



Institut Teknologi
Telkom
Purwokerto

Teknik Informatika - Fakultas Informatika

Pertemuan 7 – Hash Table

Author: **Wahyu Andi Saputra [WAA]**

Outline

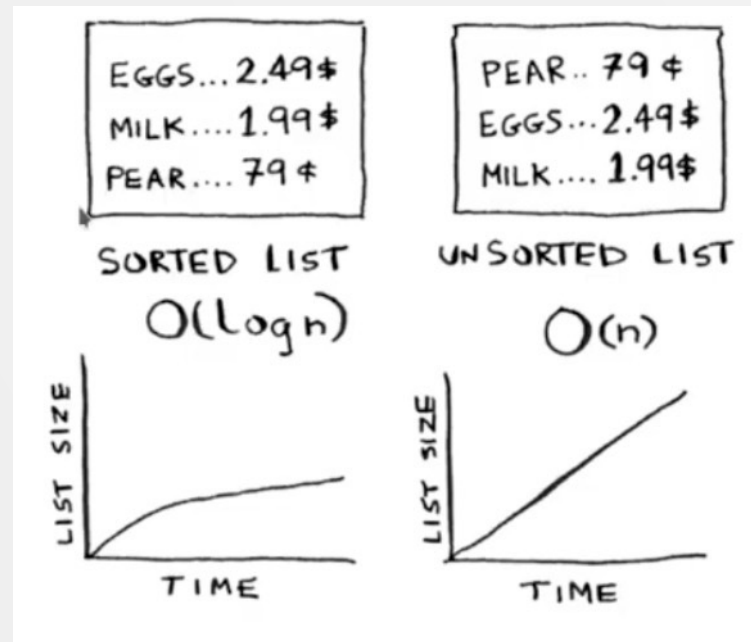
Hash Table

Hash Function

Collision

Hash Table

- Bayangkan kita sedang mencari sebuah item di list,
- Kalo belum tersusun secara alfabet: $O(n)$ time
- Kalo sudah tersusun secara alfabet: $O(\log n)$ time



Hash Table

- Bayangkan kita sedang mencari sebuah item di list,

- ... alfabet: $O(n)$ time

- ... alfa



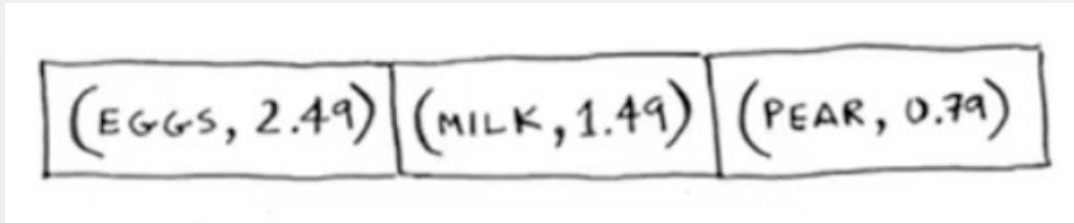
# OF ITEMS IN THE BOOK	$O(n)$	$O(\log n)$
100	10 sec	1 sec ← YOU NEED TO CHECK $\log_2 100 = 7$ LINES
1000	1.66 min	1 sec ← NEED TO CHECK $\log_2 1000 = 10$ LINES
10000	16.6 min	2 sec ← $\log_2 10000 = 14$ LINES = 2 sec

Hash Table

- Meskipun sudah terurut, namun masih tetap merepotkan kalau harus mencari item di list bahkan Ketika sudah urut abjad
- Akan lebih enak kalo semisal kita sudah punya partner yang menghafal seluruh keterangan item (nama dan harga)

Hash Table

- Struktur data utama: array (random akses) dan linked list (sequential akses)



(EGGS, 2.49)	(MILK, 1.49)	(PEAR, 0.79)
--------------	--------------	--------------

- Kita coba gunakan pendekatan array
- Misal, disini kita punya 3 slot array dengan 2 informasi (ex: nama dan harga)

