

**LAPORAN PRAKTIKUM  
STRUKTUR DATA DAN ALGORITMA**

**MODUL VIII  
ALGORITMA SEARCHING**



**DISUSUN OLEH :**  
MUHAMAD NASRULLOH  
2311102044

**DOSEN :**  
WAHYU ANDI SAPUTRA, S.PD., M.ENG.

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

**2024**

## **A. DASAR TEORI**

Searching merupakan kegiatan mencari data yang akan dibutuhkan. Searching dalam pemrograman dapat dilakukan untuk mencari data yang berada pada memory komputer. Dalam kegiatan sehari-hari kita sering melakukannya seperti pada saat mencari informasi di internet.

Ada 2 metode yang digunakan dalam searching :

### **1. Sequential/Linear Search**

Metode pencarian data dalam array dengan cara membandingkan data yang akan dicari dengan data yang terdapat dalam array secara berurutan hingga data tersebut ditemukan atau mencapai akhir array. Pencarian ini lebih efektif untuk data yang tidak terurut/acak.

Proses sequential search :

- Menentukan data yang akan dicari
- Membaca data array satu persatu secara sekuensial
- Membaca data mulai dari data pertama hingga terakhir, lalu data yang dicari akan dibandingkan dengan masing-masing data yang ada dalam array. Apabila data ditemukan maka kita dapat membuat pernyataan 'data telah ditemukan', jika tidak ditemukan maka 'data tidak ditemukan'.

### **2. Binary Search**

Metode pencarian data dengan cara mengelompokkan array menjadi bagian-bagian. Berbeda dengan sequential search, pada binary search ini hanya dapat diterapkan pada data yang telah terurut baik ascending maupun descending dalam suatu array. Sehingga butuh algoritma sorting/pengurutan.

Proses binary search :

- Membuat perulangan kemudian menentukan posisi 'low' yaitu posisi yang menandakan index paling rendah lalu menentukan posisi 'high', setelah itu mencari posisi  $\text{'mid'} = \frac{\text{high} + \text{low}}{2}$
- Membandingkan data yang dicari dengan nilai yang berada di posisi 'mid'. Jika data yang dicari sama dengan nilai yang ada di posisi 'mid' maka data telah ditemukan. Jika data yang dicari lebih kecil daripada nilai 'mid' maka pencarian data akan dilakukan dibagian kiri 'mid' dengan melakukan pembandingan dan posisi high berubah ( $\text{mid}-1$ ) namun low tetap. Jika data yang dicari lebih besar daripada nilai 'mid' maka pencarian data akan dilakukan dibagian kanan 'mid' dengan posisi low berubah ( $\text{mid}+1$ ) namun high tetap.

## B. GUIDED

### - Guided I

#### 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;

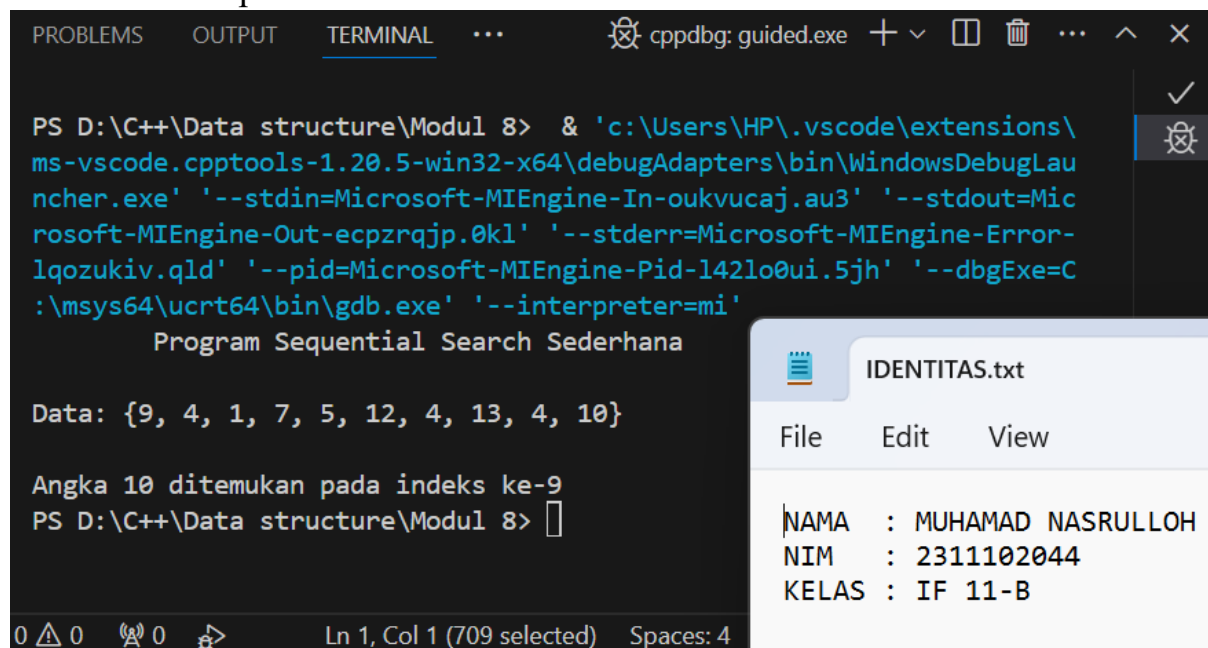
    // Algoritma Sequential Search
    for (i = 0; i < n; i++) {
        if (data[i] == cari){
            ketemu = true;
            break;
        }
    }

