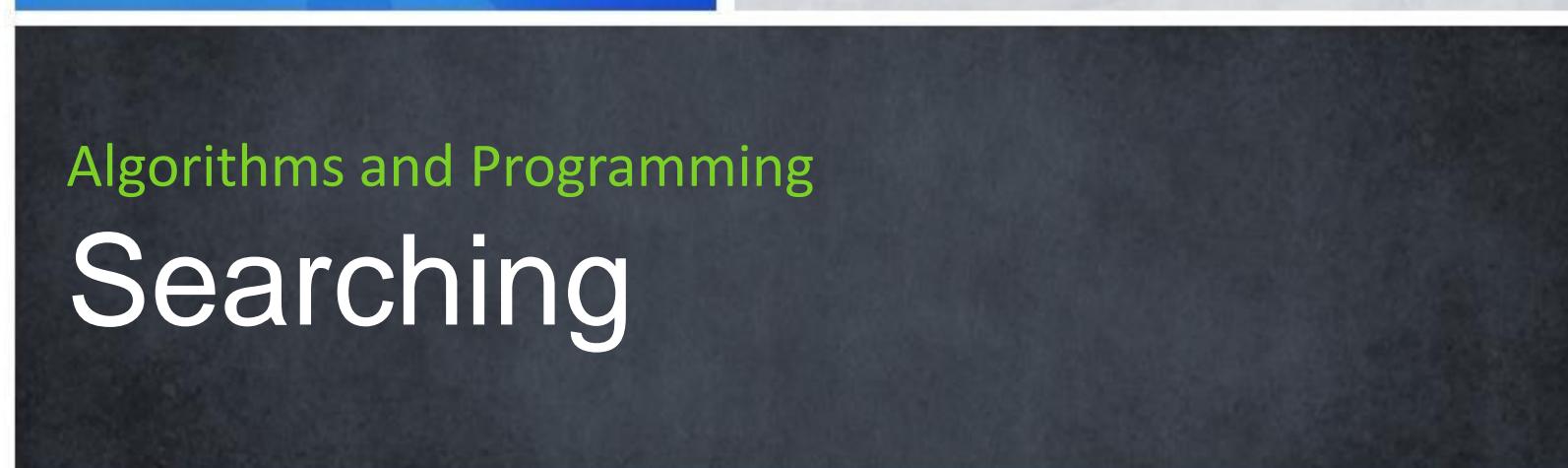


Adam Mukharil Bachtiar

English Class

Informatics Engineering 2011



Algorithms and Programming

# Searching



# Steps of the Day



Let's Start





# Definition of Searching

All About Searching



# What is Searching

Process that **search the value in group of data.**

This process can produce **FOUND** or **NOT FOUND**

**FOUND** value.



# Algorithms of Sorting

- Sequential search / Linear search
- Binary search





# Sequential Search

Definition and Structures of Sequential Search



## What is Sequential Search

- Trace group of data **one by one.**
- **Start** the process from **the first data.**
- If the data **was found** in group then **stop the searching** but **if not, search until the last data in grup.**

# Methods in Sequential Search

- Without **boolean**
  - Without **sentinel**
  - Use **sentinel**
- Use **boolean**

# Illustration of Seq. Search Without Sentinel

Given an array to be processed:

Number	5	1	9	4	2
	[1]	[2]	[3]	[4]	[5]

Data that want to be sought : **9**

- Number[1] = 9?  $i \leftarrow i + 1$
- Number[2] = 9?  $i \leftarrow i + 1$
- Number[3] = 9?  $i$  (STOP SEARCH)

Result: **9 is found in number[3]**

# Sequential Search Without Sentinel

```
1 Procedure SeqSearchTanpaSentinel (Input nama_array:tipe_array)
2 {I.S. : elemen array [1..maks_array] sudah terdefinisi}
3 {F.S. : menampilkan hasil pencarian (ditemukan/tidak) }
4 Kamus:
5     i : integer
6     data_cari : tipedata
7 Algoritma:
8     input(data_cari)
9     i  $\leftarrow$  1
10    while(nama_array [i]  $\neq$  data_cari) and (i < maks_array) do
11        i  $\leftarrow$  i + 1
12    endwhile
13    if (nama_array[i] = data_cari)
14    then
15        output(data_cari,' ditemukan pada indeks ke-' ,i)
16    else
17        output(data_cari,' tidak ditemukan' )
18    endif
19 EndProcedure
```

# Sequential Search Use Sentinel

- Place the data that want to be sought in sentinel.
- Sentinel is additional index that was placed in max array + 1.
- If the data is found in sentinel that means the result is data is not found and vice versa.

# Illustration of Seq. Search Use Sentinel

Data that was sought: 9

Number	5	1	9	4	2	9
	[1]	[2]	[3]	[4]	[5]	[6]

Result: Data was found in Number[3]

Data that was sought: 10

Number	5	1	9	4	2	10
	[1]	[2]	[3]	[4]	[5]	[6]

Result: Data was not found

# Sequential Search Use Sentinel

```
1  Procedure SeqSearchSentinel (Input nama_array:tipe_array)
2  {I.S. : elemen array [1..maks_array] sudah terdefinisi}
3  {F.S. : menampilkan hasil pencarian (ditemukan/tidak) }
4  Kamus:
5      i : integer
6      data_cari : tipedata
7  Algoritma:
8      input(data_cari)
9      i ← 1
10     nama_array(maks_array + 1) ← data_cari
11     while (nama_array [i] ≠ data_cari) do
12         i ← i + 1
13     endwhile
14     if (i < maks_array+1)
15     then
16         output(data_cari,' ditemukan pada indeks ke-' ,i)
17     else
18         output(data_cari,' tidak ditemukan' )
19     endif
20 EndProcedure
```

# Sequential Search Use Boolean

- Its searching process is similar with another sequential search method.
- Involves one boolean variable.

