

LAPORAN PRAKTIKUM

Modul VIII ALGORITMA SEARCHING



Disusun oleh:

Muhammad Irsyad : **2211102048**

Dosen

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

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

BAB I

Tujuan Pembelajaran

1. Mencari elemen dalam kumpulan data
2. Menemukan posisi elemen
3. Meningkatkan efisiensi pencarian
4. Memahami kompleksitas waktu dan ruang

BAB II

Dasar Teori

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:

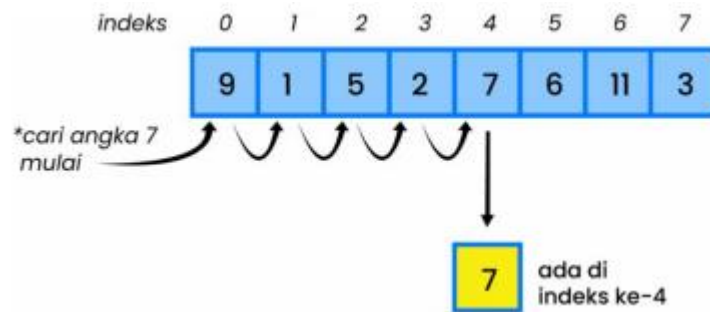
- Membandingkan setiap elemen pada array satu per satu secara berurut
- Proses pencarian dimulai dari indeks pertama hingga indeks terakhir
- Proses pencarian akan berhenti apabila data ditemukan. Jika hingga akhir array data masih juga tidak ditemukan, maka proses pencarian tetap akan dihentikan

- Proses perulangan pada pencarian akan terjadi sebanyak jumlah N elemen pada array

Contoh dari Sequential Search, yaitu:

Array

Int A[8] = {9,1,5,2,7,6,11,3}



Gambar 1. Ilustrasi Sequential Search

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.

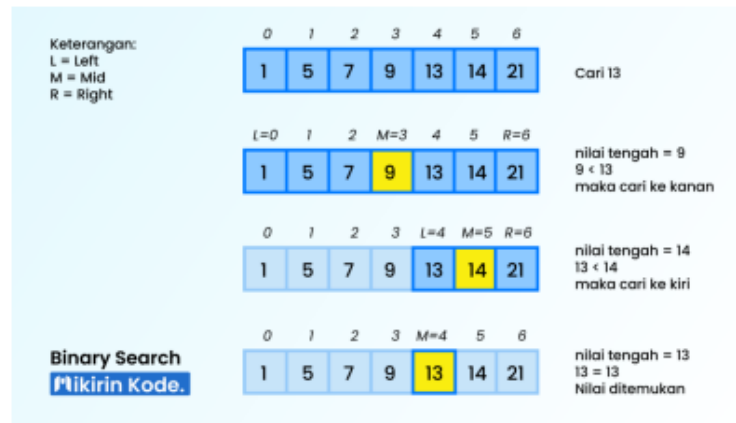
b. Binary Search

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.
- 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 posisi akhir untuk pembagian selanjutnya.
- Apabila data belum ditemukan, maka pencarian akan dilanjutkan dengan kembali membagi data menjadi dua
- Namun apabila data bernilai sama, maka data yang dicari langsung ditemukan dan pencarian dihentikan

Contoh dari Binary Search, yaitu:



Gambar 2. Ilustrasi Binary Search

- Terdapat sebuah array yang menampung 7 elemen seperti ilustrasi di atas. Nilai yang akan dicari pada array tersebut adalah 13
- Jadi karena konsep dari binary search ini adalah membagi array menjadi dua bagian, maka pertama tama kita cari nilai tengahnya dulu, total elemen dibagi 2 yaitu $7/2 = 4.5$ dan kita bulatkan jadi 4.
- Maka elemen ke empat pada array adalah nilai tengahnya, yaitu angka 9 pada indeks ke 3
- Kemudian kita cek apakah $13 > 9$ atau $13 < 9$?
- 13 lebih besar dari 9, maka kemungkinan besar angka 13 berada setelah 9 atau di sebelah kanan. Selanjutnya kita cari ke kanan dan kita dapat mengabaikan elemen yang ada di kiri
- Setelah itu kita cari lagi nilai tengahnya, didapatlah angka 14 sebagai nilai tengah. Lalu, kita bandingkan apakah $13 > 14$ atau $13 < 14$?
- Ternyata 13 lebih kecil dari 14, maka selanjutnya kita cari ke kiri
- Karna tersisa 1 elemen saja, maka elemen tersebut adalah nilai tengahnya. Setelah dicek ternyata elemen pada indeks ke-4 adalah elemen yang dicari, maka telah selesai proses pencariannya.

