



## Pertemuan XII

### - **PENCARIAN**

- Pengertian
  - ✓ Pencarian data adalah suatu proses untuk mengumpulkan informasi dalam media penyimpanan komputer dan kemudian mencari kembali informasi yang diperlukan secepat mungkin.
  - ✓ Algoritma Pencarian adalah algoritma yang menerima sebuah argumen kunci dan langkah-langkah tertentu akan mencari data yang disimpan dengan kunci tersebut. Setelah proses dilaksanakan, kemungkinannya adalah data ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).
- Jenis – jenis
  - ✓ Pencarian Berurutan (*sequential search*) adalah pencarian yang membandingkan data yang ada satu per satu secara berurutan sampai data yang dicari ditemukan atau tidak ditemukan. Untuk N elemen data, harus dilakukan pencarian sebanyak N kali juga.

Langkah-langkah algoritma pencarian berurutan :

1.  $i \leftarrow 0$  { Inisialisasi i dengan nilai NOL }
2. Ketemu  $\leftarrow$  false { Inisialisasi Ketemu dengan FALSE }
3. Selama (tidak ketemu) dan ( $i \leq N$ ) maka kerjakan nomor 4
4. Jika ( $\text{Data}[i] = X$ ) maka ketemu bernilai TRUE, jika tidak tambahkan i dengan 1
5. Jika (ketemu bernilai TRUE), maka indeks dari data yang dicari adalah i, jika tidak, maka data tidak ditemukan.

### **ALGORITMA**

{Pencarian Berurutan / Sequential Search}

```
program Pencarian;  
{I.S : Data Array sudah terdefinisi}  
{F.S : Status hasil pencarian ditampilkan}
```

#### DEKLARASI

```
const Nmaks = 100  
type larik = array[1..Nmaks] of integer
```

```
ketemu : boolean  
i,N,dicari : integer  
A : larik
```

#### ALGORITMA

```
read(N)  
for i  $\leftarrow$  1 to N do  
    write('Data ke-',i,' : '  
    read(A[i])
```



```
endfor

read(dicari)

{PROSES PENCARIAN}
ketemu ← false
i ← 0
while (ketemu = false) and (i<=N) do
    if A[i] = dicari then
        ketemu ← true
    else
        {ketemu ← false}
        i ← i + 1
    endif
endwhile

if ketemu then
    write('Nilai ',dicari,' ditemukan di indeks ke-',i)
else
    write('Nilai ',dicari,' tidak ditemukan')
endif
```

## PASCAL

```
{Pencarian Beruntun / Sequential Search}

program Pencarian;
{I.S : Data Array sudah terdefinisi}
{F.S : Status hasil pencarian ditampilkan}

const Nmaks = 100;
type larik = array[1..Nmaks] of integer;

var
    ketemu                : boolean;
    i,N,dicari            : integer;
    A                    : larik;

begin
    writeln();
    writeln('PENCARIAN BERUNTUN');
    writeln('_____');
    writeln();
    writeln('Masukan Data');
    writeln('-----');
    write('Banyak Data : '); readln(N);
    writeln();

    for i:=1 to N do
        begin
            write('Data ke-',i,' : ');
            readln(A[i]);
            end;
        writeln();
        write('Data dicari : '); readln(dicari);
```



```
{PROSES PENCARIAN}

ketemu := false;
i := 1;
while (ketemu = false) and (i<=N) do
  begin
    if A[i] = dicari then
      ketemu := true
    else
      begin
        {ketemu := false}
        i := i + 1;
      end;
  end;

writeln();
writeln('_____');
writeln();
writeln('Hasil');
writeln('-----');
if ketemu then
  writeln('Nilai ',dicari,' ditemukan di indeks ke-',i)
else
  writeln('Nilai ',dicari,' tidak ditemukan');

writeln('_____');
readln();
end.
```



✓ Pencarian Bagi Dua (*binary search*)

Sebelum pencarian ini dilakukan, data yang akan diproses harus sudah berada dalam keadaan terurut.

