LAPORAN PRAKTIKUM

MODUL VIII
“Algoritma Searching“

Disusun oleh:
Shiva Indah Kurnia
NIM 2311102035

Dosen Pengampu:
Wahyu Andi Saputra, S.Pd., M.Eng

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

TUJUAN PRAKTIKUM

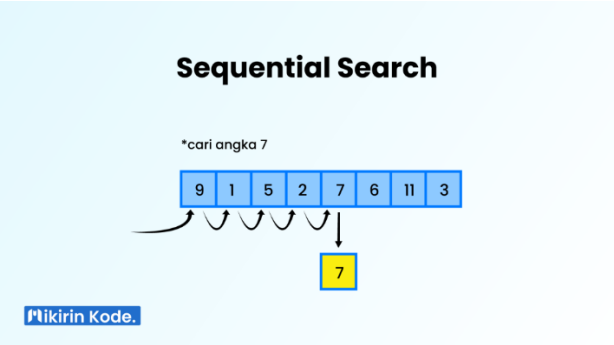
Setelah melakukan praktikum ini diharapkan mahasiswa dapat:

1. Mahasiswa diharapkan mampu menjelaskan dan memahami konsep dari Algoritma Searching itu sendiri.
2. Mahasiswa diharapkan mampu mengaplikasikan Sequential Search dan Binary Search kedalam pemrograman C++.
3. Mahasiswa diharapkan mampu menyelesaikan studi kasus menggunakan penyelesaian Sequential Search dan Binary Search.

DASAR TEORI

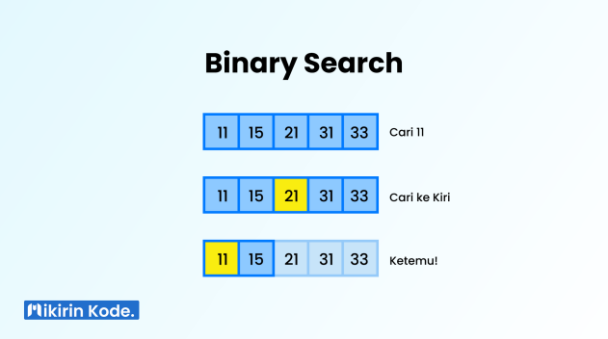
Sequential search dan binary search adalah dua jenis algoritma pencarian yang sering digunakan pada program untuk menelusuri suatu data yang dicari.

- Sequential Search
    1. Algoritma ini membandingkan setiap elemen array satu per satu secara berurutan, mulai dari elemen pertama, sampai elemen yang dicari ditemukan atau sampai semua elemen diperiksa.
    2. Jika elemen ditemukan, ia mengembalikan indeksnya, jika tidak -1.
    3. Contoh: Linear Search.



Gambar 1. Sequential Search

- Binary Search
    1. Algoritma ini hanya dapat digunakan pada data yang sudah terurut.
    2. Algoritma ini membagi data menjadi dua bagian setiap kali, dan memeriksa apakah elemen yang dicari berada di setengah kiri atau setengah kanan.
    3. Apabila ditemukan kecocokan nilai maka akan mengembalikan output, jika tidak pencarian akan terus berlanjut hingga akhir dari pembagian jumlah elemen tersebut.
    4. Contoh: Binary Search.



Gambar 2. Binary Search

Perbedaan antara sequential search dan binary search:

1. Sequential search membandingkan setiap elemen array satu per satu secara berurutan, sedangkan binary search membagi data menjadi dua bagian setiap kali dan memeriksa apakah elemen yang dicari berada di setengah kiri atau setengah kanan.
2. Sequential search dapat digunakan pada data yang tidak terurut, sedangkan binary search hanya dapat digunakan pada data yang sudah terurut.
3. Binary search lebih efisien dari sisi waktu dibandingkan dengan sequential search karena binary search hanya memerlukan logaritma basis 2 dari jumlah data yang dicari, sedangkan sequential search memerlukan waktu yang lebih lama karena harus memeriksa setiap elemen satu per satu secara berurutan.
4. Namun, binary search lebih rumit daripada sequential search dan memerlukan data yang sudah terurut.

