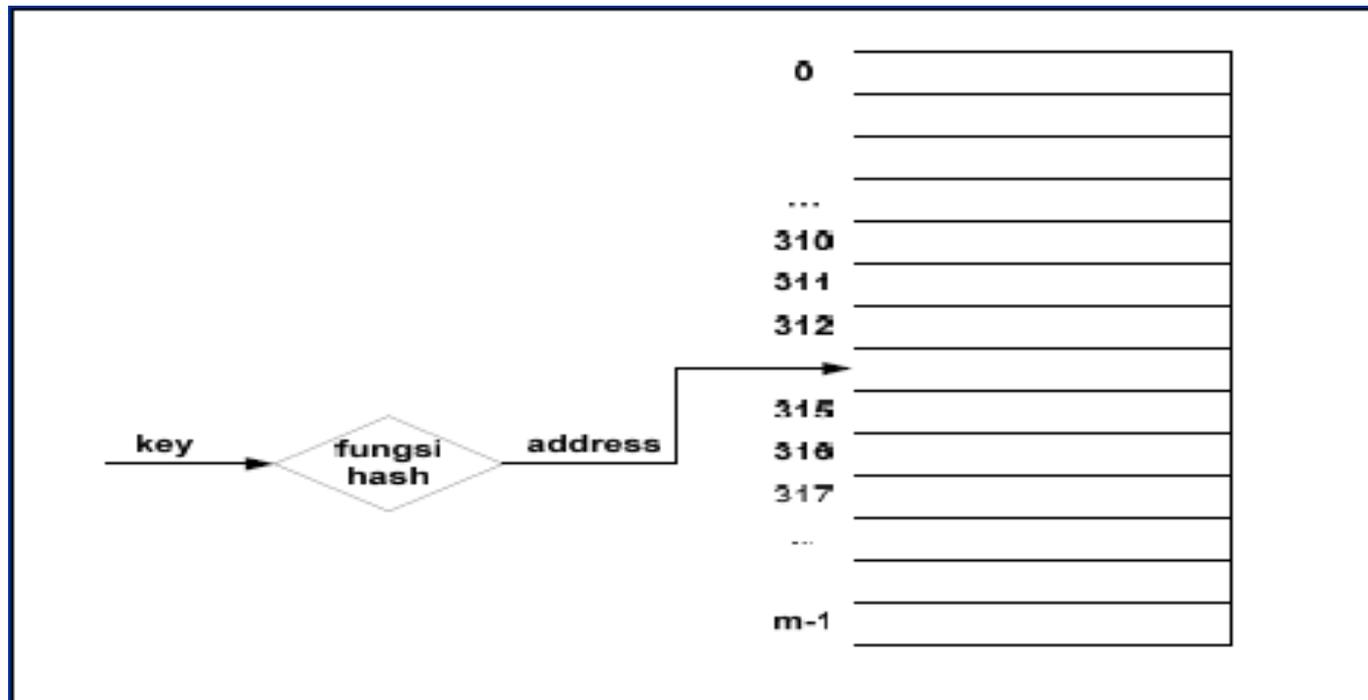


File Hash

- Konsep File Hash
- Fungsi Hash
- Collision

Konsep File Hash

- Merupakan organisasi file dengan metode akses langsung (direct access), yang menggunakan suatu fungsi untuk memetakan key menjadi address



Konsep File Hash(2)

- Fungsi yang digunakan disebut fungsi hash/KAT (key to address transformation)
- Address yang dihasilkan dari hasil perhitungan fungsi hash disebut dengan istilah home address
- Jadi, terdapat dua komponen dalam file hash :
 - Ruang rekord, yang terdiri atas m slot address
 - Fungsi hash, yang mentransformasi key menjadi address
- Transformasi key akan mudah jika key telah berupa nilai integer, untuk key berupa karakter alphanumerik terdapat proses prakondisi untuk mengubahnya menjadi suatu nilai integer

Fungsi Hash

Ada beberapa fungsi hash yang dapat digunakan, seperti :

- Key Mod N, dengan N = jumlah slot address (ukuran tabel data)

$$\text{Contoh : } 25 \bmod 11 = 3$$

$$25 \pmod{7} = 4$$

$$3 \pmod{8} = 3$$

jika key bernilai negatif, maka bagi $|key|$ dengan N untuk dapatkan sisa r :

