ORGANISASI BERKAS INDEKS SEKUENSIAL

PENGERTIAN BERKAS INDEKS SEKUENSIAL

Salah satu cara yang paling efektif untuk mengorganisasi kumpulan record-record yang membutuhkan akses record secara sekuensial maupun akses record secara individu berdasarkan nilai key adalah organisasi berkas indeks sekuensial.

Jadi berkas indeks sekuensial merupakan kombinasi dari berkas sekuensial dan berkas relatif.

Adapun jenis akses yang diperbolehkan, yaitu:

- Akses Sequential
- Akses Direct

Sedangkan jenis prosesnya adalah:

- Batch
- Interactive

Struktur berkas Index Sequential

- Index → Binary Search Tree
- Data → Sequential

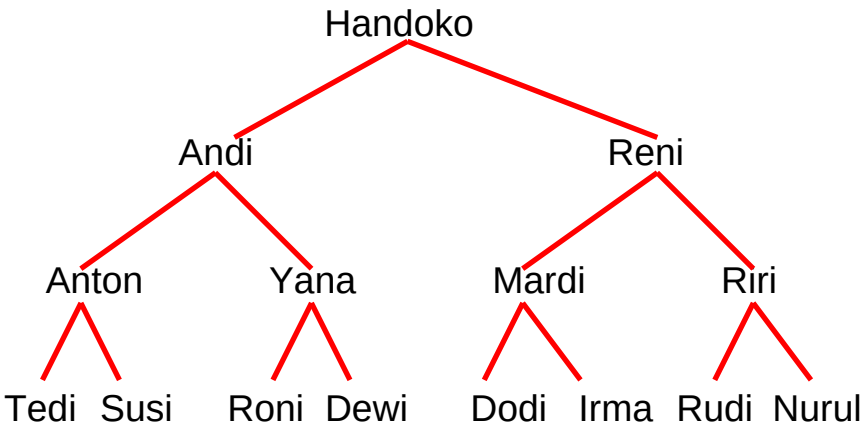
Indeksnya digunakan untuk melayani sebuah permintaan untuk mengakses sebuah record tertentu, sedangkan berkas data sekuensial digunakan untuk mendukung akses sekuensial terhadap seluruh kumpulan record-record.

STRUKTUR POHON

Sebuah pohon (*tree*) adalah struktur dari sekumpulan elemen, dengan salah satu elemennya merupakan akarnya atau root, dan sisanya yang lain merupakan bagian-bagian pohon yang terorganisasi dalam susunan berhirarki, dengan root sebagai puncaknya.

Contoh umum dimana struktur pohon sering ditemukan adalah pada penyusunan silsilah keluarga, hirarki suatu organisasi, daftar isi suatu buku dan lain sebagainya.

Contoh:

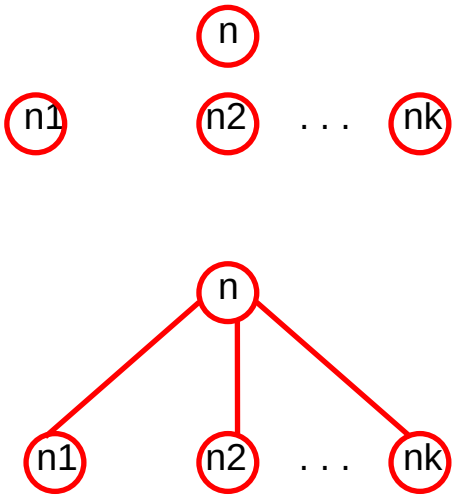


Gambar 1. Silsilah Keluarga

Akar pohon *(root)* adalah Handoko*.*

Secara rekursif suatu struktur pohon dapat didefinisikan sebagai berikut:

