

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **LAPORAN AWAL PRAKTIKUM**

**Pertemuan ke-06**

**Searching**



**Disusun Oleh:**

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566  
Tangerang Selatan - Banten

## A. RANGKUMAN MATERI

Searching atau pencarian adalah proses mencari nilai tertentu pada struktur data seperti array, list, maupun tree. Pada program C++, terdapat beberapa metode searching yang dapat digunakan seperti sequential search, binary search, dan interpolation search. Sequential search adalah metode searching sederhana dimana program mencari nilai dengan cara membandingkan setiap nilai pada struktur data satu persatu. Jika nilai ditemukan, maka program akan memberikan output indeks dari nilai tersebut. Namun, jika nilai tidak ditemukan, program akan memberikan output bahwa nilai tersebut tidak ditemukan.

Metode searching yang lebih efektif adalah binary search. Binary search bekerja dengan mengurutkan struktur data terlebih dahulu, kemudian mencari nilai dengan cara membagi struktur data menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Binary search lebih efisien daripada sequential search, terutama pada struktur data yang besar. Namun, syarat untuk menggunakan binary search adalah struktur data harus sudah terurut terlebih dahulu.

Selain itu, terdapat metode searching lain yang bisa digunakan pada data yang memiliki distribusi yang merata yaitu interpolation search. Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data. Metode ini menghitung nilai perkiraan lokasi nilai yang akan dicari, dan mencari nilai tersebut secara berulang-ulang hingga ditemukan. Interpolation search lebih cepat dibandingkan sequential search dan bisa digunakan pada data yang besar dan memiliki distribusi yang merata.

## B. TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Searching!

**Jawab :**

Searching atau pencarian adalah proses mencari nilai tertentu pada suatu struktur data, seperti array, list, maupun tree. Tujuan pencarian adalah untuk menemukan nilai yang dicari, atau memberikan pesan bahwa nilai tersebut tidak ada dalam struktur data.

2. Jelaskan perbedaan dari Sequential Search, Binary Search, dan Interpolation Search sebagai metode-metode searching!

**Jawab :**

Sequential search adalah metode pencarian sederhana yang dilakukan dengan membandingkan setiap elemen pada struktur data satu per satu. Jika nilai ditemukan, maka program akan memberikan output indeks dari nilai tersebut. Namun, jika nilai tidak ditemukan, program akan memberikan output bahwa nilai tersebut tidak ditemukan.

Binary search bekerja dengan mengurutkan struktur data terlebih dahulu, kemudian mencari nilai dengan cara membagi struktur data menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Binary search lebih efisien daripada sequential search, terutama pada struktur data yang besar. Namun, syarat untuk menggunakan binary search adalah struktur data harus sudah terurut terlebih dahulu.

Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data. Metode ini menghitung nilai perkiraan lokasi nilai yang akan dicari, dan mencari nilai tersebut secara berulang-ulang hingga ditemukan. Interpolation search lebih cepat dibandingkan sequential search dan bisa digunakan pada data yang besar dan memiliki distribusi yang merata.

3. Jelaskan apa saja yang mempengaruhi kecepatan proses pencarian data didalam penyimpanan data!

**Jawab :**

Kecepatan proses pencarian data dalam penyimpanan data dipengaruhi oleh beberapa faktor, seperti ukuran data, metode pencarian yang digunakan, jenis penyimpanan data, dan kemampuan mesin untuk memproses data. Semakin besar ukuran data, semakin lama proses pencarian data akan memakan waktu. Selain itu, metode pencarian juga mempengaruhi kecepatan proses pencarian. Binary search dan interpolation search lebih cepat daripada sequential search karena keduanya memiliki kemampuan membagi data menjadi dua bagian atau memperkirakan letak data yang

akan dicari. Jenis penyimpanan data juga mempengaruhi kecepatan proses pencarian. Penyimpanan data pada media yang lebih cepat seperti SSD akan lebih cepat dalam proses pencarian data dibandingkan media penyimpanan yang lebih lambat seperti HDD. Kemampuan mesin untuk memproses data juga mempengaruhi kecepatan proses pencarian data.

