

**MODUL MATA KULIAH**

# **ANALISIS DAN DESAIN ALGORITMA**

**PG167 – 3 SKS**



**FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS BUDI LUHUR**

**JAKARTA  
SEPTEMBER 2019**

**TIM PENYUSUN**

Ita Novita, S.Kom., M.T.I  
Dr. Achmad Solichin, S.Kom., M.T.I  
Atik Ariesta, S.Kom., M.Kom



## MODUL PERKULIAHAN #9

# SEARCHING ARRAY SATU DIMENSI

Capaian Pembelajaran	:	Mahasiswa mampu memahami konsep pencarian (searching) pada array satu dimensi
Sub Pokok Bahasan	:	1.1. Pencarian pada array satu dimensi 1.2. Contoh algoritma pencarian sekuensial (sequential searching)
Daftar Pustaka	:	1. Gaddis, nd.2011. Starting Out with C++ from Control Structures through Objects .8th. Boston: Addison-Wesley. 2. Institue of Distance & Open Learning, n.d. UNIT I Algorithms, Flowcharts & Program Design in: INTRODUCTION TO C++. p. 205 3. Sjukani,Moh .2014. Algoritma (Algoritma & Struktur Data 1) Dengan C, C++, dan Java Edisi 9", Mitra Wacana Media.

# SEARCHING ARRAY SATU DIMENSI

## 1.1. PENCARIAN PADA ARRAY SATU DIMENSI

### PENGERTIAN SEARCH

Secara umum search diartikan mencari data dengan cara menelusuri tempat penyimpanan data tersebut. Tempat penyimpanan data dapat berupa array dalam memory, bisa juga berada dalam suatu file pada external storage. Searching adalah proses mencari atau pencarian. Data yang disearch adalah data yang berada dalam array satu dimensi. Untuk data yang berada dalam array satu dimensi, ada beberapa cara searching atau proses pencarian diantaranya:

1. Sequential Search
2. Index Sequential Search
3. Binary Search
4. Fibonacci Search

Dalam modul ini yang akan dibahas adalah **Sequential Search**.

## 1.2. CONTOH ALGORITMA PENCARIAN SEKUENSIAL

### PENGERTIAN PENCARIAN SEKUENSIAL

Pencarian sekuensial merupakan metode pencarian data dalam array dengan cara membandingkan data yang dicari dengan data yang ada di dalam array secara berurutan.

Pencarian data dengan metode sequential search efektif untuk mencari data yang dalam posisi tidak teratur atau acak. Prosesnya bisa dijelaskan seperti berikut:

1. Menentukan data yang dicari
2. Membaca data array satu per satu secara sekuensial
3. Mulai dari data pertama sampai dengan data terakhir, kemudian data yang dicari dibandingkan dengan masing-masing data yang ada di dalam array. Jika data ditemukan, maka dapat dibuat statement bahwa data telah



ditemukan. Namun jika tidak ditemukan, maka dibuat statement bahwa data tidak ditemukan.

## SEQUENTIAL SEARCH MENCARI ADA ATAU TIDAK ADA SUATU NILAI DALAM ARRAY SATU DIMENSI

---

### Contoh 1:

Sudah ada array A satu dimensi yang dibuat dengan int A[11], sudah ada isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	8	3	16	19

Susun algoritma dan flowchart untuk menginput sebuah nilai integer (misal N), kemudian periksa isi array, apakah ada isi array yang (nilainya) sama dengan N. Bila ada, cetak perkataan "**ADA**", bila tidak ada cetak perkataan "**TIDAK ADA**".

