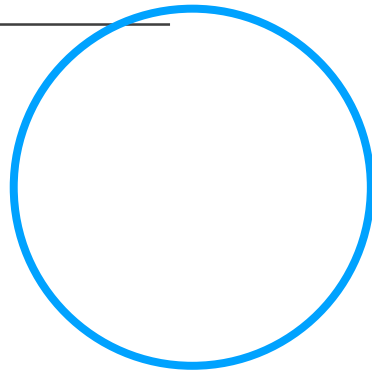




# Algoritma Pencarian (Searching)


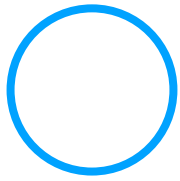
---

TIM DOSEN



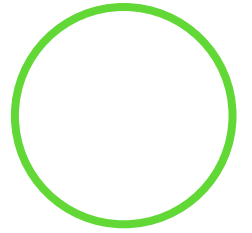


# OUTLINE

- a. Definisi Algoritma Pencarian
  - b. Spesifikasi Algoritma Pencarian
  - c. Metode Pencarian Data di dalam Array
    - Metode Pencarian Beruntun (*Sequential Search*)
    - Metode Pencarian Bagi Dua / Pencarian Biner (*Binary Search*)
- 
- 



# DEFINISI ALGORITMA PENCARIAN



- Pencarian (searching) merupakan proses yang fundamental dalam pemrograman.
- Proses pencarian adalah menemukan nilai (data) tertentu di dalam sekumpulan data yang bertipe sama (baik bertipe dasar atau bertipe bentukan).
- Mencari data dengan menelusuri tempat penyimpanan data.
- Tempat penyimpanan data dalam memory dapat berupa array/linked list.



# SPEKIFIKASI ALGORITMA PENCARIAN

- Hasil atau keluaran dari persoalan pencarian dapat bermacam-macam, bergantung pada spesifikasi rinci dari persoalan tersebut.
- Pencarian hanya untuk memeriksa keberadaan  $x$ . misal ditemukannya hasil yang dicari maka pesan ditampilkan bahwa data  $x$  yang dicari ditemukan.

Contoh : `cout << "ditemukan" ;` atau `cout << "tidak ditemukan" ;`

- Hasil pencarian adalah indeks elemen larik. Jika  $x$  ditemukan, maka indeks elemen array tempat  $x$  berada diisi ke dalam `idx`.

- Contoh : 

L	21	36	8	7	10	36	68	32	12	10	36
	1	2	3	4	5	6	7	8	9	10	11

Misalkan  $x = 68$ , maka `idx = 7`, dan bila  $x = 100$ , maka `idx = -1`.



# SPEKIFIKASI ALGORITMA PENCARIAN




➤ Hasil pencarian adalah sebuah nilai Boolean yang menyatakan status hasil pencarian. Misal jika s **ditemukan**, maka variable s bertipe Boolean diisi dengan nilai **true** atau **tidak ditemukan** bernilai **false**.

➤ Contoh :

➤ misal x = 60, maka ketemu=true, bila x =100, maka ketemu= false,

```
If ketemu then
  write(x, 'ditemukan')
else
  write(x, 'tidakditemukan')
end if
```





# METODE PENCARIAN DATA DI DALAM ARRAY


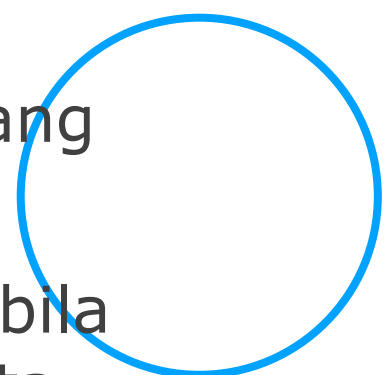

- a. Metode Pencarian Beruntun (*Sequential Search*)
- b. Metode Pencarian Bagi Dua / Pencarian Biner (*Binary Search*)
- c. Fibonacci Search (Tidak dibahas)
- d. Interpolation Search (Tidak dibahas)