Hash Table

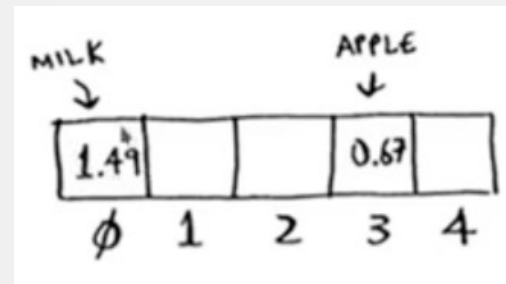
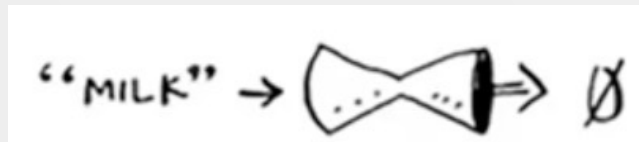
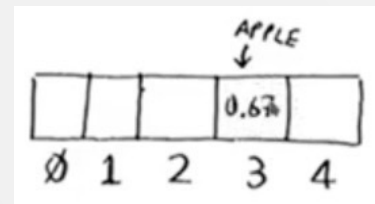
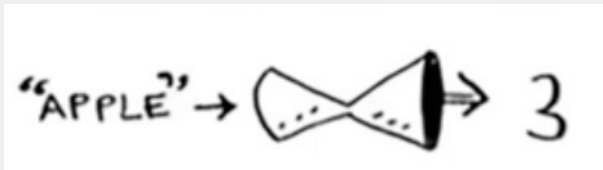
- Sekarang setiap array memiliki 2 informasi, yaitu nama dan harga item

(EGGS, 2.49)	(MILK, 1.49)	(PEAR, 0.79)
--------------	--------------	--------------

- Artinya, item bisa kita urutkan berdasar nama
- Lalu kita gunakan Binary Search Tree untuk mencari info di dalam array

Hash Table

- Ingat, hash table selalu memetakan string ke angka
- Misalkan kita memiliki peta "apple" ke angka 3,
- Maka bisa kita arahkan agar "apple" ke array index 3 dan isi array tersebut dengan harga apple

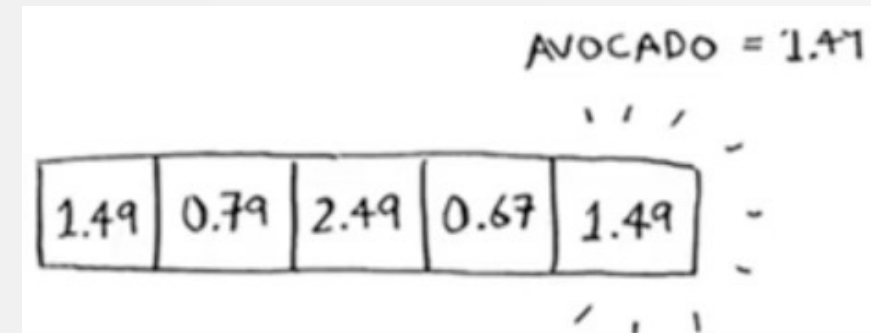


Hash Table

- Misal seluruh array sudah terisi

1.49	0.79	2.49	0.67	1.49
------	------	------	------	------

- Misalkan kita meminta data harga “avocado”, maka



Hash Table

ILUSTRASI 2:

- Linear searching: mencari kata kunci/item, satu persatu, dari yang paling kiri.
- Misalkan, kita sedang mencari kata kunci "Ada"