Berikut adalah beberapa hal yang perlu dipertimbangkan saat menggunakan Algoritma Searching:

- Kapan menggunakan sequential search: Data tidak terurut atau dalam keadaan acak, Jumlah data yang dicari relatif kecil, Data yang dicari berada di awal atau tengah array.
- Kapan menggunakan binary search: Data sudah terurut, Jumlah data yang dicari relatif besar, Data yang dicari berada di akhir array atau tidak ada dalam array.

Namun, perlu diingat bahwa binary search lebih efisien dari sisi waktu dibandingkan dengan sequential search karena binary search hanya memerlukan logaritma basis 2 dari jumlah data yang dicari, sedangkan sequential search memerlukan waktu yang lebih lama karena harus memeriksa setiap elemen satu per satu secara berurutan.

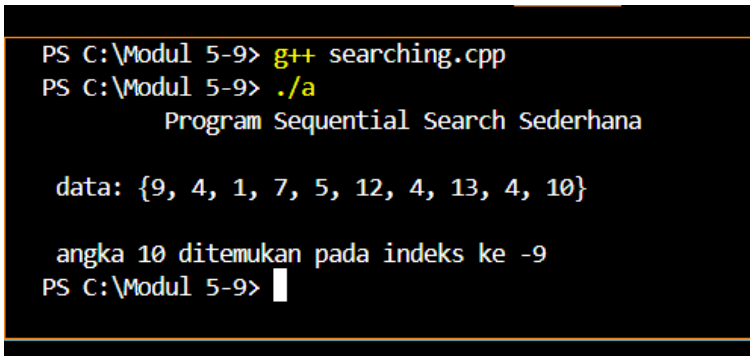
GUIDED

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;
}
```

Screenshot program



Deskripsi Program

Program di atas merupakan contoh program sequential search sederhana di C++ yang digunakan untuk mencari suatu nilai tertentu dalam array. Program menggunakan array dengan ukuran 10 dan nilai-nilai tertentu di dalamnya. Program mencari nilai tertentu (cari) dalam array menggunakan algoritma sequential search. Jika nilai ditemukan, program akan menampilkan indeks dari nilai tersebut dalam array. Jika nilai tidak ditemukan, program akan menampilkan pesan bahwa nilai tidak dapat ditemukan dalam array. Program diakhiri dengan mengembalikan nilai 0.

2. Guided 2

Source code

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

using namespace std;

int data[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 (data[j] < data[min]) {
                min = j;
```

```
            }
        }
        temp = data[i];
        data[i] = data[min];
        data[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 (data[tengah] == cari) {
            b_flag = 1;
            break;
        } else if (data[tengah] < cari) {
            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:";
    // tampilkan data awal
    for (int x = 0; x < 7; x++) {
        cout << setw(3) << data[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) << data[x];
    }
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}
```

Screenshot program

Deskripsi Program

Program di atas merupakan contoh program binary search di C++ yang digunakan untuk mencari suatu nilai tertentu dalam array yang sudah diurutkan terlebih dahulu dengan menggunakan algoritma binary search. Program menggunakan array dengan ukuran 7 dan nilai-nilai tertentu di dalamnya. Program meminta input dari user untuk nilai yang ingin dicari (cari). Program mengurutkan nilai dalam array menggunakan algoritma selection sort. Program mencari nilai tertentu (cari) dalam array menggunakan algoritma binary search. Jika nilai ditemukan, program akan menampilkan indeks dari nilai tersebut dalam array. Jika nilai tidak ditemukan, program akan menampilkan pesan bahwa nilai tidak dapat ditemukan dalam array. Program diakhiri dengan menunggu input dari user sebelum keluar. Algoritma binary search digunakan karena lebih efisien dibandingkan sequential search pada data yang sudah diurutkan.

