STRUKTUR DATA (2)

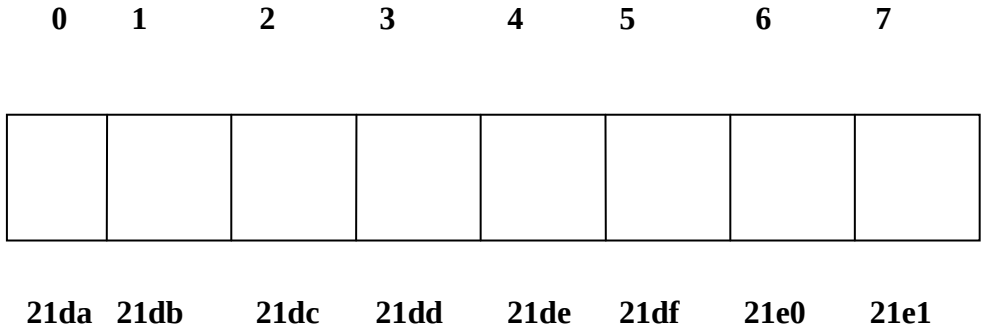
searching array

Oleh **Antonius Rachmat C, S.Kom, M.Cs**

Definisi Array

- Array : a finite ordered set of homogenous elements
- Elemen-elemen array tersusun secara **berderet** dan dapat diakses secara **random** di dalam memori.
- Array memiliki alamat yang besebelahan/berdampingan tergantung **lebar tipe datanya.**
- Array dapat berupa array 1 dimensi, 2 dimensi, bahkan n-dimensi.
- Elemen-elemen array bertipe data **sama** dan bisa berisi nilai yang sama atau **berbeda-beda.**

Ilustrasi Array 1 Dimensi char

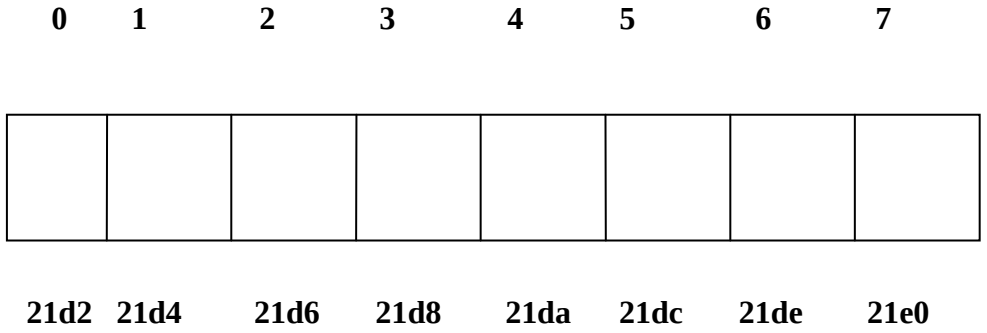



```
(Inactive C:\TC
21da
21db
21dc
21dd
21de
21df
21e0
21e1
```

```c
#include <stdio.h>

void main(){
    char a[8];
    for(int i=0;i<8;i++){
        printf("%x\n",&a[i]);
    }
}
```

%x adalah hexadesimal

Ilustrasi Array 1 Dimensi int



indeks

value

alamat

```
(Inactive C:\TCWIN
21d2
21d4
21d6
21d8
21da
21dc
21de
21e0
```

c:\tcwin45\bin\noname00.cpp

```
#include <stdio.h>

void main(){
    int a[8];
    for(int i=0;i<8;i++){
        printf("%x\n",&a[i]);
    }
}
```

%x adalah hexadesimal

Pengaksesan Elemen Array

- Elemen-elemen array dapat diakses oleh program menggunakan suatu indeks tertentu secara **random** ataupun **berurutan**
- Pengisian dan pengambilan nilai pada indeks tertentu dapat dilakukan dengan mengeset nilai atau menampilkan nilai pada indeks yang dimaksud.
- Dalam C, **tidak terdapat error handling** terhadap batasan nilai indeks, apakah indeks tersebut berada di dalam indeks array yang sudah didefinisikan atau belum. Hal ini merupakan tanggung jawab programmer. Sehingga jika programmer mengakses indeks yang salah, maka nilai yang dihasilkan akan berbeda atau rusak karena mengakses alamat memori yang tidak sesuai.

