





Struktur Data

Saniati, S.ST., M.T.

EPISODE 7

Hashing

Istilah yang Perlu Diketahui ?

- Hashing
- Hash Table
- Hash Function
- Collision
- Collision Resolution



Hashing ?

- Hashing adalah transformasi aritmatika sebuah string dari karakter menjadi nilai yang merepresentasikan string aslinya.
- Menurut bahasanya, hash berarti memenggal dan kemudian menggabungkan.
- Hashing digunakan sebagai metode untuk menyimpan data dalam sebuah larik (array) agar penyimpanan data, pencarian data, penambahan data, dan penghapusan data dapat dilakukan dengan cepat.
- Hashing digunakan untuk mengindex dan mendapatkan kembali key di database (hash table), karena lebih cepat untuk mengambil key yang sudah dihash daripada mencarinya menggunakan original value.
- Hashing juga dikenal sebagai konsep pendistribusian key dalam sebuah array (Hash Table) menggunakan fungsi yang sudah diketahui sebelumnya (Hash Function).



Hash Table ?

- Hash Table adalah array dengan sel-sel yang ukurannya telah ditentukan dan dapat berisi data atau key yang berkesesuaian dengan data.
- Hash Table adalah sebuah array untuk menyimpan original string. Index dari hash table adalah hashed key.
- Besarnya hash table lebih kecil dari jumlah key yang ada, sehingga memungkinkan ada string yang memiliki hashed key yang sama.



Hash Function ?

- Hash Function adalah cara yang kita buat untuk men-transformasi aritmetika sebuah string dari karakter menjadi nilai yang merepresentasikan string aslinya.
- Secara teori, kompleksitas waktu ($T(n)$) dari fungsi hash yang ideal adalah algoritma konstan $O(1)$. Untuk mencapai itu setiap record membutuhkan suatu kunci yang unik.



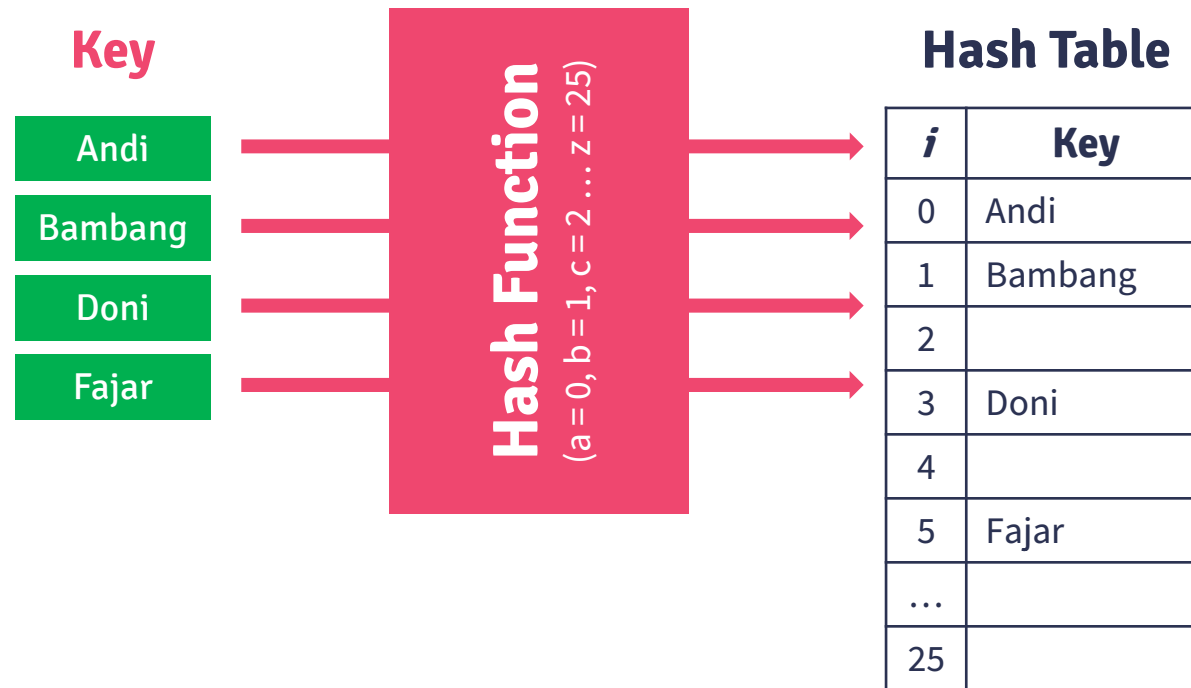
Notes ?

