

ORGANISASI BERKAS LANGSUNG

Pengguna teknologi pada umumnya dan teknologi informasi pada khususnya merasa bahwa waktu yang diperlukan untuk memperoleh suatu informasi masih terlalu lama. Metode pencarian biner maupun interpolasi masih belum dapat mengimbangi ketidaksabaran manusia terhadap penyediaan informasi yang cepat dan akurat. Perkembangan media perekam data dalam hal kemampuan untuk menampung volume data yang sangat besar harus diimbangi dengan peningkatan teknik pencarian kembali data yang tersimpan.

A. KUNCI SEBAGAI ALAMAT REKAMAN YANG UNIK

Untuk mendapat rekaman yang diasosiasikan dengan suatu kunci primer, sangat diharapkan agar proses langsung menuju ke alamat tempat rekaman dengan kunci tertentu disimpan. Hal tersebut mungkin terjadi apabila satu kunci rekaman juga merupakan alamat lokasi rekaman. Untuk aplikasi rekaman berisi nomor induk mahasiswa, terdapat 8 digit (kunci rekaman yang merupakan gabungan 2 digit tahun angkatan + 2 digit kode program studi + 4 digit nomor urut), maka diperlukan lokasi sebanyak 99.999.999. Dengan demikian waktu pencarian akan sangat baik, yaitu 1 probe untuk setiap rekaman yang dicari. Akan tetapi, teknik tersebut memiliki kerugian karena diperlukan ruang yang sangat besar untuk menampung semua rekaman. Untuk menampung data NIM mahasiswa diperlukan ruang yang besar, meski jumlah mahasiswa mungkin hanya 500, pasti ada beberapa nomor yang kosong karena beberapa alasan, misalnya sudah lulus, mengundurkan diri, drop-out, cuti, dan sebagainya. Sehingga tidak semua ruang dimanfaatkan.

B. KONVERSI KUNCI REKAMAN MENJADI SATU ALAMAT YANG UNIK

Contoh klasik yang menggambarkan fenomena konversi kunci rekaman adalah sistem reservasi penerbangan. Jika suatu maskapai penerbangan memiliki nomor penerbangan dari 1 sampai 999, ingin memantau reservasi pesawat selama setahun yang jumlah harinya 1 sampai 366 (1 tahun), maka nomor penerbangan dan hari-ke dapat dihubungkan untuk mendapatkan lokasi rekaman yang berisi data reservasi penerbangan pada hari tertentu.

$$\text{LOKASI} = \text{NOMOR PENERBANGAN} + \text{HARI-KE DALAM TAHUN INI}$$

Dengan simbol + menandakan hubungan, maka untuk menyediakan semua kemungkinan penerbangan diperlukan ruang alamat sebesar 999366 unit. Jumlah tersebut dapat direduksi sampai dengan 30% bila digunakan kombinasi:

$$\text{LOKASI} = \text{HARI-KE DALAM TAHUN INI} + \text{NOMOR PENERBANGAN}$$

Karena kecil kemungkinan dalam satu maskapai terdapat lebih dari 100 penerbangan, maka ruang alamat maksimum yang diperlukan adalah 36699.

C. MENENTUKAN ALAMAT DENGAN KONVERSI KUNCI

Diperlukan satu fungsi untuk memetakan cakupan nilai kunci yang lebih luas ke dalam cakupan yang lebih sempit nilai alamat. Fungsi yang dikenal dengan **fungsi hash** akan melakukan pemetaan sebagaimana diharapkan. Hash (kunci) → kemungkinan alamat. Keluaran dari proses hashing bukan lagi alamat yang unik, melainkan **kemungkinan alamat** bagi kunci yang di hash. Alamat untuk menempatkan alamat yang diperoleh dari fungsi hash disebut **home-address** untuk rekaman tersebut. Tidak ada batasan mengenai bentuk fungsi yang akan

memetakan kunci ke cakupan alamat, tetapi diharapkan fungsi tersebut menghasilkan kemungkinan alamat yang:

- ❑ Mampu mendistribusi kunci secara merata ke dalam cakupan alamat. Hal tersebut dimaksudkan untuk mengurangi terjadinya kolisi. Kolisi terjadi bila hasil hashing dua kunci rekaman yang berbeda menunjuk ke alamat yang persis sama.
- ❑ Dapat dieksekusi dengan efisien. Hal tersebut dimaksudkan agar waktu pembacaan dapat ditekan seminimal mungkin.

