LAPORAN PRAKTIKUM

MODUL VIII
ALGORITMA SEARCHING

Disusun oleh:
FATTAH RIZQY ADHIPRATAMA
NIM: 2311102019

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

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

BAB II
DASAR TEORI

Searching dalam pemrograman adalah proses yang sangat fundamental Guna mencari data tertentu dalam sekumpulan data tentunya yang memiliki tipe yang sama. Pencarian diperlukan untuk mencari informasi khusus dari tabel / kumpulan data pada saat lokasi yang pasti dari informasi tersebut sebelumnya tidak diketahui. Data pada tabel biasanya disimpan dengan menggunakan tipe data Array yang dimana Array memungkinkan untuk menyimpan nilai yang bertipe sama.

Adapun Metode yang digunakan dalam Searching dalam Struktur Data sebagai berikut :

1. Metode Pencarian Beruntun (Sequential/Linear Searching)
2. Metode Pencarian Bagi Dua (Binary Searching)

1. Sequential Search

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. Jika elemen ditemukan, ia mengembalikan indeksnya, jika tidak maka -1.

2. Binary Searching

Binary Searcing adalah metode dengan Prinsip dasarnya adalah melakukan proses pembagian ruang pencarian secara berulang-ulang sampai data ditemukan atau sampai ruang pencarian tidak dapat dibagi lagi (berarti ada kemungkinan data tidak ditemukan). Metode ini Memperkecil jumlah operasi pembandingan yang harus dilakukan antara data yang dicari dengan data yang ada di dalam tabel, khususnya untuk jumlah data yang sangat besar ukurannya. Syarat dalam metode ini adalah Data sudah dalam keadaan terurut (naik) ascending.

Cara Kerja Metode Binary Searching dalam Struktur data seperti berikut :

- Kunci akan selalu dibandingkan dengan data yang berada di tengah (middle).
- Bila sama berarti data ketemu, bila tidak, akan “dilihat” apakah data ada di sebelah “kiri” (artinya data lebih kecil dari data di tengah) atau di sebelah “kanan” (artinya data lebih besar dari data di tengah).
- Bila data ada di sebelah kiri, dilakukan pencarian dengan cara yang sama (sementara data yang berada di sebelah kanan akan diabaikan).
- Jadi, setiap kali pencarian, data selalu “dibelah” menjadi dua bagian (biner), sampai pada “titik tertentu”(bila sama dengan titik tengah, pencarian tidak dilakukan lagi, bila tidak, lakukan pencarian lagi sampai pada perbandingan terakhir data juga tidak sama, berarti data tidak ditemukan pada array).

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 D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
Data dan Algoritme\Modul 8\" ; if ($?) { g++ guided1.cpp -o guided1 }
         Program Sequential Search Sederhana

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

 angka 10 ditemukan pada indeks ke - 9
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
```

Deskripsi program

Program tersebut adalah implementasi sederhana dari sebuah algoritma searching dalam bahasa pemrograman C++. Program ini dapat digunakan untuk mencari nilai tertentu dalam data array. Algoritma ini bekerja dengan cara membandingkan nilai yang dicari dengan setiap elemen array secara berurutan hingga ditemukan atau seluruh elemen array telah diperiksa.

2. Guided 2

Source Code

```cpp
#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;

// Fungsi selection sort untuk mengurutkan array
void selectionSort(int arr[], int n) {
    int temp, minIndex;

    for (int i = 0; i < n - 1; i++) {
        minIndex = i;

        for (int j = i + 1; j < n; j++) {
```

```
            if (arr[j] < arr[minIndex]) {
                minIndex = j;
            }
        }

        // Tukar elemen
        temp = arr[i];
        arr[i] = arr[minIndex];
        arr[minIndex] = temp;
    }
}

// Fungsi binary search untuk mencari data dalam array yang telah
diurutkan
void binarySearch(int arr[], int n, int target) {
    int start = 0, end = n - 1, middle, found = 0;

    while (start <= end && found == 0) {
        middle = (start + end) / 2;

        if (arr[middle] == target) {
            found = 1;
        } else if (arr[middle] < target) {
            start = middle + 1;
        } else {
            end = middle - 1;
        }
    }

    if (found == 1) {
        cout << "\nData ditemukan pada indeks ke-" << middle <<
endl;
    } else {
        cout << "\nData tidak ditemukan\n";
    }
}

int main() {
    cout << "\tBINARY SEARCH" << endl;

    cout << "\nData awal: ";
```

```
    // Menampilkan data awal
    for (int i = 0; i < 7; i++) {
        cout << setw(3) << arrayData[i];
    }
    cout << endl;

    cout << "\nMasukkan data yang ingin Anda cari: ";
    cin >> cari;