Contoh array 1 dimensi

```
char huruf[9];
int umur[10];
int kondisi[2] = {0,1}
int arr_dinamis[] = {1,2,3}
```

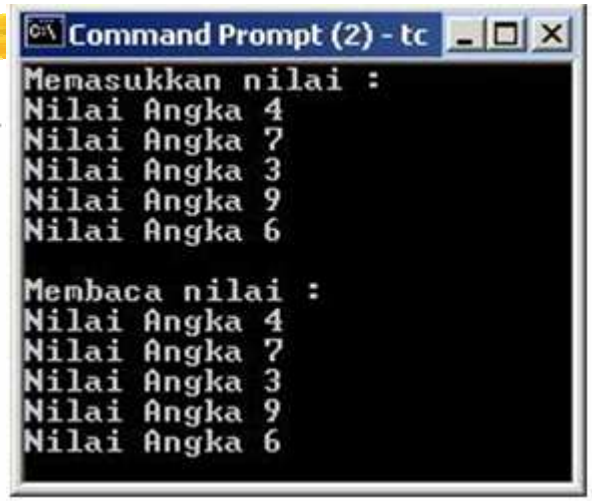
- Tanda [] disebut juga "elemen yang ke- ...". Misalnya `kondisi[0]` berarti array kondisi elemen yang ke nol.
- Array yang sudah dipesan, misalnya 10 tempat tidak harus diisi semuanya, bisa saja hanya diisi 5 elemen saja, baik secara berurutan maupun tidak. Namun pada kondisi yang tidak sepenuhnya terisi tersebut, tempat pemesanan di memori tetap sebanyak 10 tempat, jadi tempat yang tidak terisi tetap akan terpesan dan dibiarkan kosong.
- Kita **tidak dapat** mendeklarasikan array dinamis tanpa inisialisasi!

Contoh-contoh lain

- Bagaimana menginputkan dan menampilkan array?
- Manipulasi array 1 dimensi?
- Array tanpa inisialisasi langsung ditampilkan?
- Array inisialisasi dengan 0?
- Array inisialisasi hanya 2 elemen pertama?

Input – output array

```
#include <stdio.h>
#include <conio.h>

void main()
{ int nilai[5], x;
  clrscr();

  printf("Memasukkan nilai :\n");
  for(x=0;x<5;x++)
  {
     printf("Nilai Angka : "); scanf("%d",&nilai[x]);
  }
  printf("\n");

  printf("Membaca nilai :\n");
  for(x=0;x<5;x++)
  {
     printf("Nilai Angka : %d",nilai[x]);
  }

  getch();
}
```



Manipulasi array

```c
#include <stdio.h>
#include <conio.h>

int main(){
    int bil[7],i;
    printf("elemen 1 ? ");scanf("%d",&bil[0]);
    bil[1] = 5;
    bil[2] = bil[1] + 20;
    for(i=4;i<7;i++) bil[i] = i*10;
    bil[3] = bil[bil[1]];
    for(i=0;i<7;i++) printf("bil[%d] = %d dan alamatnya: %x\n",i,bil[i],&bil[i]);
    getch();
    return 0;
}
```

```
C:\TCWIN45\BIN\NONAMEOO.EXE
elemen 1 ? 25
bil[0] = 25 dan alamatnya: 2384
bil[1] = 5 dan alamatnya: 2386
bil[2] = 25 dan alamatnya: 2388
bil[3] = 50 dan alamatnya: 238a
bil[4] = 40 dan alamatnya: 238c
bil[5] = 50 dan alamatnya: 238e
bil[6] = 60 dan alamatnya: 2390
```

Inisialisasi

```
#include <stdio.h>
#include <conio.h>

int main(){
     int bil[7];    //tanpa inisialisasi
     for(int i=0;i<7;i++){
       printf("Elemen ke-%i = %d,\talamat: %x\n",i,bil[i],&bil[i]);
     }
     getch();
     return 0;
}
```

```
C:\TCWIN45\BIN\NONAMEOO.EXE
Elemen ke-0 = -1,       alamat: 21f0
Elemen ke-1 = 3224,     alamat: 21f2
Elemen ke-2 = 3182,     alamat: 21f4
Elemen ke-3 = -1,       alamat: 21f6
Elemen ke-4 = 8653,     alamat: 21f8
Elemen ke-5 = 9095,     alamat: 21fa
Elemen ke-6 = 19201,    alamat: 21fc
```

