

**LAPORAN PRAKTIKUM**  
**MODUL VIII**  
**ALGORITMA SEARCHING**



**Disusun Oleh:**

Muhammad Raafi Al Hafiidh

2311102070

**Dosen:**

Wahyu Andi Saputra, S.Pd., M.Eng.

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

## **BAB I**

### **TUJUAN PRAKTIKUM**

- Mahasiswa mampu menjelaskan definisi dan konsep dasar Sequential Search dan Binary Search.
- Mahasiswa mampu menerapkan algoritma Sequential Search dan Binary Search pada pemrograman yang dibuat.
- Mahasiswa mampu menganalisis bagaimana cara kerja dari Sequential Search dan Binary Search.

## BAB II

### DASAR TEORI

Algoritma searching adalah bagian penting dalam ilmu komputer yang digunakan untuk menemukan elemen tertentu dalam kumpulan data. Dalam bahasa pemrograman C++, terdapat berbagai algoritma searching yang dapat digunakan untuk menyelesaikan berbagai permasalahan.

#### Jenis-jenis Algoritma Searching:

- **Pencarian Linear (Sequential Search):** Algoritma ini mencari elemen secara berurutan dari awal hingga akhir kumpulan data. Algoritma ini mudah diimplementasikan, tetapi memiliki kompleksitas waktu  $O(n)$ , yang berarti waktu yang dibutuhkan untuk menemukan elemen bergantung pada ukuran ( $n$ ) kumpulan data.
- **Pencarian Biner (Binary Search):** Algoritma ini hanya dapat digunakan pada kumpulan data yang terurut. Algoritma ini bekerja dengan membagi kumpulan data menjadi dua bagian secara berulang, dan kemudian mencari elemen pada bagian yang kemungkinan besar memuat elemen tersebut. Algoritma ini memiliki kompleksitas waktu  $O(\log n)$ , yang berarti jauh lebih cepat daripada pencarian linear untuk kumpulan data besar.
- **Pencarian Hashing:** Algoritma ini menggunakan tabel hash untuk memetakan elemen ke dalam indeks tertentu. Pencarian hashing sangat cepat untuk menemukan elemen, dengan kompleksitas waktu rata-rata  $O(1)$ . Namun, algoritma ini membutuhkan memori tambahan untuk menyimpan tabel hash.
- **Pencarian Pohon (Tree Search):** Algoritma ini menggunakan struktur data pohon untuk mencari elemen. Algoritma ini dapat digunakan untuk mencari elemen dalam struktur data hierarkis, seperti direktori file atau database.

#### Pemilihan Algoritma Searching:

Pemilihan algoritma searching yang tepat tergantung pada beberapa faktor, seperti:

- Ukuran kumpulan data
- Apakah kumpulan data terurut atau tidak
- Kecepatan yang dibutuhkan
- Memori yang tersedia

## BAB III

### LATIHAN DAN TUGAS

#### A. GUIDED

##### 1. Guided1

Buatlah sebuah project dengan menggunakan sequential search sederhana untuk melakukan pencarian data.