# Illustration of Seq. Search Use Boolean

Given an array to be processed:

Number	5	1	9	4	2
	[1]	[2]	[3]	[4]	[5]

Data that want to be sought : **9**

- Number[1] = 9? FOUND  $\leftarrow$  FALSE
- Number[2] = 9? FOUND  $\leftarrow$  FALSE
- Number[3] = 9? FOUND  $\leftarrow$  TRUE (STOP SEARCH)

Result: **9 is found in number[3]**

# Sequential Search Use Sentinel

```
1  Procedure SeqSearchBoolean (Input nama_array:tipe_array)
2  {I.S. : elemen array [1..maks_array] sudah terdefinisi}
3  {F.S. : menampilkan data yg dicari ditemukan atau tidak ditemukan}
4  Kamus:
5      i : integer
6      ketemu : boolean
7      data_cari : tipedata
8  Algoritma:
9      input(data_cari)
10     i  $\leftarrow$  1
11     ketemu  $\leftarrow$  false
12     while (not ketemu) and (i  $\leq$  maks_array) do
13         if(nama_var_array(i) = data_cari)
14             then
15                 ketemu  $\leftarrow$  true
16             else
17                 i  $\leftarrow$  i + 1
18             endif
19         endwhile
20         if (ketemu)
21             then
22                 output(data_cari,' ditemukan pada indeks ke-',i)
23             else
24                 output(data_cari,' tidak ditemukan')
25             endif
26 EndProcedure
```



# Binary Search

Definition and Structures of Binary Search



# What is Binary Search

- Searching algorithm that **divide group of data into two parts** (left and right).
- **First, check data in the middle.** If same with the data that was sought then data is found. If not then continue searching process to left or right (based on condition).
- Group of data **must be sorted** before the searching process.

# Case Example of Binary Search

Data that was sought: 7

Number	3	7	12	15	29
	[1]	[2]	[3]	[4]	[5]

Result: ?

# Case Example of Binary Search

Data that was sought: 7

Number	3	7	12	15	29
	[1]	[2]	[3]	[4]	[5]

Result: ?

# Case Example of Binary Search

**Step 1 :** **Divide array into 2 parts.** Count the middle position (k) of array to start searching

$$\begin{aligned}k &= (\lvert a \rvert + \lvert b \rvert) \text{ div } 2 \\&= (1 + 5) \text{ div } 2 \\&= 3\end{aligned}$$

**la : lower bound (for index)**

**lb : upper bound (for index)**

3	7	12	15	29
[1]	[2]	[3]	[4]	[5]
la		k		lb

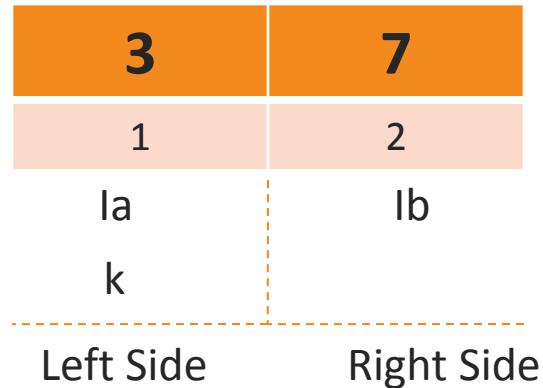
# Case Example of Binary Search

## Step 2 :

- **check data in k.** If it's same with data that was sought then **stop search** and **data is found.**
- If it's not then check whether data was **bigger** or **smaller** than data in k.
- If **it's bigger one** then **continue searching to right side and  $la = k+1$ .** if it's smaller one then **continue searching to the left side and  $lb = k-1$  (data wa sorted in ascending way).**

3	7
1	2
la	lb

# Case Example of Binary Search



Step 3 : **repeat** step 1 until step 2 until data is found or until  
 $la > lb$  then **stop searching.**

Result : 7 is found in **Number[2]** and in **the third looping.**

# Binary Search

```
1  Procedure BinarySearch (Input nama_array : tipe_array)
2  {I.S. : elemen array yang terurut secara ascending sudah terdefinisi}
3  {F.S. : menampilkan data yg dicari ditemukan atau tidak ditemukan}
4  Kamus:
5      Ia, Ib, k : integer
6      ketemu : boolean
7      data_cari : tipedata
8  Algoritma:
9      input(data_cari)
10     Ia  $\leftarrow$  1
11     Ib  $\leftarrow$  maks_array
12     ketemu  $\leftarrow$  false
13     while (not ketemu) and (Ia  $\leq$  Ib) do
14         k  $\leftarrow$  (Ia + Ib) div 2
15         if (nama_var_array[k] = data_cari)
16             then
17                 ketemu  $\leftarrow$  true
18             else
19                 if (nama_var_array[k] < data_cari)
20                     then
21                         Ia  $\leftarrow$  k + 1
22                     else
23                         Ib  $\leftarrow$  k - 1
24                     endif
25                 endif
26             endwhile
```

# Binary Search

```
27      if (ketemu)
28      then
29          output(data_cari,' ditemukan pada indeks ke-' ,k)
30      else
31          output(data_cari,' tidak ditemukan')
32      endif
33 EndProcedure
```

THANK YOU

**GRACIAS**

**Contact Person:**

Adam Mukharil Bachtiar  
Informatics Engineering UNIKOM  
Jalan Dipati Ukur Nomor. 112-114 Bandung 40132  
Email: [adfbipotter@gmail.com](mailto:adfbipotter@gmail.com)  
Blog: <http://adfbipotter.wordpress.com>