

**LAPORAN PRAKTIKUM**  
**MODUL 8**  
**ALGORITMA SEARCHING**



**Disusun oleh:**

**Annisa Al Jauhar**

**NIM: 2311102014**

**Dosen Pengampu:**

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

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

## **BAB I**

### **TUJUAN PRAKTIKUM**

#### **A. TUJUAN PRAKTIKUM**

1. Menunjukkan beberapa algoritma dalam pencarian.
2. Menunjukkan bahwa pencarian merupakan suatu persoalan diselesaikan dengan beberapa algoritma yang berbeda.
3. Dapat memilih algoritma yang paling sesuai untuk menyelesaikan suatu permasalahan pemrograman.

## **BAB II**

### **DASAR TEORI**

#### **B. DASAR TEORI**

Pencarian (Searching) yaitu proses menemukan suatu nilai tertentu pada kumpulan data. Hasil pencarian adalah salah satu dari tiga keadaan ini: data ditemukan, data ditemukan lebih dari satu, atau data tidak ditemukan. Searching juga dapat dianggap sebagai proses pencarian suatu data di dalam sebuah array dengan cara mengecek satu persatu pada setiap index baris atau setiap index kolomnya dengan menggunakan teknik perulangan untuk melakukan pencarian data. Terdapat 2 metode pada algoritma Searching, yaitu:

a. Sequential Search Sequential Search merupakan salah satu algoritma pencarian data yang biasa digunakan untuk data yang berpola acak atau belum terurut. Sequential search juga merupakan teknik pencarian data dari array yang paling mudah, dimana data dalam array dibaca satu demi satu dan diurutkan dari index terkecil ke index terbesar, maupun sebaliknya. Konsep Sequential Search yaitu:

- Membandingkan setiap elemen pada array satu per satu secara berurut.
- Proses pencarian dimulai dari indeks pertama hingga indeks terakhir.
- Proses pencarian akan berhenti apabila data ditemukan. Jika hingga akhir array data masih juga tidak ditemukan, maka proses pencarian tetap akan dihentikan.

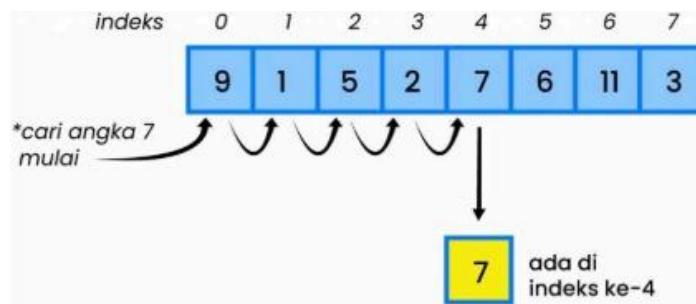
-Proses perulangan pada pencarian akan terjadi sebanyak jumlah N elemen pada array. Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

- 1)  $i \leftarrow 0$
- 2) ketemu  $\leftarrow$  false
- 3) Selama (tidak ketemu) dan ( $i \leq N$ ) kerjakan baris 4
- 4) Jika ( $\text{Data}[i] = x$ ) maka ketemu  $\leftarrow$  true, jika tidak  $i \leftarrow i + 1$
- 5) Jika (ketemu) maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.

Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian sekuensial.

```
int SequentialSearch (int x)
{
    int i = 0;
    bool ketemu = false;
    while ((!ketemu) && (i < Max)){
        if (Data[i] == x)
            ketemu = true;
        else
            i++;
    }
    if (ketemu)
        return i;
    else
        return -1;
}
```

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai  $-1$ . Contoh dari Sequential Search, yaitu:  $\text{Int } A[8] = \{9, 1, 5, 2, 7, 6, 11, 3\}$



Misalkan, dari data di atas angka yang akan dicari adalah angka 7 dalam array A, maka proses yang akan terjadi yaitu:

- Pencarian dimulai pada index ke-0 yaitu angka 9, kemudian dicocokkan dengan angka yang akan dicari, jika tidak sama maka pencarian akan dilanjutkan ke index selanjutnya.
- Pada index ke-1, yaitu angka 1, juga bukan angka yang dicari, maka pencarian akan dilanjutkan pada index selanjutnya.
- Pada index ke-2 dan index ke-3 yaitu angka 5 dan 2, juga bukan angka yang dicari, sehingga pencarian dilanjutkan pada index selanjutnya.
- Pada index ke-4 yaitu angka 7 dan ternyata angka 7 merupakan angka yang dicari, sehingga pencarian akan dihentikan dan proses selesai.

