

ORGANISASI BERKAS RELATIF



Pengertian Berkas Relatif

- ▶ Suatu cara yang efektif dalam mengorganisasi sekumpulan record yang membutuhkan akses sebuah record dengan cepat.
- ▶ Hubungan ini dinyatakan sebagai R, yang merupakan fungsi pemetaan :

$R(\text{NILAI KEY}) \longrightarrow \text{ADDRESS}$

dari nilai key ke address dalam penyimpanan sekunder.



Proses Berkas Relatif

- ▶ Pada waktu sebuah record ditulis kedalam berkas relatif, fungsi pemetaan R digunakan untuk menerjemahkan NILAI KEY DARI RECORD menjadi ADDRESS, dimana record tersebut disimpan.
- ▶ Berkas relatif harus disimpan dalam media DASD, seperti magnetic disk atau drum.



(SASD, seperti magnetic tape atau pada DASD, seperti magnetic disk)

Catatan :

- ▶ Kita tidak perlu mengakses semua record master file, cukup mengakses langsung record yang dikehendaki.
- ▶ Record dari berkas relatif dapat di update langsung tanpa perlu merekam kembali semua record.
- ▶ Keuntungan dari berkas relatif ini adalah kemampuan mengakses record secara langsung. Sebuah record dapat di retrieve, insert, modifikasi atau di delete; tanpa mempengaruhi record lain dalam berkas yang sama.



Ada 3 teknik dasar yang digunakan untuk menyatakan fungsi pemetaan R, dimana

$R(\text{NILAI KEY}) \longrightarrow \text{ADDRESS},$

yaitu :

- ▶ Teknik Pemetaan Langsung (Direct Mapping)
- ▶ Teknik Pencarian Tabel (Directory Look Up)
- ▶ Teknik Kalkulasi Alamat



Teknik Pemetaan Langsung (Direct Mapping)

- ▶ Teknik ini merupakan teknik yang sederhana untuk menerjemahkan nilai record key menjadi address. Ada 2 cara dalam pemetaan langsung, yaitu :
 - Absolute Addressing (Pengalamatan Mutlak)
 - Relative Addressing (Pengalamatan Relatif)



▶ Absolute Addressing (Pengalamatan Mutlak)

$R(\text{NILAI KEY}) \longrightarrow \text{ADDRESS}$

$\text{NILAI KEY} = \text{ALAMAT MUTLAK}$

Nilai key yang diberikan oleh pemakai program sama dengan ADDRESS SEBENARNYA dari record tersebut pada penyimpanan sekunder.



KEUNTUNGAN	KELEMAHAN
Fungsi pemetaan R sangat sederhana	Pemakai harus mengetahui dengan pasti record-record yang disimpan secara fisik.
Tidak membutuhkan waktu lama dalam menentukan lokasi record pada penyimpanan sekunder.	Merupakan device dependent. Perbaikan atau pengubahan device, dimana berkas berada akan mengubah nilai key.
	Merupakan address space dependent. Reorganisasi berkas relatif akan menyebabkan nilai key berubah.



▶ Relative Addressing (Pengalamatan Relatif)

$R(\text{NILAI KEY}) \longrightarrow \text{ADDRESS}$

NILAI KEY = ALAMAT RELATIF

KEUNTUNGAN	KELEMAHAN
Fungsi pemetaan R sangat sederhana.	bukan device dependent
Nilai key dari sebuah record dapat ditentukan lokasi recordnya dalam sebuah penyimpanan sekunder tanpa memerlukan waktu proses yang berarti.	Merupakan address space dependent Terjadinya pemborosan ruangan



Teknik Pencarian Tabel (Directory Look Up)

- ▶ Dasar pemikiran pendekatan pencarian tabel adalah sebuah tabel atau direktori dari nilai key dan address.
- ▶ *Keuntungan dari Pencarian Tabel :*
 - Sebuah record dapat diakses dengan cepat, setelah nilai key dalam direktori ditentukan.
 - Nilai key dapat berupa field yang mudah dimengerti seperti PART NUMBER, NPM, karena nilai key tersebut akan diterjemahkan menjadi alamat.
 - Nilai key adalah address space independent, dimana reorganisasi berkas tak akan memengaruhi nilai key, yang berubah adalah alamat dalam direktori.



Teknik Kalkulasi Alamat

R (NILAI KEY) \longrightarrow ADDRESS

- ▶ Adalah dengan melakukan kalkulasi terhadap nilai key, hasilnya adalah alamat relatif.
- ▶
- ▶ Salah satu kelemahan dari teknik pengalamatan relatif adalah ruang harus disediakan sebanyak jangkauan nilai key, terlepas dari berapa banyak nilai key.



- ▶ Salah satu masalah dari teknik ini adalah ditemukannya alamat relatif yang sama untuk nilai key yang berbeda.

- ▶ *Keadaan dimana :*

$R(K1) = R(K2)$,disebut benturan

$K1 \neq K2$,atau collision

- ▶ Sedangkan nilai key K1 dan K2 disebut synomin.

Synonim adalah dua atau lebih nilai key yang berbeda pada hash ke home address yang sama.



