tempat pointer yang multivalues
- ✖ Mempercepat query berdasarkan atribut yang bukan primary key

Indexed Sequential File (2)

| Index | | sno | sname | scity |
|---------|---|------|-----------|---------|
| Jakarta | → | S001 | PT Bijak | Jakarta |
| Bandung | → | S050 | PT Barata | Bandung |
| Jakarta | → | S060 | PT Kemala | Jakarta |
| Jakarta | → | S070 | PT Aman | Jakarta |
| Jogya | → | S100 | PT Citra | Jogya |



Multilevel Index

- ✦ Digunakan bila index itu sendiri sudah terlalu besar ukurannya
- ✦ IBM's ISAM merupakan 2-level index structure
- ✦ Dalam praktek, biasanya tidak melebihi 3-level
- ✦ B-tree merupakan contoh penggunaan multilevel full index



Index B+-Tree

Karakteristik:

- ✦ Struktur data berbentuk tree yang Balanced
- ✦ Struktur fleksibel: dapat menghilangkan overhead reorganisasi data saat terjadi pertumbuhan / perubahan data
- ✦ Waktu akses relatif cepat
- ✦ Berupa file index multilevel
- ✦ Perlu tambahan / space overhead untuk file index
- ✦ Perlu tambahan / time overhead untuk perawatan file index
- ✦ Banyak digunakan pada penggunaan Relational Database Management System



Index B+-Tree

Kapan dipakai?

- ✂ Bila perlu banyak akses berdasarkan *exact key match*, *pattern matching*, *range of values*, *part key specification*
- ✂ Bersifat dinamis – menyesuaikan dengan pertumbuhan files
- ✂ Memelihara urutan access key, sehingga pengambilan secara berurutan menjadi efisien
- ✂ Kalau proses update tidak sering, struktur ISAM akan lebih efisien, karena jumlah level index ISAM lebih kecil.