## SEQUENTIAL SEARCHING (1)



- Adalah proses membandingkan setiap elemen Larik satu persatu secara beruntun, dari elemen pertama sampai elemen yang dicari ditemukan.
  - Data yang ada pada suatu array dibandingkan satu persatu dengan data yang dicari.
  - Pencarian ini hanya melakukan pengulangan dari 1 s.d. dengan jumlah data.
  - Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari.
  - Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan, tidak ada yang sama, berarti data tidak ada
- 
- 
- 



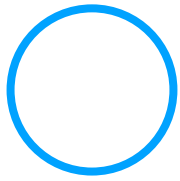

# CONTOH ILUSTRASI

Larik L dibawah ini

22	61	16	66	18	25
1	2	3	4	5	6

Misalkan, nilai yang dicari **66**

Maka elemen yang diperiksa : 22, 61, 16, 66 (ditemukan). Indeks Larik yang dikembalikan : **IX = 4**


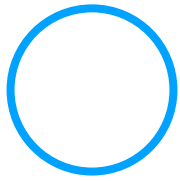






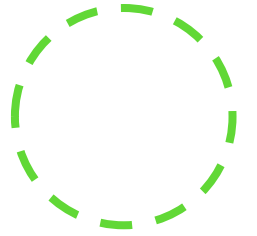
# ALGORITMA PENCARIAN SEKUENSIAL



1. Tentukan nilai  $i \leftarrow 1$ .
  2. Tentukan nilai Ketemu  $\leftarrow \text{False}$ .
  3. Selama (not ketemu) dan ( $i \leq N$ ) kerjakan baris 4.
  4. Jika ( $\text{Data}[i] = x$ ) maka ketemu  $\leftarrow \text{true}$ , jika tidak  $i \leftarrow i + 1$ .
  5. Jika (ketemu) maka  $i$  adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.
- 
- 



# ALGORITMA PENCARIAN SEKUENSIAL - LANJUT

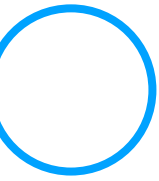


Algoritma Program :

1. Tentukan dan simpan data dalam suatu array
2. Tentukan fungsi pencarian sekuensial
3. Fungsi pencarian sekuensial adalah sebagai berikut :

```
int flag=-1;
for(int count=0; count < array_size; count++) {
    flag=count;
    break;
}
```

4. Masukkan data yang akan dicari
5. Kerjakan langkah 3, jika data ketemu kerjakan langkah 6. Jika data tidak ketemu lakukan langkah 6
6. Cetak data tersebut
7. Selesai



# Contoh Program Sequential Search pada Bahasa C++


```
/*Sequential Search*/
#include<iostream.h>
#include<conio.h>

//Fungsi
int search_01(int [], int, int);
main( )
{
    clrscr( );
    const int array_size= 11;
    int
    array[array_size]= {22,61,15,66,18,25,34,87,55,45,10};
    cout<<"*****"<<endl;
    cout<<"** S E Q U E N T I A L   S E A R C H   **"<<endl;
    cout<<"*****";
    gotoxy(1,5);
    cout<<"Isi data Array adalah : " <<endl;
    cout<<"\n  Array : " <<"\t\t      Data : " <<endl;
    for(int count=0; count<array_size; count++)
        {
            cout<<"\t" <<" Array ["<<count<<" ] " <<"\t\t";
            cout<<array[count]<<endl;
        }
    int search_elemen=0;
    int flag=0;
    cout<<"\n\n Masukkan data yang akan Anda cari ? ";
    cin>>search_elemen;

flag=search_01(array, array_size, search_elemen);
    if(flag != -1)
        cout<<"\n Data tersebut ditemukan pada posisi : array [ "<<flag<<" ]";
    else
        cout<<"\n Data yang Anda cari tidak ditemukan " ;
        getch( );
        return 0;
}

int search_01(int array[], int array_size, int elemen)
{
    int flag=-1;
    for(int count=0; count < array_size; count++)
        {
            if(elemen==array[count])
                {
                    flag=count;
                    break;
                }
        }
return flag;
}
```