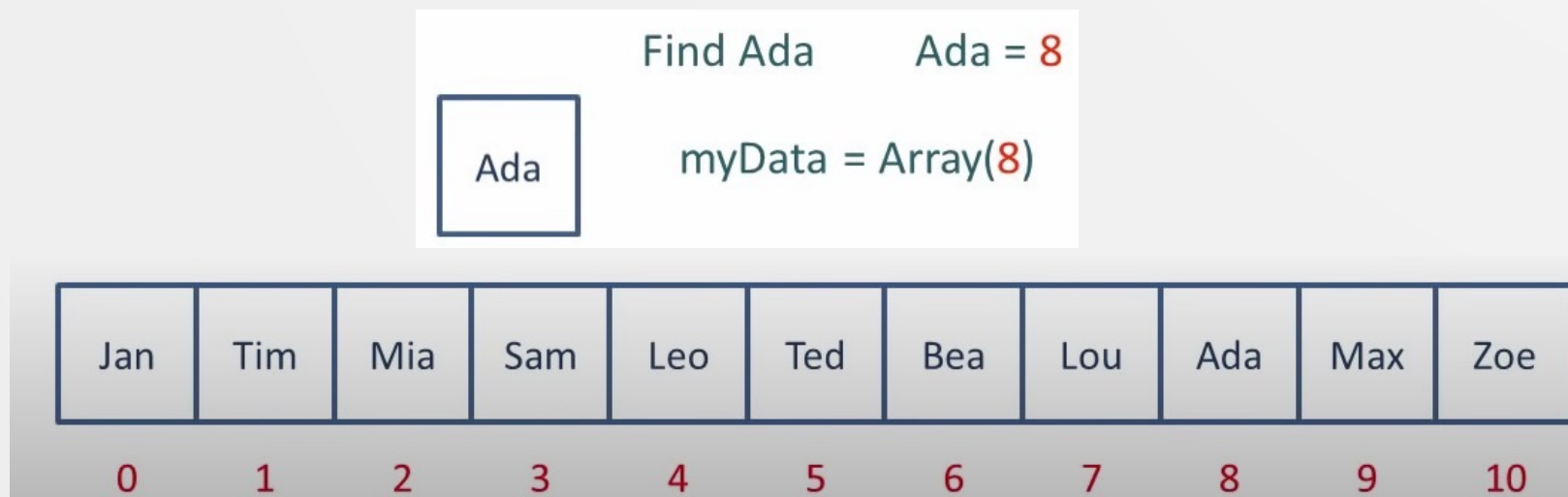
Jan	Tim	Mia	Sam	Leo	Ted	Bea	Lou	Ada	Max	Zoe
0	1	2	3	4	5	6	7	8	9	10

Jan	Tim	Mia	Sam	Leo	Ted	Bea	Lou	Ada	Max	Zoe
0	1	2	3	4	5	6	7	8	9	10

Hash Table

ILUSTRASI 2:

