

---

# File Sequential Berindeks

---

---

# File Sequential Berindeks

- Pengertian
- Struktur File Sequential Berindeks
- Indeks dan Parameter Indeks
- Algoritma Push-Trough
- Parameter Performansi File

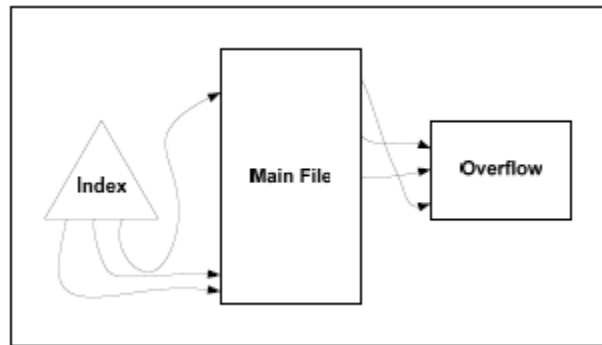
# File Sequential Berindeks

- File sequential berindeks → menambahkan fasilitas sebuah indeks tunggal terhadap file sequens sehingga pencarian dan pembaharuan rekord berdasarkan atribut yang diindeks-kan menjadi lebih efisien
- Ditujukan untuk keperluan akses data secara langsung dan juga sequential

# Struktur File Sequential Berindeks

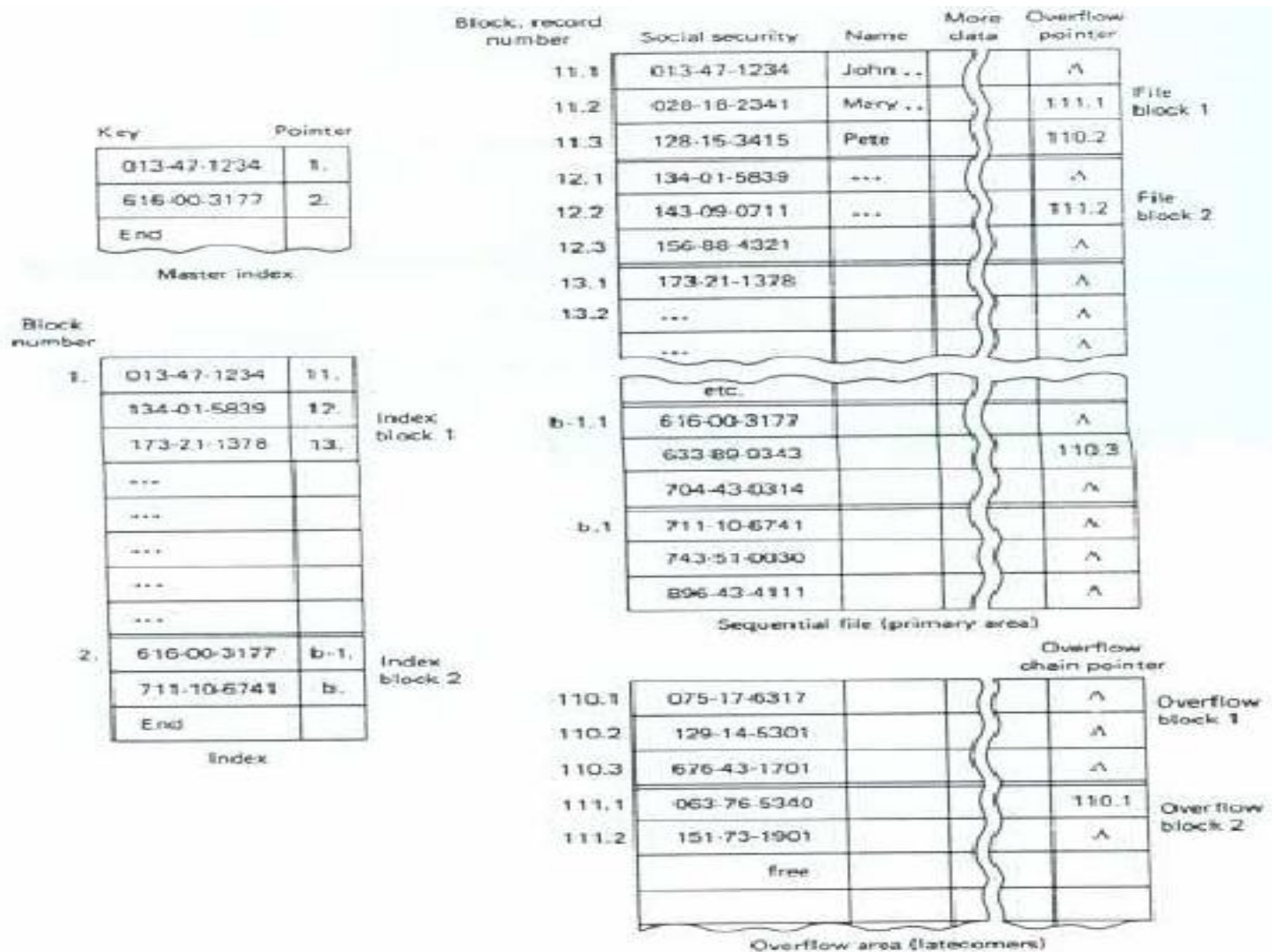
Komponen File Sequential Berindeks :