4. Buatlah contoh program sederhana menggunakan Searching!

**Jawab :**

```
#include <iostream>
using namespace std;

int binarySearch(int arr[], int n, int x) {
    int l = 0, r = n-1;
    while (l <= r) {
        int mid = l + (r - l) / 2;
        if (arr[mid] == x)
            return mid;
        if (arr[mid] < x)
            l = mid + 1;
        else
            r = mid - 1;
    }
    return -1;
}

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int n = sizeof(arr) / sizeof(arr[0]);
    int x;

    cout << "Deretan angka: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }

    cout << "\n\nMasukkan angka yang ingin dicari: ";
    cin >> x;

    int result = binarySearch(arr, n, x);

    if (result == -1) {
        cout << "Nilai tidak ditemukan." << endl;
    } else {
        cout << "Nilai ditemukan pada indeks ke-" << result << endl;
    }

    return 0;
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Deretan angka: 1 2 3 4 5 6 7 8 9 10

Masukkan angka yang ingin dicari: 5
Nilai ditemukan pada indeks ke-4
```

## C. TUGAS PRAKTIKUM

- Lat6\_1

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int i, N, angka, Bilangan;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data yang telah dimasukkan adalah : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    cout << "\nMasukkan nilai yang akan dicari : ";
    cin >> Bilangan;

    i = 0;
    do {
        if (Nilai[i] == Bilangan) angka = i;
        i++;
    } while (i < N);

    if (Nilai[angka] == Bilangan) {
        cout << "Data ditemukan pada indeks ke-" << angka << endl;
    } else {
        cout << "Data tidak ditemukan" << endl;
    }
}
```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 10
Masukkan data ke-2 : 20
Masukkan data ke-3 : 30
Masukkan data ke-4 : 40
Masukkan data ke-5 : 50
Data yang telah dimasukkan adalah : 10 20 30 40 50
Masukkan nilai yang akan dicari : 40
Data ditemukan pada indeks ke-3
```

- Lat6\_2

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int Posisi[15];
    int i, N, angka, Bilangan, Banyak = 0;
    bool ketemu;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data yang telah dimasukkan adalah : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    cout << "\nMasukkan nilai yang akan dicari : ";
    cin >> Bilangan;

    for (i = 0; i < N; i++) {
        if (Nilai[i] == Bilangan) {
            ketemu = true;
            Posisi[Banyak] = i;
            Banyak++;
        }
    }

    if (ketemu) {
        cout << "Data ditemukan pada indeks ";
        for (i = 0; i < Banyak; i++) {
            cout << "ke-" << Posisi[i] << " ";
        }
        cout << "\nBanyaknya data yang ditemukan: " << Banyak << endl;
    } else {
        cout << "Data tidak ditemukan" << endl;
    }
}
```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 11
Masukkan data ke-2 : 12
Masukkan data ke-3 : 11
Masukkan data ke-4 : 13
Masukkan data ke-5 : 14
Data yang telah dimasukkan adalah : 11 12 11 13 14
Masukkan nilai yang akan dicari : 11
Data ditemukan pada indeks ke-0 ke-2
Banyaknya data yang ditemukan: 2

```

- Lat6\_3

```

#include <iostream>
#include <string>
using namespace std;

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    int Nilai[15];
    int i, j, N;
    int temp, awal, akhir, tengah, Bilangan;

    cout << "Masukkan banyaknya data : ";
    cin >> N;

    for (i = 0; i < N; i++) {
        cout << "Masukkan data ke-" << i + 1 << " : ";
        cin >> Nilai[i];
    }

    cout << "Data sebelum diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }

    for (i = 0; i < N - 1; i++) {
        for (j = i + 1; j < N; j++) {
            if (Nilai[i] > Nilai[j]) {
                temp = Nilai[i];
                Nilai[i] = Nilai[j];
                Nilai[j] = temp;
            }
        }
    }

    cout << "\nData setelah diurutkan : ";
    for (i = 0; i < N; i++) {
        cout << Nilai[i] << " ";
    }
}