**Jawab:**

**Cara 1:**

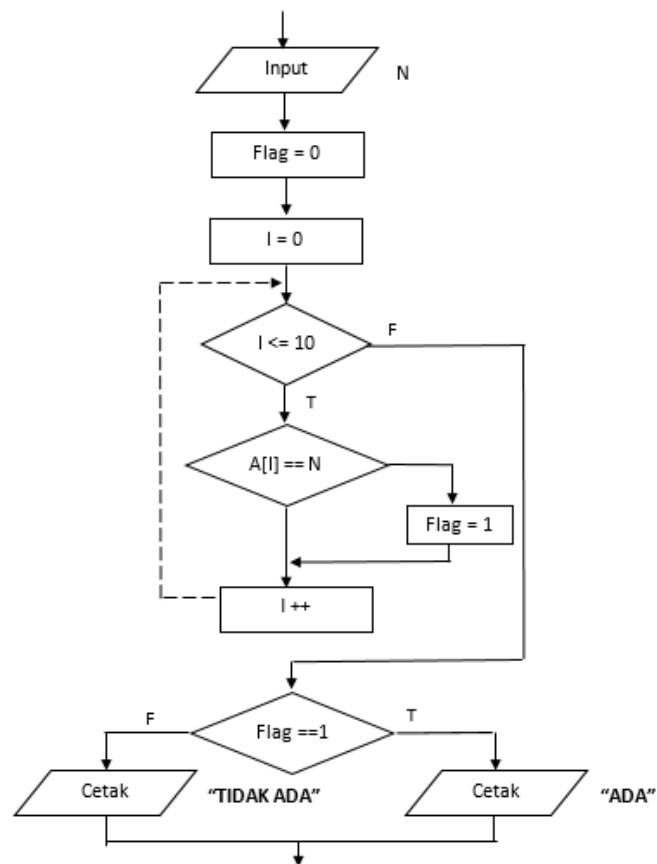
**Algoritma:**

```
Deklarasi variabel A[11]={12,17,10,5,15,25,11,8,3,16,19}, N,
I, Flag

Input Nilai N
Flag = 0
I=0
Lakukan selama I<=10
    Jika array A[I] == N
        Maka Flag = 1
    I=I+1

Cek apakah Flag == 1
    Jika Ya, Cetak "ADA"
    Jika Tidak, Cetak "TIDAK ADA"
```

## Flowchart:



## Program 11.1 Program Contoh 1 (Cara 1) Dalam Bahasa C

```
#include<stdio.h>
void main()

{ int A[[11]={ 12,17,10,515,25,11,8,3,16,19}, N,I, Flag;

  scanf ("%i", &N);
  Flag= 0;
  I=0;
  while (I<=10)
  { if (A[I] == N)
    Flag = 1;
    I++;
  }

  if (Flag == 1)
    printf ("ADA");
  else
    printf ("TIDAK ADA");
}
```



## Cara 2:

### Algoritma:

```
Deklarasi variabel A[11]={12,17,10,515,25,11,8,3,16,19}, N,  
I, Jum
```

```
Input Nilai N
```

```
Jum = 0
```

```
I=0
```

```
Lakukan selama I<=10
```

```
    Jika array A[I] == N
```

```
        Maka Jum=Jum+1
```

