

LAPORAN PRAKTIKUM

MODUL VIII Searching



**Disusun oleh:
Aryo Tegar Sukarno
NIM: 2311102018**

**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. Menunjukkan beberapa algoritma dalam Pencarian.
- b. Dapat memilih algoritma yang paling sesuai untuk menyelesaikan suatu permasalahan pemrograman.

BAB II

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.

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 inisiering 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 datatengah.
- 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 menjadidua bagian pada bagian kanan dengan acuan posisi data tengah akan menjadi posisi awal untuk pembagian tersebut.
- Apabila data yang dicari lebih kecil dari data tengah, maka dapat dipastikan bahwa data yang dicari kemungkinan berada di sebelah kiri dari data tengah. Proses pencarian selanjutnya akan dilakukan pembagian data menjadi dua bagian pada bagian kiri. Dengan acuan posisi data tengah akan menjadi posisiakhir untuk pembagian selanjutnya.
- Apabila data belum ditemukan, maka pencarian akan dilanjutkan dengankembali membagi data menjadi dua.
- Namun apabila data bernilai sama, maka data yang dicari langsungditemukan dan pencarian dihentikan.

BAB III

GUIDED

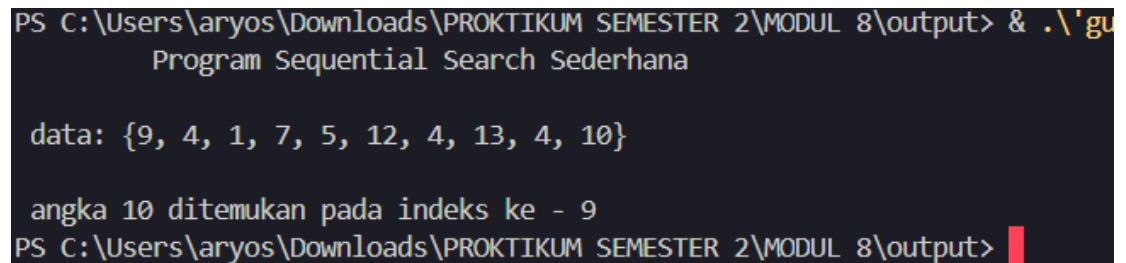
1. Guided 1

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



```
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output> & .\'gu  
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\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output>
```

DESKRIPSI PROGRAM

Program ini melakukan pencarian nilai dalam array menggunakan algoritma pencarian berurutan (sequential search). Array `data` berisi 10 elemen, dan program mencari nilai `cari` yang diset ke 10. Program melakukan iterasi melalui elemen-elemen array dan memeriksa apakah ada elemen yang sama dengan nilai yang dicari. Jika ditemukan, program menampilkan indeks elemen tersebut; jika tidak, program menyatakan bahwa nilai tersebut tidak ditemukan dalam array.

1. Guided

Source code

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

// Deklarasi array dan variabel untuk pencarian
int arrayData[7] = {1, 8, 2, 5, 4, 9, 7};
int cari;

void selection_sort(int arr[], int n)
{
    int temp, min;
    for (int i = 0; i < n - 1; i++)
    {
        min = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[min])
            {
                min = j;
            }
        }
        // Tukar elemen
        temp = arr[i];
        arr[i] = arr[min];
        arr[min] = temp;
    }
}

void binary_search(int arr[], int n, int target)
{
    int awal = 0, akhir = n - 1, tengah, b_flag = 0;
    while (b_flag == 0 && awal <= akhir)
    {
        tengah = (awal + akhir) / 2;
```

```

        if (arr[tengah] == target)
        {
            b_flag = 1;
            break;
        }
        else if (arr[tengah] < target)
        {
            awal = tengah + 1;
        }
        else
        {
            akhir = tengah - 1;
        }
    }
    if (b_flag == 1)
        cout << "\nData ditemukan pada index ke-" << tengah <<
endl;
    else
        cout << "\nData tidak ditemukan\n";
}

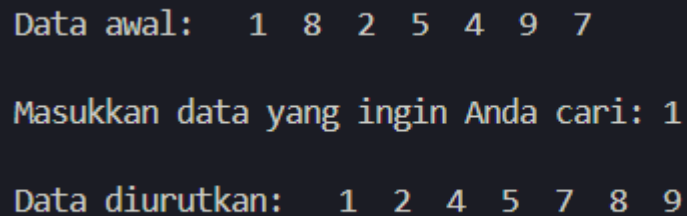
int main()
{
    cout << "\tBINARY SEARCH" << endl;
    cout << "\nData awal: ";
    // Tampilkan data awal
    for (int x = 0; x < 7; x++)
    {
        cout << setw(3) << arrayData[x];
    }
    cout << endl;
    cout << "\nMasukkan data yang ingin Anda cari: ";
    cin >> cari;
    // Urutkan data dengan selection sort
    selection_sort(arrayData, 7);

```



```
cout << "\nData diurutkan: ";  
// Tampilkan data setelah diurutkan  
for (int x = 0; x < 7; x++)  
{  
    cout << setw(3) << arrayData[x];  
}  
cout << endl;  
// Lakukan binary search  
binary_search(arrayData, 7, cari);  
return 0;  
}
```

Screenshoot program