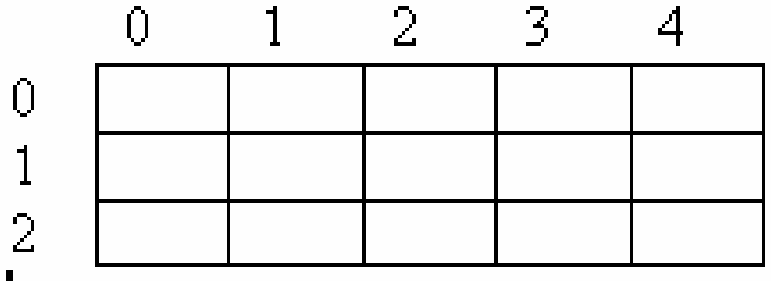
```
#include <stdio.h>
#include <conio.h>

int main(){
     int bil[7] = {0};    //inisialisasi 0
     for(int i=0;i<7;i++){
       printf("Elemen ke-%i = %d\talamat: %x\n",i,bil[i],&bil[i]);
     }
     getch();
     return 0;
}
```

```
C:\TCWIN45\BIN\NONAMEOO.EXE
Elemen ke-0 = 0 alamat: 21fc
Elemen ke-1 = 0 alamat: 21fe
Elemen ke-2 = 0 alamat: 2200
Elemen ke-3 = 0 alamat: 2202
Elemen ke-4 = 0 alamat: 2204
Elemen ke-5 = 0 alamat: 2206
Elemen ke-6 = 0 alamat: 2208
```

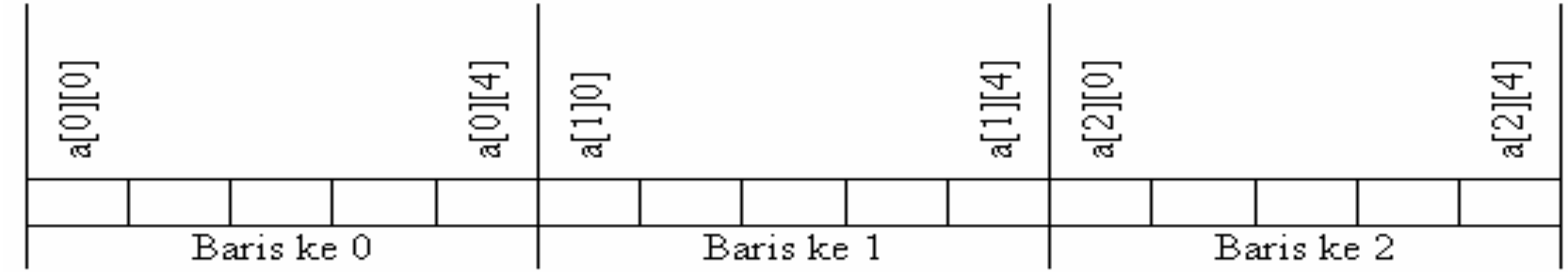
Operasi-operasi Array

- Penambahan elemen array
- Menampilkan elemen array
- Pencarian elemen array
  - Cari, jika ditemukan, katakan KETEMU!
- Penghapusan elemen array
  - Cari, jika ditemukan kemudian dihapus!
- Pengeditan elemen array
  - Cari, jika ditemukan kemudian diedit!

Arrray 2 dimensi



- **char a[3][5]**
- Sama dengan matriks berukuran 3x5
- Pada kenyataan di memory:



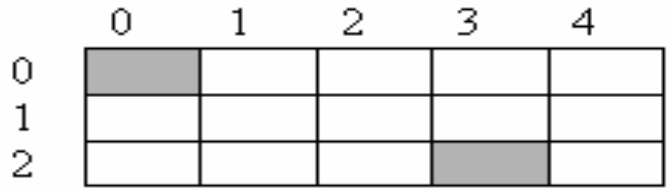
Ilustrasi array 2 dimensi

```c
#include <stdio.h>
#include <conio.h>

int main(){
    char a[3][5];
    for(int i=0;i<3;i++){
        for(int j=0;j<5;j++){
            printf("%x ",&a[i][j]);
        }
        printf("\n");
    }
    getch();
    return 0;
}
```

```
21d4 21d5 21d6 21d7 21d8
21d9 21da 21db 21dc 21dd
21de 21df 21e0 21e1 21e2
```

