

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL 8
ALOGARITMA SEARCHING**



Disusun Oleh :
MARSHELY AYU ISWANTO
2311102073

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

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

A. Dasar Teori

Pencarian atau searching merupakan mencari data yang akan kita butuhkan. Searching dalam pemrograman dilakukan untuk mencari data yang berada pada memori komputer. 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 pencarian nilai tertentu dalam kumpulan data, seperti array atau struktur data lainnya. Terdapat 2 metode pada algoritma searching:

a. Sequential Search

Sequential Search adalah pencarian data dalam array dengan cara membandingkan data yang dicari dengan data yang terdapat dalam array secara berurutan, 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.

Algoritma pencarian berurutan dapat dituliskan sebagai berikut :

1. $i \leftarrow 0$ Praktikum Struktur Data dan Algoritma 1
2. $ketemu \leftarrow false$
3. Selama (tidak $ketemu$) dan $(i \leq N)$ kerjakan baris 4
4. Jika $(Data[i] = x)$ maka $ketemu \leftarrow true$, jika tidak $i \leftarrow i + 1$
5. Jika $(ketemu)$ maka i adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.

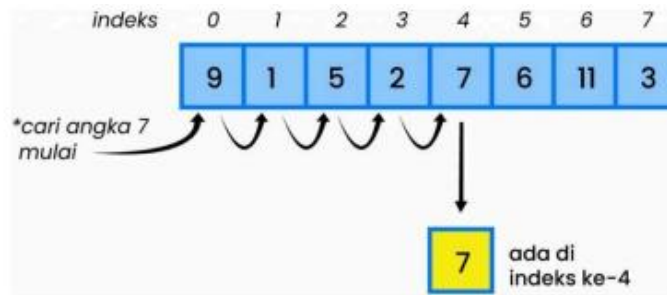
Di bawah ini merupakan fungsi untuk mencari data menggunakan pencarian sekuensial.

```
int SequentialSearch (int x)
{
    int i = 0;
    bool ketemu = false;
    while ((!ketemu) && (i < Max)){
        if (Data[i] == x)
            ketemu = true;
        else
            i++;
    }
    if (ketemu)
        return i;
    else
        return -1;
}
```

Fungsi diatas akan mengembalikan indeks dari data yang dicari. Apabila data tidak ditemukan maka fungsi diatas akan mengembalikan nilai -1.

Contoh dari Sequential Search, yaitu:

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



Gambar 1. Ilustrasi Sequential Search

b. Binary Search

Binary search merupakan pencarian data dengan mengelompokkan array menjadi bagian-bagian, Binary Search termasuk ke dalam interval search. *Binary Search* hanya dapat diterapkan pada data yang telah terurut baik ascending maupun descending dalam suatu array. 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

Algoritma pencarian biner dapat dituliskan sebagai berikut :

1) $L \leq 0$

2) $R \leq N - 1$

3) ketemu \square false

4) Selama ($L \leq R$) dan (tidak ketemu) kerjakan baris 5 sampai dengan 8

5) $m \square (L + R) / 2$

6) Jika ($\text{Data}[m] = x$) maka ketemu \square true

7) Jika ($x < \text{Data}[m]$) maka $R \square m - 1$

8) Jika ($x > \text{Data}[m]$) maka $L \square m + 1$

9) Jika (ketemu) maka m adalah indeks dari data yang dicari, jika tidak data tidak ditemukan.