- Main/Primary File (File Utama)
- Indeks
- Overflow Area



# Struktur File Sequential Berindeks

## ■ Contoh ilustrasi :



# Indeks dan Parameter Indeks

- Indeks berisi pasangan nilai atribut dengan pointer acuan
- File Indeks adalah kumpulan isian indeks untuk mengacu record di file utama
- Jenis Indeks ada 2 yaitu :
  - a. Indeks Statis → indeks yang diciptakan saat reorganisasi dan tidak berubah karena pembaharuan. Record baru ditempatkan di file overflow dan dikaitkan ke record sebelumnya
  - b. Indeks Dinamis → Pembaharuan dilakukan pada file utama bukan pada overflow

# Indeks dan Parameter Indeks

- Ada dua tipe pengaksesan pada indeks :
  - a. Block Anchor → Indeks tidak menunjuk langsung ke record tetapi ke blok yang memuat record
  - b. Record Anchor → Indeks menunjuk langsung ke record
- Jika terdapat  $n$  rekord data, maka banyak entri indeks diperlukan =  $n/Bfr$
- Tiap entri berukuran  $V+P$  byte ( $V$ =atribut kunci,  $P$  = TID Pointer)

# Parameter indeks

- Fanout Ratio : Perbandingan antara block size  $B$  dengan besar space yang dibutuhkan untuk menyimpan sebuah entri indeks

$$y = \left\lfloor \frac{B}{V + P} \right\rfloor$$

- Height : Jumlah level indeks yang diperlukan untuk mengakses seluruh rekord pada file