Soal alamat array

- Soal: int A[10], diket. &A[0] = H1000
  - Berapa &A[7]?
- Jawab:
  - int berukuran 2 byte
  - Perpindahan 7-0 = 7 * 2 byte = 14 byte
  - Maka H1000 + 7 = H100E
- Soal: int A[3][5], &A[0][0] = H1000
  - Berapa &A[2][3]?
- Jawab:
  - int 2 byte
  - Perpindahan baris: 2-0 = 2 * 5 (kolomnya) = 10
  - Perpindahan kolom: 3-0 = 3
  - Total perpindahan: 10 + 3 = 13 * 2 byte = 26 byte
  - Maka H1000 + H1A = H101A



Searching

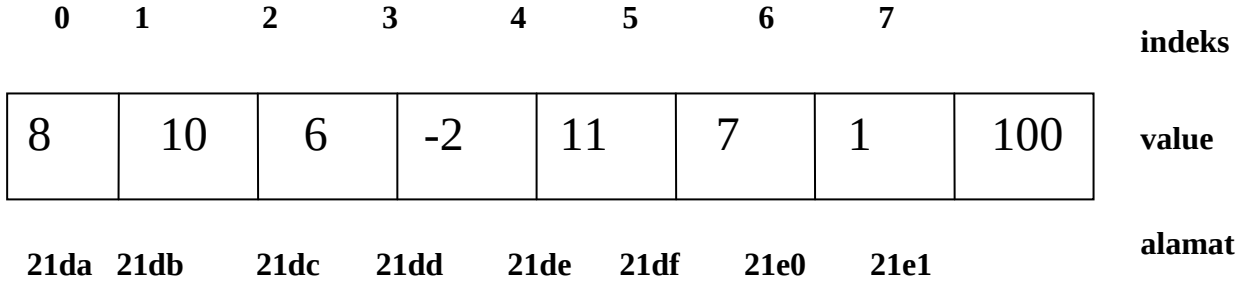
- Pada suatu data seringkali dibutuhkan pembacaan kembali informasi (**retrieval information**) dengan cara searching.
- Searching adalah pencarian data dengan cara menelusuri data-data tersebut.
- Tempat pencarian data dapat berupa array dalam memori, bisa juga pada file pada external storage.

Sequential Search

- Adalah suatu teknik pencarian data dalam array (1 dimensi) yang akan menelusuri semua elemen-elemen array dari awal sampai akhir, dimana data-data **tidak perlu** diurutkan terlebih dahulu.
- Kemungkinan terbaik (**best case**) adalah jika data yang dicari terletak di indeks array terdepan (elemen array pertama) sehingga waktu yang dibutuhkan untuk pencarian data sangat sebentar (minimal).
- Kemungkinan terburuk (**worst case**) adalah jika data yang dicari terletak di indeks array terakhir (elemen array terakhir) sehingga waktu yang dibutuhkan untuk pencarian data sangat lama (maksimal).

Sequential Search (2)

- Misalnya terdapat array satu dimensi sebagai berikut:



- Kemudian program akan meminta data yang akan dicari, misalnya **6**.
- Jika ada maka akan ditampilkan tulisan “ADA”, sedangkan jika tidak ada maka akan ditampilkan tulisan “TIDAK ADA”.

Detail Program

```
#include <stdio.h>

void main(){
    int data[8] = {5,2,1,6,7,9,8,3};
    int cari = 6;
    int i=0;
    int flag = 0; //blm ketemu
    while(i<8){
        if(data[i] == cari){
            flag=1;
            break;
        }
        i++;
    }
    if(flag==1) printf("ketemu");
    else printf("tidak ketemu");
}
```

Pembahasan Program

- Program menggunakan sebuah variabel flag yang berguna untuk menadai ada atau tidaknya data yang dicari dalam array data. Hanya bernilai 0 atau 1.
- Flag pertama kali diinisialiasasi dengan nilai 0.
- Jika ditemukan, maka flag akan diset menjadi 1, jika tidak ada maka flag akan tetap bernilai 0.
- Semua elemen array data akan dibandingkan satu persatu dengan data yang dicari dan diinputkan oleh user.
- **Question**: Bagaimana jika data yang dicari ditemukan dan ditanyakan terletak di indeks ke berapa?