BAB III

LATIHAN KELAS – GUIDED

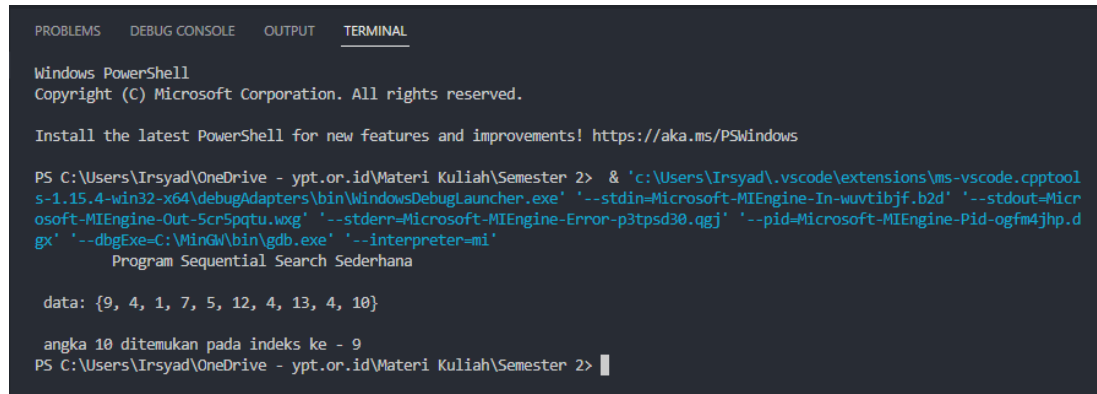
Guided 1

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



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptool
s-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-wuvtibjf.b2d' '--stdout=Micro
soft-MIEngine-Out-5cr5pqtu.wgx' '--stderr=Microsoft-MIEngine-Error-p3tpsd30.qgj' '--pid=Microsoft-MIEngine-Pid-ogfm4jhp.d
gx' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
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\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> |
```

Deskripsi Program

- 1) Baris 1: Mendefinisikan header file iostream yang digunakan untuk input/output stream.
- 2) Baris 2: Menggunakan namespace std untuk menggunakan objek dan fungsi standar dari C++.
- 3) Baris 3: Mendefinisikan fungsi main sebagai titik masuk utama program.
- 4) Baris 5: Mendeklarasikan variabel n dengan nilai 10. Variabel n digunakan sebagai ukuran array data.
- 5) Baris 6: Mendeklarasikan array data dengan ukuran n dan menginisialisasi elemen-elemennya dengan nilai {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}.
- 6) Baris 7: Mendeklarasikan variabel cari dengan nilai 10. Variabel cari merupakan nilai yang ingin dicari dalam array data.
- 7) Baris 8: Mendeklarasikan variabel ketemu dengan nilai awal false. Variabel ini akan digunakan untuk menandai apakah nilai cari ditemukan dalam array.
- 8) Baris 9: Mendeklarasikan variabel i yang akan digunakan sebagai indeks saat melakukan pencarian.

- 9) Baris 12-17: Melakukan iterasi sebanyak n kali menggunakan loop for. Pada setiap iterasi, dilakukan pengecekan apakah nilai pada indeks ke-i dari array data sama dengan nilai cari. Jika iya, maka ketemu diset menjadi true dan loop dihentikan dengan break.
- 10) Baris 19-22: Menampilkan pesan judul program ke layar.
- 11) Baris 23: Menampilkan isi array data ke layar.
- 12) Baris 24-28: Mengecek apakah nilai cari ditemukan dalam array. Jika ketemu bernilai true, maka menampilkan pesan bahwa nilai cari ditemukan pada indeks ke-i. Jika tidak ditemukan, menampilkan pesan bahwa nilai cari tidak dapat ditemukan dalam data.
- 13) Baris 30: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

Guided 2

Source Code

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>
int angka[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 (angka[j] < angka[min])
            {
                min = j;
            }
        }
    }
}
```



```

        }

        }

        temp = angka[i];
        angka[i] = angka[min];
        angka[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 (angka[tengah] == cari)
        {
            b_flag = 1;
            break;
        }
        else if (angka[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) << angka[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) << angka[x];
    cout << endl;

    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```

Screenshot Program