- Salah satu kegunaan hashing adalah untuk mengamankan tulisan yang biasanya diterapkan untuk menyembunyikan password asli.
- Teknik dan cara hashing semakin berkembang seiring dengan keamanan data yang terus bisa diretas.



Cara-cara Hash Function ?

- Untuk dasarnya, biasanya mengambil dari satu karakter awal pada string original dan diubah menjadi angka index yang nantinya disimpan ke hash table.



Cara-cara Hash Function ?

- Banyak cara untuk melakukan hash sebuah string menjadi hashed key. Berikut adalah cara-cara penting untuk membuat hash function :

- **Mid-Square**
- **Division** (Most Common)
- **Folding**
- **Digit Extraction**
- **Rotating Hash**
- **Truncation**
- dll..



Mid-Square ?

- Pangkatkan key, dan ambil bit pada bagian tengah dari hasil pangkat untuk dijadikan hash-key.
- Jika key berupa string maka ubah menjadi number dengan cara :

U **D** **I** **N**

21 + 4 + 9 + 14

$48^2 = 2304$

30

A **D** **I**

1 + 4 + 9

$14^2 = 196 = 0196$

19



Division ?

- Lakukan pembagian pada key dengan operator modulus (sesuai dengan jumlah hash table).
- Jika key berupa string maka ubah menjadi number dengan cara :

U **D** **I** **N**

21 + 4 + 9 + 14

48 % 20

8

A **D** **I**

1 + 4 + 9

14 % 20

14



Folding ?

- Hash Function folding akan men-transformasi string ke hashed key dengan 2 Langkah :

1. Bagi key menjadi bagian-bagian yang memiliki jumlah digit sama.
2. Jumlahkan setiap bagian.

U	D	I	N
21	4	9	14
$21 + 49 + 14$			
84			

O	K	I
15	11	9
$01 + 51 + 19$		
71		

O	K	I
15	11	9
$51 + 19$		
70		



Folding ?

- Hash Function folding akan men-transformasi string ke hashed key dengan 2 Langkah :

1. Bagi key menjadi bagian-bagian yang memiliki jumlah digit sama.
2. Jumlahkan setiap bagian.

U	D	I	N
---	---	---	---

21	4	9	14
----	---	---	----

41 + 94 + 12

147

O	K	I
---	---	---

15	11	9
----	----	---

91 + 15

106



Digit Extraction ?

- Mendapatkan hashed key dengan cara mengambil digit-digit tertentu dari sebuah key .

U **D** **I** **N**

21 4 9 14

2 4 1

O **K** **I**

15 11 9

1 1 9



Rotating Hash ?

- Mendapatkan hashed key dengan cara membalik urutan dari key.

U D I N

21 4 9 14

4 1 9 4 2 1

O K I

15 11 9

9 1 1 5 1



Truncation ?

- Mendapatkan hashed key dengan cara memenggal key sebanyak k (tidak boleh dilewati) dari n digit, dimana $k < n$.

U	D	I	N
---	---	---	---

21	4	9	14
----	---	---	----

2 1 4

U	D	I	N
---	---	---	---

21	4	9	14
----	---	---	----

1 4 9

U	D	I	N
---	---	---	---

21	4	9	14
----	---	---	----

4 9 1 4



Hash Function yang Baik ?