UNGUIDED

1. Unguided 1

```
//Shiva Indah Kurnia, 2311102035
#include <iostream>
using namespace std;

void selectionSort(string &huruf, int n)
{
    int i, j, min;
    for (i = 0; i < n - 1; i++)
    {
        min = i;
        for (j = i + 1; j < n; j++)
            if (huruf[j] < huruf[min])
                min = j;
        char temp = huruf[i];
        huruf[i] = huruf[min];
        huruf[min] = temp;
    }
}

int binarySearch(string huruf, int kiri, int kanan, char target)
{
    while (kiri <= kanan)
    {
        int mid = kiri + (kanan - kiri) / 2;
        if (huruf[mid] == target)
            return mid;
        if (huruf[mid] < target)
            kiri = mid + 1;
        else
            kanan = mid - 1;
    }
    return -1;
}

int main()
{
    string kalimat;
    char input;
    cout << "=============================================" <<
endl;
```

```
    cout << "======PROGRAM MENENTUKAN INDEKS PADA HURUF=====" <<
endl;
    cout << "===============================================" <<
endl;
    cout << "Masukkan kalimat yang anda inginkan: ";
    getline(cin, kalimat);
    cout << "Masukkan huruf yang anda ingin cari: ";
    cin >> input;
    cout << endl;
    selectionSort(kalimat, kalimat.size());
    int result = binarySearch(kalimat, 0, kalimat.size() - 1,
input);
    if (result == -1)
    {
        cout << "Huruf yang anda cari tidak ditemukan!" << endl;
    }
    else
    {
        cout << "Huruf setelah diurutkan: " << kalimat << endl;
        cout << "Huruf ditemukan pada indeks ke- " << result <<
endl;
    }
    return 0;
}
```

Screenshot program

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

PS C:\Modul 5-9> g++ searchingunguided.cpp
PS C:\Modul 5-9> ./a
===============================================
======PROGRAM MENENTUKAN INDEKS PADA HURUF=====
===============================================
Masukkan kalimat yang anda inginkan: Shiva
Masukkan huruf yang anda ingin cari: v

Huruf setelah diurutkan: Sahiv
Huruf ditemukan pada indeks ke- 4
PS C:\Modul 5-9>
```

Deskripsi program

Program di atas merupakan contoh program binary search di C++ yang digunakan untuk mencari indeks dari suatu huruf tertentu dalam sebuah kalimat. Program meminta input dari user berupa sebuah kalimat dan sebuah huruf yang ingin dicari indeksnya. Program mengurutkan huruf-huruf dalam kalimat menggunakan algoritma selection sort. Program mencari huruf tertentu (target) dalam kalimat menggunakan algoritma binary search. Jika huruf ditemukan, program akan menampilkan indeks dari huruf tersebut dalam kalimat. Jika huruf tidak ditemukan, program akan menampilkan pesan bahwa huruf tidak dapat ditemukan dalam kalimat. Program diakhiri dengan menampilkan hasil pencarian indeks huruf dan menunggu input dari user sebelum keluar.

2. Unguided 2

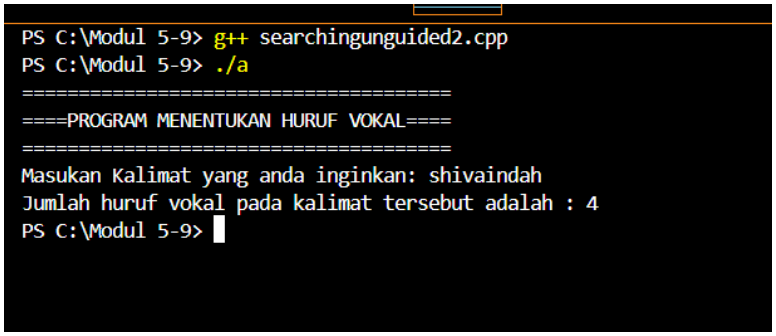
```
//Shiva Indah Kurnia, 2311102035
#include <iostream>
using namespace std;

int main()
{
   string kalimat;
   int count=0;
   cout << "=====================================" << endl;
   cout << "====PROGRAM MENENTUKAN HURUF VOKAL====" << endl;
   cout << "=====================================" << endl;
   cout << "Masukan Kalimat yang anda inginkan: "   ;
   cin  >> kalimat;

   for (int i=0; i<kalimat.length(); i++)
   {
       if(kalimat[i]=='a'|| kalimat[i]=='i'|| kalimat[i]=='u'||
kalimat[i]=='e'|| kalimat[i]=='o')
       {
             count++;
       }
   }
   cout << "Jumlah huruf vokal pada kalimat tersebut adalah : "
<< count;