```

PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptool
s-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ou3e0zcd.sl2' '--stdout=Micr
osoft-MIEngine-Out-gih1tn4w.0xm' '--stderr=Microsoft-MIEngine-Error-ydidjyto.umt' '--pid=Microsoft-MIEngine-Pid-hmaj0mk4.v
v1' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
    BINARY SEARCH

    Data :   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 index ke-6
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2>

```

Deskripsi Program

Kode program di atas adalah contoh implementasi dari algoritma Binary Search dalam bahasa C++.

1. Baris 1-4: Mendefinisikan beberapa header file yang digunakan dalam program ini, yaitu `iostream`, `conio.h`, dan `iomanip`. `iostream` digunakan untuk input dan output dalam program, `conio.h` digunakan untuk mendapatkan input karakter dari pengguna, dan `iomanip` digunakan untuk mengatur tata letak output.
2. Baris 6: Mendefinisikan sebuah array angka dengan 7 elemen dan menginisialisasinya dengan nilai tertentu.
3. Baris 7: Mendefinisikan variabel `cari` yang akan digunakan untuk mencari elemen dalam array.
4. Baris 9-23: Fungsi `selection_sort()` digunakan untuk mengurutkan elemen-elemen dalam array angka dengan menggunakan algoritma Selection Sort. Algoritma ini secara berulang memilih elemen terkecil dan menukar posisinya dengan elemen pertama, kemudian memilih elemen terkecil dari sisa array dan menukar posisinya dengan elemen kedua, dan seterusnya. Akibatnya, array akan diurutkan secara ascending (terurut naik).
5. Baris 25-46: Fungsi `binarysearch()` digunakan untuk mencari elemen dengan menggunakan algoritma Binary Search. Algoritma ini bekerja dengan membagi array menjadi dua bagian dan membandingkan elemen tengah dengan elemen yang dicari. Jika elemen tengah sama dengan elemen yang dicari, pencarian selesai. Jika elemen tengah lebih kecil, maka pencarian dilakukan pada setengah kanan array. Jika elemen tengah lebih besar, maka pencarian dilakukan pada setengah kiri array. Proses ini diulang hingga elemen ditemukan atau tidak ada elemen yang tersisa.
6. Baris 49-71: Fungsi `main()` adalah fungsi utama dalam program. Di sini, terdapat langkah-langkah berikut:
 - Menampilkan judul program.
 - Menampilkan data awal dalam array angka.
 - Menerima input dari pengguna untuk nilai yang ingin dicari (variabel `cari`).
 - Menampilkan data dalam array angka setelah diurutkan menggunakan selection sort.

- Memanggil fungsi `binarysearch()` untuk mencari elemen yang dicari dalam array.
- Menggunakan `_getche()` untuk menahan tampilan output agar pengguna dapat melihat hasilnya.
- Mengembalikan nilai `EXIT_SUCCESS` untuk menandakan bahwa program berjalan dengan sukses.

Program ini mencari elemen yang dicari dalam array menggunakan algoritma Binary Search setelah array diurutkan menggunakan Selection Sort. Hasil pencarian akan ditampilkan apakah elemen ditemukan atau tidak.

TUGAS – UNGUIDED

Unguided 1

Source Code

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

using namespace std;

int binarySearch(const 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;
        }
    }
}
```

```

        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);

    transform(sentence.begin(), sentence.end(), sentence.begin(),
::tolower);

    cout << "Masukkan huruf yang ingin dicari: ";
    cin >> target;
    target = tolower(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;
}

```

Screenshot Program

```

PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptool
s-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-qddzpgheh.fjb' '--stdout=Micr
osoft-MIEngine-Out-350wokfb.00x' '--stderr=Microsoft-MIEngine-Error-3n5wian3.0bu' '--pid=Microsoft-MIEngine-Pid-5d0f1wzz.b
du' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat: Irsyad
Masukkan huruf yang ingin dicari: i
Huruf 'i' ditemukan pada indeks ke-0
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2>

```

Deskripsi Program

Kode program di atas adalah implementasi dari algoritma binary search dalam bahasa pemrograman C++. Algoritma binary search digunakan untuk mencari suatu elemen dalam array yang sudah terurut secara menaik.