```



```

}

cout << "\nMasukkan nilai yang akan dicari : ";
cin >> Bilangan;

awal = 0;
akhir = N - 1;

do {
    tengah = (awal + akhir) / 2;
    if (Nilai[tengah] == Bilangan) {
        cout << "Data ditemukan pada indeks ke-" << tengah << endl;
        break;
    } else if (Nilai[tengah] < Bilangan) {
        awal = tengah + 1;
    } else {
        akhir = tengah - 1;
    }
} while (awal <= akhir);
}

```

```

Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan banyaknya data : 5
Masukkan data ke-1 : 15
Masukkan data ke-2 : 11
Masukkan data ke-3 : 17
Masukkan data ke-4 : 12
Masukkan data ke-5 : 10
Data sebelum diurutkan : 15 11 17 12 10
Data setelah diurutkan : 10 11 12 15 17
Masukkan nilai yang akan dicari : 12
Data ditemukan pada indeks ke-2

```

- Lat6\_4

```

#include <iostream>
#include <string>
using namespace std;

// * Struct Declaration
struct Pegawai {
    string NIP;
    string Nama;
    string Alamat;
    string Agama;
    string Jabatan;
};

int main()
{
    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    const int MAX_PEGAWAI = 10;

```

```

Pegawai daftar_pegawai[MAX_PEGAWAI]; // * Array of struct

int n;
string nip_cari;
bool found = false;

cout << "Masukkan jumlah pegawai: ";
cin >> n;

for (int i = 0; i < n; i++) {
    cout << "Pegawai ke-" << i+1 << endl;
    cout << "NIP\t: ";
    cin >> daftar_pegawai[i].NIP;

    cout << "Nama\t: ";
    cin.ignore();
    getline(cin, daftar_pegawai[i].Nama);

    cout << "Alamat\t: ";
    getline(cin, daftar_pegawai[i].Alamat);

    cout << "Agama\t: ";
    getline(cin, daftar_pegawai[i].Agama);

    cout << "Jabatan\t: ";
    getline(cin, daftar_pegawai[i].Jabatan);

    cout << endl;
}

// * Sequential search
cout << "Masukkan NIP pegawai yang ingin dicari: ";
cin >> nip_cari;

for (int i = 0; i < n; i++) {
    if (daftar_pegawai[i].NIP == nip_cari) {
        found = true;
        cout << "Biodata Pegawai dengan NIP " << nip_cari << endl;
        cout << "Nama\t: " << daftar_pegawai[i].Nama << endl;
        cout << "Alamat\t: " << daftar_pegawai[i].Alamat << endl;
        cout << "Agama\t: " << daftar_pegawai[i].Agama << endl;
        cout << "Jabatan\t: " << daftar_pegawai[i].Jabatan << endl;
        break;
    }
}

if (!found) {
    cout << "Pegawai dengan NIP " << nip_cari << " tidak ditemukan" << endl;
}

return 0;
}

```

```
Nama      : Nova Ardiansyah
NIM       : 211011401309
=====

Masukkan jumlah pegawai: 2
Pegawai ke-1
NIP       : 211011
Nama      : Nova
Alamat    : Tangerang
Agama     : Islam
Jabatan   : Developer

Pegawai ke-2
NIP       : 211012
Nama      : RA
Alamat    : Serang
Agama     : Islam
Jabatan   : Administrator

Masukkan NIP pegawai yang ingin dicari: 211011
Biodata Pegawai dengan NIP 211011
Nama      : Nova
Alamat    : Tangerang
Agama     : Islam
Jabatan   : Developer
```

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **LAPORAN AKHIR PRAKTIKUM**

**Pertemuan ke-06**

**Searching**



**Disusun Oleh:**

Nama Lengkap : Nova Ardiansyah

NIM : 211011401309

Kelas : 04-TPLE008

**TEKNIK INFORMATIKA**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS PAMULANG**

Jl. Raya Puspitek No. 11 Buaran, Serpong Telp. (021) 7412566, Fax. (021) 7412566  
Tangerang Selatan - Banten

## A. TUGAS AKHIR

1. Buatlah program untuk melakukan pencarian terhadap data 12, 15, dan 37, dari sederetan data 34, 8, 50, 74, 87, 90, 12, 25, 20, 30, 35, 45, 40, 22, 29, 72, 60, 55, 53, 12, 32, 33, 12, 41, 12, Jika data yang dicari terdapat lebih dari satu, tentukan banyaknya dan sebutkan berada pada posisi berapa saja data yang dicari berada pada sederetan data.