$$X = \lceil y \log \lceil n / Bfr \rceil \rceil$$

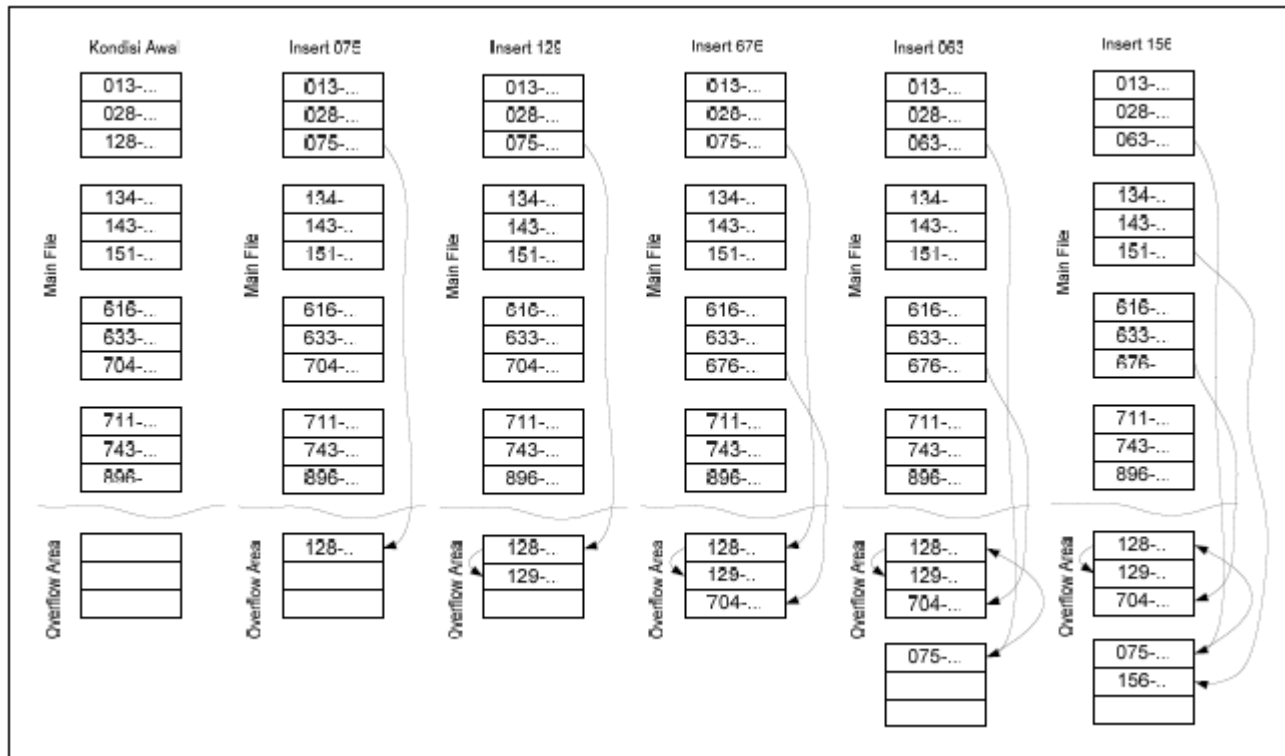


---

# Algoritma Push-Trough

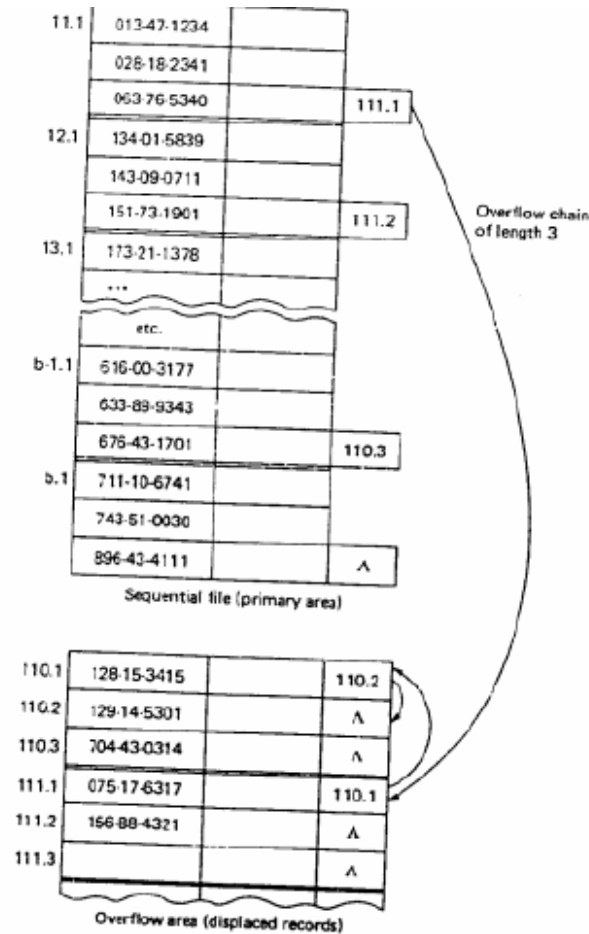
- Rekord baru diinsert di posisi yang sesuai
- Rekord sisa setelah rekord baru di-push hingga akhir block
- Jika melebihi akhir block, push ke overflow area

# Contoh Implementasi Algoritma Push-Trough



# Contoh Implementasi Algoritma Push-Trough

Hasil akhir :



# Parameter Performansi File

- Ukuran Rekord (R)  
 $R = aV + P$   
a : jumlah atribut  
V : panjang rata-rata nilai atribut  
P : ukuran pointer
- Jumlah rekord file sekuensial berindeks (n)  
 $n = nm + o'$   
nm : jumlah rekord di file utama  
o' : jumlah rekord di overflow area

# Parameter Performansi File

Jumlah entri indeks level pertama ( $i_1$ )

$$i_1 = n_m / Bfr$$

Jumlah entri indeks level berikutnya :

$$i_{level} = \left\lceil \frac{i_{level-1}}{y} \right\rceil$$

Jumlah block yang dibutuhkan pada suatu

level :  $b_{level} = \left\lceil \frac{i_{level}}{y} \right\rceil = i_{level} + 1$

## Total ruang Indeks

$$SI = (b_{i1} + b_{i2} + \dots + 1)B = (i_1 + i_2 + \dots + 1)B$$

## Total ruang per rekord

$$R_{total} = \frac{nm.R + oR + SI}{n}$$

# Parameter Performansi File

- Waktu Fetch record tertentu ( $T_F$ ), ditentukan oleh :
  - Akses indeks,  $T_{An} = x(s+r+btt)$
  - Akses main file,  $T_{Aq} = s+r+btt$
  - Akses overflow,  $TAo = o'/n(s+r+btt)$dengan  $\theta$  = jumlah record yang diinsertkan

Jika overflow masih kosong, maka

$$\begin{aligned}TF &= T_{An} + T_{Aq} = x(s+r+btt) + s+r+btt \\ &= (x+1)(s+r+btt)\end{aligned}$$

Jika overflow telah ada isinya, maka

$$\begin{aligned}TF &= T_{An} + T_{Aq} + TA\theta \\ &= x(s+r+btt) + s+r+btt + \theta'/n (s+r+btt) \\ &= (x+1+o'/n)(s+r+btt)\end{aligned}$$