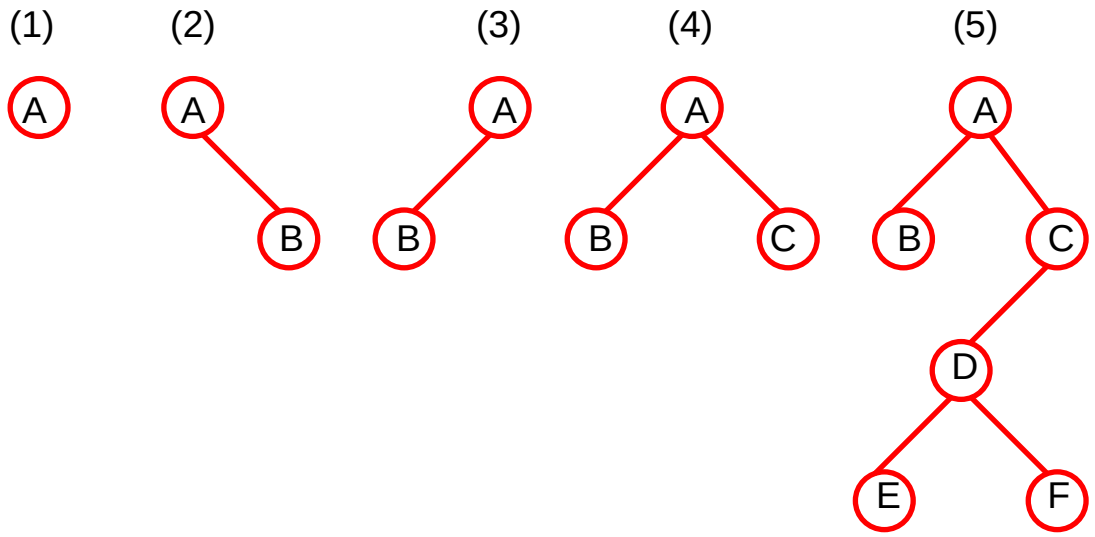
- Sebuah simpul tunggal adalah sebuah pohon.
- Bila terdapat simpul n, dan beberapa sub-pohon T1,T2,...,Tk, yang tidak saling berhubungan, yang masing-masing akarnya adalah n1,n2,..., nk, dari simpul/sub pohon ini dapat dibuat sebuah pohon baru dengan n sebagai akar dari simpul-simpul n1,n2,...,nk.



Gambar 2. Definisi Struktur Pohon

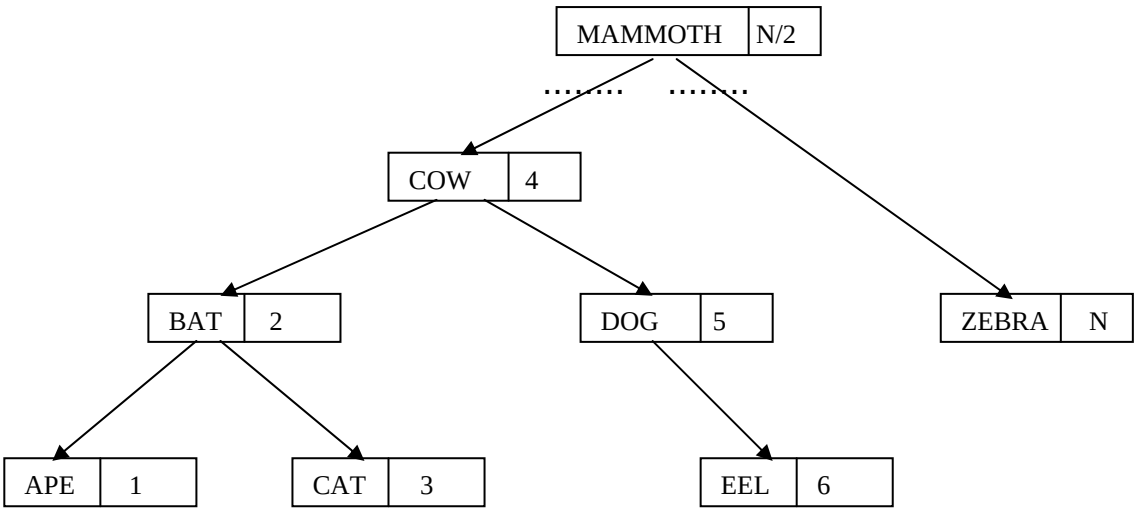
POHON BINER

Salah satu tipe pohon yang paling banyak dipelajari adalah pohon biner. Pohon Biner adalah pohon yang setiap simpulnya memiliki paling banyak dua buah cabang/anak.



Gambar 3. Beberapa Contoh Pohon Biner

Perhatikan gambar 4 berikut:



Posisi	Sequential Data File
1	APE
2	BAT
3	CAT
4	COW
5	DOG
6	EEL
:	:
N	ZEBRA

Gambar 4. Binary Search Tree dan Sequential File

Pada gambar 4 memperlihatkan struktur berkas indeks sekuensial dengan sebuah indeks berikut pointer yang menuju ke berkas data sekuensial. Pada contoh gambar tersebut, indeksnya disusun berdasarkan binary search tree. Indeksnya digunakan untuk melayani sebuah permintaan untuk mengakses sebuah record tertentu, sedangkan berkas data sekeunsial digunakan untuk mendukung akses sekuensial terhadap seluruh kumpulan record-record.