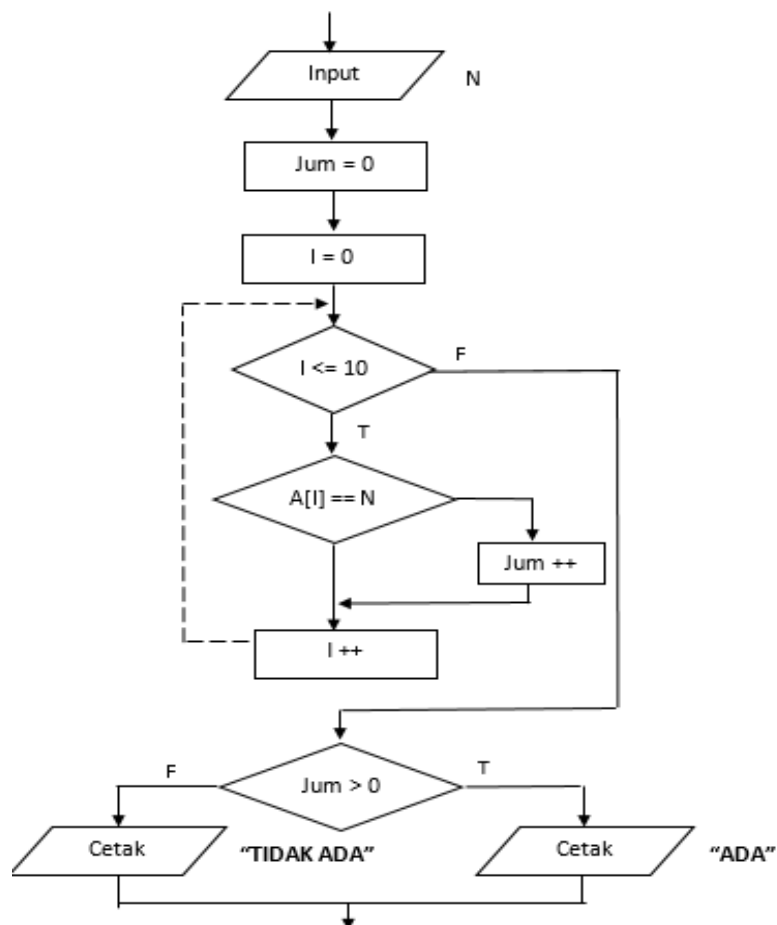
```
    I=I+1
```

```
Cek apakah Jum > 0
```

```
    Jika Ya, Cetak "ADA"
```

```
    Jika Tidak, Cetak "TIDAK ADA"
```

### Flowchart:



## Program 11.2 Program Contoh 1 (Cara 2) Dalam Bahasa C

```
#include<stdio.h>
void main()

{ int A[[11]={ 12,17,10,515,25,11,8,3,16,19}, N,I, Jum;

    scanf ("%i", &N);
    Jum= 0;
    I=0;
    while (I<=10)
    {   if (A[I] == N)
        Jum++;
        I++;
