



# 10. Algoritma Pencarian (*Searching Algorithm*)

---

ARNA FARIZA

YULIANA SETIOWATI

# Capaian Pembelajaran

---

Mahasiswa memahami algoritma pencarian yaitu sequential search dan binary search.

Mahasiswa dapat mengimplementasikan fungsi algoritma pencarian.

# Materi

---

Algoritma Pencarian Sekuensial (*sequential search*)

Algoritma Pencarian Biner (*binary search*).

# Algoritma pencarian

---

Algoritma yang digunakan untuk mencari lokasi dari sebuah data yang diberikan (disebut kata kunci) pada kumpulan data yang telah ada.

Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

# Algoritma pencarian

---

Ada dua macam teknik pencarian yaitu pencarian sekuensial (*sequential search*) dan pencarian biner (*binary search*).

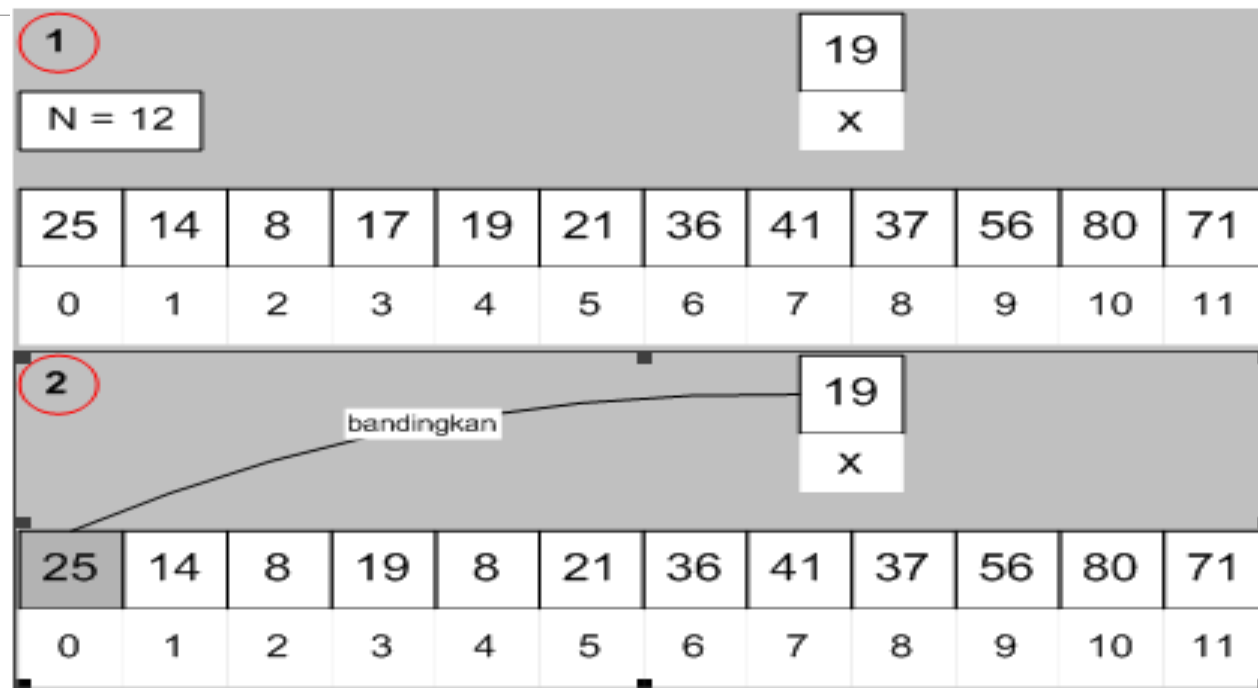
Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