    // Mengurutkan data dengan selection sort
    selectionSort(arrayData, 7);

    cout << "\nData diurutkan: ";
    // Menampilkan data setelah diurutkan
    for (int i = 0; i < 7; i++) {
        cout << setw(3) << arrayData[i];
    }
    cout << endl;

    // Melakukan binary search
    binarySearch(arrayData, 7, cari);

    return 0;
}
```

Screenshoot Program

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
me\Modul 8\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunner
        BINARY SEARCH

Data awal:   1  8  2  5  4  9  7

Masukkan data yang ingin Anda cari: 9

Data diurutkan:   1  2  4  5  7  8  9

Data ditemukan pada indeks ke-6
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
```

Deskripsi Program

Program tersebut adalah implementasi sederhana dari sebuah algoritma searching dalam bahasa pemrograman C++. Program ini menunjukkan contoh penggunaan

algoritma Binary Search untuk mencari nilai tertentu dalam data array yang telah diurutkan. Algoritma ini umumnya lebih efisien untuk data yang besar dibandingkan dengan algoritma pencarian lain seperti Sequential Search.

LATIHAN KELAS - UNGUIDED

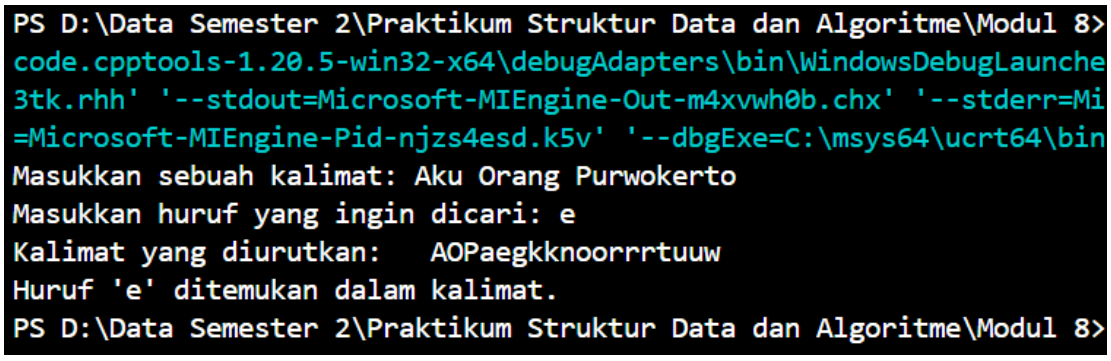
1. **Unguided 1**

   Source code

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

using namespace std;

// Fungsi untuk melakukan binary search pada array karakter
bool binarySearch(const string& arr, char target) {
    int left = 0;
    int right = arr.size() - 1;

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

        // Jika target ditemukan
        if (arr[mid] == target) {
            return true;
        }

        // Jika target lebih besar dari nilai tengah, abaikan
bagian kiri
        if (arr[mid] < target) {
            left = mid + 1;
        }
        // Jika target lebih kecil dari nilai tengah, abaikan
bagian kanan
        else {
            right = mid - 1;
        }
    }
    return false;
}

int main() {
    string kalimat;
```

```
    char target;

    // Input kalimat dari pengguna
    cout << "Masukkan sebuah kalimat: ";
    getline(cin, kalimat);

    // Input huruf yang ingin dicari
    cout << "Masukkan huruf yang ingin dicari: ";
    cin >> target;

    // Ubah kalimat menjadi string terurut
    string sorted_kalimat = kalimat;
    sort(sorted_kalimat.begin(), sorted_kalimat.end());

    // Tampilkan kalimat yang sudah diurutkan (untuk verifikasi)
    cout << "Kalimat yang diurutkan: " << sorted_kalimat << endl;

    // Lakukan pencarian binary search
    if (binarySearch(sorted_kalimat, target)) {
        cout << "Huruf '" << target << "' ditemukan dalam
kalimat." << endl;
    } else {
        cout << "Huruf '" << target << "' tidak ditemukan dalam
kalimat." << endl;
    }