    }

    if (Jum > 0)
        printf ("ADA");
    else
        printf ("TIDAK ADA");
}
```

### Cara 3:

#### Algoritma:

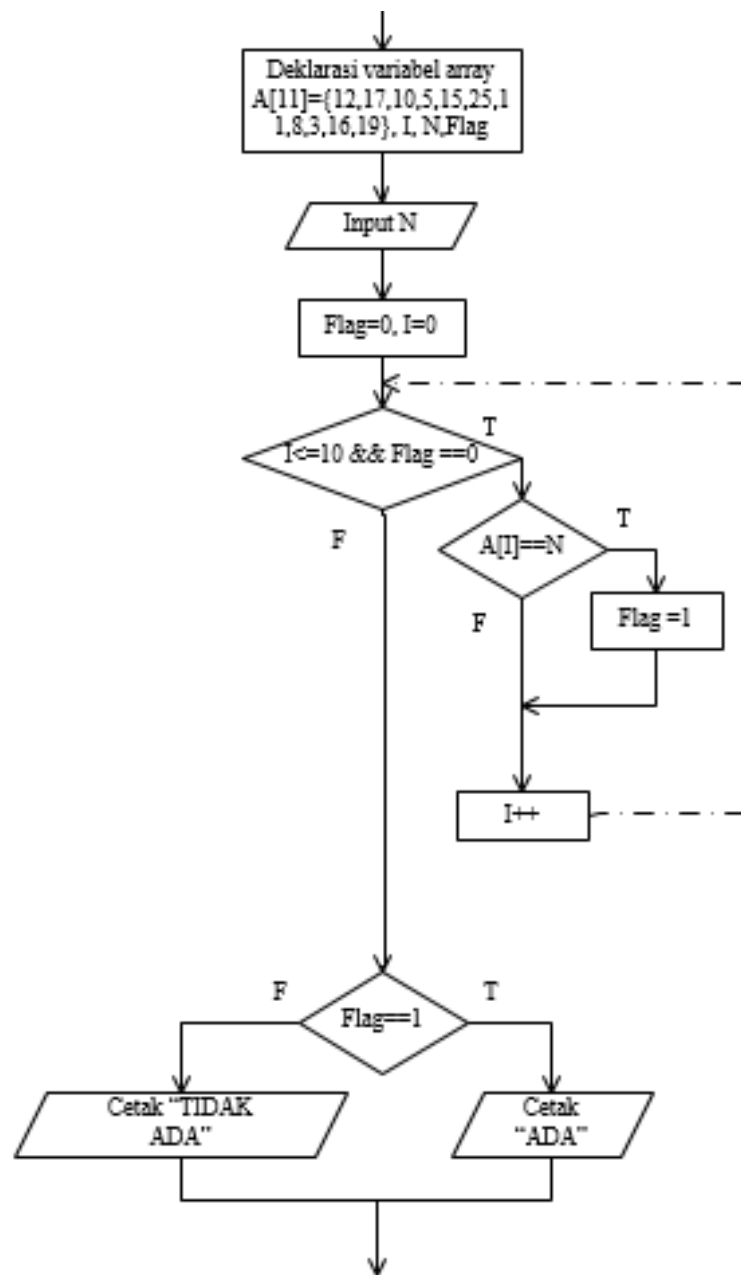
```
Deklarasi variabel A[11]={12,17,10,5,15,25,11,8,3,16,19},
N, I, Flag

Input Nilai N
Flag = 0
I=0
Lakukan selama I<=10 dan Flag == 0
    Jika array A[I] == N
        Maka Flag = 1
    I=I+1

Cek apakah Flag == 1
    Jika Ya, Cetak "ADA"
    Jika Tidak, Cetak "TIDAK ADA"
```