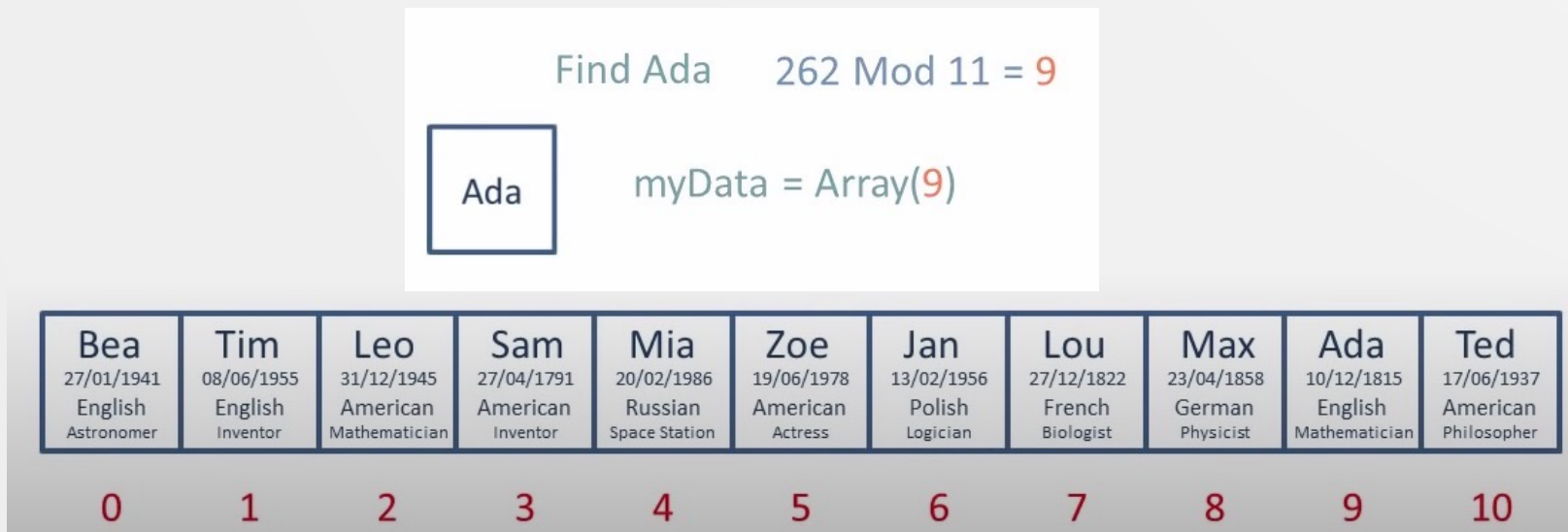
- Bahkan dengan menggunakan array, saat kita tahu lokasi item dan indexnya, maka akan mudah memanggil item tersebut



Hash Table

ILUSTRASI 2:

- Dari nilai indeks tadi lalu digunakan untuk memanggil karakter yang ingin dicari
- Setiap indeks dan array juga bisa digunakan untuk menyimpan beberapa informasi lain di dalamnya



Hash Table

- **ILUSTRASI 3:**

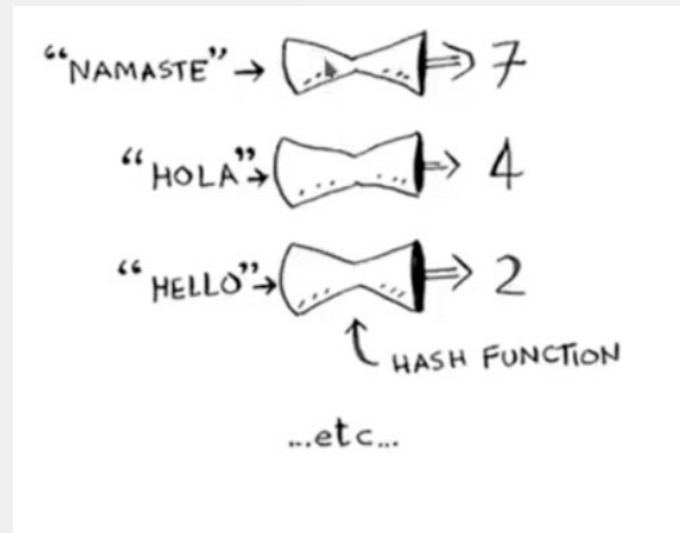
- Dianalogikan, seseorang bernama “Laura Basuki Situmorang”.
- Kalau kita menghafalkan nama panjang “Laura Basuki Situmorang”, maka akan sulit diingat
- Tetapi, kalau kita menggunakan nama “Laura” saja, maka akan lebih mudah dan cepat diingat

Hash Table

- Hash function sangat membantu kita karena tidak perlu melakukan pencarian sama sekali
- Hal ini karena:
 1. HF pasti memetakan secara konsisten string ke index yang pasti
 2. HF selalu memetakan string ke index/angka yang berbeda
 3. HF selalu tahu kapasitas array kita. Jadi kalau array kita 5 slot, HF tersebut tidak akan mengembalikan nilai 100 karena tidak valid
- Array dan list langsung dipetakan ke memori secara langsung, sedangkan HT menggunakan HF untuk menentukan posisi nilai yang akan disimpan