IMPLEMENTASI ORGANISASI BERKAS INDEKS SEKUENSIAL

Ada 2 pendekatan dasar untuk mengimplementasikan konsep dari organisasi berkas indeks sekuensial:

- Blok Indeks dan Data (Dinamik)
- Prime dan Overflow Data Area (Statik)

Kedua pendekatan tersebut mengunakan sebuah bagian indeks dan sebuah bagian data, dimana masing-masing menempati berkas yang terpisah.

Alasannya:

Karena mereka diimplementasikan pada organisasi internal yang berbeda. Masing-masing berkas tersebut harus menempati pada alat penyimpan yang bersifat Direct Access Storage Device (DASD).

BLOK INDEKS DAN DATA

Pada pendekatan ini berkas indeks dan berkas data diorganisasikan dalam blok. Berkas indeks mempunyai struktur tree, sedangkan berkas data mempunyai struktur sekuensial dengan ruang bebas yang didistribusikan antar populasi record.

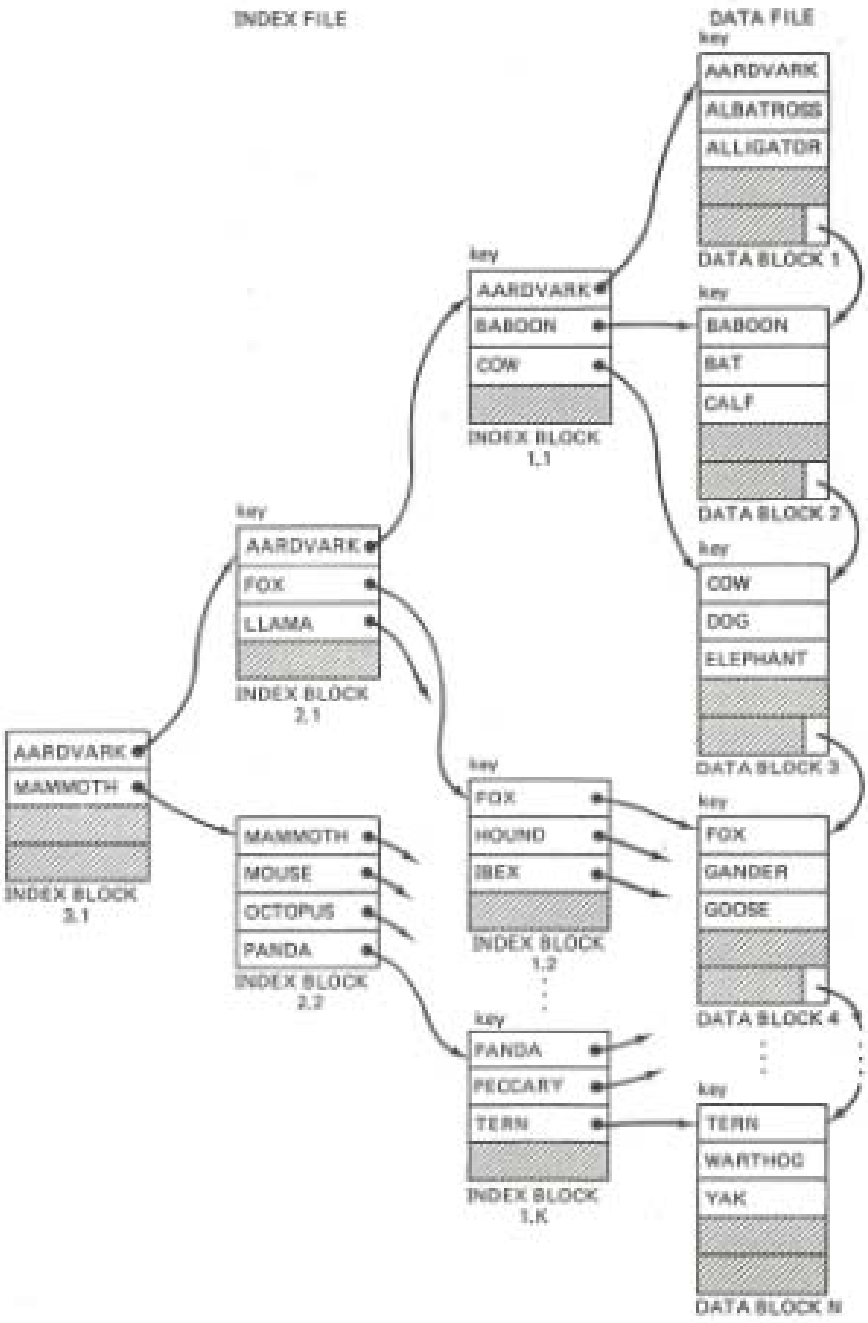


Figure 15–3 Example indexed sequential file structured using a B⁺-tree.