    cout << "\tProgram Sequential Search Sederhana\n" << endl;
    cout << "Data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}" << endl;

    if (ketemu) {
        cout << "\nAngka " << cari << " ditemukan pada indeks ke-" << i <<
endl;
    } else {
        cout << "\nAngka " << cari << " tidak dapat ditemukan pada data." <<
endl;
    }

    return 0;
}
```

## Screenshot Output



The screenshot shows a VS Code interface. The terminal window at the top displays the command prompt for a C++ program. The command executed is: `PS D:\C++\Data structure\Modul 8> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-oukvucaj.au3' '--stdout=Microsoft-MIEngine-Out-ecpzrqjp.0kl' '--stderr=Microsoft-MIEngine-Error-lqozukiv.qld' '--pid=Microsoft-MIEngine-Pid-l42lo0ui.5jh' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'`. The output shows the program title "Program Sequential Search Sederhana", the data array `Data: {9, 4, 1, 7, 5, 12, 4, 13, 4, 10}`, and the result "Angka 10 ditemukan pada indeks ke-9". Below the terminal, a text editor window titled "IDENTITAS.txt" is open, showing a form with the following text: `NAMA : MUHAMAD NASRULLOH`, `NIM : 2311102044`, and `KELAS : IF 11-B`.

```
PS D:\C++\Data structure\Modul 8> & 'c:\Users\HP\.vscode\extensions\
ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLau
ncher.exe' '--stdin=Microsoft-MIEngine-In-oukvucaj.au3' '--stdout=Mic
rosoft-MIEngine-Out-ecpzrqjp.0kl' '--stderr=Microsoft-MIEngine-Error-
lqozukiv.qld' '--pid=Microsoft-MIEngine-Pid-l42lo0ui.5jh' '--dbgExe=C
:\msys64\ucrt64\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 D:\C++\Data structure\Modul 8> █
```

IDENTITAS.txt

File Edit View

NAMA : MUHAMAD NASRULLOH  
NIM : 2311102044  
KELAS : IF 11-B

## Deskripsi

Program diatas merupakan implementasi sederhana sequential search untuk mencari angka dalam suatu data array. Dimana ada data berjumlah 10 dalam array yang bertipe data interger. Program akan melakukan perulangan (looping) for sebanyak n kali, lalu disetiap iterasinya nilai indeks i array ‘data’ dibandingkan dengan nilai ‘cari’ menunjukkan perbandingan antara data yang dicari dengan data yang ada pada array. Jika ditemukan kesamaan, variabel ‘ketemu’ diubah menjadi ‘true’ dan looping akan berhenti dengan adanya ‘break’.

## - Guided II

## Source Code

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>
int datas[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 (datas[j] < datas[min])
        {
            min = j;
        }
    }
    temp = datas[i];
    datas[i] = datas[min];
    datas[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 (datas[tengah] == cari)
        {
            b_flag = 1;
            break;
        }
        else if (datas[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) << datas[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) << datas[x];
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```

## Screenshot Output