# Output Sequential Search



```
*****
* S E Q U E N T I A L   S E A R C H   *
*****

Isi data Array adalah :

Array :                               Data :
Array [0 ]                             22
Array [1 ]                             61
Array [2 ]                             15
Array [3 ]                             66
Array [4 ]                             18
Array [5 ]                             25
Array [6 ]                             34
Array [7 ]                             87
Array [8 ]                             55
Array [9 ]                             45
Array [10 ]                            10

Masukkan data yang akan Anda cari ?
```



```
*****
* S E Q U E N T I A L   S E A R C H   *
*****

Isi data Array adalah :

Array :                               Data :
Array [0 ]                             22
Array [1 ]                             61
Array [2 ]                             15
Array [3 ]                             66
Array [4 ]                             18
Array [5 ]                             25
Array [6 ]                             34
Array [7 ]                             87
Array [8 ]                             55
Array [9 ]                             45
Array [10 ]                            10

Masukkan data yang akan Anda cari ? 87

Data tersebut ditemukan pada posisi : array [ 7]
```

# ALGORITMA PROGRAM

- BUAT  
PROGRAMNYA?

## PROGRAM Pencarian

{ Program untuk mencari nilai tertentu di dalam larik }

### DEKLARASI

const Nmaks = 100 { jumlah maksimum elemen larik }

type LarikInt : array[1..Nmaks] of integer

L : LarikInt

x : integer { elemen yang dicari }

found : boolean { true jika x ditemukan, false jika tidak }

n : integer { ukuran larik }

procedure BacaLarik(output L : LarikInt, input n : integer )

{ Mengisi elemen larik L[1..n] dengan nilai yang dibaca dari piranti masukan }

procedure SeqSearch1(input L : LarikInt, input n : integer,

input x : integer, output ketemu : boolean)

{ Mencari keberadaan nilai x di dalam larik L[1..n]. }

### ALGORITMA:

read(n) { tentukan banyaknya elemen larik }

BacaLarik(L, n) { baca elemen-elemen larik L }

read(x) { baca nilai yang dicari }

SeqSearch1(L, n, x, found) { cari }

if found then { found = true }

write(x, ' ditemukan!')

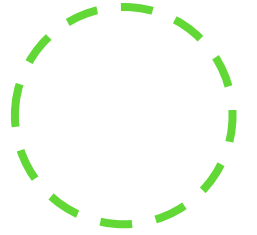
else

write(x, ' tidak ditemukan!')

endif



# BINARY SEARCHING (2)



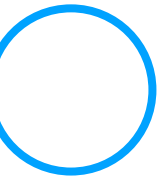
Pencarian sebuah elemen dalam sebuah array satu dimensi dengan cara selalu membandingkan dengan nilai yang berada di tengah array tersebut

Apabila tidak sama maka array akan dibagi dua dan pencarian diulang pada bagian dimana nilai yang dicari.

Salah satu syarat pencarian bagi dua (binary search) adalah **data sudah dalam keadaan terurut**

Apabila data belum keadaan terurut, pencarian biner tidak dapat dilakukan


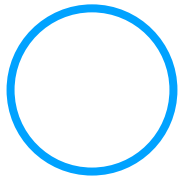
**Data yang terurut** merupakan **syarat mutlak** penerapan pencarian  
**Algoritma pencarian Bagi Dua**