► *Teknik-teknik yang terdapat pada kalkulasi alamat :*

- Scatter storage techniques
- Randomizing techniques
- Key-to-address transformation methods
- Direct addressing techniques
- Hash table methods
- Hashing



Metode Hashing

- ▶ Hashing merupakan kalkulasi terhadap nilai key untuk mendapatkan sebuah alamat (address)

Keuntungan pemakaian hashing

- ▶ Nilai key yang sebenarnya dapat dipakai karena diterjemahkan ke dalam sebuah alamat.
- ▶ Nilai key adalah address space berubah, tetapi nilai key akan tetap.



Kelemahan :

- ▶ Membutuhkan waktu proses dalam mengimplementasikan fungsi hash
- ▶ Membutuhkan waktu proses dan akses I/O dalam mengatasi benturan (Collision)

Penampilan fungsi hash bergantung pada :

- ▶ Distribusi nilai key yang dipakai
- ▶ Banyaknya nilai key yang dipakai relatif terhadap ukuran dari ruang alamat.
- ▶ Banyaknya record yang dapat disimpan pada alamat tertentu tanpa menyebabkan benturan
- ▶ Teknik yang dipakai untuk mengatasi benturan



Fungsi hash yang umum digunakan :

- ▶ Hashing dengan Kunci Modulus N
- ▶ Hashing dengan Kunci Modulus P
- ▶ Hashing dengan Pemotongan
- ▶ Hashing dengan Lipatan
- ▶ Hashing dengan Pergeseran
- ▶ Hashing dengan Pengkuadratan
- ▶ Hashing dengan Konversi Radix



1. Hashing dengan Modulus N

- ▶ Merupakan fungsi hashing yang paling populer

- ▶ Rumus :

$$f(\text{kunci}) = \text{kunci mod } N$$

dengan :

N : ukuran tabel atau berkas

Mod : sisa pembagian

- ▶ Contoh :

misal $N = 12$

$30 \text{ mod } N = 6$, diperoleh dari 30 dibagi 12 menghasilkan 2 dgn sisa 6

$40 \text{ mod } N = 4$, diperoleh dari 40 dibagi 12 menghasilkan 3 dgn sisa 4



2. Hashing dengan Modulus P

- ▶ Fungsi hashing Kunci mod P merupakan variasi Kunci modulus N.

- ▶ Rumus :

$$f(\text{kunci}) = \text{kunci mod } P$$

dimana :

P : merupakan bilangan prima terkecil yang lebih besar atau sama dengan N

- ▶ Contoh :

Jika N = 12 maka P = 13

$30 \text{ mod } P = 4$, diperoleh dari $30 \text{ mod } 13$ hasilnya 2 dgn sisa 4

$40 \text{ mod } P = 1$, diperoleh dari $40 \text{ mod } 13$ hasilnya 2 dgn sisa 1



3. Hashing dengan Pemotongan

- ▶ melakukan pemenggalan sejumlah digit yang pertama atau yang terakhir dari sejumlah digit.
- ▶ Keuntungan, cepat dan mudah dalam implementasinya
- ▶ Kerugian, terbatasnya ukuran ruang alamat

▶ Contoh 1:

Jika fungsi F menghapus 6 digit akhir dari digit 123456789
maka $f(123456789) = 123$

hashing memetakan 123456789 ke alamat 123



4. Hashing dengan Lipatan

- ▶ Nilai kunci dibagi menjadi beberapa bagian, masing-masing memiliki jumlah digit yang sama (kecuali bagian awal atau akhir). Bagian-bagian ini kemudian dilipat antara satu bagian dengan bagian lain. Hasil penjumlahan setelah dilipat dan digit dengan orde paling tinggi dipenggal menjadi alamat relative.

- ▶ Contoh :

Nilai kunci 253456718, target alamat relative menggunakan 4 digit

artinya Nilai Kunci dibagi menjadi 4 digit



4. Hashing dengan Lipatan...(2)

Jawab :

2 5345 6718

dilipat dengan urutan terbalik, hasilnya :

2

5345

8176

15521

Digit dengan order tertinggi dihilangkan menjadi 5521



5. Hashing dengan Pengkuadratan

- ▶ Hashing dengan pengkuadratan adalah fungsi hashing dengan cara mengkuadratkan kunci.
- ▶ Hasil pengkuadratan ini kemudian dapat dikombinasi dengan pemotongan atau lipatan untuk mendapatkan alamat yang diijinkan.
- ▶ Contoh : Pengkuadratan kunci 782 akan menghasilkan kemungkinan alamat 117

diperoleh dari :

$$7^2 + 8^2 + 2^2 = 49 + 64 + 4 = 117$$



6. Hashing dengan Konversi Radix

- ▶ Dalam konversi radix, kunci dianggap dalam base selain 10 yang kemudian dikoversi ke dalam basis 10, misal 5678 dalam base 13 akan menghasilkan 12098, diperoleh dari :

Posisi : $\begin{matrix} 3 & 2 & 1 & 0 \\ 5 & 6 & 7 & 8 \end{matrix}$

$$(5 \times 13^3) + (6 \times 13^2) + (7 \times 13^1) + (8 \times 13^0) = 10985 + 1013 + 91 + 8 = 12098$$