Contoh dari Binary Search, yaitu:



Gambar 2. Ilustrasi Binary Search

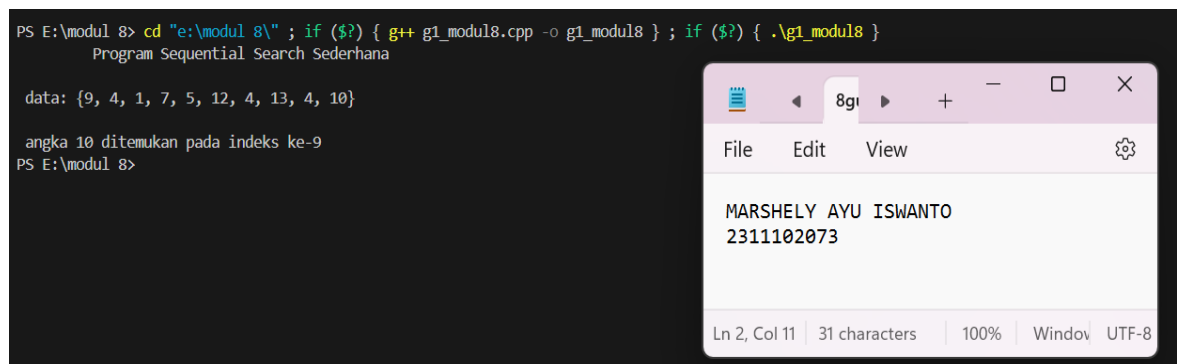
B. Guided

Guided 1 : Buatlah sebuah project dengan menggunakan sequential search sederhana untuk melakukan pencarian data.

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;
}
```

Screenshots Output



```
PS E:\modul 8> cd "e:\modul 8\" ; if ($?) { g++ g1_modul8.cpp -o g1_modul8 } ; if ($?) { .\g1_modul8 }
Program Sequential Search Sederhana

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

angka 10 ditemukan pada indeks ke-9
PS E:\modul 8>
```

The Notepad++ editor shows the following 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;
}
```

Deskripsi Program:

Program diatas merupakan program c++ yaitu program Sequential Search Sederhana.

Program ini dimulai dari mendeklarasikan array 'data' yang berisi 10 bilangan bulat, Selanjutnya program ini mencari angka tertentu yang disimpan dalam variable 'cari' dengan menggunakan algoritma pencarian sekuen. Proses pencarian dilakukan dalam iterasi melalui setiap elemen dalam array 'data'. Output dari program ini adalah apakah angka yang dicari ditemukan dan menampilkan indeksinya jika ditemukan.

Guided 2 : Buatlah sebuah project untuk melakukan pencarian data dengan menggunakan Binary Search.

Source Code

```
#include <iostream>
using namespace std;
#include <conio.h>
#include <iomanip>
int dataa[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 (dataa[j] < dataa[min])
            {
                min = j;
            }
        }
        temp = dataa[i];
        dataa[i] = dataa[min];
        dataa[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 (dataa[tengah] == cari)
        {
            b_flag = 1;
            break;
        }
        else if (dataa[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 dataa awal
    for (int x = 0; x < 7; x++)
        cout << setw(3) << dataa[x];
    cout << endl;
    cout << "\n Masukkan dataa yang ingin Anda cari :";
    cin >> cari;
    cout << "\n Data diurutkan : ";
    // urutkan dataa dengan selection sort
    selection_sort();
    // tampilkan dataa setelah diurutkan
    for (int x = 0; x < 7; x++)
        cout << setw(3) << dataa[x];
    cout << endl;
    binarysearch();
    _getche();
    return EXIT_SUCCESS;
}

```

Screenshoots Output

The screenshot shows two windows. The terminal window on the left displays the output of the C++ program. It starts with the command prompt 'PS E:\modul 8>' followed by the compilation command 'cd "e:\modul 8\" ; if (\$?) { g++ g2.cpp -o g2 } ; if (\$?) { .\g2 }'. The program output is as follows:

```

BINARY SEARCH

Data :  1  8  2  5  4  9  7

Masukkan dataa yang ingin Anda cari :7

Data diurutkan :  1  2  4  5  7  8  9

Data ditemukan pada index ke- 4

```

The Notepad++ editor window on the right shows the input 'MARSHELY AYU ISWANTO' and '2311102073' entered into a text field. The status bar at the bottom of the editor indicates 'Ln 2, Col 11 | 31 characters | 100% | Window | UTF-8'.