Gambar 5. Indexed Sequential File Structured dengan B-Tree

Pada gambar 5 ada N blok data dan 3 tingkat dari indeks. Setiap entry pada indeks mempunyai bentuk (nilai key terendah, pointer), dimana pointer menunjuk pada blok yang lain, dengan nilai key-nya sebagai nilai key terendah. Setiap tingkat dari blok indeks menunjuk seluruh blok, kecuali blok indeks pada tingkat terendah yang menunjuk ke blok data.

Jika sebuah permintaan untuk mengakses record tertentu, misal kita ingin mengakses dengan nilai key BAT, indeks dengan tingkat tertinggi (dalam hal ini blok indeks 3-1) yang pertama yang akan dicari pada contoh ini, pointer dari AARDVARK menunjuk blok indeks 2-1. Pointer yang ditunjuk pada kotak tersebut adalah pointer yang berisikan AARDVARK, yang akan menunjuk ke blok indeks 1-1. Pointer berikutnya yang akan ditunjuk adalah pointer yang berisi BABOON, yang selanjutnya akan menunjuk blok data 2. Blok data ini akan mencari untuk record dengan key tujuan, yaitu BAT, dimana pada blok ini record tersebut ditemukan.

Permintaan untuk akses data dalam urutan sekuensial dilayani dengan mengakses blok data dalam urutan sekuensial. Sebagai catatan blok data merupakan urutan secara logik dan bukan urutan secara fisik. Dalam hal ini, blok data harus dihubungkan secara bersama dalam urutan secara logik, seperti terlihat pada gambar 5.

Misal:

Setiap blok data mempunyai ruang yang cukup untuk menampung 5 record dan setiap blok indeks mempunyai ruang yang cukup untuk menyimpan 4 pasang (nilai key, pointer).

Parameter ini biasanya sudah dilengkapi dengan rutin dukungan sistem manajemen data, pada saat berkas binatang ini dibentuk.

Jika kita menginginkan penyisipan maupun penghapusan terhadap isi berkas, maka blok indeks dan blok data akan dibuat dengan sejumlah ruang bebas, yang biasanya disebut sebagai padding dan pada gambar ditunjukkan sebagai irisan.

Permintaan:

INSERT APE
INSERT AIREDALE

Hanya blok data 1 yang digunakan dan hasilnya ditunjukkan pada gambar 6.

Figure 15–4 Data block 1 of Fig. 15–3 after inserting APE and AIREDALE

Gambar 6. Hasil Insert APE dan AIREDALE

Entry pada blok harus diletakan berdasarkan urutan sekuensial ascending.

Permintaan:

INSERT ARMADILLO

Pencarian dari struktur indeks menyatakan bahwa ARMADILLO seharusnya menempati blok data 1, tetapi blok tersebut sudah penuh.

Untuk mengatasi keadaan tersebut, blok data 1 dipecah dengan memodifikasi blok indeks 1-1.

Separuh dari isi blok data, tetap menempati blok tersebut dan separuhnya lagi dipindahkan ke blok yang baru dibuat, yaitu blok data 1A. Hasilnya ditunjukkan pada gambar 7.

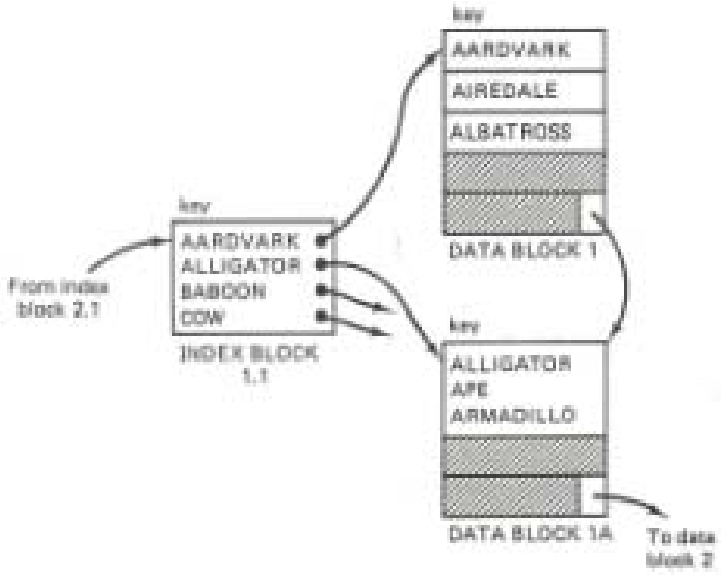


Figure 15–5 Splitting to accommodate insertion of ARMADILLO into data block 1 of Fig. 15–4.