## Flowchart:





### Program 11.3 Program Contoh 1 (Cara 3) Dalam Bahasa C

```
#include<stdio.h>
void main()

{ int A[[11]={ 12,17,10,515,25,11,8,3,16,19}, N,I, Flag;

  scanf ("%i", &N);
  Flag= 0;
  I=0;
  while (I<=10 && Flag == 0)
  {   if (A[I] == N)
      Flag = 1;
      I++;
  }

  if (Flag == 1)
      printf ("ADA");
  else
      printf ("TIDAK ADA");
}
```

#### Contoh 2:

Sudah ada array satu dimensi yang dibuat dengan int A[11]. Sudah ada isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	8	3	16	19

Susun algoritma dan flowchart untuk menginput sebuah nilai integer (misal N), kemudian periksa isi array, dan cetak ada berapa buah isi array yang nilainya sama dengan nilai N.



## Jawab:

Deklarasi variabel  $A[11]=\{12,17,10,515,25,11,8,3,16,19\}$ ,  $N$ ,  
 $I$ ,  $Jum$

Input Nilai  $N$

$Jum = 0$

$I=0$

Lakukan selama  $I \leq 10$

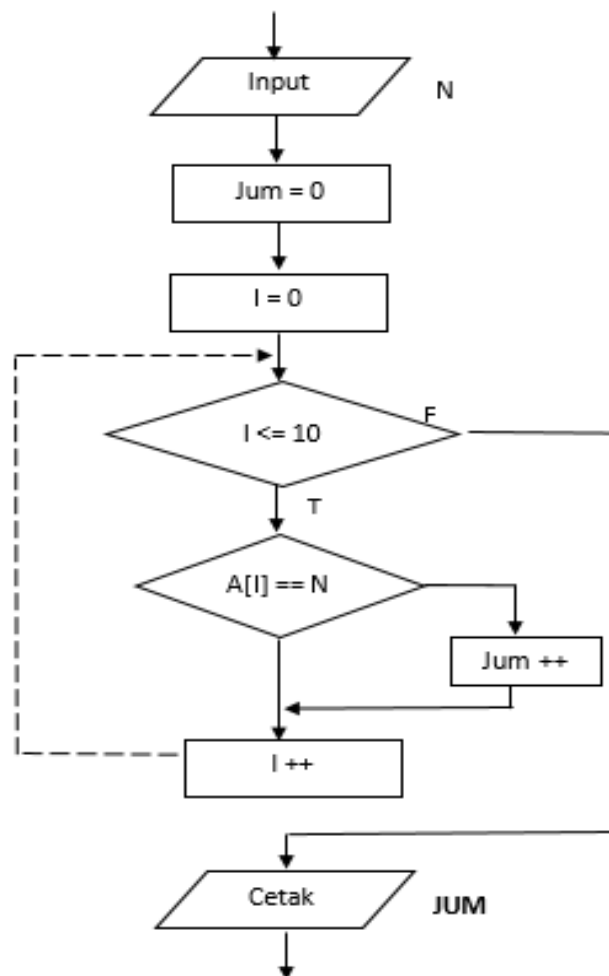
    Jika array  $A[I] == N$

        Maka  $Jum=Jum+1$

$I=I+1$

Cetak  $JUM$

## Flowchart:



### Program 11.4 Program Contoh 2 Dalam Bahasa C

```
#include<stdio.h>
void main()

{ int A[[11]={ 12,17,10,515,25,11,8,3,16,19}, N,I, Jum;

  scanf ("%i", &N);
  Jum= 0;
  I=0;
  while (I<=10)
  {   if (A[I] == N)
        Jum++;
        I++;
  }

  printf("%i Buah", Jum);
}
```

#### Contoh 3:

Sudah ada array satu dimensi yang dibuat dengan int A[11]. Sudah ada isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9	10
12	17	10	5	15	25	11	8	3	16	19

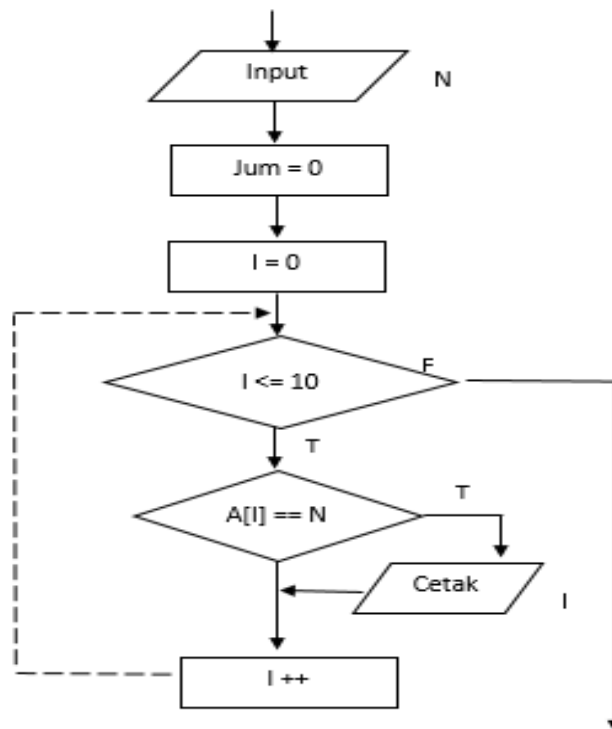
Susun algoritma dan flowchart untuk menginput sebuah nilai integer (misal N), kemudian periksa isi array, dan cetak ada di lokasi mana saja isi array yang nilainya sama dengan N.

#### Jawab:

```
Deklarasi variabel A[11]={12,17,10,515,25,11,8,3,16,19}, N, I, Jum
Input Nilai N
Jum = 0
I=0
Lakukan selama I<=10
    Jika array A[I] == N
        Cetak I
I=I+1
```



## Flowchart:



## Program 11.5 Program Contoh 3 Dalam Bahasa C

```
#include<stdio.h>
void main()

{ int A[[11]={ 12,17,10,515,25,11,8,3,16,19}, N,I, Jum;

  scanf ("%i", &N);
  Jum= 0;
  I=0;
  while (I<=10)
  {  if (A[I] == N)
      printf("%i", I);
      I++;
  }
}
```