    return 0;
}
```

Screenshoot program



Deskrirpsi Program

Program ini adalah implementasi dari dari algoritma pencarian binary search untuk mencari keberadaan sebuah karakter dalam sebuah kalimat yang dimasukkan oleh pengguna. Program ini sangat berguna ketika pengguna ingin mencari apakah sebuah huruf tertentu ada dalam sebuah kalimat dan ingin mengetahui posisinya dalam kalimat tersebut.

2. Unguided 2

Source Code

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

using namespace std;

// Fungsi untuk menghitung jumlah huruf vokal dalam sebuah
kalimat
int hitungVokal(const string& kalimat) {
    int count = 0;
    for (char c : kalimat) {
        // Periksa apakah karakter saat ini adalah huruf vokal
        if (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c
== 'u' ||
            c == 'A' || c == 'E' || c == 'I' || c == 'O' || c
== 'U') {
            count++;
        }
    }
    return count;
}

int main() {
    string kalimat;

    // Input kalimat dari pengguna
    cout << "Masukkan sebuah kalimat: ";
    getline(cin, kalimat);

    // Hitung jumlah huruf vokal
    int vocalCount = hitungVokal(kalimat);
```

```
    // Tampilkan hasilnya
    cout << "Jumlah huruf vokal dalam kalimat: " << vocalCount
<< endl;

    return 0;
}
```

Screenshoot Program

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher
fpv.kut' '--stdout=Microsoft-MIEngine-Out-cb0c0gzo.5tk' '--stderr=Mic
=Microsoft-MIEngine-Pid-j3afm5bk.uiv' '--dbgExe=C:\msys64\ucrt64\bin\
Masukkan sebuah kalimat: Aku orang purwokerto kecamatan patikraja
Jumlah huruf vokal dalam kalimat: 16
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
```

Deskripsi Program

Program ini merupakan implementasi dari sebuah program sederhana yang menghitung jumlah huruf vokal dalam sebuah kalimat yang dimasukkan oleh pengguna. Program ini meminta pengguna untuk memasukkan sebuah kalimat, Kemudian menggunakan fungsi hitungVokal untuk menghitung jumlah huruf vokal dalam kalimat tersebut.

3. **Unguided 3**

Source Code

```
#include <iostream>

using namespace std;

// Fungsi untuk menghitung jumlah angka 4 menggunakan
Sequential Search
int hitungAngka(const int arr[], int size, int target) {
    int count = 0;
    for (int i = 0; i < size; ++i) {
        if (arr[i] == target) {
            count++;
        }
    }
    return count;
}
```

```cpp
int main() {
    // Data array yang diberikan
    int data[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int size = sizeof(data) / sizeof(data[0]);
    int target = 4;

    // Hitung jumlah angka 4
    int count = hitungAngka(data, size, target);

    // Tampilkan hasilnya
    cout << "Jumlah angka " << target << " terdapat sebanyak :
" << count << endl;

    return 0;
}
```

Screenshoot Program

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher
de0.4lv' '--stdout=Microsoft-MIEngine-Out-ozvrulee.lao' '--stderr=Mic
=Microsoft-MIEngine-Pid-pv3mqzab.h41' '--dbgExe=C:\msys64\ucrt64\bin\
Jumlah angka 4 terdapat sebanyak : 4
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 8>
```

Deskripsi Program

Program tersebut adalah implementasi dari algoritma Sequential Search untuk menghitung jumlah angka 4 yang ada dalam sebuah array. Program dimulai dengan mendefinisikan data array yang berisi 10 elemen. Fungsi hitungAngka kemudian digunakan untuk mengiterasi setiap elemen dalam array dan menghitung berapa kali angka 4 muncul.

BAB IV

KESIMPULAN

Algoritma searching adalah teknik untuk mencari suatu nilai tertentu dalam sebuah kumpulan data atau struktur data. Terdapat berbagai macam algoritma searching, seperti Sequential Search, Binary Search, dan lain-lain. Sequential Search adalah algoritma searching sederhana yang melakukan pencarian elemen satu per satu secara berurutan dari awal sampai akhir kumpulan data.

Binary Search adalah algoritma searching yang lebih efisien daripada Sequential Search, terutama untuk kumpulan data yang besar dan terurut. Binary Search membagi kumpulan data menjadi dua bagian setiap kali melakukan pencarian, sehingga memperkecil jumlah data yang perlu diproses pada setiap iterasi.

DAFTAR PUSTAKA

[1] Daisma, 2019. Metode Searching dalam Struktur Data dan Implementasi Pemrogramannya. Diakses pada tanggal 31 Mei 2024, dari https://daismabali.medium.com/metode-searching-dalam-struktur-data-dan-implementasi-pemrogramannya-d97582084866

[2] Trivusi, 2022. Algoritma Pencarian: Pengertian, Karakteristik, dan Jenis-Jenisnya. Diakses pada tanggal 31 Mei 2024, dari https://www.trivusi.web.id/2022/11/pengertian-algoritma-pencarian.html