7. Baris 1-4: Mendefinisikan header file yang akan digunakan, yaitu iostream, string, dan algorithm.
8. Baris 6: Menggunakan namespace std untuk menggunakan objek dan fungsi standar dari C++.
9. Baris 8-18: Mendefinisikan fungsi binarySearch yang menerima parameter string sentence (kalimat) dan char target (huruf yang dicari). Fungsi ini akan mengembalikan indeks dari huruf target jika ditemukan dalam kalimat, atau mengembalikan -1 jika tidak ditemukan. Pada algoritma binary search, variabel left dan right akan membatasi rentang pencarian, sedangkan variabel mid digunakan untuk mencari tengah-tengah rentang tersebut. Pada setiap iterasi, dilakukan pengecekan apakah huruf pada indeks mid dari kalimat sama dengan

huruf target. Jika iya, maka indeks mid dikembalikan. Jika huruf target lebih besar, maka rentang pencarian dipindahkan ke sebelah kanan tengah. Jika huruf target lebih kecil, maka rentang pencarian dipindahkan ke sebelah kiri tengah. Jika rentang pencarian tidak dapat dibagi lagi ($left > right$), maka huruf target tidak ditemukan dan -1 dikembalikan.

10. Baris 20-38: Mendefinisikan fungsi main sebagai titik masuk utama program. Pada fungsi ini, dilakukan langkah-langkah sebagai berikut:

- a. Membaca kalimat dari pengguna dan menyimpannya dalam variabel `sentence` menggunakan `getline(cin, sentence)`. `getline` digunakan untuk membaca satu baris teks yang mungkin mengandung spasi.
- b. Mengubah semua huruf dalam kalimat menjadi huruf kecil menggunakan `transform(sentence.begin(), sentence.end(), sentence.begin(), ::tolower)`. Ini dilakukan untuk memastikan pencarian tidak bersifat case-sensitive.
- c. Meminta pengguna memasukkan huruf yang ingin dicari dan menyimpannya dalam variabel `target`.
- d. Mengubah huruf target menjadi huruf kecil menggunakan `target = tolower(target)`. Ini dilakukan untuk memastikan pencarian tidak bersifat case-sensitive.
- e. Memanggil fungsi `binarySearch` dengan parameter kalimat dan huruf target, dan menyimpan hasilnya dalam variabel `index`.
- f. Mengecek nilai `index`. Jika `index` tidak sama dengan -1, maka huruf target ditemukan dalam kalimat dan menampilkan pesan yang menyatakan huruf tersebut ditemukan pada indeks ke-`index`. Jika `index` sama dengan -1, maka huruf target tidak ditemukan dalam kalimat dan menampilkan pesan yang menyatakan huruf tersebut tidak ditemukan.

11. Baris 40: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

Unguided 2

Source Code

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

using namespace std;

int hitungVokal(const string &sentence)
{
    int count = 0;

    for (char ch : sentence)
    {
        ch = tolower(ch);

        if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch
== 'u')
        {
            count++;
        }
    }
    return count;
}

int main()
{
    string sentence;

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

    int jumlahVokal = hitungVokal(sentence);
```



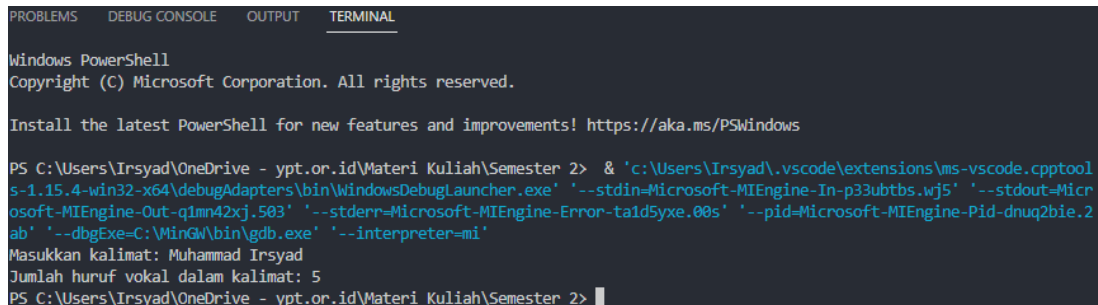
```