#### b. Binary Search

Binary Search termasuk ke dalam interval search, dimana algoritma ini merupakan algoritma pencarian pada array/list dengan elemen terurut.

Pada metode ini, data harus diurutkan terlebih dahulu dengan cara data dibagi menjadi dua bagian (secara logika), untuk setiap tahap pencarian. Dalam penerapannya algoritma ini sering digabungkan dengan algoritma sorting karena data yang akan digunakan harus sudah terurut terlebih dahulu. Konsep Binary Search:

- Data diambil dari posisi 1 sampai posisi akhir N.
- Kemudian data akan dibagi menjadi dua untuk mendapatkan posisi data tengah.
- Selanjutnya data yang dicari akan dibandingkan dengan data yang berada di posisi tengah, apakah lebih besar atau lebih kecil.
- Apabila data yang dicari lebih besar dari data tengah, maka dapat dipastikan bahwa data yang dicari kemungkinan berada di sebelah kanan dari data tengah. Proses pencarian selanjutnya akan dilakukan pembagian data menjadi dua bagian pada bagian kanan dengan acuan posisi data tengah akan menjadi posisi awal untuk pembagian tersebut.

## BAB III

### GUIDED

#### 1. Guided 1

##### Source code

```
#include <iostream>
using namespace std;
int main()
{
    int n = 10;
    int data[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 << "\t Program Sequential Search Sederhana\n " << endl;
    cout << " data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}" << endl;

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

## Screenshoot program

```
ve\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data Rabu\Modul 8\ ; if ($?) { g++ guided_1.c
($?) { .\guided_1 }
    Program Sequential Search Sederhana

    data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}

    angka 10 ditemukan pada indeks ke - 9
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data Rabu\Modul 8>
```

## Deskripsi program

Program ini mencari keberadaan suatu angka dalam sebuah array dengan cara menelusuri setiap elemen satu per satu dari awal hingga akhir array. Pada awalnya, program mendefinisikan sebuah array dengan panjang 10 dan beberapa angka sebagai data yang akan dicari. Kemudian, program mencari angka tertentu dalam array tersebut dengan menggunakan algoritma Sequential Search. Jika angka tersebut ditemukan, program akan mengeluarkan pesan bahwa angka tersebut ditemukan beserta indeksnya dalam array. Jika tidak ditemukan, program akan memberikan pesan bahwa angka tersebut tidak ditemukan dalam data yang ada.



## GUIDED

### 2. Guided 2

#### Source code

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

int arr[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 (arr[j] < arr[min]) {
                min = j;
            }
        }
        temp = arr[i];
        arr[i] = arr[min];
        arr[min] = temp;
    }
}

void binarysearch() {
    // searching
    int awal, akhir, tengah, b_flag = 0;
    awal = 0;
    akhir = 6; // Corrected the value from 7 to 6
    while (b_flag == 0 && awal <= akhir) {
        tengah = (awal + akhir) / 2;
        if (arr[tengah] == cari) {
            b_flag = 1;
            break;
        } else if (arr[tengah] < cari) {
            awal = tengah + 1;
        } else {
            akhir = tengah - 1;
        }
    }
}
```

```

    }

    }

    if (b_flag == 1) {
        cout << "\narray ditemukan pada index ke " << tengah << endl;
    } else {
        cout << "\nData tidak ditemukan\n";
    }
}

int main() {
    cout << "\tBINARY SEARCH" << endl;
    cout << "\nData:";
    // tampilkan data awal
    for (int x = 0; x < 7; x++) {
        cout << setw(3) << arr[x];
    }
    cout << endl;
    cout << "\nMasukkan data yang ingin Anda cari : ";
    cin >> cari;
    cout << "\nData diurutkan : ";

    // urutkan data dengan selection sort
    selection_sort();
    // tampilkan data setelah diurutkan

    for (int x = 0; x < 7; x++) {
        cout << setw(3) << arr[x];
    }
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```

## Screenshoot program

```
ve\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 8\" ; if ($?) {  
($?) { .\guided_2 }  
    BINARY SEARCH  
  
