

### **PERTEMUAN**

Struktur Data & Algorin



What is an MD5 hash?

## **HASHING**



# **Hashing**

- Hashing
- Hash Tables
- Fungsi (Hash Functions)
- □ Collision Resolution
- Menerapakan Hash Table



# **Hashing**

#### **Pengertian Hash**

Kita dapat membangun struktur data yang dapat dicari dalam waktu O (1).
Konsep ini disebut sebagai hashing



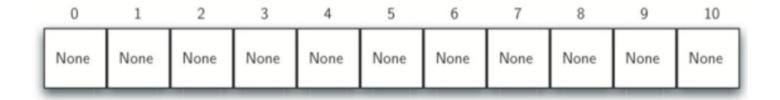
# **Hash Table**

- Hash table adalah kumpulan item yang disimpan sedemikian rupa agar mudah ditemukan nanti.
- Setiap posisi Hash table, slot, dapat menampung item dan dinamai dengan nilai integer mulai dari 0.
- Misalnya, kita akan memiliki slot bernama 0, slot bernama 1, slot bernama 2, dan seterusnya.
- Awalnya, tabel hash tidak berisi item sehingga setiap slot kosong.



# **Hash Table**

- Untuk mengimplementasikan tabel hash dengan menggunakan daftar dengan setiap elemen diinisialisasi ke nilai khusus None.
- $\Box$  Berikut adalah tabel hash kosong dengan ukuran m = 11 value None.





### **Hash Table**

- Pemetaan antara item dan slot tempat item tersebut berada dalam tabel hash disebut hash function.
- Hash function akan mengambil item apa pun dalam koleksi dan mengembalikan integer dalam kisaran nama slot, antara 0 dan m-1.
- Bagaimana kita harus menggunakan fungsi hash?



- Salah satu fungsi hash (hash function) yang bisa kita gunakan adalah metode sisanya.
- Saat disajikan dengan item, fungsi hash adalah item dibagi dengan ukuran tabel, ini kemudian nomor slotnya.



- Asumsikan bahwa kita memiliki himpunan bilangan integer 54, 26, 93, 17, 77, dan 31.
- Tetapkan tabel hash kosong dari m=11
- Selanjutnya, fungsi hash: h(item)= item % 11
  - $\Box$  54 = 54 / 11 = 10 Sisa 10
  - $\square$  26 = 26/11 = 4
  - 93 = 93/11 = 5
  - □ 17 = 17/11 = 6
  - 77 = 77/11 = 0
  - 31 = 31/11 = 9



Item	Hash Value
54	10
26	4
93	5
17	6
77	0
31	9



- Bagaimana dapat menempati 6 dari 11 slot.
- Ini disebut sebagai file faktor beban, and is biasanya dilambangkan dengan

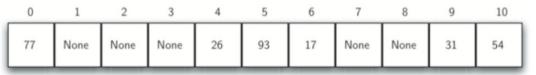
$$\lambda =$$
.

 $\Box$  contoh ini,  $\lambda = 6/11$ .



#### **Hash Function – Remainder Methode**

· Tabel hash.





#### **Hash Function – Remainder Methode**

- Saat user ingin mencari item, kita cukup menggunakan fungsi hash untuk menghitung nama slot untuk item tersebut dan kemudian memeriksa tabel hash untuk melihat apakah ada.
- Operasi pencarian ini adalah 0 (1), karena jumlah waktu yang konstan diperlukan untuk menghitung nilai hash dan kemudian mengindeks tabel hash di lokasi itu.



#### **Hash Function – Remainder Methode**

- bagaimana jika Anda memiliki dua item yang akan menghasilkan lokasi yang sama?
- Misalnya 44% 11 dan 77% 11 adalah sama.
- Ini dikenal sebagai tabrakan / collision (juga dikenal sebagai benturan).
- fungsi hash secara umum!



- Fungsi hash yang memetakan setiap item ke dalam slot unik disebut sebagai fungsi hash yang sempurna.
- Tujuan membuat fungsi hash adalah yang meminimalkan jumlah tabrakan, mudah dihitung, dan mendistribusikan item dalam tabel hash secara merata.
- Ada beberapa teknik untuk pembahaan ini!



- Jika dicontohkan pada nomor telepon 436-555-4601
- Kami akan mengambil digit dan membaginya menjadi kelompok 2 (43,65,55,46,01).
- Setelah penjumlahan, 43 + 65 + 55 + 46 + 01, kita mendapatkan 210.
- Jika kita mengasumsikan tabel hash kita memiliki 11 slot, maka kita perlu melakukan langkah ekstra untuk membagi dengan 11 dan menyimpan sisanya.
- 210% 11 adalah 1 jadi nomor telepon 436-555-4601 di-4 3 6 5 5 5 4 6 0 1



- Untuk metode kuadrat tengah (Mid Square Method), pertama kita mengkuadratkan item, lalu mengekstrak beberapa bagian dari digit yang dihasilkan.
- Misalnya, jika itemnya 44, pertama-tama kita akan menghitung 442 = 1.936.
- Dengan mengekstrak dua digit tengah, 93, dan melakukan langkah sisanya, kita mendapatkan 93% 11 = 5



- Untuk metode kuadrat tengah (Mid Square Method), pertama kita mengkuadratkan item, lalu mengekstrak beberapa bagian dari digit yang dihasilkan.
- Misalnya, jika itemnya 44, pertama-tama kita akan menghitung 442 = 1.936.
- Dengan mengekstrak dua digit tengah, 93, dan melakukan langkah sisanya, kita mendapatkan 93% 11 = 5



#### Comparison Table

Item	Remainder	Mid-Square
54	10	3
26	4	7
93	5	9
17	6	8
77	0	4
31	9	6



#### **Collision Resolution**

- Salah satu metode untuk menyelesaikan tabrakan melihat ke tabel hash dan mencoba menemukan slot terbuka lain untuk menampung item yang menyebabkan tabrakan.
- Kita bisa mulai dari posisi nilai hash asli dan kemudian bergerak secara berurutan melalui slot sampai kita menemukan slot pertama yang kosong.
- Proses resolusi tabrakan ini disebut sebagai pengalamatan terbuka yang mencoba menemukan slot atau alamat terbuka berikutnya dalam tabel hash.



#### **Collision Resolution**

Dengan mengunjungi setiap slot secara sistematis satu per satu, kami melakukan teknik pengalamatan terbuka yang disebut *linear probing*.



### **SELESAI**