        cout << "Jumlah huruf vokal dalam kalimat: " << jumlahVokal <<
endl;

        return 0;
}

```

Screenshot Program



```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptools-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-p33ubtbs.wj5' '--stdout=Microsoft-MIEngine-Out-q1mn42xj.503' '--stderr=Microsoft-MIEngine-Error-ta1d5yxe.00s' '--pid=Microsoft-MIEngine-Pid-dnuq2bie.2ab' '-dbgExe=C:\WinG\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat: Muhammad Irsyad
Jumlah huruf vokal dalam kalimat: 5
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2>

```

Deskripsi Program

Kode program di atas bertujuan untuk menghitung jumlah huruf vokal dalam sebuah kalimat yang diinputkan oleh pengguna.

1. Baris 1-4: Mendefinisikan header file yang akan digunakan, yaitu `iostream`, `string`, dan `algorithm`.
2. Baris 6: Menggunakan namespace `std` untuk menggunakan objek dan fungsi standar dari C++.
3. Baris 8-14: Mendefinisikan fungsi `hitungVokal` yang menerima parameter string `sentence` (kalimat). Fungsi ini akan mengembalikan jumlah huruf vokal dalam kalimat. Pada awalnya, variabel `count` diinisialisasi dengan nilai 0. Kemudian, dilakukan iterasi menggunakan loop `for-each` untuk setiap karakter (`ch`) dalam kalimat. Setiap karakter `ch` diubah menjadi huruf kecil menggunakan `tolower(ch)`. Kemudian, dilakukan pengecekan apakah karakter tersebut adalah huruf vokal (`a`, `e`, `i`, `o`, `u`). Jika iya, maka variabel `count` ditambah 1. Setelah selesai iterasi, fungsi mengembalikan nilai `count`.
4. Baris 16-24: Mendefinisikan fungsi `main` sebagai titik masuk utama program. Pada fungsi ini, dilakukan langkah-langkah sebagai berikut:

- a. Membaca kalimat dari pengguna dan menyimpannya dalam variabel `sentence` menggunakan `getline(cin, sentence)`. `getline` digunakan untuk membaca satu baris teks yang mungkin mengandung spasi.
 - b. Memanggil fungsi `hitungVokal` dengan parameter kalimat dan menyimpan hasilnya dalam variabel `jumlahVokal`.
 - c. Menampilkan pesan yang menyatakan jumlah huruf vokal dalam kalimat.
5. Baris 26: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

Unguided 3

Source Code

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

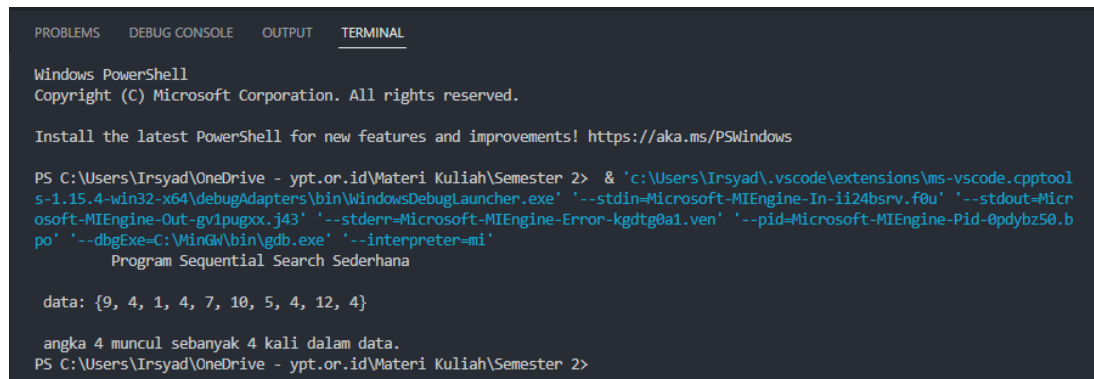
int main()
{
    int n = 10;
    int data[n] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int cari = 4;
    int count = 0;

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

    cout << "\t Program Sequential Search Sederhana\n " << endl;
    cout << " data: {9, 4, 1, 4, 7, 10, 5, 4, 12, 4} " << endl;
    cout << "\n angka " << cari << " muncul sebanyak " << count <<
    " kali dalam data." << endl;
```

```
    return 0;
}
```