Gambar 7. Hasil Insert ARMADILLO

Permintaan:

INSERT CAT
INSERT BEAR
INSERT BOBCAT

Akan mengisi blok data 2, tetapi blok data tersebut harus dipecah menjadi blok data 2 dan 2A.

Blok indeks 1-1 sudah penuh dan tidak dapat memuat pointer pada blok data 2A, sehingga inipun harus dipecah, dengan cara mengubah penafsiran indeks tingkat 2.

Jika blok indeks pada tingkat paling tinggi (dalam hal ini indeks tingkat 3) sudah penuh, maka harus dipecah, sehingga sebuah indeks tingkat yang baru akan ditambahkan pada indeks tree.
Maka seluruh pencarian langsung, memerlukan pengaksesan empat blok indeks dan sebuah blok data. Hasilnya ditunjukkan pada gambar 8.

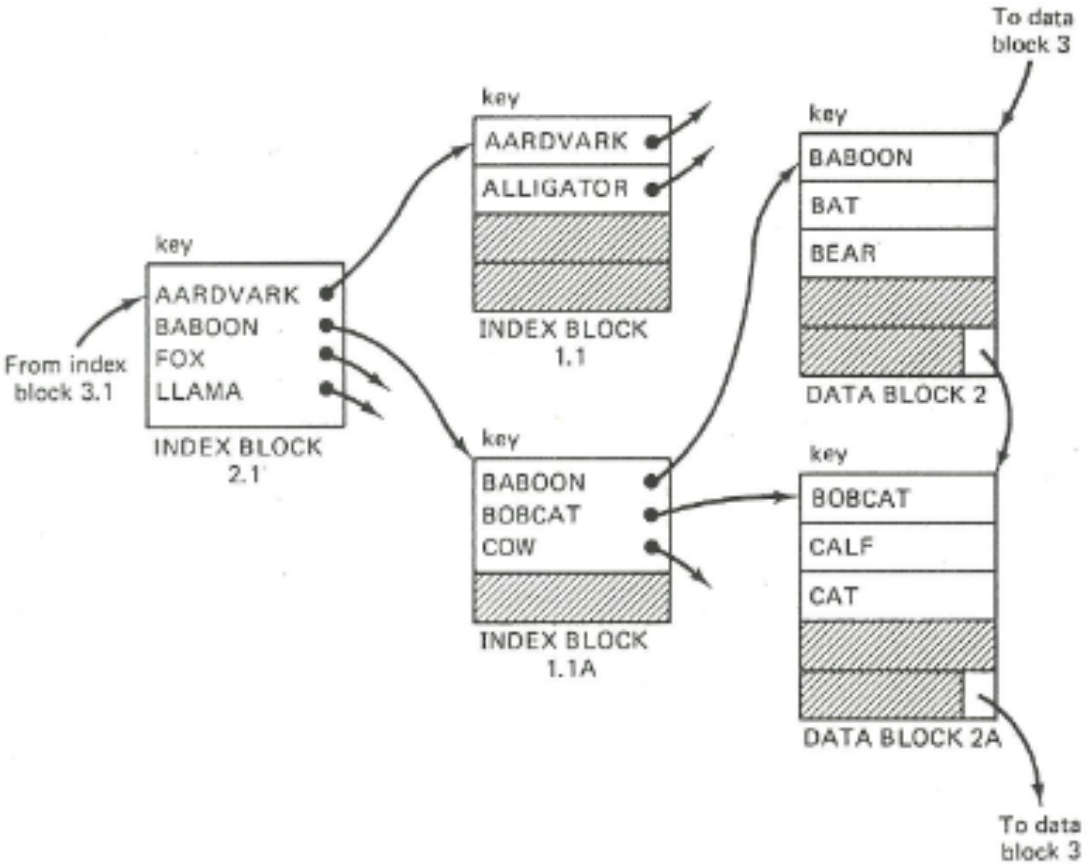


Figure 15–6 Splitting to accommodate insertion of CAT, BEAR, and BOBCAT into data block 2 of Fig. 15–3.