## SEQUENTIAL SEARCH MEMERIKSA APAKAH DIANTARA 2 BUAH ARRAY ADA NILAI YANG SAMA

---

### Contoh:

Sudah ada array satu dimensi A dan B yang dibuat dengan char A[5] dan char B[7]. Sudah ada isinya huruf – huruf kapital tanpa spasi dengan contoh ilustrasi sebagai berikut:

	0	1	2	3	4	
A	B	O	G	O	R	

	0	1	2	3	4	5	6
B	J	A	K	A	R	T	A

Susun algoritma dan flowchart untuk memeriksa apakah diantara isi array A ada yang sama dengan isi array B. Bila ada cetak perkataan "ADA", bila tidak cetak perkataan "TIDAK ADA".

### Jawab:

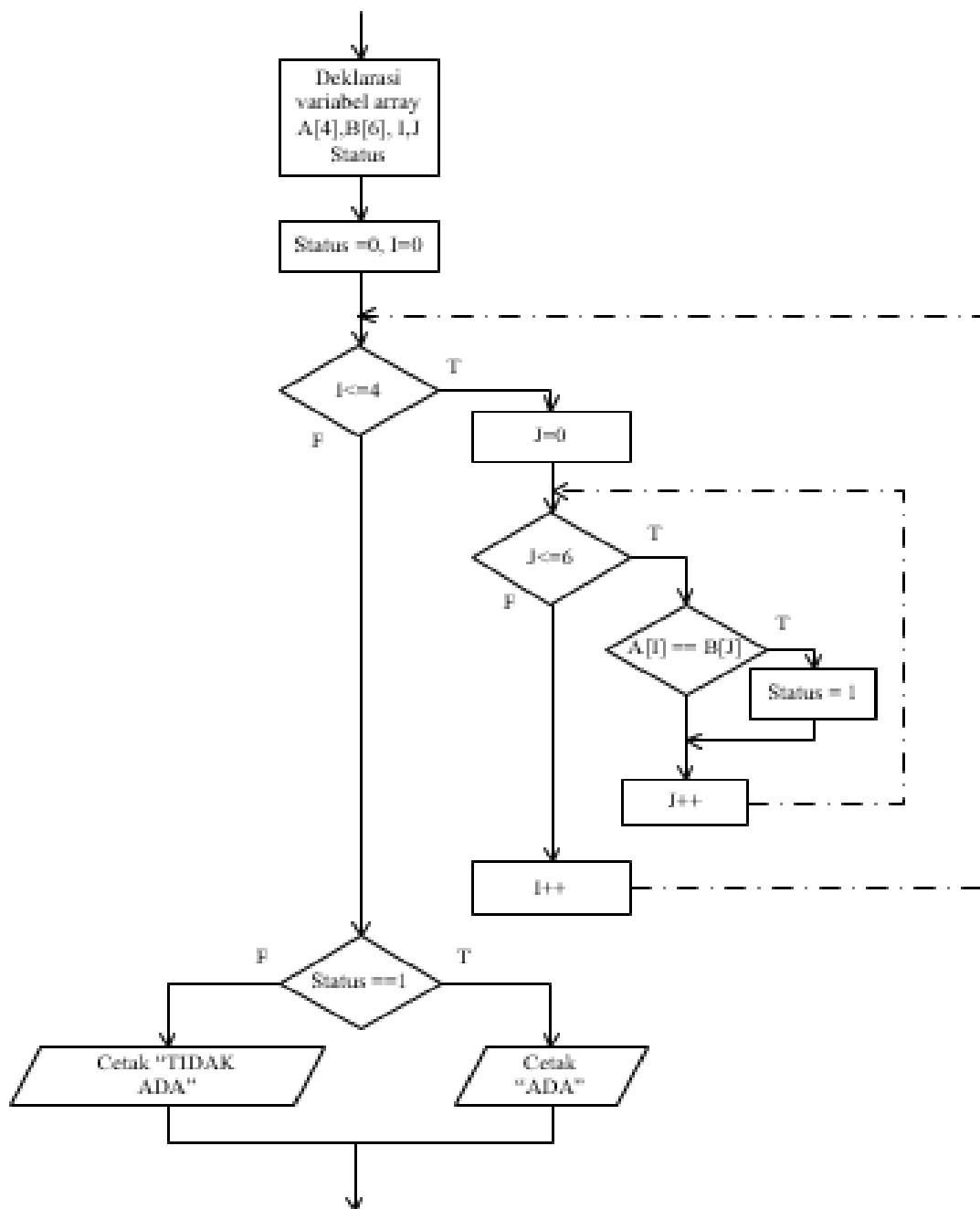
```
Deklarasi variabel A[5]="BOGOR", B[7]="JAKARTA", Status, I, J

Status = 0
I=0
Lakukan selama I<=4
    J=0
    Lakukan selama J<=6
        Cek apakah A[I] == B[J]
            Jika Ya, maka Status = 1
            J++
    I++

Cek jika Status == 1
    Jika benar, maka Cetak "ADA"
    Jika salah, maka cetak "TIDAK ADA"
```