Screenshot Program



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptool
s-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ii24bsrv.f0u' '--stdout=Micr
osoft-MIEngine-Out-gv1pugxx.j43' '--stderr=Microsoft-MIEngine-Error-kdgtg0a1.ven' '--pid=Microsoft-MIEngine-Pid-0pdybz50.b
po' '--dbgExe=C:\WinGW\bin\gdb.exe' '--interpreter=mi'
Program Sequential Search Sederhana

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

angka 4 muncul sebanyak 4 kali dalam data.
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2>
```

Deskripsi Program

Kode program di atas adalah sebuah implementasi dari algoritma Sequential Search dalam bahasa pemrograman C++. Tujuannya adalah untuk mencari berapa kali suatu angka muncul dalam sebuah array.

1. Baris 1: Mendefinisikan header file iostream yang digunakan untuk input/output stream.
2. Baris 2: Menggunakan namespace std untuk menggunakan objek dan fungsi standar dari C++.
3. Baris 4: Mendefinisikan fungsi main sebagai titik masuk utama program.
4. Baris 6: Mendeklarasikan variabel n dengan nilai 10. Variabel n digunakan sebagai ukuran array data.
5. Baris 7: Mendeklarasikan array data dengan ukuran n dan menginisialisasi elemen-elemennya dengan nilai {9, 4, 1, 4, 7, 10, 5, 4, 12, 4}.
6. Baris 8: Mendeklarasikan variabel cari dengan nilai 4. Variabel cari merupakan angka yang ingin dicari dalam array.
7. Baris 9: Mendeklarasikan variabel count dengan nilai awal 0. Variabel ini akan digunakan untuk menghitung berapa kali angka cari muncul dalam array.

8. Baris 11-16: Melakukan iterasi sebanyak n kali menggunakan loop for. Pada setiap iterasi, dilakukan pengecekan apakah nilai pada indeks ke- i dari array data sama dengan nilai cari. Jika iya, maka variabel count ditambah 1.
9. Baris 18-22: Menampilkan pesan judul program ke layar.
10. Baris 23: Menampilkan isi array data ke layar.
11. Baris 24-26: Menampilkan pesan yang menyatakan berapa kali angka cari muncul dalam array data.
12. Baris 28: Mengembalikan nilai 0, menandakan bahwa program berakhir dengan sukses.

BAB IV

KESIMPULAN

1. Kode Program Pencarian Sequential Search:
 - Kode program ini mengimplementasikan algoritma pencarian sekuensial untuk mencari suatu nilai dalam array.
 - Iterasi dilakukan secara linier dari awal array hingga nilai yang dicari ditemukan atau mencapai akhir array.
 - Jika nilai yang dicari ditemukan, program mengembalikan indeks tempat nilai tersebut ditemukan. Jika tidak ditemukan, program memberikan pesan bahwa nilai tidak dapat ditemukan.
2. Kode Program Pencarian Binary Search:
 - Kode program ini mengimplementasikan algoritma pencarian biner untuk mencari suatu elemen dalam array terurut.
 - Algoritma biner membagi rentang pencarian menjadi dua bagian setiap iterasinya, membandingkan elemen tengah dengan elemen target, dan memperbarui rentang pencarian sesuai hasil perbandingan.
 - Jika elemen target ditemukan, program mengembalikan indeksnya. Jika tidak ditemukan, program mengembalikan -1.

3. Kode Program Penghitung Jumlah Huruf Vokal:

- Kode program ini menghitung jumlah huruf vokal dalam sebuah kalimat yang diinputkan.
- Menggunakan iterasi untuk setiap karakter dalam kalimat, dan menghitung jumlah huruf vokal dengan membandingkan setiap karakter dengan huruf vokal yang telah ditentukan.
- Hasil perhitungan jumlah huruf vokal ditampilkan pada layar.

Secara keseluruhan, ketiga kode program ini menunjukkan berbagai teknik pencarian dan pengolahan data. Sequential Search digunakan untuk mencari nilai dalam array, Binary Search digunakan untuk mencari elemen dalam array terurut, dan penghitungan jumlah huruf vokal digunakan untuk mengolah data dalam string.