# Pencarian Berurutan (*Sequential Searching*)

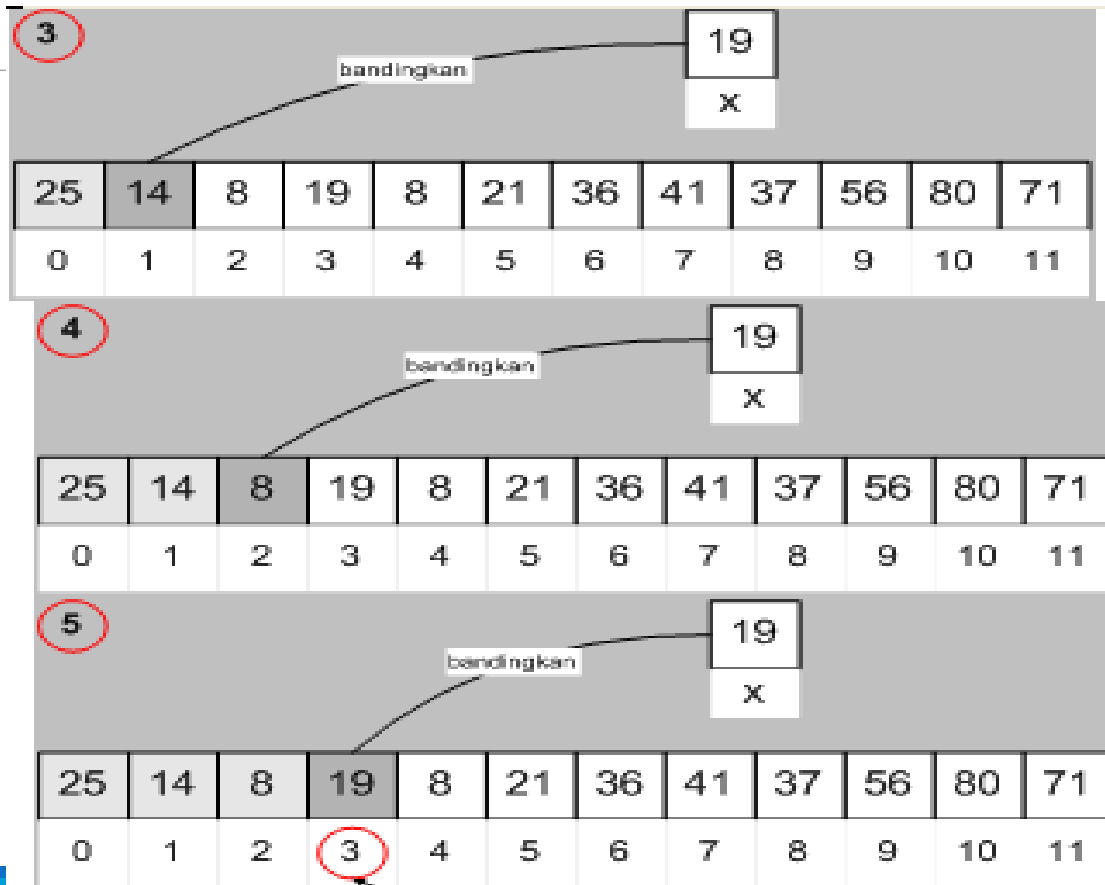
---

1.  $i \leftarrow 0$
2. ketemu  $\leftarrow$  false
3. Selama (tidak ketemu) dan ( $i \leq N$ ) kerjakan baris 4
4. Jika ( $\text{Data}[i] = x$ ) maka ketemu  $\leftarrow$  true, jika tidak  $i \leftarrow i + 1$
5. Jika (ketemu) maka  $i$  adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