- untuk $r = 0$, maka $k \bmod N = 0$

- untuk $r \neq 0$, maka $k \bmod N = N - r$

$$\text{Contoh : } -26 \pmod{7} = 7 - 5 = 2$$

$$-371 \pmod{8} = 8 - 3 = 5$$

Fungsi Hash(2)

- Truncation/substring, cara transformasi yang dilakukan dengan mengambil hanya sebagian digit dari key
- Misal Nomor Kartu anggota club MAKRO Bandung 05 533107 70.
- Apabila diinginkan dipetakan ke tabel 10000 alamat, maka 4 digit nomor dapat digunakan untuk alamat, yaitu misal 3107

Fungsi Hash (3)

Folding (Metoda Pelipatan), dapat dilakukan dengan cara:

- Folding by boundary

contoh jika key = 123456789, maka transformasi ke 3 digit address dengan teknik folding by boundary dapat dilakukan dengan membagi digit key tsb dengan cara seolah-olah melipat batas pembagian digit seperti berikut :

3 2 1

4 5 6

9 8 7 +

6 5 4

- apabila kode – kode itu ditambahkan (tanpa carry) , maka diperoleh 654.

Fungsi Hash(4)

Folding (Metoda Pelipatan), dapat dilakukan dengan cara:

- Folding by boundary

contoh jika key = 123456789, maka transformasi ke 3 digit address dengan teknik folding by boundary dapat dilakukan dengan membagi digit key tsb dengan cara seolah-olah melipat batas pembagian digit seperti berikut :

3 2 1

4 5 6

9 8 7 +

6 5 4

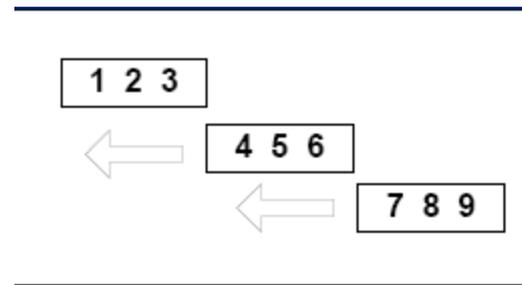
- apabila kode – kode itu ditambahkan (tanpa carry) , maka diperoleh 654.

Fungsi Hash(5)

- Folding by shifting

contoh jika key = 123456789, maka transformasi ke 3 digit address dengan teknik folding by boundary dapat dilakukan dengan membagi digit key tsb dengan cara seolah-olah menggeser batas pembagian digit seperti berikut :

$$\begin{array}{r} 1\ 2\ 3 \\ 4\ 5\ 6 \\ \underline{7\ 8\ 9} + \\ 2\ 5\ 8 \end{array}$$



- apabila kode – kode itu ditambahkan (tanpa carry) , maka diperoleh 258

Fungsi Hash (5)

- ***Squaring***

1. Metoda menentukan alamat dengan cara pengkuadratan kunci kemudian hasilnya akan dilakukan truncation
2. Cara ini dilakukan agar terjadi kombinasi nilai kunci terdistribusi dengan baik keseluruhan alamat yang mungkin
3. Fungsi mid square adalah menghitung kwadrat key dan kemudian menggunakan string bit tengah dari hasil sebagai alamat ke tabel

Contoh :

Key enam digit : 123456
Dikwadratkan : 15241383936

Apabila dikehendaki alamat 3 digit, maka dapat dipilih posisi 5 sampai dengan 7, dengan demikian menghasilkan alamat 138.

Fungsi Hash (6)

- **Radix Conversion**

Pada metoda ini kunci dianggap berbasis selain 10 dan kemudian dikonversi dalam basis 10

Contoh :1234 dianggap berbasis 11 berarti :

$$= (1*11^3) + (2*11^2) + (3*11^1) + (4*11^0)$$

$$= 1331 + 242 + 33 + 4$$

$$= \mathbf{1610}$$

- **Multiplicative Hashing**

Floor dan Cell

$\lfloor x \rfloor$ adalah bilangan bulat terbesar yang tidak melebihi x

$\lceil x \rceil$ adalah bilangan bulat terkecil yang tidak lebih kecil dibanding x

apabila x adalah bilangan bulat maka :

$$\lfloor x \rfloor = \lceil x \rceil = x$$

apabila x bukan bilangan bulat maka :

$$\lfloor x \rfloor + 1 = \lceil x \rceil$$

contoh :

$$\lfloor 3.14 \rfloor = 3$$

$$\lceil 3.14 \rceil = 4$$