```

PS D:\C++\Data structure\Modul 8> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-po3uzmw2.cce' '--stdout=Microsoft-MIEngine-Out-if33mc1e.kil' '--stderr=Microsoft-MIEngine-Error-sanopqec.jr0' '--pid=Microsoft-MIEngine-Id-engine-Pid-t15ecffq.shv' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
    BINARY SEARCH

Data :   1   8   2   5   4   9   7

Masukkan data yang ingin Anda cari : 2

Data diurutkan :   1   2   4   5   7   8   9

Data ditemukan pada index ke-1

```

## Deskripsi

Program diatas implementasi dari binary search sekaligus selection sort yang ada pada fungsi ‘void selection\_sort’. Selection sort digunakan untuk mengurutkan array terlebih dahulu, lalu pencarian data yang di inginkan menggunakan binary search yang ada pada fungsi ‘void binarysearch’ dari data yang telah di urutkan sebelumnya.

## C. UNGUIDED

### - Unguided I

Buatlah sebuah program untuk mencari sebuah huruf pada kalimat yang sudah di input dengan menggunakan binary search!

Source Code

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

using namespace std;

bool binarySearch(string kata, char cari, int awal, int akhir) {
    if (awal > akhir) return false;
    int tengah = (awal + akhir) / 2;
    if (kata[tengah] == cari) {
        cout << "\nHuruf '" << cari << "' ditemukan pada indeks ke-" << tengah
        << endl;
        return true;
    } else if (kata[tengah] < cari) {
        return binarySearch(kata, cari, tengah + 1, akhir);
    } else {
        return binarySearch(kata, cari, awal, tengah - 1);
    }
}

bool selectionSort(string& kata) {
    if (kata.empty()) return false;
    for (int i = 0; i < kata.size() - 1; i++) {
        int minIndex = i;
        for (int j = i + 1; j < kata.size(); j++) {
            if (kata[j] < kata[minIndex]) {
                minIndex = j;
            }
        }
        swap(kata[i], kata[minIndex]);
    }
    return true;
}

int main() {
    string kata;
    char cari;
    cout << "Masukkan kata/kalimat: ";
    getline(cin, kata);
    selectionSort(kata);
    cout << "Kata/kalimat setelah diurutkan: " << kata << endl;
```

```

bool found = false;

do {
    cout << "\nMasukkan huruf yang ingin dicari: ";
    cin >> cari;
    found = binarySearch(kata, cari, 0, kata.size() - 1);
    if (!found) {
        cout << "\nHuruf '" << cari << "' tidak ditemukan." << endl;
        cout << "\nApakah ingin mencari huruf lain? (Y/T): ";
        char pilihan;
        cin >> pilihan;
        found = (pilihan == 'Y' || pilihan == 'y');
    }
} while (found);
return 0;
}

```

## Screenshot Output

```

PS D:\C++\Data structure\Modul 8> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0jgdnbhe.24z' '--stdout=Microsoft-MIEngine-Out-itj0uw0t.3pj' '--stderr=Microsoft-MIEngine-Error-tmdhnm5m.w4h' '--pid=Microsoft-MIEngine-Pid-mwpeg4wu.b0x' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan kata/kalimat: Halo gue Nasrul
Kata/kalimat setelah diurutkan: HNaaegllorsuu

Masukkan huruf yang ingin dicari: N
Huruf 'N' ditemukan pada indeks ke-3

Masukkan huruf yang ingin dicari: n
Huruf 'n' tidak ditemukan.

Apakah ingin mencari huruf lain? (Y/T): T
PS D:\C++\Data structure\Modul 8>

```

IDENTITAS.txt

```

File Edit View
NAMA : MUHAMAD NASRULLOH
NIM : 2311102044
KELAS : IF 11-B

```

## Deskripsi

Program diatas dibuat dengan algoritma binary search yang juga mengaplikasikan selection sort untuk mengurutkan data yang telah di inputkan. Ketika user menginputkan kata/kalimat, langsung algoritma ‘selectionSort’ mengurutkan data kata/kalimat tersebut dari yang terkecil ke terbesar. Lalu ‘binarySearch’ akan mencari huruf yang di iinputkan user tadi. Selain itu, program ini juga menggunakan looping untuk memberi kesempatan user mencari huruf lainnya bila huruf itu tidak ditemukan.