# Ilustrasi



# Ilustrasi





# Pencarian Biner (Binary Search)

---

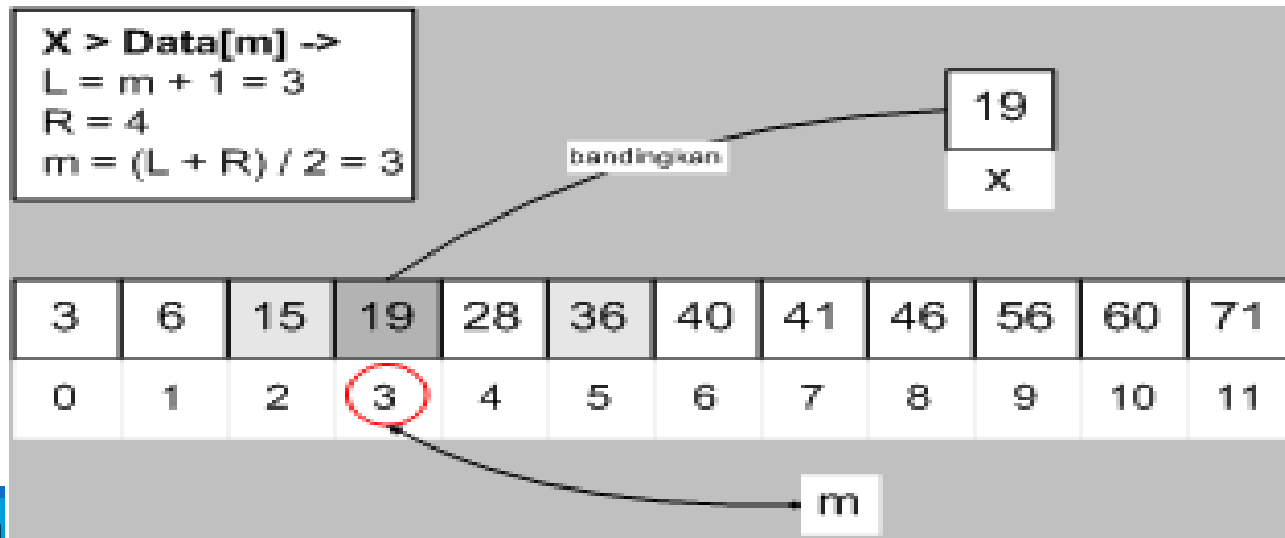
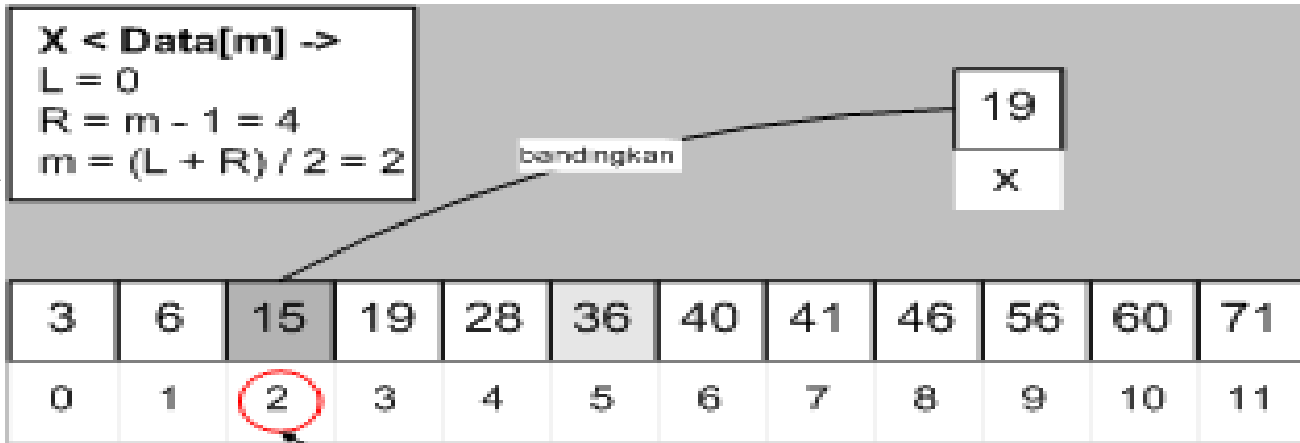
1.  $L \leftarrow 0$
2.  $R \leftarrow N - 1$
3.  $ketemu \leftarrow false$
4. Selama ( $L \leq R$ ) dan (tidak ketemu) kerjakan baris 5 sampai dengan 8
5.  $m \leftarrow (L + R) / 2$
6. Jika ( $Data[m] = x$ ) maka  $ketemu \leftarrow true$
7. Jika ( $x < Data[m]$ ) maka  $R \leftarrow m - 1$
8. Jika ( $x > Data[m]$ ) maka  $L \leftarrow m + 1$
9. Jika (ketemu) maka  $m$  adalah indeks dari data yang dicari, jika tidak data tidak ditemukan

# Ilustrasi Studi Kasus 1

N = 12						19						
						x						
3	6	15	19	28	36	40	41	46	56	60	71	
0	1	2	3	4	5	6	7	8	9	10	11	

L = 0 R = N - 1 = 11 m = (L + R) / 2 = 5									19			
						bandingkan						x
3	6	15	19	28	36	40	41	46	56	60	71	
0	1	2	3	4	5	6	7	8	9	10	11	
											m	

# Ilustrasi



# Binary Search

## - Studi Kasus 2

---

Pencarian data dengan target=7


[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

# Binary Search

---

Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

$$m = (0+6) / 2 = 3$$


# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

Is 7 = m ? NO.

# Binary Search

---

Example: sorted array of integer keys. Target=7.