# ALGORITMA PENCARIAN BINER


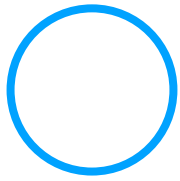


1. Pertama diambil posisi awal =1 dan posisi akhir =N, kemudian dicari posisi data tengah dengan rumus  $(\text{posisi awal} + \text{posisi akhir}) / 2$ .
  2. Kemudian data yang dicari dibandingkan dengan data tengah.
  3. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1.
  4. Jika lebih besar, proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah +1.
  5. Demikian seterusnya sampai data tengah sama dengan yang dicari.
- 
- 



# CONTOH : ALGORITMA PENCARIAN



1. Input N.
  2. Tentukan  $Lo=0$ ;  $Hi=N-1$ ; dan  $Flag=0$  (untuk tanda tidak ditemukan yang dicari)  $\rightarrow Mid = (Lo + Hi)/2$ .
  3. Selama  $Lo \leq Hi$  dan  $Flag==0$ 
    - Hitung  $Mid=(Lo+Hi)/2$
    - Jika  $N==A[Mid]$ , maka isi  $Flag=1$ ; (tanda ditemukan)
    - Jika  $N<A[Mid]$ , maka isi  $Hi=Mid-1$ , proses pencarian dibagian kiri
    - Jika  $N>A[Mid]$ , maka isi  $Hi=Mid+1$ , proses pencarian dibagian kanan
  4. Proses pencarian selesai.
- 
- 



# ILUSTRASI

$A[10] = \{4, 7, 10, 11, 16, 22, 24, 28, 63, 64\}$

Dicari angka  $N = 10$  dari array  $A[..]$

Hitung Mid :

$$MID = (Lo + Hi)/2 = (0+9)/2 = 4$$

$$A[Mid] = A[4] = 16$$

Periksa nilai N:

$$N > A[Mid] = 10 < 16$$

= true cari sebelah kiri

$$Hi = Mid-1 \rightarrow 9-1 = 8$$

Hitung Mid :

$$MID = (Lo + Hi)/2 = (0+8)/2 = 4$$

$$A[Mid] = A[4] = 16$$

Periksa nilai N:

$$N > A[Mid] = 10 < 16$$

= true cari sebelah kiri

$$Mid \rightarrow Lo = Mid-1 \rightarrow 8-1 = 7$$

Hitung Mid :

$$MID = (Lo + Hi)/2 = (0+7)/2 = 3$$

$$A[Mid] = A[3] = 11$$

Periksa nilai N:

$$N > A[Mid] = 10 < 11$$

= true cari sebelah kiri

$$Mid \rightarrow Lo = Mid-1 \rightarrow 7-1 = 6$$

Hitung Mid :

$$MID = (Lo + Hi)/2 = (0+6)/2 = 3$$

$$A[Mid] = A[3] = 11$$

Periksa nilai N:

$$N > A[Mid] = 10 < 11$$

= true cari sebelah kiri

$$Hi = Mid-1 \rightarrow 6-1 = 5$$



# ILUSTRASI



$A[10] = \{4, 7, 10, 11, 16, 22, 24, 28, 63, 64\}.$

Dicari angka  $N = 10$  dari array  $A[..]$

Hitung Mid :

$$MID = (Lo + Hi)/2 = (0+5)/2 = 2$$

$$A[Mid] = A[2] = 10$$

Periksa nilai  $N$ :

$$N = A[Mid] = 10 = 10$$

Maka  $N=10$  ditemukan



# ILUSTRASI

Sederatan data dalam array nilai sebanyak 10 elemen dan akan dilakukan pencarian data 87 terhadap array. Nilai[0..9] = 12,45,23,87,90,55,15,25,40,21

Urutkan elemen array secara menaik, sehingga diperoleh:

**0 1 2 3 4 5 6 7 8 9**

Nilai[0..9] = 12,15,21,23,**25**,40,45,55,**87**,90

Data yang akan dicari = 87(bilangan)