*Source Code*

```
#include <iostream>

using namespace std;

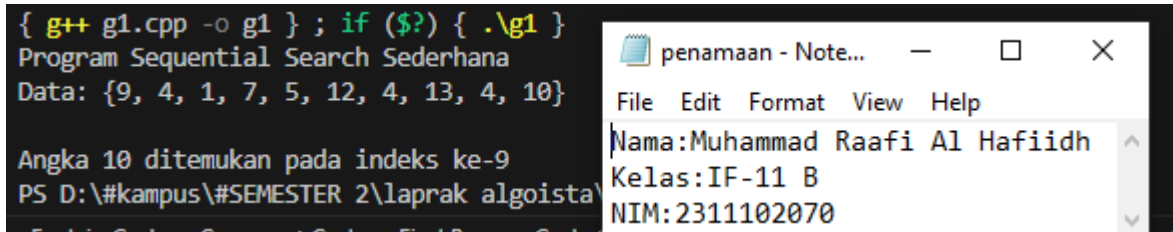
int main()
{
    const int n = 10;
    int *data = new int[n]{9, 4, 1, 7, 5, 12, 4, 13, 4, 10};
    int cari = 10;
    bool ketemu = false;
    int i;

    // algoritma sequential search
    for (i = 0; i < n; i++)
    {
        if (data[i] == cari)
        {
            ketemu = true;
            break;
        }
    }

    cout << "Program Sequential Search Sederhana" << endl;
    cout << "Data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}" << endl;

    if (ketemu)
    {
        cout << "\nAngka " << cari << " ditemukan pada indeks ke-" << i << endl;
    }
    else
    {
        cout << cari << "tidak dapat ditemukan pada data." << endl;
    }
    return 0;
}
```

*Screenshoot Program*



```
{ g++ g1.cpp -o g1 } ; if ($?) { .\g1 }
Program Sequential Search Sederhana
Data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}

Angka 10 ditemukan pada indeks ke-9
PS D:\#kampus\#SEMESTER 2\laprak algoista>
```

penamaan - Note... — □ ×

File Edit Format View Help

Nama:Muhammad Raafi Al Hafiidh  
Kelas:IF-11 B  
NIM:2311102070

### Deskripsi Program

Kode di atas menggunakan sequential search untuk mencari angka yang berada pada array yang sudah dibuat dan diberikan nilai sebelumnya, untuk angka yang dicari juga sudah ditentukan didalam program jadi program nantinya akan mencari nilai yang sama dengan yang ada didalam data array lalu menampilkan data tersebut berada pada posisi indeks nomor berapa.

## 2. Guided2

Buatlah sebuah project untuk melakukan pencarian data dengan menggunakan Binary Search.

### Source Code

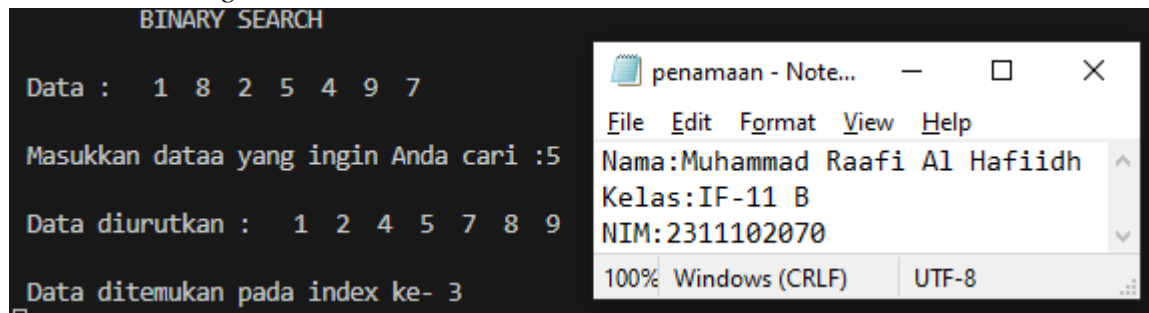
```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>
int dataa[7] = {1, 8, 2, 5, 4, 9, 7};
int cari;
void selection_sort()
{
    int temp, min, i, j;
    for (i = 0; i < 7; i++)
    {
        min = i;
        for (j = i + 1; j < 7; j++)
        {
            if (dataa[j] < dataa[min])
            {
                min = j;
            }
        }
        temp = dataa[i];
        dataa[i] = dataa[min];
        dataa[min] = temp;
    }
}
void binarysearch()
{
    // searching
    int awal, akhir, tengah, b_flag = 0;
    awal = 0;
    akhir = 7;
    while (b_flag == 0 && awal <= akhir)
    {
```

```

        tengah = (awal + akhir) / 2;
        if (dataa[tengah] == cari)
        {
            b_flag = 1;
            break;
        }
        else if (dataa[tengah] < cari)
            awal = tengah + 1;
        else
            akhir = tengah - 1;
    }
    if (b_flag == 1)
        cout << "\n Data ditemukan pada index ke- " <<
tengah << endl;
    else
        cout << "\n Data tidak ditemukan\n";
}
int main()
{
    cout << "\t BINARY SEARCH " << endl;
    cout << "\n Data : ";
    // tampilkan dataa awal
    for (int x = 0; x < 7; x++)
        cout << setw(3) << dataa[x];
    cout << endl;
    cout << "\n Masukkan dataa yang ingin Anda cari :";
    cin >> cari;
    cout << "\n Data diurutkan : ";
    // urutkan dataa dengan selection sort
    selection_sort();
    // tampilkan dataa setelah diurutkan
    for (int x = 0; x < 7; x++)
        cout << setw(3) << dataa[x];
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```

### Screenshoot Program



### Deskripsi Program

Kode di atas menggunakan binary search untuk mencari angka yang berada pada array yang sudah dibuat dan diberikan nilai sebelumnya, untuk angka yang dicari nantinya pengguna akan diminta memasukkan angka yang sudah tertera diatas untuk mencari angka tersebut ada diposisi indeks berapa.

## B. UNGUIDED

### 1. Unguided 1

Buatlah sebuah program untuk mencari sebuah huruf pada sebuah kalimat yang sudah diinput dengan menggunakan Binary Search.

*Source Code*