### - Unguidd II

Buatlah sebuah program yang dapat menghitung banyaknya huruf vokal dalam suatu kalimat!



## Source Code

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

using namespace std;

int main() {
    string kata;
    int jumlahVokal = 0;

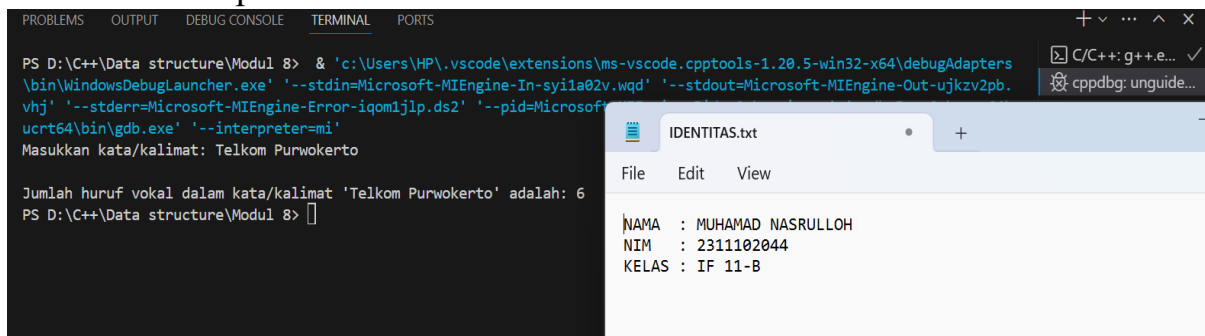
    cout << "Masukkan kata/kalimat: ";
    getline(cin, kata);

    for (char huruf : kata) {
        if (tolower(huruf) == 'a' || tolower(huruf) == 'e' || tolower(huruf) == 'i' ||
            tolower(huruf) == 'o' || tolower(huruf) == 'u') {
            jumlahVokal++;
        }
    }

    cout << "\nJumlah huruf vokal dalam kata/kalimat '" << kata << "' adalah: "
    << jumlahVokal << endl;

    return 0;
}
```

## Screenshot Output



## Deskripsi

Program diatas adalah program sederhana untuk menghitung jumlah karakter (huruf vokal) yang telah di inputkan oleh user. Dengan huruf vokal yang sudah di setting dalam perulangan for. Program lalu akan mengecek apakah karakter yang di inputkan memiliki huruf vokal (a, i, u, e, o). Jika iya maka status 'jumlahVokal' akan bertambah 1.

### - Unguided III

Diketahui data = 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. Hitunglah berapa banyak angka 4 dengan algoritma sequential search!

#### Source Code

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

int main() {
    int data[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int n = sizeof(data) / sizeof(data[0]); // Ukuran array
    cout << "Data array : ";
    for (int i = 0; i < 10; i++) {
        cout << data[i] << " ";
    }
    cout << endl;

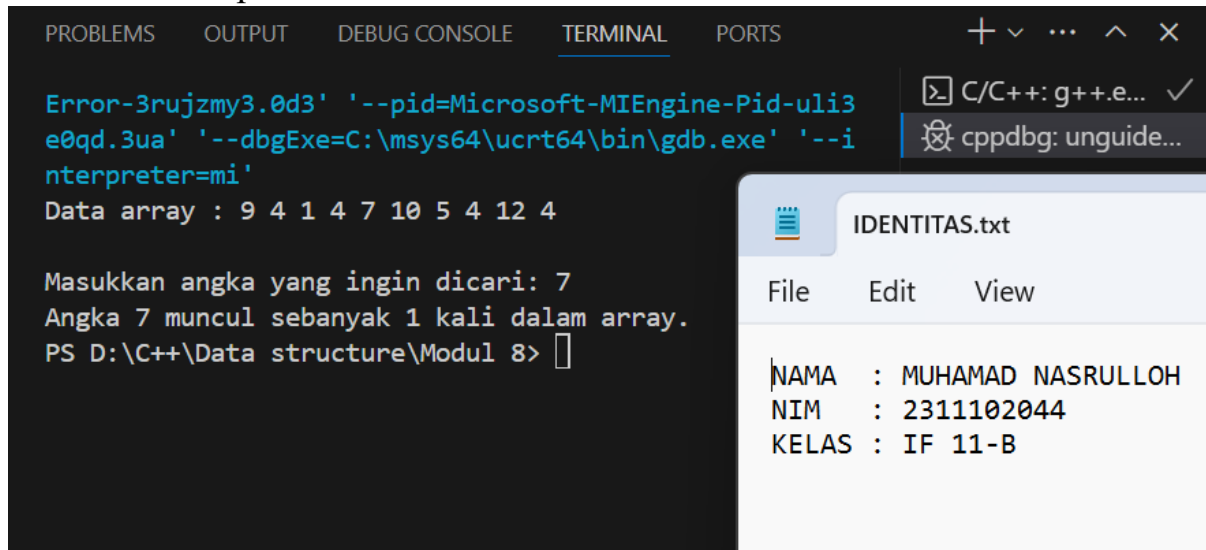
    int angkaDicari;
    cout << "\nMasukkan angka yang ingin dicari: ";
    cin >> angkaDicari;
    int kemunculan = 0;

    // Algoritma Sequential Search
    for (int i = 0; i < n; i++) {
        if (data[i] == angkaDicari) {
            kemunculan++;
        }
    }

    // Output hasil pencarian
    if (kemunculan > 0) {
        cout << "Angka " << angkaDicari << " muncul sebanyak " << kemunculan <<
" kali dalam array." << endl;
    } else {
        cout << "Angka " << angkaDicari << " tidak ditemukan dalam array." <<
endl;
    }

    return 0;
}
```