Data: 1 8 2 5 4 9 7  
  
Masukkan data yang ingin Anda cari : 5  
  
Data diurutkan : 1 2 4 5 7 8 9  
  
array ditemukan pada index ke 3
```

## Deskripsi program

Program ini bertujuan untuk mencari keberadaan suatu angka dalam array yang telah diurutkan secara menaik. Pada awalnya, program mendefinisikan sebuah array dengan panjang 7 dan beberapa angka sebagai data yang akan dicari. Kemudian, program mengimplementasikan fungsi `selection_sort()` untuk mengurutkan data dalam array menggunakan algoritma Selection Sort. Setelah itu, program meminta pengguna untuk memasukkan angka yang ingin dicari. Setelah angka dimasukkan, program akan menampilkan data yang telah diurutkan menggunakan Selection Sort. Selanjutnya, program mencari angka yang dimasukkan oleh pengguna dalam array yang telah diurutkan menggunakan algoritma Binary Search. Jika angka tersebut ditemukan, program akan menampilkan indeks tempat angka tersebut ditemukan. Jika tidak ditemukan, program akan memberikan pesan bahwa data tidak ditemukan dalam array.

## LATIHAN KELAS - UNGUIDED

### 1. Unguided 1

#### A. Source code

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

using namespace std;

// Fungsi untuk melakukan pencarian linear pada string
int linearSearch(const string& kalimat, char hurufDicari) {
    for (int i = 0; i < kalimat.length(); ++i) {
        if (kalimat[i] == hurufDicari) {
            return i; // Mengembalikan indeks jika huruf
ditemukan
        }
    }
    return -1; // Mengembalikan -1 jika huruf tidak ditemukan
}

int main() {
    string kalimat;
    char hurufDicari;

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

    cout << "Masukkan huruf yang ingin dicari: ";
    cin >> hurufDicari;

    // Melakukan pencarian linear pada kalimat
    int hasilPencarian = linearSearch(kalimat, hurufDicari);
```

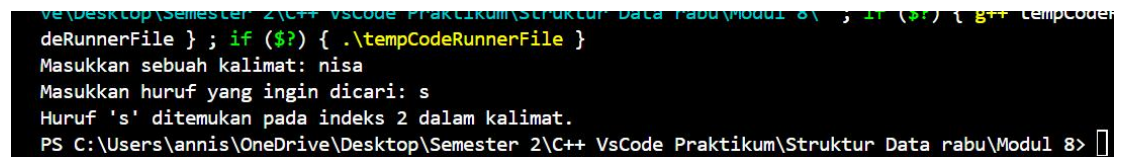
```

        if (hasilPencarian != -1) {
            cout << "Huruf '" << hurufDicari << "' ditemukan pada
indeks " << hasilPencarian << " dalam kalimat." << endl;
        } else {
            cout << "Huruf '" << hurufDicari << "' tidak ditemukan
dalam kalimat." << endl;
        }

        return 0;
    }
}

```

## Screenshoot program



```

PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data Rabu\Modul 8> g++ tempCode
deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan sebuah kalimat: nisa
Masukkan huruf yang ingin dicari: s
Huruf 's' ditemukan pada indeks 2 dalam kalimat.
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data Rabu\Modul 8>

```

## Deskripsi program

Program ini bertujuan untuk mencari indeks pertama kemunculan suatu huruf dalam sebuah kalimat yang dimasukkan oleh pengguna. Pada awalnya, program menggunakan `#include` untuk menyertakan pustaka yang diperlukan seperti `iostream` dan `string`. Selanjutnya, program mendefinisikan sebuah fungsi bernama `linearSearch()` yang menerima dua parameter, yaitu sebuah string (kalimat) dan sebuah karakter (hurufDicari). Fungsi ini melakukan pencarian linear pada string untuk mencari huruf yang diinginkan, dan mengembalikan indeks pertama kemunculan huruf tersebut dalam kalimat. Dalam fungsi `main()`, program meminta pengguna untuk memasukkan sebuah kalimat dan sebuah huruf yang ingin dicari. Setelah itu, program memanggil fungsi `linearSearch()` untuk mencari huruf yang dimasukkan dalam kalimat, kemudian menampilkan hasilnya ke layar. Jika huruf tersebut ditemukan, program akan menampilkan

indeks pertama kemunculannya dalam kalimat. Jika tidak ditemukan, program akan memberikan pesan bahwa huruf tersebut tidak ditemukan dalam kalimat.