}
```

Screenshot program



Deskripsi program

Program di atas merupakan contoh program untuk menghitung jumlah huruf vokal pada sebuah kalimat yang dimasukkan oleh user. Program meminta input dari user berupa sebuah kalimat.Program menghitung jumlah huruf vokal dalam kalimat menggunakan loop for dan kondisi if. Jika huruf dalam kalimat adalah huruf vokal (a, i, u, e, o), program akan menambahkan jumlah huruf vokal. Program menampilkan jumlah huruf vokal pada kalimat tersebut.

3. Unguided 3

```cpp
#include <iostream>

using namespace std;

int HitungAngka( const int array[], int size, int target) {
    int count = 0;

    for (int i = 0; i < size; i++) {
        if (array[i] == target) {
            count++;
        }
    }

    return count;
}

int main() {
```

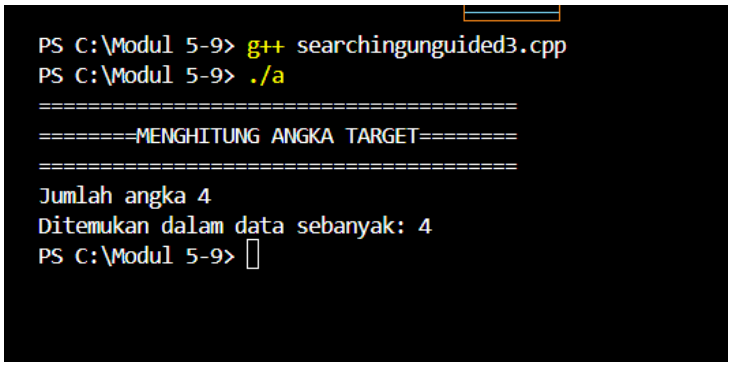
```
    const int size = 10;
    int array[size] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int target = 4;

    int count = HitungAngka(array, size, target);

    cout << "=======================================" << endl;
    cout << "========MENGHITUNG ANGKA TARGET========" << endl;
    cout << "=======================================" << endl;
    cout << "Jumlah angka " << target <<  endl;
    cout << "Ditemukan dalam data sebanyak: " << count << endl;

    return 0;
}
```

Screenshot program



Deskripsi program

Program di atas merupakan contoh program untuk menghitung jumlah kemunculan suatu angka tertentu dalam sebuah array menggunakan sequential search. Program menggunakan array dengan ukuran 10 dan nilai-nilai tertentu di dalamnya. Program meminta input dari user untuk angka yang ingin dicari (target). Program menghitung jumlah kemunculan angka tertentu (target) dalam array menggunakan loop for dan fungsi HitungAngka. Fungsi HitungAngka menerima tiga parameter, yaitu array, ukuran array, dan target. Fungsi ini menghitung jumlah kemunculan target dalam array. Program menampilkan jumlah kemunculan angka target dalam array.

KESIMPULAN

Setelah mempelajari algoritma searching sequential search dan binary search dapat diketahui Sequential search adalah metode pencarian data dengan cara membandingkan setiap elemen dengan elemen yang dicari secara berurutan mulai dari elemen pertama sampai elemen yang dicari ditemukan. Metode ini disarankan untuk digunakan pada data yang sedikit saja. Sedangkan untuk Binary search sendiri adalah metode pencarian data atau elemen di dalam suatu array dengan kondisi data dalam keadaan terurut. Proses pencarian binary search hanya dapat dilakukan pada kumpulan data yang sudah diurutkan terlebih dahulu (menaik atau menurun). Algoritma ini biasanya banyak digunakan untuk mencari di program dengan jumlah data yang banyak, dimana kompleksitas dari algoritma ini adalah O(log n) di mana n adalah jumlah item.

Binary search dapat lebih efisien dari sisi waktu jika dibandingkan dengan sequential search, namun binary search ini memerlukan data yang sudah terurut terlebih dahulu. Adapun algoritma binary search lebih rumit daripada sequential search. Pada saat menggunakan binary search, data yang berada di dalam array harus diurutkan terlebih dahulu menggunakan teknik sorting seperti bubble sort.