Hash Table

- Hash Function + Array = Hash Table
- Array dan list langsung dipetakan ke memori secara langsung, sedangkan HT menggunakan HF untuk menentukan posisi nilai yang akan disimpan

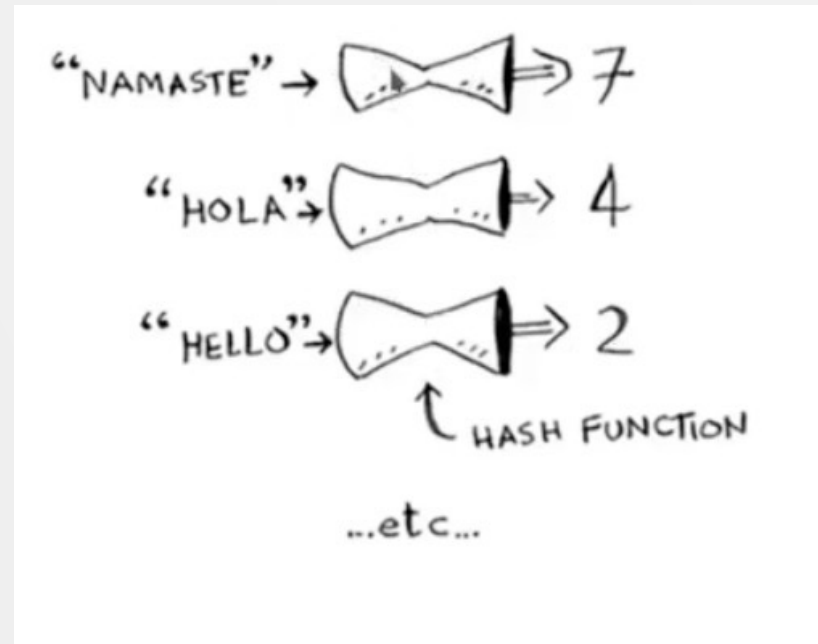




HASH FUNCTION

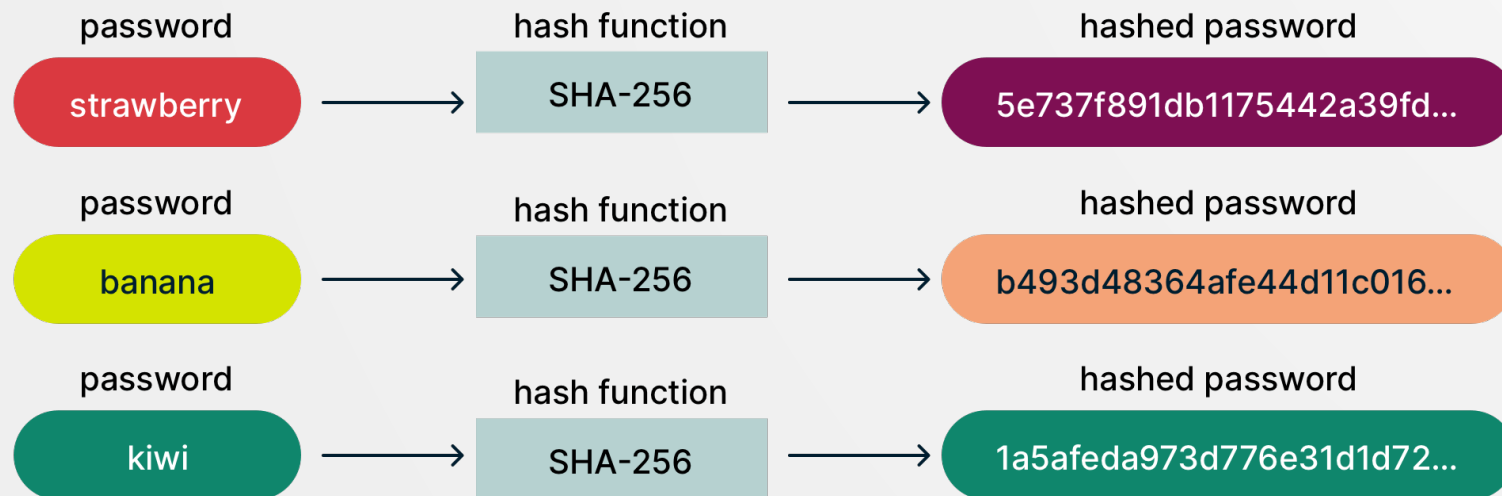
Hash Function

- Hash function adalah fungsi dimana kita bisa menginputkan suatu string, diinputkan ke hash function, dan outputnya suatu angka
- Cara mentransformasi suatu karakter, dari string menjadi angka



Hash Function

- Kegunaan hashing: mengamankan tulisan. Contoh: password
- Pada hash yang modern, output dapat berupa perpaduan angka dan alphabet



Hash Function: Cara Konversi

- Dalam mengubah string menjadi number, bisa menggunakan penomoran sesuai urutan abjad atau menggunakan kode ASCII

A = 1	H = 8	O = 15	V = 22
B = 2	I = 9	P = 16	W = 23
C = 3	J = 10	Q = 17	X = 24
D = 4	K = 11	R = 18	Y = 25
E = 5	L = 12	S = 19	Z = 26
F = 6	M = 13	T = 20	
G = 7	N = 14	U = 21	



Character	Decimal Value	Character	Decimal Value
A	65	a	97
B	66	b	98
C	67	c	99
D	68	d	100
E	69	e	101
F	70	f	102
G	71	g	103
H	72	h	104
I	73	i	105
J	74	j	106
K	75	k	107
L	76	l	108
M	77	m	109
N	78	n	110
O	79	o	111
P	80	p	112
Q	81	q	113
R	82	r	114
S	83	s	115
T	84	t	116
U	85	u	117
V	86	v	118
W	87	w	119
X	88	x	120