## 2. Unguided 2

### B. Source code

```
#include <iostream>
#include <string>
#include <cctype>

using namespace std;

// Fungsi untuk menghitung jumlah huruf vokal dalam sebuah
kalimat
int hitungVokal(const string& teks) {
    int jumlahVokal = 0;
    for (char karakter : teks) {
        // Mengubah huruf menjadi huruf kecil untuk memudahkan
perbandingan
        char huruf = tolower(karakter);
        // Memeriksa apakah karakter tersebut adalah huruf vokal
        if (huruf == 'a' || huruf == 'e' || huruf == 'i' || huruf
== 'o' || huruf == 'u') {
            jumlahVokal++;
        }
    }
    return jumlahVokal;
}

int main() {
    string teks;

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

```

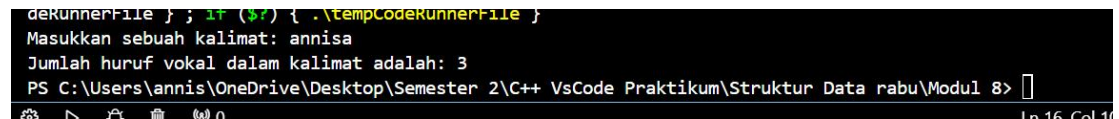
// Menghitung jumlah huruf vokal dalam teks
int totalVokal = hitungVokal(teks);

cout << "Jumlah huruf vokal dalam kalimat adalah: " <<
totalVokal << endl;

return 0;
}

```

## Screenshoot program



```

dekunnerFile } ; if ($?) { .\tempCodeKunnerFile }
Masukkan sebuah kalimat: annisa
Jumlah huruf vokal dalam kalimat adalah: 3
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 8>

```

## Deskripsi program

Program di atas adalah contoh implementasi dari sebuah progra yang bertujuan untuk menghitung jumlah huruf vokal dalam sebuah kalimat yang dimasukkan oleh pengguna. Pada awalnya, program menggunakan `#include` untuk menyertakan pustaka yang diperlukan seperti `iostream`, `string`, dan `cctype`. Kemudian, program mendefinisikan sebuah fungsi bernama `hitungVokal()` yang menerima sebuah string sebagai parameter dan mengembalikan jumlah huruf vokal dalam string tersebut. Fungsi ini menggunakan perulangan `for` untuk mengiterasi setiap karakter dalam string dan menggunakan fungsi `tolower()` untuk mengubah karakter menjadi huruf kecil agar memudahkan perbandingan. Selanjutnya, program di dalam fungsi `main()` meminta pengguna untuk memasukkan sebuah kalimat menggunakan fungsi `getline()` dan menyimpannya dalam variabel `teks`. Setelah itu, program memanggil fungsi `hitungVokal()` untuk menghitung jumlah huruf vokal dalam kalimat yang dimasukkan, kemudian menampilkannya ke layar.

### 3. Unguided 3

#### C. Source code

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

// Fungsi untuk melakukan pencarian Sequential Search
int cariSequensial(int data[], int ukuran, int target) {
    int hitung = 0;
    for (int i = 0; i < ukuran; ++i) {
        if (data[i] == target) {
            hitung++;
        }
    }
    return hitung;
}

int main() {
    int data[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int ukuran = sizeof(data) / sizeof(data[0]);
    int target = 4;

    // Melakukan pencarian Sequential
    int jumlah = cariSequensial(data, ukuran, target);

    cout << "Banyaknya angka 4 dalam data adalah: " << jumlah <<
endl;

    return 0;
}
```



## Screenshoot program

```
deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
Banyaknya angka 4 dalam data adalah: 4  
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 8> █  
⚙ ▶ 🔍 🗑 🔊 0
```

## Deskripsi program

Program ini bertujuan untuk mencari berapa kali suatu angka tertentu muncul dalam sebuah array. Pada awalnya, program mendefinisikan sebuah array dengan beberapa angka sebagai data yang akan dicari. Kemudian, program menghitung ukuran dari array tersebut. Selanjutnya, program memanggil fungsi cariSequensial() untuk melakukan pencarian berulang kali terhadap angka target dalam array menggunakan algoritma Sequential Search. Fungsi ini akan mengembalikan jumlah kemunculan angka target dalam array. Setelah pencarian selesai, program akan menampilkan jumlah kemunculan angka target tersebut dalam array.

## **BAB IV**

### **KESIMPULAN**

Kesimpulan dari program ini adalah sebagai berikut:

1. Program memanfaatkan struktur dasar bahasa C++, termasuk penggunaan tipe data string dan fungsi-fungsi bawaan dari pustaka standar seperti `getline()` untuk mengambil input string dari pengguna.
2. Program mengimplementasikan algoritma pencarian linear untuk mencari sebuah karakter dalam sebuah string. Algoritma ini sederhana dan efektif untuk ukuran data yang kecil atau tidak terurut.
3. Program menunjukkan cara menggunakan fungsi-fungsi dasar bahasa C++ seperti perulangan `for` dan `if` untuk melakukan iterasi dan pengecekan kondisi.
4. Program memberikan pengalaman interaktif kepada pengguna, memungkinkan mereka untuk melihat hasil pencarian huruf dalam kalimat yang mereka masukkan.