

# Pencarian (Searching)

---

BAB 8

# Pencarian (Searching 2)

---

Sequential Searching

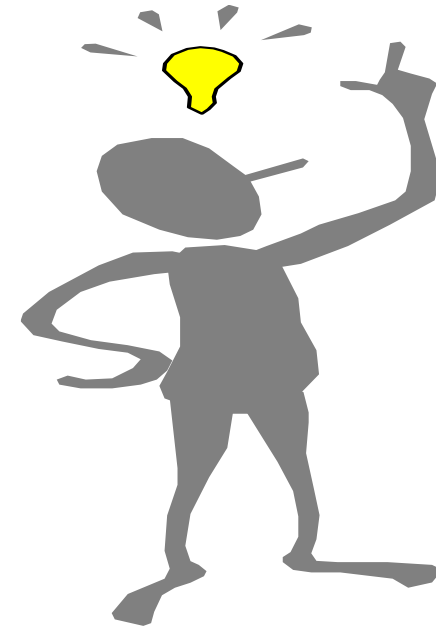
Binary Search

# SEARCH (PENCARIAN)

---

## Goal:

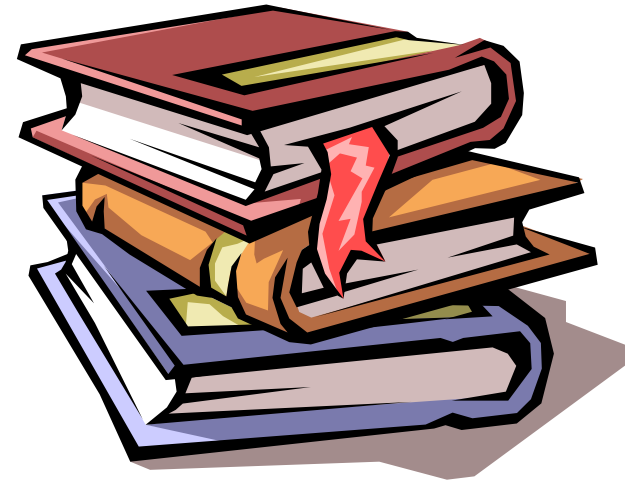
- *Dapat memahami konsep dan algoritma proses pencarian data (searching)*



# SEARCHING

---

- Sequential search
- Binary search



# Searching

---

Adalah proses mendapatkan (retrieve) information berdasarkan kunci (key) tertentu dari sejumlah informasi yang telah disimpan

Kunci (key) digunakan untuk melakukan pencarian record yang diinginkan didalam suatu list

# Searching

---

- Single match

Siapa mahasiswa dengan Nim 0800123456

- Multiple match

Siapa saja yang mendapat nilai Algoritma  $\geq 85$

# Metode Searching

---

**Sequential Search**

**Binary Search**

# Sequential Search

Merupakan teknik yang sederhana dan langsung dapat digunakan pada struktur data baik array maupun linked-list.

Pencarian data **secara urut** mulai dari data pertama sampai kunci yang dicari ditemukan atau sampai seluruh data telah dicari dan tidak ditemukan

Dilakukan pada data yang tidak terurut



# **(1) Sequential Search**

---

Disebut juga linear search atau Metode pencarian beruntun.

Tidak efisien untuk data yang list yang besar

Adalah suatu teknik pencarian data yang akan menelusuri tiap elemen satu per-satu dari awal sampai akhir.

Data awal = tidak harus dalam kondisi terurut.

# Algoritma Sequential Search

---

1. Input  $x$  (data yang dicari)
2. Bandingkan  $x$  dengan data **ke- $i$  sampai  $n$**
3. Jika ada data yang sama dengan  $x$  maka cetak pesan “Ada”
4. Jika tidak ada data yang sama dengan  $x$  cetak pesan “tidak ada”

# Ilustrasi Sequential Search

Misalnya terdapat array satu dimensi sebagai berikut:

0	1	2	3	4	5	6	7	indeks
8	10	6	-2	11	7	1	100	value

Kemudian program akan meminta data yang akan dicari, misalnya **6 (x = 6)**.

Iterasi :

6 = 8 (tidak!)

6 = 10 (tidak!)

6 = 6 (Ya!) => output : “Ada” pada index ke-2

Jika, sampai data terakhir tidak ditemukan data yang sama maka output : “ data yang dicari tidak ada”.

# Best & Worst Case

---

**Best case** : jika data yang dicari terletak di depan sehingga waktu yang dibutuhkan minimal.

**Worst case** : jika data yang dicari terletak di akhir sehingga waktu yang dibutuhkan maksimal.

Contoh :

**DATA = 5 6 9 2 8 1 7 4**

bestcase ketika  $x = 5$

worstcase ketika  $x = 4$

\* $x$  = key/data yang dicari

# Contoh Sequential Search

	Nim	Nama	IPK
[0]	2207023006	Mulyadi	2.94
[1]	2207023004	Willy Johan	3.15
[2]	2207023003	Anthony Liberty	2.78
[3]	2207023007	Ferry Santoso	3.37
[4]	2207023005	Jaya Mulya	2.93
[5]	2207023001	Budi Santoso	3.01
[6]	2207023008	Indra Gunawan	3.56
[7]	2207023002	M. Rudito W	3.44

