

Nama : Moh. Syahrul Aziz Imastara

Nim : 1203220084

Kelas : IF-02-01

```
#include <iostream>
using namespace std;
int metode_ini(int arr[], int n, int key)
{
    for (int i = 0; i < n; ++i)
    {
        if (arr[i] == key)
            return i;
    }
    return -1;
}
int main()
{
    int arr[] = {10, 20, 30, 40, 50};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 30;
    int result = metode_ini(arr, n, key);
    if (result != -1)
        cout << "Berhasil: " << result << endl;
    else
        cout << "Tidak berhasil" << endl;
    return 0;
}
```

Tugas

1. Apa yang dilakukan oleh program di atas?
2. Apa nama algoritma tersebut?
3. Berapakah Kompleksitas algoritma tersebut? (Dalam Big O)
4. Apakah ada algoritma lain yang lebih baik dari algoritma di atas? Jika ada maka sebutkan dan berikan kode programnya! (SS kode dan hasil)

Jawaban

1. Melakukan pencarian berurutan yang dimulai dari satu ujung dan menelusuri setiap elemen hingga elemen yang diinginkan ditemukan.
2. Linear Search, karena mencari nilai secara berurutan dari awal hingga akhir array
3. $O(N)$, Dimana n adalah jumlah elemen dalam array, algoritma harus memeriksa setiap elemen dalam array satu per satu untuk memulai nilai yang dicari

4.

```
#include <iostream>
using namespace std;
// menggunakan namespace 'std' sehingga tidak perlu lagi menuliskan std::

int binarySearch(int arr[], int low, int high, int key)
// fungsi yang mengembalikan indeks dari nilai key ke dalam array jika
ditemukan
{
    while (low <= high)
        // agar low tidak melebihi high
        {
            int mid = low + (high - low) / 2;
            // menghitung indeks tengah
            if (arr[mid] == key)
                return mid;
            else if (arr[mid] < key)
                low = mid + 1;
            else
                high = mid - 1;
            // jika nilai key lebih dari mid maka +1 jika kurang dari mid maka -1
        }
    return -1;
    // jika nilai tidak ditemukan maka fungsi mengembalikan -1
}

int main()
{
    int arr[] = {10, 20, 30, 40, 50};
    int n = sizeof(arr) / sizeof(arr[0]);
    int key = 30;
    int result = binarySearch(arr, 0, n - 1, key);
    if (result != -1)
        cout << "Berhasil: " << result << endl;
    else
        cout << "Tidak berhasil" << endl;
    // jika nilai yang dicari ditemukan maka print "Berhasil" jika tidak maka
    "Tidak berhasil"
    return 0;
}
```