```
Data awal:  1  8  2  5  4  9  7  
  
Masukkan data yang ingin Anda cari: 1  
  
Data diurutkan:  1  2  4  5  7  8  9
```

Deskripsi Program

Program diatas adalah program yang mengurutkan array menggunakan selection sort, lalu mencari nilai dalam array yang telah diurutkan menggunakan binary search.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>
#include <algorithm>
#include <cstring>

using namespace std;

void Selection_Sort(char arr[], int n) {
    int i, j, min_idx;
    char temp;
    for (i = 0; i < n-1; i++) {
        min_idx = i;
        for (j = i+1; j < n; j++) {
            if (arr[j] < arr[min_idx]) {
                min_idx = j;
            }
        }
        temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}

bool BinarySearch(char arr[], int n, char cari, int &index) {
    int awal = 0, akhir = n - 1, tengah;
    while (awal <= akhir) {
        tengah = (awal + akhir) / 2;
        if (arr[tengah] == cari) {
            index = tengah;
            return true;
        } else if (arr[tengah] < cari) {
```

```

        awal = tengah + 1;
    } else {
        akhir = tengah - 1;
    }
}
return false;
}

int main() {
    string kalimat;
    char cari;

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

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

    int n = kalimat.length();
    char arr[n + 1];
    strcpy(arr, kalimat.c_str());

    Selection_Sort(arr, n);

    cout << "\nKalimat setelah diurutkan: ";
    for (int i = 0; i < n; i++) {
        cout << arr[i];
    }
    cout << endl;

    int index;
    if (BinarySearch(arr, n, cari, index)) {
        cout << "\nHuruf '" << cari << "' ditemukan pada indeks ke-" << index << " setelah pengurutan." << endl;
    }
}

```

```

    } else {
        cout << "\nHuruf '" << cari << "' tidak ditemukan." <<
endl;
    }

    return 0;
}

```

Screenshoot program

```

PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8> cd 'c:\Users\aryos\Downloads\p
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output> & .\'unguided1.exe'
Masukkan kalimat: hai
Masukkan huruf yang ingin dicari: i

Kalimat setelah diurutkan: ahi

Huruf 'i' ditemukan pada indeks ke-2 setelah pengurutan.
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output>

```

Deskripsi Program

Kode program yang diberikan adalah program C++ yang mengimplementasikan algoritma pengurutan seleksi (Selection Sort) dan pencarian biner (Binary Search) untuk mencari sebuah karakter dalam sebuah kalimat. Berikut adalah penjelasan singkat dari setiap bagian kode

2. Unguided 2

Source code

```

#include <iostream>
#include <cctype>

using namespace std;

// Fungsi untuk memeriksa apakah karakter adalah huruf vokal

```

```
bool cekVokal(char c)
{
    c = tolower(c);
    return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c ==
'u');
}

// Fungsi untuk menghitung jumlah huruf vokal dalam string
int hitungVokal(const string &str)
{
    int count = 0;
    for (char c : str)
    {
        if (cekVokal(c))
        {
            count++;
        }
    }
    return count;
}

int main()
{
    string kalimat;

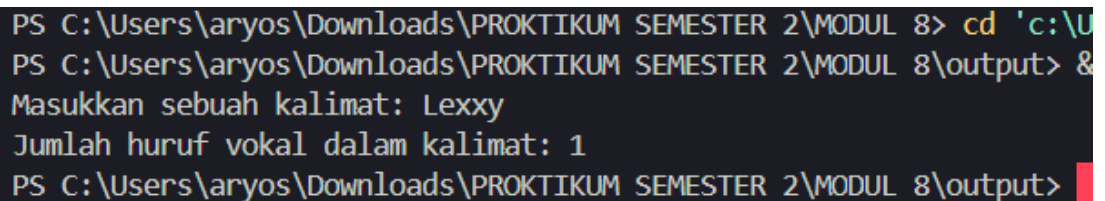
    cout << "Masukkan sebuah kalimat: ";
    getline(cin, kalimat);

    // Hitung jumlah huruf vokal dalam kalimat
    int jumlahVokal = hitungVokal(kalimat);

    // Tampilkan jumlah huruf vokal
    cout << "Jumlah huruf vokal dalam kalimat: " << jumlahVokal
<< endl;
```

```
    return 0;
}
```

Screenshoot program



```
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8> cd 'c:\U
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output> &
Masukkan sebuah kalimat: Lexxy
Jumlah huruf vokal dalam kalimat: 1
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output>
```

Deskripsi Program

Kode yang diberikan adalah program C++ yang menghitung jumlah huruf vokal dalam sebuah kalimat yang dimasukkan oleh pengguna. Berikut adalah penjelasan singkat dari setiap bagian kode

3. Unguided 3

Source code

```
#include <iostream>

using namespace std;

int hitungAngka(int data[], int n, int cari) {
    int count = 0;
    for (int i = 0; i < n; i++) {
        if (data[i] == cari) {
            count++;
        }
    }
}
```

```

        return count;
    }

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

        int jumlah = hitungAngka(data, n, cari);

        cout << "Data: {9, 4, 1, 4, 7, 10, 5, 4, 12, 4}" << endl;
        cout << "Jumlah angka " << cari << " dalam data adalah: " <<
        jumlah << endl;

        return 0;
    }

```

Screenshoot program

```

PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8> cd 'c:\Users\
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output> & .\
Data: {9, 4, 1, 4, 7, 10, 5, 4, 12, 4}
Jumlah angka 4 dalam data adalah: 4
PS C:\Users\aryos\Downloads\PROKTIKUM SEMESTER 2\MODUL 8\output> 

```

Deskripsi Program

Kode yang diberikan adalah program C++ yang menghitung berapa kali angka tertentu muncul dalam sebuah array.

BAB IV

KESIMPULAN