Maka fungsi hashing dapat dipandang dari 2 aspek, yaitu:

- ❑ Fungsi hash itu sendiri
- ❑ Metode untuk meresolusi kolisi

Mekanisme resolusi kolusi diperlukan untuk mengatasi terjadinya jumlah rekaman yang dikonversikan ke suatu lokasi melebihi kapasitasnya. Dalam bahasan ini diasumsikan bahwa satu lokasi memiliki kapasitas satu rekaman. Berikut adalah beberapa fungsi hash, dimulai dari yang paling sering digunakan.

a) Hashing dengan Kunci Modulus N

Satu fungsi hash yang paling populer dan paling sering diimplementasikan adalah modulus N,

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

dengan N sebagai ukuran tabel atau berkas. Hasil fungsi modulus adalah sisa pembagian kunci oleh N. Sebagai contoh untuk $N=12$ maka: $30 \bmod N = 6$ 40

$\text{mod } N = 4$ Keuntungan fungsi ini hanya menghasilkan nilai dalam rentang ruang alamat (0) sampai dengan (N-1)

b) Hashing dengan Kunci Modulus P

Fungsi hashing Kunci mod P merupakan variasi fungsi modulus N, rumusnya adalah:

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

dengan P sebagai bilangan prima terkecil yang lebih besar atau sama dengan N. dan N adalah ukuran tabel. P ini kemudian menjadi ukuran tabel baru yang menggantikan N. Contoh: untuk $N=12$ maka $P=13$ $30 \bmod P = 6$ $40 \bmod P = 4$

c) Hashing dengan Pemotongan

Alternatif lain untuk fungsi hashing adalah pemotongan. Sebagai contoh adalah para pegawai negeri sipil di Indonesia. Para pegawai memiliki NIP (Nomor Induk Pegawai) dengan panjang total 9 digit, terdiri atas 3 digit pertama sebagai identitas departemen, sementara 6 digit terakhir adalah nomor urutnya. Bila ingin dijadikan 6 digit saja, maka bisa dilakukan pemotongan jumlah digit. Pemotongan bisa dilakukan dibagian mana saja, tentunya dengan konsekuensi masing-masing. Pada kasus NIP, jika pemotongan dilakukan pada bagian belakang, akan diperoleh sejumlah alamat yang memiliki 3 digit yang sama (identitas departemen) sehingga kemudian kolisinya akan lebih besar dibanding bila pemotongan dilakukan di bagian depan.

d) Hashing dengan Lipatan

Fungsi ini melipat digit pada batasan yang ditentukan berdasarkan kondisi digit awal dan digit yang akan dihasilkan. Sebagai contoh 9 digit NIP akan direduksi

menjadi 3 digit, maka digit awal di bagi 3, kemudian dilipat pada batas antar-bagian.

Contoh :

5	8	3	9	7	6	1	2	4
---	---	---	---	---	---	---	---	---

Kunci asli tersebut ditulis pada selembar kertas. Batasan dimana lipatan akan dilakukan ditandai dengan garis .

Penjumlahan dari susunan tersebut adalah:

```

385
976
421
----- +

```

Jika penjumlahan dilakukan dengan mengabaikan carry maka diperoleh alamat 672, sedangkan jika tidak mengabaikan carry maka hasilnya 782.

e) Hashing dengan Penggeseran

Hashing dengan penggeseran memiliki proses yang serupa dengan hashing lipatan, bedanya setelah ditentukan batasan. Digit asli dipotong kemudian digeser dan dihitung hasil jumlahnya. 583 976 124 ----- + Yang diperoleh alamat yang berbeda yaitu 573 (tanpa carry). Jika kedua hasil penjumlahan tersebut diterapkan dengan menggunakan carry dan hanya tiga digit yang paling kanan saja yang digunakan, maka diperoleh 683.

f) Hashing dengan Pengkuadratan

Hashing dengan penguadratan adalah fungsi hashing dengan menguadratkan kunci. Hasil penguadratan ini kemudian dapat dikombinasi dengan pemotongan atau lipatan untuk mendapatkan alamat yang diperbolehkan. Sebagai contoh,

penguadratan kunci 782 akan menghasilkan kemungkinan alamat 117. $F(782) = 117$

g) Hashing dengan Konversi Radix

Dalam konversi radix, kunci dianggap dalam base selain 10 yang kemudian dikonversi ke dalam basis 10, misal kunci 5 6 7 8 dalam base 13 akan menghasilkan 12098, diperoleh dari: 5 6 7 8 Posisi: 3 2 1 0 Hasil tersebut masih dapat dikombinasi dengan fungsi hash lain (pemotongan atau lipatan) untuk mendapatkan digit alamat yang diinginkan.