

1. Linear Search

Linear Search merupakan sebuah teknik pencarian data dengan menelusuri semua data satu per satu. Apabila ditemukan kecocokan data maka program akan mengembalikan output, jika tidak pencarian akan terus berlanjut hingga akhir dari array tersebut

Sebagai Contoh :

Code in java :

```
int[] a = {70, 60, 30, 50, 40,20};
int target = 40;
for( int i=0; i<this.a.length; i++) {
    if(a[i]==target) {
        System.out.println ( "Element found at index "+i);
    }
}
```

Penjelasan :

Misalkan angka yang dicari adalah angka 40, berikut gambaran dari implementasi LinearSearch:

Tahap pertama:

(70, 60, 30, 50, 40,20)

(70==40) //FALSE

Tahap kedua:

(70, 60, 30, 50, 40,20)

(60==40) //FALSE

Tahap ketiga:

(70, 60, 30, 50, 40,20)

(30==40) //FALSE

Tahap keempat:

(70, 60, 30, 50, 40,20)

(50==40) //FALSE

Tahap kelima:

(70, 60, 30, 50, 40,20)

(40==40) //TRUE

Jika data ditemukan, maka program akan keluar dari looping. Jika kita ingin menampilkan index dari data yang dicari, kita tinggal menyimpan index dari array tersebut dan menampilkan nya.

2. Binary Search

Binary Search merupakan sebuah teknik pencarian data dengan cara berulang kali membagi separuh dari jumlah data yang dicari sampai sehingga memperkecil lokasi pencarian menjadi satu data. Dengan teknik ini kita akan membuang setengah dari jumlah data. Apabila ditemukan kecocokan data maka program akan mengembalikan output, jika tidak pencarian akan terus berlanjut hingga akhir dari pembagian jumlah data tersebut.

Sebagai Contoh :

Code in java :

```
int[] a = { 20,30,40,50,60,70 };
int target = 40;

int low = 0;
int high = a.length - 1;
while (low <= high) {
    int mid = (low + high) / 2;
    if (a[mid] == target) {
        System.out.println("Element found at index " + mid);
        break;
    } else if (a[mid] < target) {
        low = mid + 1;
    } else if (a[mid] > target) {
        high = mid - 1;
    }
}
```

Misalkan kita memiliki `int a[] = {20,30,40,50,60,70}` Apabila angka yang dicari adalah angka 40, berikut gambaran dari implementasi Binary Search:

Tahap pertama :

(20,30,40,50,60,70)

LOW = 0

HIGH = N

MID = (LOW + HIGH)/2 = (0+6)/2 = 3

(a[MID] == 40)

(20,30,40,50,60,70)

(50==40) // FALSE

HIGH = MID-1

Tahap kedua:

(20,30,40,50,60,70)

$MID = (LOW + HIGH)/2 = (0+2)/2 = 1$

$(a[MID] == 40)$

(20,30,40,50,60,70)

$(30==40) // \text{FALSE}$

$LOW = MID+1$

Tahap ketiga :

(20,30,40,50,60,70)

$MID = (LOW + HIGH)/2 = (2+2)/2=2$

$(a[MID] == 40)$

(20,30,40,50,60,70)

$(40==40) // \text{TRUE}$

Jika data ditemukan, maka program akan keluar dari looping. Jika kita ingin menampilkan index dari data yang dicari, kita tinggal menyimpan index dari array tersebut dan menampilkan nya.

3. Bubble Sort

Bubble sort merupakan sebuah teknik pengurutan data dengan cara menukar dua data yang bersebelahan jika urutan dari data tersebut salah. Algorithm ini dapat mengurutkan data dari besar ke kecil (Ascending) dan kecil ke besar (Descending).

Sebagai contoh :

Code in java :

```
int[] a = {70, 60, 30, 50, 40,20};
for (int i = 0; i < a.length - 1; i++) {
    for (int j = 0; j < a.length - 1 - i; j++) {
        if (a[j + 1] < a[j]) {
            int temp = a[j];
            a[j] = a[j + 1];
            a[j + 1] = temp;
        }
    }
}
```

Berikut gambaran dari implementasi Bubble Sort:

Tahap pertama:

(70, 60, 30, 50, 40,20) -> (70, 60, 30, 50, 20,40)

(70, 60, 30, 50, 20, 40) -> (70, 60, 30, 20, 50, 40)

(70, 60, 30, 20,50, 40) -> (70, 60, 20, 30,50, 40)

(70, 60, 20,30,50, 40) -> (70, 20, 60,30,50, 40)

(70, 20,60,30,50, 40) -> (20, 70,60,30,50, 40)

(20,70,60,30,50, 40)

Tahap kedua:

(20,70,60,30,50, 40) ->(20,70,60,30,40,50)

(20,70,60,30,40,50) -> (20,70,60,30,40,50)

(20,70,60,30,40,50) -> (20,70,30, 60,40,50)

(20,70,30,60,40,50) -> (20,30,70, 60,40,50)

(20,30,70, 60,40,50)

Tahap ketiga:

(20,30,70, 60,40,50) -> (20,30,70, 60,40,50)

(20,30,70, 60,40,50) ->(20,30,70,40,60,50)

(20,30,70,40,60,50) -> (20,30,40, 70,60,50)

(20,30,40,70,60,50)

Tahap keempat:

(20,30,40,70,60,50) -> (20,30,40,70,50,60)

(20,30,40,70,50,60) -> (20,30,40,50, 70,60)

(20,30,40,50,70,60)

Tahap kelima:

(20,30,40,50,70,60) -> (20,30,40,50,60, 70)

(20,30,40,50,60,70)