# Contoh Sequential Search

---

Kunci pencarian? 2207023007

NIM[0] == kunci? → tidak

NIM[1] == kunci? → tidak

NIM[2] == kunci? → tidak

NIM[3] == kunci? → ya → Ferry Santoso, 3.37

# Contoh Sequential Search

Kunci pencarian? 2207023010

---

NIM[0] == kunci? → tidak

NIM[1] == kunci? → tidak

NIM[2] == kunci? → tidak

NIM[3] == kunci? → tidak

NIM[4] == kunci? → tidak

NIM[5] == kunci? → tidak

NIM[6] == kunci? → tidak

NIM[7] == kunci? → tidak

Semua data telah di cari, kunci tidak ditemukan

# BINARY SEARCH

Pencarian data dimulai **dari pertengahan data** yang telah terurut

---

Jika kunci pencarian lebih kecil daripada kunci posisi tengah, maka kurangi lingkup pencarian pada separuh data pertama

Begitu juga sebaliknya jika kunci pencarian lebih besar daripada kunci tengah, maka pencarian ke separuh data kedua

Teknik Binary Search hanya dapat **digunakan pada sorted array**, yaitu array yang elemen-elemennya telah terurut.



## (2) Binary Search

---

Lebih cepat dari sequential search

Teknik pencarian = data dibagi menjadi dua bagian untuk setiap kali proses pencarian.

Data awal harus dalam kondisi terurut. Sehingga harus dilakukan proses sorting terlebih dahulu untuk data awal.

Mencari posisi tengah :

$$\text{Posisi tengah} = (\text{posisi awal} + \text{posisi akhir}) / 2$$

# Algoritma Binary Search

---

1. Data diambil dari posisi awal 1 dan posisi akhir N
2. Kemudian cari posisi data tengah dengan rumus: **(posisi awal + posisi akhir) / 2**
3. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar?
4. Jika data sama, berarti ketemu.
5. Jika lebih besar, maka ulangi langkah 2 dengan posisi awal adalah **posisi tengah + 1**
6. Jika lebih kecil, maka ulangi langkah 2 dengan posisi akhir adalah **posisi tengah - 1**

# Algoritma Binary Search

N: banyak record array x

- 
1. kiri = 0 dan kanan = N-1
  2. tengah = (kiri+kanan)/2
  3. jika  $x[\text{tengah}] = \text{kunci}$  maka indeks = tengah. Selesai
  4. jika  $x[\text{tengah}] < \text{kunci}$  maka kiri = tengah+1
  5. jika  $x[\text{tengah}] > \text{kunci}$  maka kanan = tengah-1
  6. jika  $\text{kiri} \leq \text{kanan}$  dan  $x[\text{tengah}] \neq \text{kunci}$   
maka ulangi mulai dari 2
  7. jika  $x[\text{tengah}] \neq \text{kunci}$  maka indeks=-1
  8. selesai

# Ilustrasi

---

## Contoh Data:

Misalnya data yang dicari **23** ( $X = 23$ )

### Iterasi 1

0	1	2	3	4	5	6	7	8	
3		9	11	12	15	17	23	31	35
A					B				C

Karena  $23 > 15$  (data tengah), maka: awal = tengah + 1

### Iterasi 2

0	1	2	3	4	5	6	7	8	
3		9	11	12	15	17	23	31	35
					A	B		C	

$X = B$  (sama dengan data tengah). Output = "Data ditemukan"

# Best & Worst Case

---

**Best case** : jika data yang dicari terletak di posisi tengah.

**Worst case** : jika data yang dicari tidak ditemukan.

Contoh :

**DATA = 5 6 9 2 8 1 7 4 3**

bestcase ketika  $x = 8$  ( $T(n)=1$ )

worstcase ketika  $x = 25$  ( $T(n) = 5$  atau  $n/2$ )

\* $x$  = key/data yang dicari

# Contoh Binary Search

---

	Nim	Nama	IPK
[0]	2207023010	Mulyadi	2.94
[1]	2207023020	Willy Johan	3.15
[2]	2207023030	Anthony Liberty	2.78
[3]	2207023040	Ferry Santoso	3.37
[4]	2207023050	Jaya Mulya	2.93
[5]	2207023060	Budi Santoso	3.01
[6]	2207023070	Indra Gunawan	3.56
[7]	2207023080	M. Rudito W	3.44

# Contoh Binary Search

Kunci pencarian? 2207023060

---

[0]	2207023010	← Lo	
[1]	2207023020		
[2]	2207023030		
[3]	2207023040	← Mid	
[4]	2207023050		← Lo
[5]	2207023060		← Mid
[6]	2207023070		
[7]	2207023080	← Hi	← Hi

Ditemukan pada indeks [5] → Budi Santoso 3.01

# Contoh Binary Search

Kunci pencarian? 2207023022

---

[0]	2207023010	← Lo	← Lo	
[1]	2207023020		← Mid	← Hi, Mid
[2]	2207023030		← Hi	← Lo, Hi, Mid
[3]	2207023040			← Lo
[4]	2207023050	← Mid		
[5]	2207023060			
[6]	2207023070			
[7]	2207023080			

← Hi

NIM 2207023022 tidak ada pada data



# Terima Kasih

---

Pertanyaan?