- Hash Function harus memiliki sifat berikut :
 - Mudah dihitung
 - Dua key yang berbeda akan dipetakan pada dua sel yang berbeda pada array. Tapi secara umum tidak bisa berlaku karena bisa jadi dua key yang berbeda mempunyai hasil hashed key yang sama (tabrakan).
 - Membagi key secara rata pada seluruh sel.
- Sebuah hash function sederhana adalah menggunakan fungsi modulus (sisa bagi) dengan bilangan prima.
- Dapat menggunakan manipulasi digit dengan kompleksitas rendah dan distribusi key yang rata.



Memilih Hash Function ?

- Sebuah hash function yang bagus memiliki dua kriteria :
 - Mudah dan cepat dihitung
 - Harus meminimalkan juga collisions (tabrakan) yang terjadi.



Collision ?

- Dikatakan terjadi collision (tabrakan) jika dua buah keys dipetakan pada sebuah sel yang sama.
- Collision bisa terjadi saat melakukan insertion (penambahan data).
- Penyelesaian bila terjadi collision (tabrakan) disebut *Collision Resolution*.
- Dibutuhkan prosedur tambahan untuk mengatasi terjadinya collision (tabrakan).



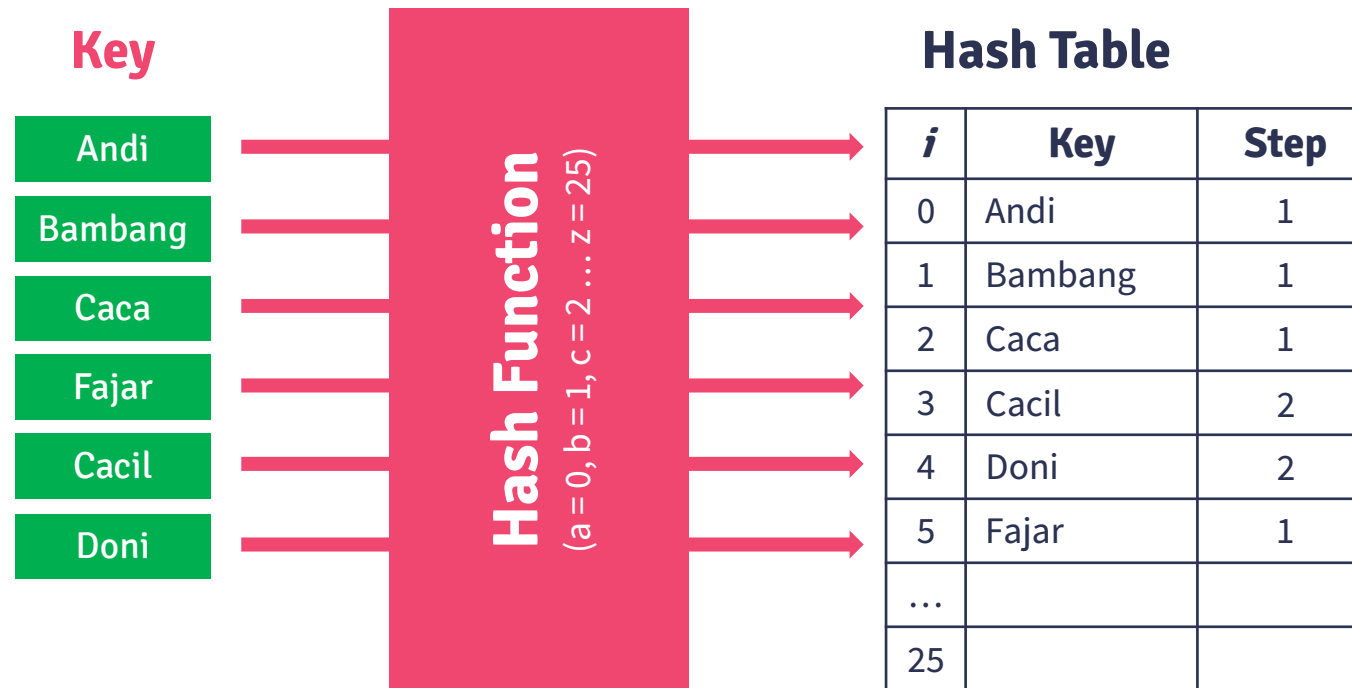
Collision Resolution ?