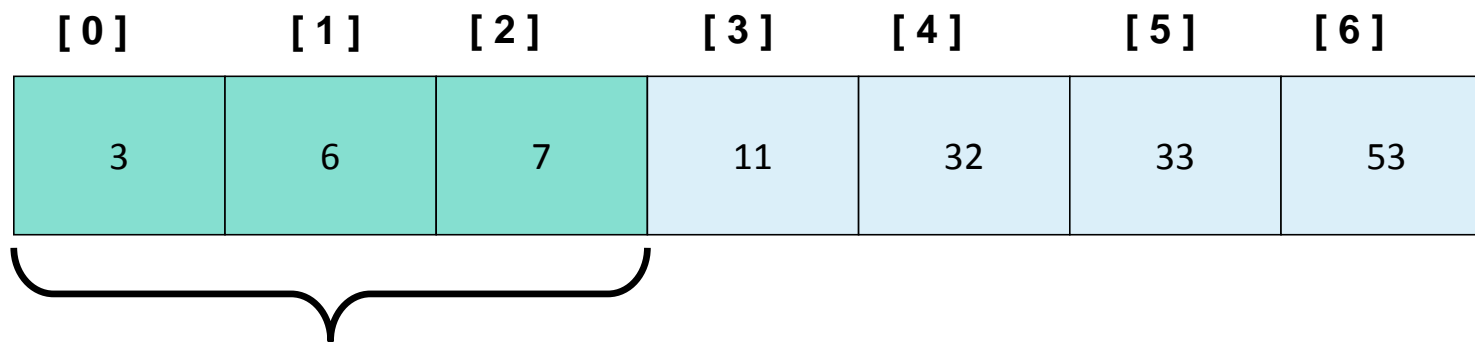
[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

↑  
Is  $7 < m$  ? YES.

# Binary Search

---

Example: sorted array of integer keys. Target=7.



Pencarian target pada area sebelum m




# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

$m = (0+2)/2 = 1$



# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Target = m ? NO.

# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Target < m ? NO.

# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



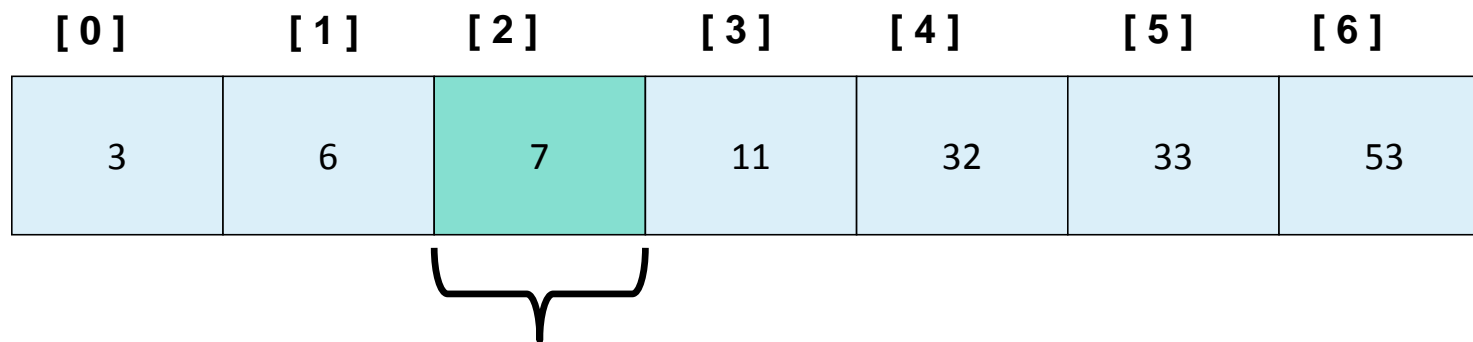
Target > m ? YES.

# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53



Pencarian target pada area setelah m

# Binary Search

---

Example: sorted array of integer keys. Target=7.

[0]	[1]	[2]	[3]	[4]	[5]	[6]
3	6	7	11	32	33	53

Is target = m? YES.

# Analisa Binary Search

---

Worst case complexity?

What is the maximum depth of recursive calls in binary search as function of  $n$ ?

Each level in the recursion, we split the array in half (divide by two).

Therefore maximum recursion depth is  $\text{floor}(\log_2 n)$  and worst case =  $O(\log_2 n)$ .

Average case is also =  $O(\log_2 n)$ .

# Can we do better than $O(\log_2 n)$ ?

---

Average and worst case of serial search =  $O(n)$

Average and worst case of binary search =  $O(\log_2 n)$



# Kesimpulan

---

Ada dua macam teknik pencarian yaitu pencarian sekuensial (*sequential search*) dan pencarian biner (*binary search*).

Algoritma yang digunakan untuk mencari lokasi dari sebuah data yang diberikan (disebut kata kunci) pada kumpulan data yang telah ada.

Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak terurut. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

# Latihan Soal

---

Dengan Pencarian Sekuensial, carilah data 9 dari data berikut

10 1 2 11 8 7 5 4 15 16

Dengan Pencarian Biner, carilah data 9 dari data berikut

2 5 6 9 10 12 15 18 20 22