```
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

bool binarySearch(string sentence, char target)
{
    int left = 0;
    int right = sentence.length() - 1;

    while (left <= right)
    {
        int mid = left + (right - left) / 2;

        if (sentence[mid] == target)
        {
            return true;
        }

        if (sentence[mid] < target)
        {
            left = mid + 1;
        }
        else
        {
            right = mid - 1;
        }
    }

    return false;
}

int main()
{
    string sentence;
    char target;

    cout << "Masukkan kalimat : ";
    getline(cin, sentence);
```

```

        cout << "Masukkan huruf yang akan dicari : ";
        cin >> target;

        // Mengubah seluruh huruf menjadi lowercase untuk
        pencarian yang tidak case-sensitive
        transform(sentence.begin(), sentence.end(),
        sentence.begin(), ::tolower);
        target = tolower(target);

        if (binarySearch(sentence, target))
        {
            cout << "Huruf " << target << " ditemukan
            didalam kalimat." << endl;
        }
        else
        {
            cout << "Huruf tidak ditemukan didalam kalimat."
            << endl;
        }

        return 0;
    }

```

### Screenshoot Program

```

PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 8>
{ g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if
Masukkan kalimat : muhammad raafi al hafiidh
Masukkan huruf yang akan dicari : h
Huruf h ditemukan didalam kalimat.
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 8>

```

penamaan - Note... — □ ×

File Edit Format View Help

Nama:Muhammad Raafi Al Hafiidh

Kelas:IF-11 B

NIM:2311102070

### Deskripsi Program

Kode diatas menggunakan binary search untuk mencari apakah ada salah satu huruf didalam kalimat yang diinputkan sebelumnya. Untuk kalimat dan kata akan diinputkan oleh pengguna dan diakhir akan ada aksi dimana kondisi bernilai benar dan salah.

## 2. Unguided 2

Buatlah sebuah program yang dapat menghitung banyaknya huruf vocal dalam sebuah kalimat!

### Source Code

```

#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

int countVowels(string sentence)
{
    int count = 0;
    transform(sentence.begin(), sentence.end(),
    sentence.begin(), ::tolower);

```



```

        for (char c : sentence)
        {
            if (c == 'a' || c == 'e' || c == 'i' || c == 'o'
|| c == 'u')
            {
                count++;
            }
        }

        return count;
    }

    int main()
    {
        string sentence;

        cout << "Masukkan kalimat yang ingin di cari huruf
vokalnya : ";
        getline(cin, sentence);

