LAPORAN PRAKTIKUM
MODUL 8
ALGORITMA SEARCHING

Disusun oleh :

Diva Octaviani

NIM : 2311102006

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

1. Pengertian Searching

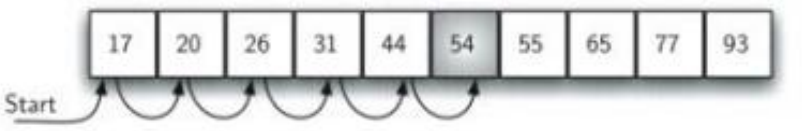
Searching adalah tindakan mengambil data dari kumpulan data berdasarkan kunci (key) atau referensi data. Di kehidupan sehari-hari, kita sering berurusan dengan pencarian. Misalnya, mencari nomor telepon seseorang di buku telepon, atau mencari kata pada kamus, dan masih banyak lagi. Pencarian sering dilakukan dalam aplikasi komputer. Misalnya, jika ingin menghapus atau mengubah data/record tertentu dalam suatu file, harus terlebih dahulu menentukan apakah data tersebut ada dalam file. Jika ada, maka data dapat dihapus atau diubah.

2. Jenis-jenis Searching

a) Sequential Search

Sequential Search adalah proses membandingkan setiap elemen array satu persatu secara beruntun dimulai dari elemen pertama hingga elemen yang dicari ditemukan atau hingga elemen terakhir dari array. Metode Sequential Search atau disebut pencarian beruntun dapat digunakan untuk melakukan pencarian data baik pada array yang sudah terurut maupun yang belum terurut.

Proses pencarian data dengan metode ini cukup sederhana dan mudah. Proses pencarian data dilakukan dengan mencocokkan data yang dilakukan secara berurut satu demi satu dimulai dari data ke-1 hingga data pada urutan terakhir. Jika data yang dicari mempunyai nilai yang sama dengan data yang ada dalam kelompok data, berarti data telah ditemukan. Jika data yang dicari tidak ada yang cocok dengan data dalam sekelompok data, data tersebut tidak ada dalam sekelompok data. Selanjutnya kita tinggal menampilkan hasil yang diperoleh tersebut.



Gambar 1. Sequential Search

b) Binary Search

Binari Search merupakan teknik pencarian data dalam dengan cara membagi data menjadi dua bagian setiap kali terjadi proses pencarian. Dalam Binary Search, data yang ada harus diurutkan terlebih dahulu berdasarkan suatu urutan tertentu yang dijadikan kunci pencarian.

Data diambil dari posisi 1 sampai posisi akhir N Kemudian cari posisi data tengah dengan rumus: (posisi awal + posisi akhir) / 2. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar? Jika lebih besar, maka proses pencarian dicari dengan posisi awal adalah posisi tengah + 1 Jika lebih kecil, maka proses pencarian dicari dengan posisi akhir adalah posisi tengah - 1 Jika data sama, berarti ketemu.

Gambar 2. Binary Search

BAB III

GUIDED

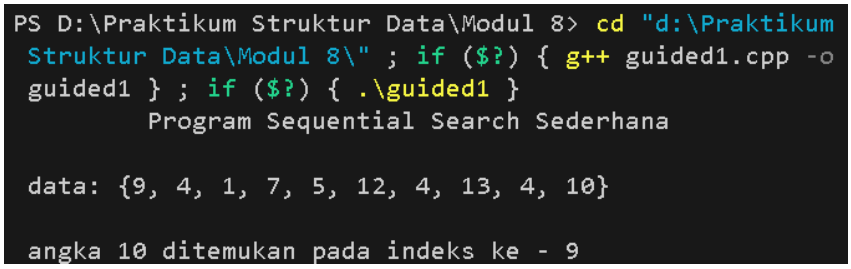
1. Guided 1

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



Deskripsi program

Pada program tersebut, terdapat deklarasi variabel n dengan nilai 10, dan array data dengan elemen-elemen {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}. Program tersebut menggunakan algoritma Sequential Search untuk mencari nilai cari dalam array data. Jika nilai cari ditemukan, program akan mencetak pesan bahwa nilai tersebut ditemukan pada indeks ke - i. Jika nilai cari tidak ditemukan, program akan mencetak pesan bahwa nilai tersebut tidak dapat ditemukan pada data.

