

Pertemuan 12

STRUKTUR SEARCHING

Linier Searching (Sequential Searching)

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

Misalnya terdapat array satu dimensi sebagai berikut:

0	1	2	3	4	5	6	7	
								indeks
8	10	6	-2	11	7	1	100	value
21da	21db	21dc	21dd	21de	21df	21e0	21e1	alamat

Kemudian program akan meminta data yang akan dicari, misalnya **6**.

Jika ada maka akan ditampilkan “Data ada”, jika tidak ada maka akan ditampilkan “Data tidak ada”.

Linier Searching (Sequential Searching) Dengan Program C++

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

void main(){
    clrscr();
    int data[8] = {8,10,6,-2,11,7,1,100};
    int cari;
    int flag=0;
    printf("masukkan data yang ingin dicari = ");    scanf("%d",&cari);
    for(int i=0;i<8;i++){
        if(data[i] == cari) flag=1;
    }
    if(flag==1) printf("Data ada!\n");
        else printf("Data tidak ada!\n");
}
```

Binary Searching

Merupakan metode terbaik dalam search (pencarian), karena memulai pencarian dari lokasi tengah (m). Kemudian berdasarkan posisi tengah tersebut terdapat 3 kemungkinan :

- a. Jika $cari < data[m]$, maka informasi yang dicari berada disebelah kiri dari lokasi tengah (m)
- c. Jika $cari > data[m]$, maka informasi yang dicari berada disebelah kanan dari lokasi tengah (m)
- b. Jika $cari = data[m]$ maka data tengah tersebut adalah data yang dicari

Binary Searching Dengan Program 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 di kanan\n");
        }
    }
    if (ktm==1) printf("data ada\n");
    else printf("data tidak ada\n");
}
```