Y	89	y	121
Z	90	z	122

Hash Function: Cara Konversi

- Dalam mengubah string menjadi number, bisa menggunakan penomoran sesuai urutan abjad atau menggunakan **kode ASCII**

Karakter	Perhitungan						SUM
Leo	L	76	e	101	o	111	288
Sam	S	83	a	97	m	109	289

Character	Decimal Value	Character	Decimal Value
A	65	a	97
B	66	b	98
C	67	c	99
D	68	d	100
E	69	e	101
F	70	f	102
G	71	g	103
H	72	h	104
I	73	i	105
J	74	j	106
K	75	k	107
L	76	l	108
M	77	m	109
N	78	n	110
O	79	o	111
P	80	p	112
Q	81	q	113
R	82	r	114
S	83	s	115
T	84	t	116
U	85	u	117
V	86	v	118
W	87	w	119
X	88	x	120
Y	89	y	121
Z	90	z	122

Teknik 1: Mid Square

- Mid square:
 - Jumlahkan semua angka, Pangkatkan key, ambil 2 nilai tengah

U	D	I	N			
21	+	4	+	9	+	14
$48^2 = 2304$						
30						

A	D	I		
1	+	4	+	9
$14^2 = 196 = 0196$				
19				

Teknik 2: Division

- Division
 - Lakukan pembagian modulus dengan angka yang dipilih
 - 20 diperoleh dari kapasitas hash yang kita gunakan

Contoh 1: Menggunakan angka sesuai nomor urut abjad

U	D	I	N			
21	+	4	+	9	+	14
48	%	20				
8						

A	D	I		
1	+	4	+	9
14	%	20		
14				

Teknik 2: Division

- Ambil kode ASCII dari tiap karakter, jumlahkan, lalu bagi dengan total angka, dan ambil modulo nya

$$\text{Index number} = \text{sum ASCII codes} \text{ Mod } \text{size of array}$$