Q & A

- **Problem:** Apakah cara di atas efisien? Jika datanya ada 10000 dan semua data dipastikan unik?
- **Solution**: Untuk meningkatkan efisiensi, seharusnya jika data yang dicari sudah ditemukan maka perulangan harus dihentikan!
  - **Hint**: Gunakan **break**!
- **Question**: Bagaimana cara menghitung ada berapa data dalam array yang tidak unik, yang nilainya sama dengan data yang dicari oleh user?
  - **Hint**: Gunakan variabel counter yang nilainya akan selalu bertambah jika ada data yang ditemukan!

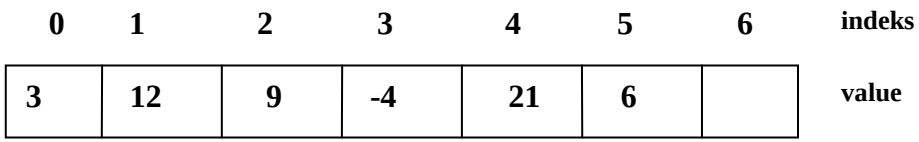
Contoh

```
#include <conio.h>
#include <stdlib.h>

void main(){
    clrscr();
    int data[100];
    int cari=20;
    int counter=0;
    int flag=0;
    randomize();
    printf("generating 100 random number...\n");
    for(int i=0;i<100;i++){
        data[i]=random(100)+1;
        printf("%d ",data[i]);
    }
    printf("\ndone.\n");

    for(i=0;i<100;i++){
        if(data[i]==cari){
          counter++;
          flag=1;
        }
    }
    if(flag==1) printf("Data ada, sebanyak %d!\n",counter);
          else printf("Data tidak ada!\n");
}
```

```
(Inactive C:\TCWIN45\BIN\NONAME00.EXE)
generating 100 random number...
46 6 98 94 46 60 44 85 66 39 61 16 68 44 100 14 11 74 32 11 7 35 20 91 53 42 30
71 69 53 28 66 25 94 90 19 100 7 64 65 52 83 13 13 24 11 88 84 99 11 33 29 97 66
 4 43 37 40 87 94 75 92 90 5 9 59 11 86 91 38 44 60 20 24 35 94 53 46 41 62 33 4
9 2 48 40 73 100 41 81 68 88 2 26 91 73 70 91 54 47 85
done.
Data ada, sebanyak 2!
```

Sequential Search with Sentinel

- Perhatikan array data berikut ini:



- Terdapat 6 buah data dalam array (dari indeks 0 s/d 5) dan terdapat 1 indeks array tambahan (indeks ke 6) yang belum berisi data (disebut sentinel)
- Array pada indeks ke 6 berguna untuk menjaga agar indeks data berada pada indeks 0 s/d 5 saja. Bila pencarian data sudah mencapai array indeks yang ke-6 maka berarti data TIDAK ADA, sedangkan jika pencarian tidak mencapai indeks ke-6, maka data ADA.

Program

```
#include <stdio.h>

void main(){
    int data[8] = {5,2,1,6,7,9,8,3};
    int cari = 11;
    int i=0;
    data[7]=cari;

    while(data[i] !=cari){
        i++;
    }
    if(i<7) printf("ketemu");
    else printf("tidak ketemu");
}
```

Binary Search

- Data yang ada harus diurutkan terlebih dahulu berdasarkan suatu urutan tertentu yang dijadikan kunci pencarian.
- Adalah teknik pencarian data dalam dengan cara membagi data menjadi dua bagian setiap kali terjadi proses pencarian.
- Prinsip pencarian biner adalah:
  - Data diambil dari posisi 1 sampai posisi akhir N
  - Kemudian cari posisi data tengah dengan rumus: (posisi awal + posisi akhir) / 2
  - Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar?
  - Jika lebih besar, maka proses pencarian dicari dengan posisi awal adalah posisi tengah + 1
  - Jika lebih kecil, maka proses pencarian dicari dengan posisi akhir adalah posisi tengah – 1
  - Jika data sama, berarti ketemu.