Untuk N elemen data, dibutuhkan pencarian sebanyak  $2 \log N$ .

Untuk kasus data yang terurut menurun, maka :

1. Ketemu  $\leftarrow$  False {inisialisasi variabel ketemu dengan FALSE}  
Awal  $\leftarrow$  1 {Inisialisasi variabel awal dengan 1}  
Akhir  $\leftarrow$  N {Inisialisasi variabel akhir dengan N, N merupakan banyaknya data}
2. Tengah  $\leftarrow$  (awal + akhir)/2 {Tentukan indeks dari data yang berada di tengah}
3. Bandingkan data yang dicari (X) dengan data yang berada di tengah (A[tengah]).  
Jika sama, maka data ditemukan.  
Jika tidak sama, maka bandingkan lagi data yang dicari (X) dengan data yang berada di tengah (A[tengah]).  
Jika data yang dicari (X) lebih kecil, maka ubah indeks awal dengan nilai indeks data tengah+1.  
Jika tidak, ubah indeks akhir dengan nilai indeks data tengah-1.
4. Ulangi proses 2 dan 3 sampai data tengah sama dengan data yang dicari, selama indeks tengah tidak lebih besar dari N.

## ALGORITMA

{Pencarian Bagi Dua / Binary Search}

```
program Pencarian;  
{I.S : Data Array sudah terdefinisi}  
{F.S : Status hasil pencarian ditampilkan}
```

### DEKLARASI

```
const Nmaks = 100  
type larik = array[1..Nmaks] of integer
```

```
ketemu : boolean  
i,N,dicari : integer  
A : larik
```

### ALGORITMA

```
read(N)  
for i  $\leftarrow$  1 to N do  
    write('Data ke-',i,' : '  
    read(A[i])  
endfor  
  
read(dicari)
```



```
{PROSES PENCARIAN}
ketemu ← false
awal ← 1
akhir ← N
while (ketemu = false) and (awal<=akhir) do
  tengah ← (awal + akhir)div 2;
  if A[tengah] = dicari then
    ketemu ← true
  else
    {ketemu ← false}
    if dicari < A[tengah] then
      awal ← tengah+1
    else
      akhir ← tengah-1
    endif
  endif
endif
endwhile

if ketemu then
  write('Nilai ',dicari,' ditemukan di indeks ke-',tengah)
else
  write('Nilai ',dicari,' tidak ditemukan')
endif
```

## PASCAL

```
{Pencarian Bagi Dua / Binary Search - Menurun -}

program Pencarian;
{I.S : Data Array sudah terurut menurun}
{F.S : Status hasil pencarian terdefinisi}

const Nmaks = 100;
type larik = array[1..Nmaks] of integer;

var
  ketemu           : boolean;
  i,N,dicari       : integer;
  awal, akhir, tengah : integer;
  A                : larik;

begin
  writeln();
  writeln('PENCARIAN BAGI DUA');
  writeln('_____');
  writeln();
  writeln('Masukan Data Terurut Menurun');
  writeln('-----');
```



```
write('Banyak Data : '); readln(N);
writeln();
for i:=1 to N do
begin
write('Data ke-',i,' : ');
readln(A[i]);
end;
writeln();
write('Data dicari : '); readln(dicari);

{PROSES PENCARIAN}

ketemu := false;
awal := 1;
akhir := N;
while (ketemu = false) and (awal<=akhir) do
begin
tengah := (awal + akhir)div 2;
if A[tengah] = dicari then
begin
ketemu := true;
end
else
begin
ketemu := false;
if A[tengah] > dicari then
awal := tengah+1
else
akhir := tengah-1;
end;
end;

writeln();
writeln('_____');
writeln();
writeln('Hasil');
writeln('-----');
if ketemu then
writeln('Nilai ',dicari,' ditemukan di indeks ke-',tengah)
else
writeln('Nilai ',dicari,' tidak ditemukan');

writeln('_____');
readln();
end.
```