

# PENCARIAN DATA

OLEH : AKHI HARUNI

# PENGANTAR

Dalam kehidupan sehari-hari, kita sering mencari (jodoh #eh) suatu data terutama dalam gadget/komputer yang kita punya. Apabila data hanya 1, secara otomatis kita tidak perlu melakukan pencarian. Tapi bagaimana bila data yang kita punya jumlahnya ribuan? Mungkinkah kita akan mengecek folder satu demi satu? Jawabannya adalah mungkin, namun tidak efisien waktu, tenaga, pikiran, (amal, dan perbuatan). Oleh karena itu diciptakanlah algoritma pencarian yang memudahkan pengguna dalam menemukan apa yang sedang dia cari tanpa harus mengecek satu-satu data yang dia punya.

# TAHAPAN PENCARIAN DATA

1. Membuat sekumpulan data
2. Membuat nilai default data tidak ada
3. Memasukkan data yang akan dicari
4. Membandingkan apakah data yang dicari ada di dalam sekumpulan data
5. Menampilkan ada atau tidaknya data yang dicari

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     //membuat sekumpulan data
6     int nilaipemdas[5];
7     int a,b;
8     for (a=0; a<5; a++) {
9         cout<<"masukkan nilai : ";
10        cin>>nilaipemdas[a];
11    }
12
13    //deklarasi default tidak ditemukan
14    int z=0;
15
```

```
15
16 //memasukkan data yang akan dicari
17 cout<<"masukkan nilai yang akan dicari : ";
18 cin>>b;
19
20 //membandingkan apakah data ada atau tidak
21 for (a=0; a<5; a++) {
22     if(b==nilaipemdas[a]){ //jika data ada, maka
23         z=1; //ganti nilai default tidak ada menjadi ada
24     }
25 }
26
```

```
26  
27 //menampilkan ada tau tidaknya data  
28 if(z==1){  
29     cout<<"data ditemukan";  
30 } else{  
31     cout<<"data tidak ada";  
32 }  
33 }
```

# PENJELASAN KODE PROGRAM

## 1. Membuat Sekumpulan Data

- ❑ Sekumpulan data disimpan dalam array (cek materi array). Agar efisien, proses membuat sekumpulan data dibantu dengan perulangan FOR
- ❑ `int nilaipemdas[0] = 5;`
- ❑ `int nilaipemdas[1] = 6;`
- ❑ Kode di atas kurang efisien jika jumlah data puluhan, ratusan, ribuan, bahkan jutaan. Oleh karena itu dibantu dengan FOR
- ❑ `for (a = 0; a < 5; a++)` maksudnya jumlah data nantinya 5, dengan batas awal 0 dan batas akhir 4 (kurang dari 5). Apabila datanya 1000 maka `for (a=0;a<1000;a++)`
- ❑ Variabel masukan disesuaikan dengan variabel perulangan terdekat. Karena paling dekat adalah perulangan `for (a ..... )` maka variabel masukannya adalah **`nilaipemdas[a]`** bukan lagi `nilaipemdas[5]`

## 2. Membuat default nilai tidak ada

- ❑ Dalam pencarian, hasilnya hanya ada 2 yaitu ada atau tidak ada. Bilangan yang merepresentasikan ada dan tidak adalah bilangan biner yaitu 0 dan 1. Dalam program ini, data tidak ada didefinisikan dengan 0 sehingga  $\text{int } z = 0$ ;

## 3. Memasukkan data yang akan dicari

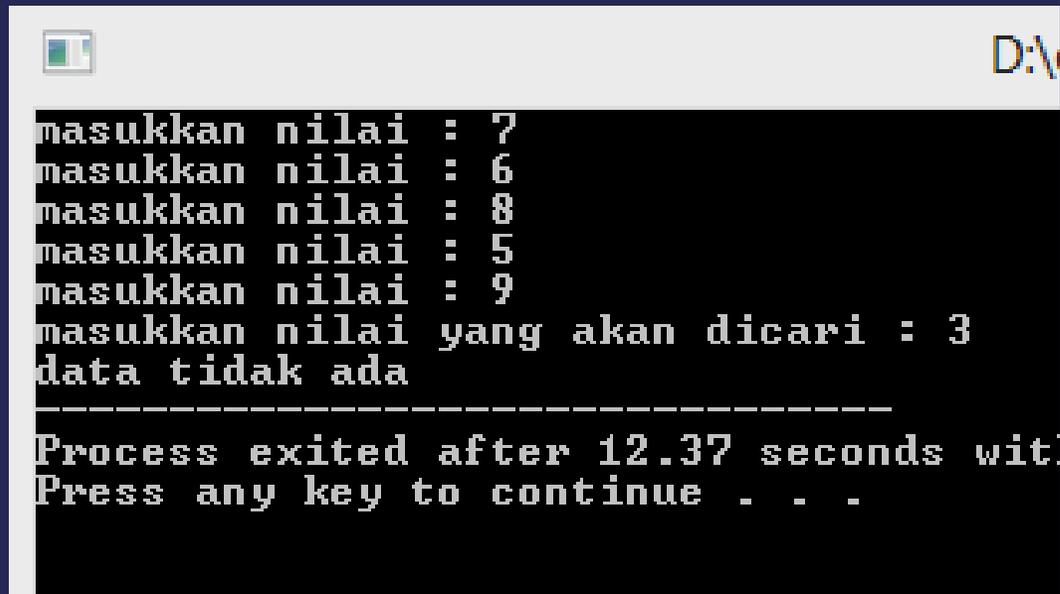
- ❑ Proses ini dilakukan untuk memasukkan data yang akan dicari. Data yang dicari disimpan dalam variabel **b**. Mengapa tidak berbentuk array? Karena data yang dicari hanya satu, jadi cukup **b** saja.

## 4. Membandingkan apakah data yang dicari ada di dalam sekumpulan data atau tidak

- ❑ Proses perbandingannya menggunakan bantuan IF dengan bahasa lain : jika data tersebut ada di dalam sekumpulan data, maka ubah default  $z = 0$  menjadi  $z = 1$  yang berarti data ADA. Jumlah sama dengan ( $==$ ) yang digunakan ada 2 karena untuk membandingkan. Jika jumlahnya satu maka fungsinya hanya sebagai pemberi nilai misal  $\text{int } z = 0$  berarti memberi nilai  $z$  dengan 0.

#### 4. Menampilkan ada atau tidaknya data

- ❑ Menampilkan hasil menggunakan bantuan IF. Sintaknya adalah `if (z==1)` dengan membandingkan apakah nilai default (variabel z) berisi 1? Dengan bahasa lain : jika nilai default berubah menjadi 1, maka data ditemukan. Selain itu, data tidak ditemukan.



```
D:\V
masukkan nilai : 7
masukkan nilai : 6
masukkan nilai : 8
masukkan nilai : 5
masukkan nilai : 9
masukkan nilai yang akan dicari : 3
data tidak ada
-----
Process exited after 12.37 seconds with
Press any key to continue . . .
```