2. Guided 2

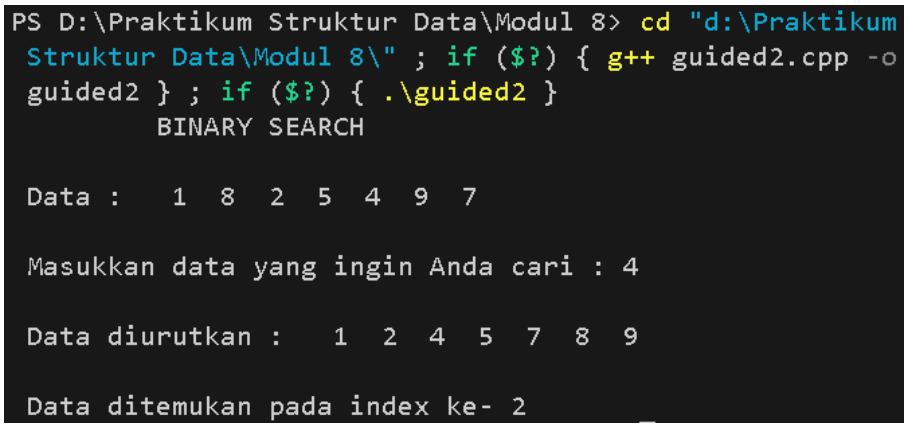
Source code

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>

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 = 7;
```

```
    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 << "\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 data awal
    for (int x = 0; x < 7; x++)
    cout << setw(3) << data[x];
    cout << endl;
    cout << "\n Masukkan data yang ingin Anda cari : ";
    cin >> cari;
    cout << "\n Data 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;
}
```

Screenshoot program



Deskripsi program

Pada program tersebut, terdapat deklarasi array data dengan elemen-elemen {1, 8, 2, 5, 4, 9, 7} dan variabel cari yang akan digunakan untuk mencari nilai dalam array tersebut. Program menggunakan algoritma Selection Sort untuk mengurutkan array data secara ascending. Setelah itu, program menggunakan algoritma Binary Search untuk mencari nilai cari dalam array yang sudah diurutkan. Program mencetak data awal sebelum diurutkan, meminta input dari pengguna untuk nilai yang ingin dicari, mencetak data setelah diurutkan, dan mencetak hasil pencarian menggunakan Binary Search.

LATIHAN KELAS – UNGUIDED

1. Unguided 1

Buatlah sebuah program untuk mencari sebuah huruf pada sebuah kalimat yang sudah di input dengan menggunakan Binary Search!

Source code

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

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

    return -1;
```

```
}

int main()
{
    string sentence;
    char target;

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

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

    int index = binarySearch(sentence, target);

    if (index != -1)
    {
        cout << "Huruf " << target << " ditemukan pada indeks
ke-" << index << endl;
    }
    else
    {
        cout << "Huruf " << target << " tidak ditemukan dalam
kalimat" << endl;
    }

    return 0;
}
```

Screenshoot program

```
PS D:\Praktikum Struktur Data\Modul 8> cd "d:\Praktikum Struktur
 Data\Modul 8\" ; if ($?) { g++ ungided1.cpp -o ungided1 } ; if
($?) { .\ungided1 }
Masukkan kalimat : diva
Masukkan huruf yang ingin dicari : d
Huruf d ditemukan pada indeks ke-0
```

Deskripsi program

Program di atas menggunakan algoritma Binary Search untuk mencari huruf tertentu dalam sebuah kalimat. Pengguna diminta untuk memasukkan kalimat dan huruf yang ingin dicari. Program akan mencari huruf tersebut dalam kalimat menggunakan Binary Search dan mengembalikan indeks huruf tersebut jika ditemukan, atau memberikan pesan bahwa huruf tersebut tidak ditemukan dalam kalimat.

2. Unguided 2

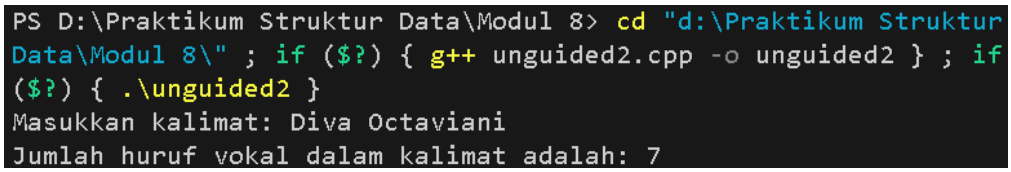
Buatlah sebuah program yang dapat menghitung banyaknya huruf vocal dalam sebuah kalimat!