- Ada dua strategi umum dalam melakukan Collision Resolution :
 - **Closed Hashing** (Open Addressing)
 - **Linear Probing.**
 - Quadratic Probing.
 - Double Hashing.
 - **Open Hashing** (Chaining)



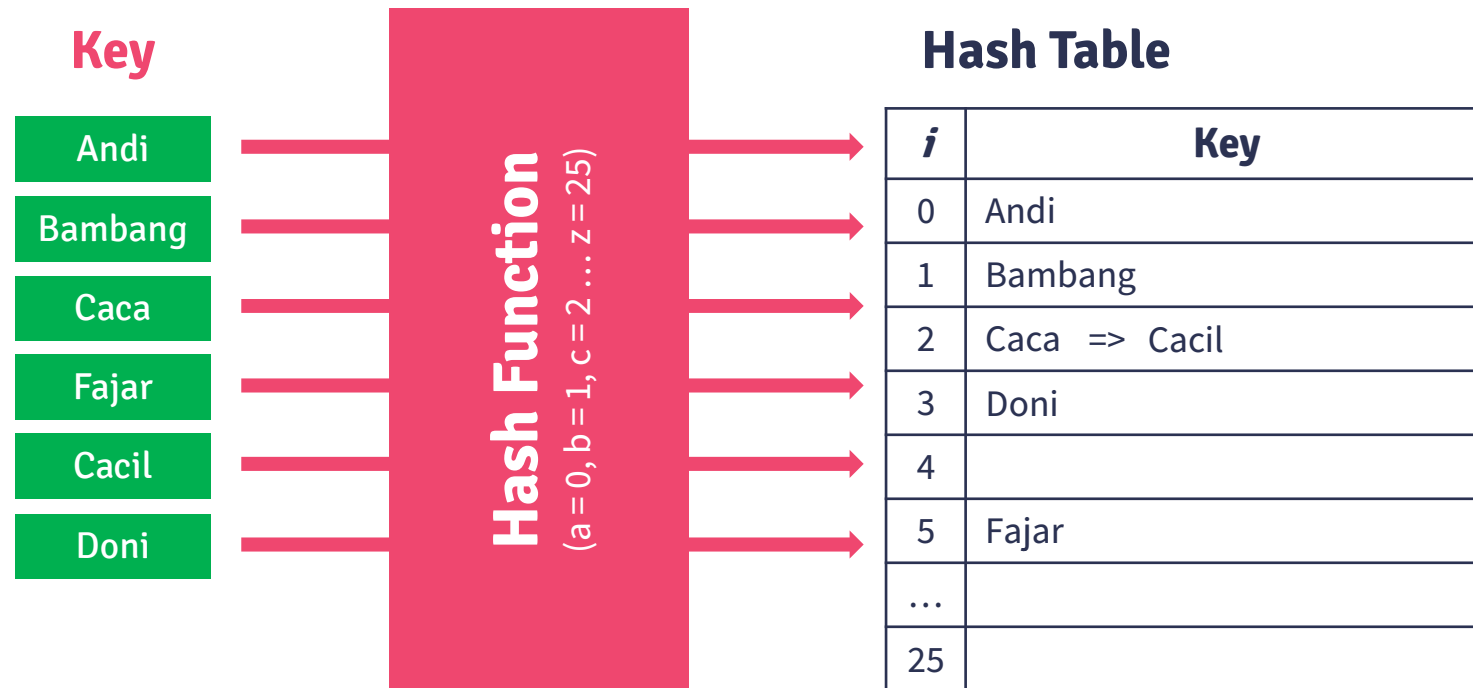
Closed Hashing – Linear Probing ?

- Idanya adalah mencari alternatif sel lain pada hash table.



Open Hashing (Chaining) ?

- Idanya meletakkan key pada table menggunakan linked list.





Video Selanjutnya

**Ngoding Program Login Sederhana
Dengan Hashing C++**



Thank you

**#KEEPLARNING
#KEEPSPIRITS**