**Jawab :**

```
#include <iostream>
using namespace std;

int main() {
    int data[] = {34, 8, 50, 74, 87, 90, 12, 25, 20, 30, 35, 45, 40, 22, 29, 72, 60, 55, 53, 12, 32,
33, 12, 12, 41, 12};
    int n = sizeof(data) / sizeof(data[0]); // * Menghitung ukuran array
    int count_12 = 0, count_15 = 0, count_37 = 0;
    int posisi_12[10], posisi_15[10], posisi_37[10];

    cout << "Nama \t: Nova Ardiansyah\n";
    cout << "NIM \t: 211011401309\n";
    cout << "=====\n\n";

    cout << "Data yang ada pada array: " << endl;
    for (int i = 0; i < n; i++) {
        cout << data[i] << " ";
    }
    cout << endl << endl;

    for (int i = 0; i < n; i++) {
        if (data[i] == 12) {
            count_12++;
            cout << "Data 12 ditemukan pada posisi ke-" << i << endl;
        } else if (data[i] == 15) {
            count_15++;
            cout << "Data 15 ditemukan pada posisi ke-" << i << endl;
        } else if (data[i] == 37) {
            count_37++;
            cout << "Data 37 ditemukan pada posisi ke-" << i << endl;
        }
    }

    cout << endl;
    cout << "Banyaknya data 12 yang ditemukan: " << count_12 << endl;
    cout << "Banyaknya data 15 yang ditemukan: " << count_15 << endl;
    cout << "Banyaknya data 37 yang ditemukan: " << count_37 << endl;

    return 0;
}
```

```
Nama    : Nova Ardiansyah
NIM     : 211011401309
=====
```

```
Data yang ada pada array:
```

```
34 8 50 74 87 90 12 25 20 30 35 45 40 22 29 72 60 55 53 12 32 33 12 12 4
```

```
Data 12 ditemukan pada posisi ke-6
```

```
Data 12 ditemukan pada posisi ke-19
```

```
Data 12 ditemukan pada posisi ke-22
```

```
Data 12 ditemukan pada posisi ke-23
```

```
Data 12 ditemukan pada posisi ke-25
```

```
Banyaknya data 12 yang ditemukan: 5
```

```
Banyaknya data 15 yang ditemukan: 0
```

```
Banyaknya data 37 yang ditemukan: 0
```

## 2. KESIMPULAN

Searching atau pencarian adalah proses untuk mencari nilai tertentu pada suatu struktur data. Metode-metode searching yang dapat digunakan antara lain sequential search, binary search, dan interpolation search. Sequential search adalah metode sederhana yang dilakukan dengan membandingkan setiap elemen pada struktur data satu per satu, sedangkan binary search bekerja dengan mengurutkan struktur data terlebih dahulu dan membaginya menjadi dua bagian secara terus menerus sampai nilai yang dicari ditemukan. Interpolation search mencari nilai dengan mengestimasi lokasi nilai tersebut berada di dalam struktur data.

Kecepatan proses pencarian data dipengaruhi oleh beberapa faktor, seperti ukuran data, metode pencarian yang digunakan, jenis penyimpanan data, dan kemampuan mesin untuk memproses data. Binary search dan interpolation search lebih cepat dibandingkan sequential search karena keduanya memiliki kemampuan membagi data menjadi dua bagian atau memperkirakan letak data yang akan dicari. Jenis penyimpanan data juga mempengaruhi kecepatan proses pencarian.