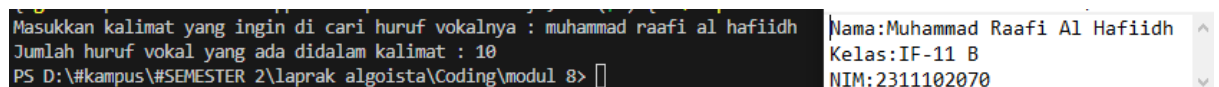
        int vowelCount = countVowels(sentence);

        cout << "Jumlah huruf vokal yang ada didalam kalimat
: " << vowelCount << endl;

        return 0;
    }

```

### *Screenshoot Program*



### **Deskripsi Program**

Untuk program diatas menggunakan metode binary searching untuk mencari huruf vocal pada kalimat yang pengguna inputkan diawal. Setelah kalimat diinputkan nantinya akan muncuk hasil dari jumlah huruf vocal yang ada pada kalimat yang pengguna inputkan.

### **3. Unguided 3**

Diketahui data = 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. Hitunglah berapa banyak angka 4 dengan menggunakan algoritma Sequential Search!

#### *Source Code*

```

#include <iostream>
#include <vector>

// Fungsi untuk menghitung berapa banyak angka 4 dalam
data

```

```

int countNumber(const std::vector<int> &data, int
target)
{
    int count = 0;

    for (int num : data)
    {
        if (num == target)
        {
            count++;
        }
    }

    return count;
}

int main()
{
    std::vector<int> data = {9, 4, 1, 4, 7, 10, 5, 4,
12, 4};
    int target = 4;

    // int ukuran = sizeof(data) / sizeof(data[0]);
    int ukuran = 10;

    // Menghitung berapa banyak angka 4 dalam data
    int count = countNumber(data, target);
    std::cout << "\nData : ";
    for (int i = 0; i < ukuran; i++)
    {
        std::cout << data[i] << ", ";
    }
    std::cout << "\nBanyaknya angka " << target << "
dalam data: " << count << std::endl;

    return 0;
}

```

### *Screenshoot Program*

```

Data : 9, 4, 1, 4, 7, 10, 5, 4, 12, 4,
Banyaknya angka 4 dalam data: 4
PS D:\#kampus\#SEMESTER 2\laprak algoista\Coding\modul 8>

```

```

Nama:Muhammad Raafi Al Hafiidh
Kelas:IF-11 B
NIM:2311102070

```

### **Deskripsi Program**

Untuk program diatas menggunakan sequential search untuk menghitung berapa kali angka 4 muncul dalam sebuah data. Data yang diberikan adalah 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. Program akan melakukan perulangan pada setiap elemen data dan memeriksa apakah angka tersebut sama dengan 4. Jika iya, maka counter akan bertambah satu. Setelah perulangan selesai, program akan mengembalikan jumlah angka 4 yang ditemukan dan menampilkannya.

Dengan demikian, program ini dapat digunakan untuk menghitung kemunculan angka 4 dalam data menggunakan algoritma Sequential Search.

## **BAB IV**

### **KESIMPULAN**

Kesimpulan yang dapat diambil dari laporan praktikum dengan “*Modul 8 : ALGORITMA SEARCHING*” adalah bahwa praktikum ini bertujuan untuk mempelajari fungsi dari penggunaan Sequential Searching dan Binary Searching dalam pemrograman. Praktisi jadi bisa memahami penggunaan dari konsep tersebut dan bagaimana kegunaan dari konsep tersebut dalam program yang dibuat, serta setelah melakukan pratikum, mahasiswa mampu menerapkan konsep tersebut didalam program yang mereka buat.

### **REFERENSI :**

"A Survey of Efficient String Matching Algorithms" oleh Gonzalo Navarro