Ilustrasi

Contoh Data:
Misalnya data yang dicari **17**

- 0 1 2 3 4 5 6 7 8
- **3 9 11 12 15 17 23 31 35**
- **A B C**
- Karena 17 > 15 (data tengah), maka: awal = tengah + 1

- 0 1 2 3 4 5 6 7 8
- **3 9 11 12 15 17 23 31 35**
- **A B C**
- Karena 17 < 23 (data tengah), maka: akhir = tengah – 1

- 0 1 2 3 4 5 6 7 8
- **3 9 11 12 15 17 23 31 35**
- **A=B=C**
- Karena 17 = 17 (data tengah), maka KETEMU!

Program

```
data tengah: 4
cari dikanan
data tengah: 6
cari dikiri
data tengah: 5
data ada
```

```c
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>

void main(){
     clrscr();
     int data[9] = {3,9,11,12,15,17,23,31,35};
     int l,r,m;
     int n=9;
     int cari=17;
     l = 0;
     r = n-1;
     int ktm = 0;
     while(l<=r && ktm==0){
          m = (l+r)/2;
          printf("data tengah: %d\n",m);
          if(data[m] == cari) ktm=1;
          else if (cari < data[m]){
               printf("cari dikiri\n");
               r=m-1;
          }
          else{
               l=m+1;
               printf("cari dikanan\n");
          }
     }
     if(ktm==1) printf("data ada\n");
     else printf("data tidak ada\n");
}
```

Interpolation Search

- Teknik ini dilakukan pada data yang sudah terurut berdasarkan **kunci tertentu**
- Teknik searching ini dilakukan dengan **perkiraan letak data**.
  - Contoh ilustrasi: jika kita hendak mencari suatu nama di dalam buku telepon, misal yang berawalan dengan huruf T, maka kita tidak akan mencarinya dari awal buku, tapi kita langsung membukanya pada 2/3 atau ¾ dari tebal buku.
- Rumus posisi relatif kunci pencarian dihitung dengan rumus:

$$Posisi = \frac{kunci - data[low]}{data[high] - data[low]} x(high - low) + low$$

- Jika data[posisi] > data yg dicari, high = pos – 1
- Jika data[posisi] < data yg dicari, low = pos + 1

Kasus

- Misal terdapat data sebagai berikut:

Kode	Judul Buku	Pengarang
025	The C++ Programming	James Wood
034	Mastering Delphi 6	Marcopolo
041	Professional C#	Simon Webe
056	Pure JavaScript v2	Michael Bolton
063	Advanced JSP & Servlet	David Dunn
072	Calculus Make it Easy	Gunner Christian
088	Visual Basic 2005 Express	Antonie
096	Artificial Life : Volume 1	Stephen Seagle

Penyelesaian

- Kunci Pencarian ? **088**
- Low ? **0**
- High ? **7**
- Posisi = **(088 - 025) / (096 - 025) * (7 - 0) + 0 = [6]**
- Kunci[6] = kunci pencarian, data ditemukan : **Visual Basic 2005**

- Kunci Pencarian ? **060**
- Low ? **0**
- High ? **7**
- Posisi = **(060 – 025) / (096 – 025) * (7 – 0) + 0 = [3]**
- Kunci[3] < kunci pencarian, maka teruskan
- Low = **3 + 1 = 4**
- High = **7**
- Ternyata Kunci[4] adalah **063** yang lebih besar daripada **060**.
- Berarti tidak ada kunci **060.**

Program

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

void main(){
   clrscr();
   int data[9] = {3,9,11,12,15,17,23,31,35};
   int low,high;
   int flag=0,d=36;
   float pos1;
   int pos,n=9;
   low=0;
   high=n-1;
   do{
      pos1 = (float) (d-data[low])/(data[high]-data[low]) * (high-low) + low;
      pos = floor(pos1);
      if (data[pos] == d){
         flag=1;
         break;
      }
      if (data[pos] > d) high = pos-1;
      else
         if (data[pos] < d) low = pos + 1;
   } while(d >= data[low] && d <= data[high]);
   if(flag==0) printf("data tidak ada\n"); else printf("data ada\n");
}
```

Soal-soal

- Cari tahu tentang cara penggunaan dan teknologi dari website-website pencari (search engine) yang ada di Internet!
- Cari tahu tentang teknik teknik pencarian yang lain!

- NEXT : **Sorting Array**!