Gambar 8. Hasil Insert CAT, BEAR, dan BOBCAT

PRIME DAN OVERFLOW DATA AREA

Pendekatan lain untuk mengimplementasikan berkas indek sekuensial adalah berdasarkan struktur indek dimana struktur indek ini lebih ditekankan pada karakteristik fisik dari penyimpanan, dibandingkan dengan distribusi secara logik dari nilai key.

Indeksnya ada beberapa tingkat, misalnya tingkat cylinder index dan tingkat track index. Berkas datanya secara umum diimplementasikan sebagai 2 berkas, yaitu prime area dan overflow area.

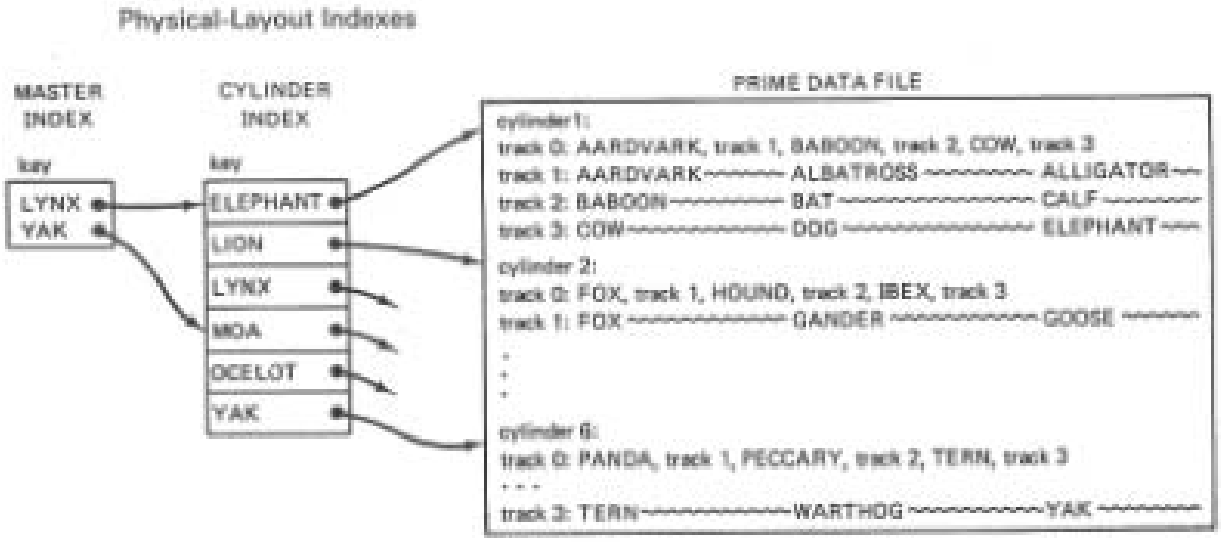


Figure 15–7 Example animal file structured using physical-layout index approach.

Gambar 9. Struktur File Binatang menggunakan pendekatan Physical-Layout Index

Misal setiap cylinder dari alat penyimpanan mempunyai 4 track.
Pada berkas binatang ada 6 cylinder yang dialokasikan pada prime data area. Track pertama (nomor 0) dari setiap cylinder berisi sebuah indeks pada record key dalam cylinder tersebut.

Entry pada indeks ini adalah dalam bentuk:

nilai key terendah, nomor track

Dalam sebuah track data, tracknya disimpan secara urut berdasarkan nilai key.

Tingkat pertama dari indeks dalam berkas indeks dinamakan *master index*.

Entry pada indeks ini adalah dalam bentuk:

nilai key tertinggi, pointer

Tingkat kedua dari indeks dinamakan *cylinder index.*
Indeks ini berisi pointer pada berkas prime data dan entry-nya dalam bentuk:

nilai key tertinggi, nomor cylinder

Jika sebuah permintaan untuk mengakses record tertentu, misal kita akan mengakses dengan nilai key BAT, pertama akan dicari pada master index. Karena BAT ada di depan LYNX, maka pointer dari LYNX akan menunjuk ke cylinder index. Karena BAT ada di depan ELEPHANT, maka pointer dari ELEPHANT akan menunjuk ke track 0 dari cylinder 1. Karena BAT ada di belakang BABOON dan di depan COW, maka pointer dari BABOON akan menunjuk ke track 2, yang mencari secara sekuensial, sampai BAT ditemukan.