# Parameter Performansi File

## ■ Waktu Pembacaan Next Record ( $T_N$ )

- Jika next rekord di main file :

$$T_N = (1/Bfr)(s+r+btt)$$

- Jika next rekord di overflow

$$T_{OF} = (\theta'/Nm)(r+btt)$$

$$T_N = (1/Bfr)(s+r+btt) + (\theta'/Nm)(r+btt)$$

## ■ Waktu Penyisipan Rekord ( $T_I$ )

Proses :

- Baca dan tulis ulang rekord predesesor
- Baca dan tulis ulang block overflow untuk rekord yang dipush atau diinsert

$$\begin{aligned} \rightarrow T_I &= T_F + T_{RW} + r + btt + T_{RW} \\ &= T_F + 5r + btt \end{aligned}$$

# Parameter Performansi File

- Waktu Update Rekord ( $T_U$ )

- Update terhadap bukan kunci

$$T_U = T_F + T_{RW}$$

- Update terhadap kunci

Proses :

- Rekord yang diubah ditandai untuk dihapus
- Rekord baru disisipkan

$$T_U = T_F + T_{RW} + T_I$$



# Parameter Performansi File

- Waktu Pembacaan seluruh rekord  
Proses : file dibaca serial mulai dari awal hingga akhir file, dengan mengikuti rantai overflow jika ada.  
→ 
$$T_X = T_F + (n + o' - 1)T_N$$
$$= [(n + o' \cdot B_{fr}) / B_{fr}] (r + b_{tt})$$
  
- Waktu Reorganisasi  
Reorganisasi dilakukan jika :
  - Telah terjadi 'overflow' pada overflow area
  - Telah terjadi rantai yang panjang
  - Proses akses data lambat
  
- Proses Reorganisasi :
  - Baca file secara serial
  - tulis kembali ke file baru tanpa menyertakan rekord yang ditandai untuk didelete
  - Buat indeks
  - Buat indeks untuk tiap block indeks hingga block indeks terakhir selesai dibuat
  - Hapus main file dan indeks lama

# Parameter Performansi File

- Waktu Reorganisasi ( $T_Y$ )

$$= [(n+o'Bfr)/Bfr](r+btt) + (n+o'-d)(R/t') + (SI/t')$$

dengan  $n$  = jumlah rekord pada main file

$o'$  = jumlah rekord pada overflow area

$d$  = jumlah rekord yang ditandai delete

$[(n+o'Bfr)/Bfr](r+btt) \rightarrow$  waktu pembacaan main file secara serial

$(n+o'-d)(R/t') \rightarrow$  waktu penulisan kembali ke file baru

$(SI/t') \rightarrow$  waktu pembantuan indeks

# Tugas

Diketahui File sequensial berindeks:

## Parameter Hardisk

- Putaran disk = 8000 rpm
- Seek time = 5 ms
- Transfer rate = 2048 byte/ms
- TRW =  $2r$

## Parameter Penyimpanan

- Ukuran blok = 4096 byte
- Ukuran Pointer blok = 8 byte
- IBG = 1024 byte

## Parameter File

- Jumlah rekord di file = 1000000 rekord
- Jumlah field = 8 field
- Panjang nilai = 25 byte

## Parameter indeks

- Ukuran indeks  $(V+P) = 20$  byte

## Parameter Reorganisasi

- File log transaksi = 5000 rekord

## Parameter Pemrosesan

- Waktu pemrosesan block = 2 ms

## Hitung :

fan-out ( $y$ ), height ( $x$ ),  $R_{Total}$ ,  $T_F$ ,  $T_N$ ,  
 $T_I$ ,  $T_U$ ,  $T_x$ ,  $T_y$  jika metode bloking :  
Variable length spanned blocking !

# Tugas

Diketahui File sequensial berindeks:

## Parameter Hardisk

- Putaran disk = 8000 rpm
- Seek time = 5 ms
- Transfer rate = 2048 byte/ms
- TRW =  $2r$

## Parameter Penyimpanan

- Ukuran blok = 4096 byte
- Ukuran Pointer blok = 8 byte
- IBG = 1024 byte

## Parameter File

- Jumlah rekord di file = 1000000 rekord
- Jumlah field = 8 field
- Panjang nilai = 25 byte

## Parameter indeks

- Ukuran indeks  $(V+P) = 20$  byte

## Parameter Reorganisasi

- File log transaksi = 5000 rekord

## Parameter Pemrosesan

- Waktu pemrosesan block = 2 ms

## Hitung :

fan-out ( $y$ ), height ( $x$ ),  $R_{Total}$ ,  $T_F$ ,  $T_N$ ,  
 $T_I$ ,  $T_U$ ,  $T_x$ ,  $T_y$  jika metode bloking :  
Variable length spanned blocking !