## Flowchart:



## Program 11.6 Program Sequential Search Dalam Bahasa C

```
#include<stdio.h>
void main()

{ char A[5]="BOGOR", B[7]="JAKARTA";
  int Status,I, J;

  Status = 0;
  I=0;
  while (I<=4)
  { J=0;
    while (J<=6)
      { if (A[I] == B[J])
        Status = 1;
        J++;
      }
    I++;
  }

  if (Status == 1)
    printf("ADA");
  else
    printf("TIDAK ADA");
}
```



## SOAL LATIHAN

1. Sudah ada array A yang dibuat dengan `int A[11]`, sudah ada isinya yaitu nilai-nilai ujian mahasiswa. Susun algoritma untuk memeriksa apakah ada mahasiswa yang mendapat nilai 90 keatas. Bila ada cetak perkataan "ADA", bila tidak ada cetak perkataan "TIDAK ADA".
2. Sudah ada array A satu dimensi yang dibuat dengan `int A[10]`. Sudah ada isinya dengan ilustrasi sebagai berikut:

0	1	2	3	4	5	6	7	8	9
12	15	7	10	25	2	17	25	5	20

Susun algoritma untuk menginput sebuah nilai integer (misal N), kemudian periksa isi array.

- a. Bila ada isi array yang nilainya sama dengan N, maka cetak perkataan "FOUND", bila tidak ada isi array yang nilainya sama dengan N, maka cetak perkataan "NOT FOUND"
- b. Bila ada isi array yang nilainya sama dengan N, maka cetak perkataan "FOUND", dan mencetak ada berapa buah nilai yang sama dengan N. Bila tidak ada isi array yang nilainya sama dengan N, maka cetak perkataan "NOT FOUND"
- c. Bila ada isi array yang nilainya sama dengan N, maka cetak perkataan "FOUND", dan mencetak ada berapa buah nilai yang sama dengan N, serta mencetak nomor lokasi tempat nilai yang sama dengan N berada. Bila tidak ada isi array yang nilainya sama dengan N, maka cetak perkataan "NOT FOUND"





## KESIMPULAN

1. Untuk mencari data dalam suatu deret array, kita bisa menggunakan konsep searching salah satunya yaitu sequential searching, yaitu mencari data secara urut dari suatu array.
2. Sequential searching memiliki kekurangan dan kelebihan. Kelebihannya adalah lebih mudah untuk diimplementasikan dalam pemrograman. Sedangkan kekurangannya, jika data yang terdapat dalam array jumlahnya banyak, maka akan diperlukan waktu yang lebih lama untuk membandingkan data yang dicari dengan jumlah data yang sangat banyak dalam suatu array.





**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS BUDI LUHUR**

Jl. Raya Ciledug, Petukangan Utara, Pesanggrahan  
Jakarta Selatan, 12260  
Telp: 021-5853753 Fax : 021-5853752  
<http://fti.budiluhur.ac.id>