Permintaan untuk mengakses data secara sekuensial akan dilayani dengan mengakses cylinder dan track dari berkas data prime secara urut.

Misal setiap track dari berkas prime data mempunyai ruang yang cukup untuk menampung 5 record (jika penyisipan dan penghapusan terhadap berkas dilakukan, maka akan dibentuk padding).

Permintaan:

INSERT APE
INSERT AIREDALE

Akan mudah dilayani. Hanya track data 1 dari cylinder 1 yang akan digunakan dan hasilnya ditunjukkan pada gambar 10.

Figure 15–8 Cylinder 1 of Fig. 15–7 after inserting APE and AIREDALE.

Gambar 10. Hasil Insert APE dan AIREDALE

Permintaan:

INSERT ARMADILLO

Agak sulit ditangani. Pencarian struktur indeks menyatakan bahwa ARMADILLO seharusnya menempati track 1 dari cylinder 1, tetapi track tersebut sudah penuh.

Untuk mengatasi keadaan tersebut diperlukan overflow data area. Overflow data area ini merupakan berkas yang terpisah dari prime data area, tetapi overflow area ini ditunjukan oleh entry prime data area. Hasilnya ditunjukkan pada gambar 11.

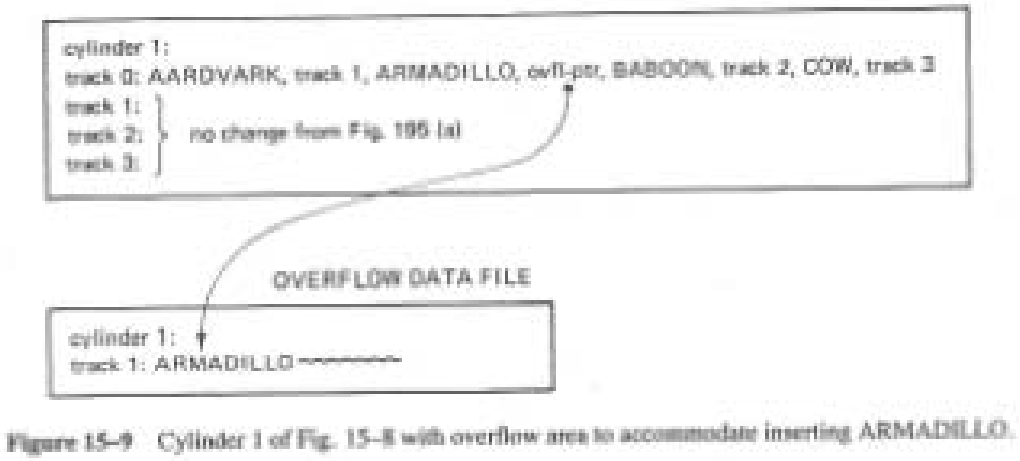


Figure 15–9 Cylinder 1 of Fig. 15–8 with overflow area to accommodate inserting ARMADILLO.

Gambar 11. Hasil Insert ARMADILLO

Karena ARMADILLO seharusnya berada setelah kelima entry pada track 1 dari cylinder 1, tetapi karena track ini sudah penuh, maka ARMADILLO dipindahkan ke overflow data area. Indeks track dari cylinder 1 harus dimodifikasi untuk memperlihatkan bahwa ada sebuah record pada overflow area yang secara logik seharusnya menempati pada akhir dari track 1, sehingga penambahan dari entry itu adalah:

<ARMADILLO,ovfl-ptr>

Dengan ovfl-ptr adalah:

<cylinder, track, record>

Permintaan:

INSERT CAT
INSERT BEAR
INSERT BOBCAT

Akan mengisi track 2 dari cylinder 1 pada prime data area, tetapi pengisian tersebut mengakibatkan penggunaan overflow area. Perhatikan CAT dipindahkan ke overflow area, karena entry pada prime track tidak hanya harus dalam urutan, tetapi juga entry tersebut harus mendahului suatu entry overflow dari track tersebut.
Hasilnya ditunjukkan pada gambar 12.