## Screenshot Output



```
Error-3rujzmy3.0d3' '--pid=Microsoft-MIEngine-Pid-uli3
e0qd.3ua' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--i
nterpreter=mi'
Data array : 9 4 1 4 7 10 5 4 12 4

Masukkan angka yang ingin dicari: 7
Angka 7 muncul sebanyak 1 kali dalam array.
PS D:\C++\Data structure\Modul 8> 
```

IDENTITAS.txt

	File	Edit	View
NAMA	: MUHAMAD NASRULLOH		
NIM	: 2311102044		
KELAS	: IF 11-B		

## Deskripsi

Program diatas mengimplementasikan sequential search untuk mencari data pada suatu array. Di program ini juga tidak hanya mencari namun melihat kemungkinan berapa kali data tersebut muncul dalam array. Dilihat pada perulangan for disana melakukan pengecekan apakah data indeks (dalam array) = data yang dicari, jika sama maka nilai kemunculan tambah 1 dari yang semula disetting 0. Jika ternyata kemunculan tersebut lebih dari 0, maka akan ada pesan bahwa angka yang dicari telah ditemukan dan menampilkan berapa kali kemunculannya. Namun jika 0, maka akan ada pesan bahwa angka yang dicari tidak ditemukan.

## D. KESIMPULAN

Searching merupakan salah satu operasi penting dalam pemrograman C++ dengan teknik yang sesuai untuk berbagai jenis data dan kebutuhan. Sequential/linear search sederhana dan efektif untuk dataset kecil atau yang tidak terurut. Sementara binary search hadir dengan kinerja yang lebih cepat untuk data yang sudah terurut. Masing-masing metode memiliki kelebihan dan kekurangannya.

## **E. REFERENSI**

- [1] Bocah Alpro (2018). Searching dalam Bahasa Pemrograman C++  
<https://sinaualpro.blogspot.com/2018/12/searching-dalam-bahasa-pemrograman-c.html>.
- [2] Admin Sarjana (2019). Belajar C++ Part 6 (Searching)  
<https://www.sarjanapedia.com/2019/02/metode-searching-dalam-cpp.html>.
- [3] Koding Akademi. Pengenalan Searching dalam C++  
<https://www.kodingakademi.id/pengenalan-searching-dalam-c/>
- [4] Asisten Praktikum (2024). Modul 8 Algoritma Searching