Source code

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

int countVowels(string sentence)
{
    int count = 0;
    string vowels = "aiueoAIUEO";

    for (char c : sentence)
    {
        if (vowels.find(c) != string::npos)
        {
            count++;
        }
    }

    return count;
}

int main()
{
    string sentence;

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

    int vowelCount = countVowels(sentence);
```

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

    return 0;
}
```

Screenshoot program



Deskripsi program

Program di atas akan meminta pengguna untuk memasukkan sebuah kalimat. Kemudian, program akan menghitung jumlah huruf vokal dalam kalimat tersebut menggunakan algoritma searching. Program akan mencari setiap karakter dalam kalimat apakah termasuk dalam huruf vokal (a, i, u, e, o, A, I, U, E, O). Jika ditemukan huruf vokal, program akan mengincrement variabel count. Pada akhirnya, program akan menampilkan jumlah huruf vokal dalam kalimat.

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>
using namespace std;

int sequentialSearch(int arr[], int n, int key)
{
    int count = 0;
    for (int i = 0; i < n; i++)
    {
        if (arr[i] == key)
        {
            count++;
        }
    }
    return count;
}

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

    int count = sequentialSearch(data, n, key);

    cout << "Banyaknya angka 4 dalam data adalah : " << count
<< endl;
    return 0;
}
```

Screenshoot program

```
PS D:\Praktikum Struktur Data\Modul 8> cd "d:\Praktikum Struktur
Data\Modul 8\" ; if ($?) { g++ unguided3.cpp -o unguided3 } ; if
($?) { .\unguided3 }
Banyaknya angka 4 dalam data adalah : 4
```

Deskripsi program

Program ini mencari berapa kali suatu nilai kunci (key) muncul dalam sebuah array menggunakan algoritma sequential search. Pertama, program mendefinisikan sebuah fungsi `sequentialSearch` yang menerima array, ukuran array, dan nilai kunci sebagai parameter. Di dalam fungsi, program melakukan iterasi melalui setiap elemen array dan menghitung jumlah kemunculan nilai kunci. Kemudian, di dalam fungsi `main`, program mendeklarasikan sebuah array `data` dan menetapkan nilai kunci yang ingin dicari, yaitu 4. Fungsi `sequentialSearch` dipanggil dengan argumen array, ukuran array, dan nilai kunci. Hasilnya, yang merupakan jumlah kemunculan nilai kunci dalam array, dicetak ke layar.

BAB IV

KESIMPULAN

Pencarian (searching) adalah proses untuk mencari suatu data dalam kumpulan data berdasarkan suatu kunci atau acuan data. Terdapat dua jenis pencarian yang umum digunakan, yaitu sequential search dan binary search.

Sequential search adalah metode pencarian yang dilakukan secara berurutan dari awal hingga akhir data. Pada setiap langkah, data dibandingkan dengan elemen yang sedang diperiksa. Jika data ditemukan, pencarian berhenti dan indeks data tersebut dikembalikan. Sequential search cocok digunakan pada data yang tidak terurut atau jumlah data yang kecil.

Sementara itu, binary search adalah metode pencarian yang efisien pada data yang terurut. Metode ini bekerja dengan membagi data menjadi dua bagian dan membandingkan nilai target dengan nilai tengah data. Jika nilai target lebih kecil, pencarian dilanjutkan pada bagian kiri data, dan sebaliknya. Binary search memiliki kompleksitas waktu yang lebih rendah dibandingkan dengan sequential search, namun syarat utama adalah data harus terurut terlebih dahulu.

DAFTAR PUSTAKA

Asisten Praktikum, "*Modul 8 Algoritma Searching*", Learning Management System, 2024.

Putra, M. T. D., Munawir, Ana, R. Y. (2023). Belajar *Pemrograman Lanjut dengan C++*. Bandung : Widina Media Utama.

Firliana, Rina dan Patmi Kasih. (2018). *Algoritma & Pemrograman C++*. Bandung : Widina Media Utama. Nganjuk : Adjie Media Nusantara.

Sonita Anisya dan Mayang Sari. *Implementasi Algoritma Sequential Searching untuk Pencarian Nomor Surat Pada Sistem Arsip Elektronik*. Jurnal Pseudocode, 5(1), 1-9.