Contoh 2: Menggunakan angka sesuai KODE ASCII

Karakter	Perhitungan						SUM	Modulo
Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1
Bea	B	66	e	101	a	97	264	0
Zoe	Z	90	o	111	e	101	302	5
Jan	J	74	a	97	n	110	281	6
Ada	A	65	d	100	a	97	262	9
Leo	L	76	e	101	o	111	288	2
Sam	S	83	a	97	m	109	289	3
Lou	L	76	o	111	u	117	304	7
Max	M	77	a	97	x	120	294	8
Ted	T	84	e	101	d	100	285	10

Bea	Tim	Leo	Sam	Mia	Zoe	Jan	Lou	Max	Ada	Ted
0	1	2	3	4	5	6	7	8	9	10

*modulo:

$$279 \% 11 = 4$$

$$298 \% 11 = 1$$

→ 11 adalah banyaknya slot array yang dideklarasikan

Teknik 3: Folding

- Folding
 - Membagikan key menjadi jumlah digit yang sama, lalu jumlahkan
 - Bisa dibalik, dihilangkan suatu angka, dan ditambahkan suatu angka agar jumlah key menjadi sama
 - Angka yang dipilih sesuai dengan keinginan programmer

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		

U	D	I	N
21	4	9	14
41 + 94 + 12			
147			

O	K	I
15	11	9
91 + 15		
106		

Teknik 4: Digit Extraction

- Digit Extraction
 - Mengambil digit digit tertentu dari sebuah key
 - Angka yang dipilih sesuai dengan keinginan programmer

U	D	I	N		O	K	I
21	4	9	14		15	11	9
2	4	1			1	1	9

Teknik 5: Rotating Hash

- Rotating Hash
 - membalik urutan key

U	D	I	N		O	K	I
21	4	9	14		15	11	9
4	1	9	4	2	1	9	1

Teknik 6: Truncation

- Truncation
 - Mengambil sebagian key dengan cara memenggal
 - Urutan pemenggalan sesuai dengan keinginan programmer

U	D	I	N		O	K	I
21	4	9	14		15	11	9
4	1	9	4	2	1	9	1



COLLISION

COLLISION

- Akan ada suatu kondisi dimana indeks hasil hash function memiliki nilai yang kembar dan mengakibatkan tubrukan indeks (**COLLISION**). Hal ini akan mengakibatkan pemetaan indeks menjadi kacau.
- Jika terdapat nilai indeks yang kembar, bisa dilakukan teknik untuk mengatasi indeks yang saling bertubrukan, Hal ini dinamakan dengan teknik **COLLISION RESOLUTION**
- Terdapat 2 teknik yang kerap digunakan, yakni **Open Addressing** dan **Closed Addressing**

COLLISION



Institut Teknologi
Telkom
Purwokerto

- **Contoh**

- Dari ilustrasi di samping, hasil dari hash function, "Mia" memiliki nilai indeks 4
- "Sue" juga memiliki nilai indeks 4
- Agar tidak mengalami tubrukan, Sue dialihkan ke slot yang kosong setelah Mia.
- "Sue" sekarang menempati indeks 6

Mia	M	77	i	105	a	97	279	4
Tim	T	84	i	105	m	109	298	1
Bea	B	66	e	101	a	97	264	0
Zoe	Z	90	o	111	e	101	302	5
Sue	S	83	u	117	e	101	301	4

Bea	Tim			Mia	Zoe					
0	1	2	3	4	5	6	7	8	9	10

Bea	Tim			Mia	Zoe	Sue				
0	1	2	3	4	5	6	7	8	9	10

COLLISION

CHALLENGE

- Buatlah rangkuman mengenai teknik collision resolution berikut:
 - a. (closed hashing) Linear probing
 - b. (closed hashing) Double hashing
 - c. (Open hashing) Chaining



Institut Teknologi
Telkom
Purwokerto

TERIMA KASIH

SOURCE:

https://www.youtube.com/watch?v=KyUTuwz_b7Q&t=48s