1. Tentukan nilai awal = 0, akhir = N-1=9
2. Hitung tengah =  $(9+0)/2=4$ , nilai[4] = 25 nilai indeks ke-4 (tengah)
  - Bandingkan **Bilangan** = **Nilai[tengah]** ->  $87=25$  -> false
  - Bandingkan **Bilangan** < **Nilai[tengah]** ->  $87<25$  -> false
  - Bandingkan **Bilangan** > **Nilai[tengah]** ->  $87>25$  -> true maka pencarian dilakukan ke sebelah kanan dengan nilai awal = tengah+1 = 5

**0 1 2 3 4 5 6 7 8 9**

Nilai[0..9] = 12,15,21,23,25,40,45,55,**87**,90



# ILUSTRASI- LANJUTAN

3. Karena awal masih lebih kecil dari akhir maka ulangi kembali mulai menghitung tengah.

1) Nilai awal = tengah+1 =  $4 + 1 = 5$

2) Hitung tengah =  $(9+5)/2=7$

- Bandingkan Bilangan < Nilai[tengah] ->  $87=55$ ->false
- Bandingkan Bilangan < Nilai[Tengah]->  $87<55$ ->false
- Bandingkan Bilangan < Nilai[tengah]->  $87>55$ ->true maka pencarian dilakukan ke sebelah kanan dengan nilai awal = tengah+1 = 8

4. Karena awal masih lebih kecil dari akhir maka ulangi kembali mulai menghitung tengah

1) Nilai awal = tengah+1 = 8

2) Hitung tengah =  $(9+8)/2 = 8$

- Bandingkan Bilangan < Nilai[tengah]->  $87=87$ ->true



5. Karena sudah ditemukan hasilnya maka proses pencarian berhenti.



# CONTOH PROGRAM BINARY SEARCHING PADA BAHASA C++

## Output Binary Searching

```
(Inactive E:\ALSTRU~1\SEARCHBLEX.E)
DAFTAR NILAI/ANGKA
4
7
10
11
16
22
24
28
63
66
Masukkan Data Yang dicari : 63
Nilai 63 ditemukan pada kolom ke 9
```

```
//Binary Search
//tidak menggunakan fungsi

#include<iostream.h>
#include<conio.h>
#define n 10

void main()
{
    int N;
    int A[n] = {4,7,10,11,16,22,24,28,63,66};
    int i,Lo, Hi, Mid, Flag;
    //Menampilkan daftar nilainya
    cout<<"DAFTAR NILAI/ANGKA "<<endl;
    for(i=0; i<=n-1; i++)
    {
        cout<< A[i] << endl;
    }
    //Menginput data yang mau dicari
    cout<<"Masukkan Data Yang dicari : "; cin>>N;
    Lo=0, Hi=n-1; Flag=0;
    while(Lo <= Hi && Flag==0)
    {
        Mid=(Lo+Hi)/2;
        if(N==A[Mid])
            Flag=1;
        else
            if(N<A[Mid])
                Hi=Mid-1;
            else
                Lo=Mid+1;
    }
    if(Flag==1)
        cout<<"Nilai "<<N<<" ditemukan pada kolom ke "<<(Mid+1)<<endl;
    else
        cout<<"Nilai tidak ditemukan"<<endl;
}
```