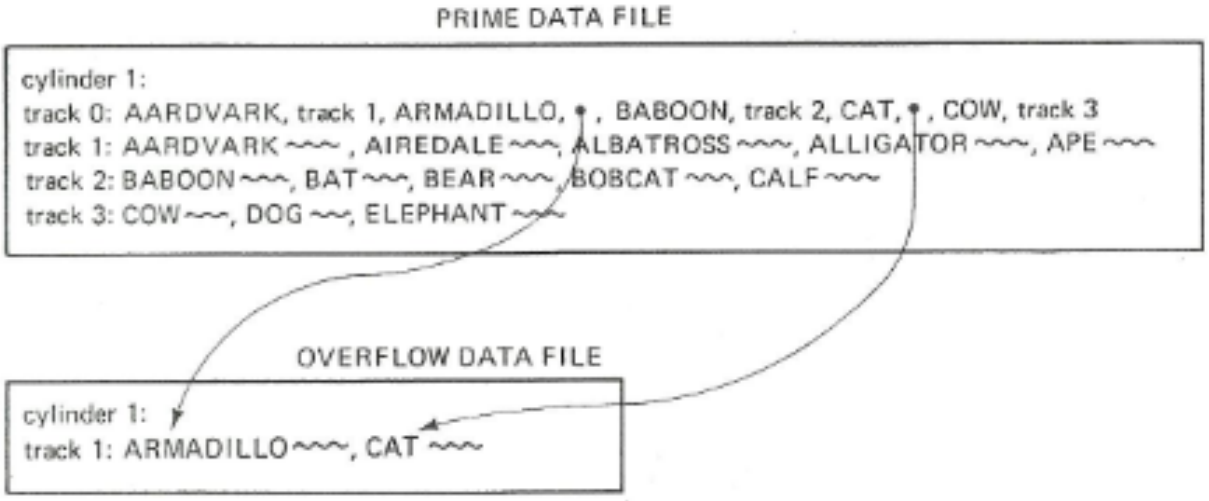


Figure 15–10 Cylinder 1 of Fig. 15–9 with overflow area after inserting CAT, BEAR, and BOBCAT.

Gambar 12. Hasil Insert CAT, BEAR dan BOBCAT

Permintaan:

INSERT ANT

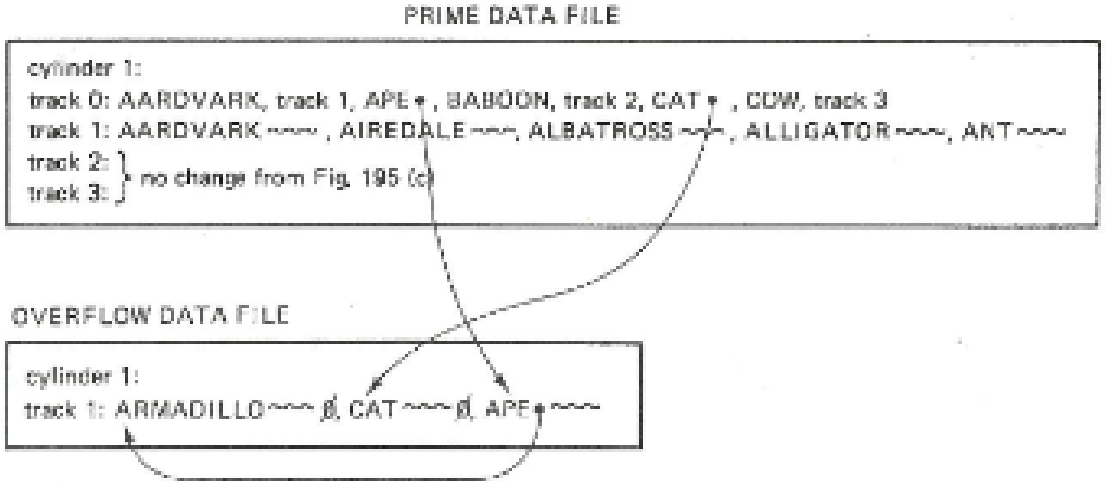


Figure 15–11 Cylinder 1 of Fig. 15–10 with overflow area after inserting ANT.

Gambar 13. Hasil Insert ANT

Deklarasi Berkas Indeks Sekuensial Dalam Bahasa COBOL

```
SELECT filename ASSIGN TO implementor-name[,implementor-name2]
    [ RESERVE integer [ AREA | AREA ] ]
    ORGANIZATION IS INDEXED
    [ ACCESS MODE IS { SEQUENTIAL | RANDOM | DYNAMIC } ]
    RECORD KEY IS dataname-1
    [FILE STATUS IS dataname-2].
```

LATIHAN

1. Dari contoh soal pada gambar 8, tunjukkan dengan gambar Blok Indeks dan Data, bila ada permintaan:

   INSERT DRAGON
   INSERT CROCODILE
   INSERT DEER

2. Dari contoh soal pada gambar 13, tunjukkan dengan gambar Prime dan Overflow Data Area, bila ada permintaan:

   INSERT BEE