- CONTOH PROGRAM
- FUNGSI BINARY SEARCHING
- PADA BAHASA C++

## Output Binary Searching

```
(Inactive E:\ALSTRU~1\SEARCHBLEXE)
DAFTAR NILAI/ANGKA
4
7
10
11
16
22
24
28
63
66
Masukkan Data Yang dicari : 63
Nilai 63 ditemukan pada kolom ke 9
```

```
#include<iostream.h>
#include<conio.h>
#define n 10

void BinSearch(int A[], int N)
{
    int i,Lo, Hi, Mid, Flag;
    Lo=0, Hi=n-1; Flag=0;
    while (Lo <= Hi && Flag==0)
    {
        Mid=(Lo+Hi)/2;
        if (N==A[Mid])
            Flag=1;
        else
            if (N<A[Mid])
                Hi=Mid-1;
            else
                Lo=Mid+1;
    }
    if (Flag==1)
        cout<<"Nilai "<<N<<" ditemukan pada kolom ke "<<(Mid+1)<<endl;
    else
        cout<<"Nilai tidak ditemukan"<<endl;
}

void main()
{
    int i, N;
    int A[n] = {4,7,10,11,16,22,24,28,63,66};
    //Menampilkan daftar nilainya
    cout<<"DAFTAR NILAI/ANGKA "<<endl;
    for (i=0; i<=n-1; i++)
    {
        cout<< A[i] << endl;
    }
    //Menginput data yang mau dicari
    cout<<"Masukkan Data Yang dicari : "; cin>>N;
    BinSearch(A,N);
}
```

●  
●  
●  
●  
●  
●

# CONTOH PROGRAM FUNGSI BINARY SEARCHING PADA BAHASA C++ - LANJUT

```
/*Binary Search*/
#include<iostream.h>
#include<conio.h>

int search02(int[], int,int);
main( )      (
clrscr( );
const int array_size=11;
int array[array_size]={10,15,18,22,25,34,44,55,61,66,87};
cout<<"*****"<<endl;
cout<<"  b i n a r y    s e a r c h    "<<endl;
cout<<"*****";
gotoxy(1,5);
cout<<"\n Isi Data dalam Array adalah : " <<endl;
cout<<" \n Array : " <<"\t\t Data : " <<endl;
for(int count=0; count < array_size; count++) {
    cout<<"\t" <<"Array [" <<count<<" ]" <<"\t\t";
    cout<<array[count] <<endl;
}
int search_elemen=0;
int flag=0;
cout<<"\n Masukkan Data Yang Akan Anda Cari ? "; cin>>search_elemen;
flag=search02(array, array_size, search_elemen);
if(flag != -1)
    cout<<"\n Data ditemukan pada posisi array [" <<flag<<" ]";
else
    cout<<"\n Data tidak ditemukan ";
    getch( );      return 0;
}

int search02(int array[], int array_size, int elemen) {
    int start=0;    int end=array_size-1; int middle;    int position=-1;
    middle=(start+end)/2;
do {
    if(elemen < array[middle])
        end=middle-1;
    else if(elemen > array[middle])
        start=middle+1;
        middle=(start+end)/2;
} while (start <= end && array[middle] != elemen);
if(array[middle] == elemen)
    position=middle;
return position;
}
```

# Output Binary Searching

```
D:\KULIAH\ALGORI~1\ALGOBY~1\ALGOI~1\PRAK03~1\CARIO2.EXE
*****
b i n a r y   s e a r c h
*****

Isi Data dalam Array adalah :

Array :           Data :
  Array [0]         10
  Array [1]         15
  Array [2]         18
  Array [3]         22
  Array [4]         25
  Array [5]         34
  Array [6]         44
  Array [7]         55
  Array [8]         61
  Array [9]         66
  Array [10]        87

Masukkan Data Yang Akan Anda Cari ?
```



```
D:\KULIAH\ALGORI~1\ALGOBY~1\ALGOI~1\PRAK03~1\CARIO2.EXE
*****
b i n a r y   s e a r c h
*****

Isi Data dalam Array adalah :

Array :           Data :
  Array [0]         10
  Array [1]         15
  Array [2]         18
  Array [3]         22
  Array [4]         25
  Array [5]         34
  Array [6]         44
  Array [7]         55
  Array [8]         61
  Array [9]         66
  Array [10]        87

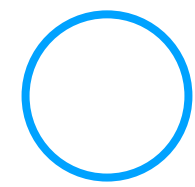
Masukkan Data Yang Akan Anda Cari ?  34

Data dietemukan pada posisi array [5]_
```





# QUESTIONS





# TUGAS LMS