Deskripsi Program

Program diatas merupakan program c++ yaitu program Binary Search Sederhana untuk mencari sebuah angka dalam array. Dalam program ini pengguna diminta untuk memasukkan angka yang ingin dicari, lalu mengurutkan array menggunakan selection sort, Kemudian program mencari angka yang dicari setelah diurutkan menggunakan

binary search. Output dari program ini adalah memberitahu apakah angka yang dicari itu ada atau tidak dan menampilkan indeks setelah diurutkan.

C. Unguided

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 <algorithm>
#include <vector>

using namespace std;

// Fungsi untuk melakukan Binary Search
bool binarySearch(const vector<char>& sortedVec, char
target) {
    int left = 0;
    int right = sortedVec.size() - 1;

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

        if (sortedVec[middle] == target) {
            return true;
        } else if (sortedVec[middle] < target) {
            left = middle + 1;
        } else {
            right = middle - 1;
        }
    }

    return false;
}

int main() {
    string kalimat;
    char huruf;

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

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

    vector<char> vec(kalimat.begin(), kalimat.end());
```



```

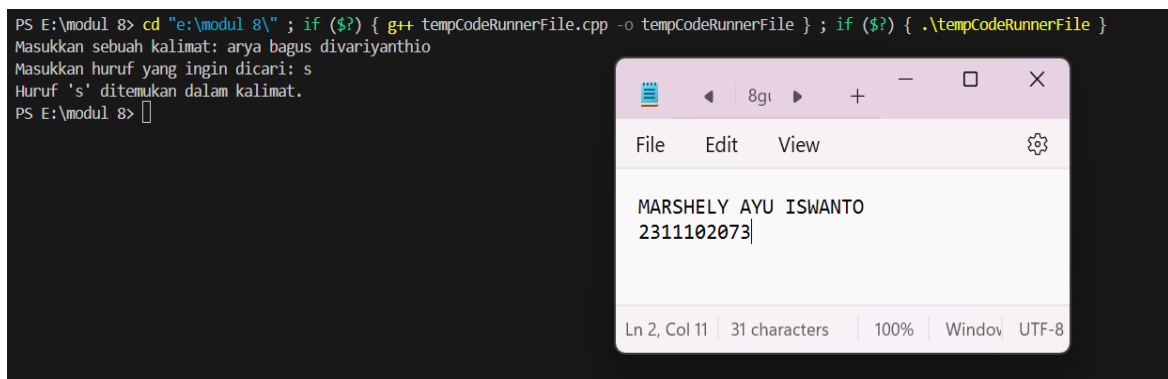
        sort(vec.begin(), vec.end());

        // Mencari huruf menggunakan Binary Search
        if (binarySearch(vec, huruf)) {
            cout << "Huruf '" << huruf << "' ditemukan dalam
kalimat." << endl;
        } else {
            cout << "Huruf '" << huruf << "' tidak ditemukan
dalam kalimat." << endl;
        }

        return 0;
    }
}

```

Screenshots Output



Deskripsi Program :

Program diatas merupakan program c++ yaitu program memanfaatkan fungsi `binarySearch` untuk melakukan pencarian biner pada vektor karakter yang telah diurutkan. Fungsi ini bekerja dengan memeriksa elemen tengah dari rentang yang tersisa, kemudian mempersempit rentang pencarian berdasarkan perbandingan antara elemen tengah dan huruf yang dicari. Jika huruf ditemukan, fungsi mengembalikan 'true' jika tidak, fungsi mengembalikan 'false'. Berdasarkan hasil pencarian, program akan menampilkan output yang menunjukkan apakah huruf yang dicari terdapat dalam kalimat atau tidak.

Unguided 2

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

```

#include <iostream>
#include <string>
#include <cctype>

using namespace std;

int hitungVokal(const string &kalimat, string &vokal) {
    int jumlahVokal = 0;
    for (char huruf : kalimat) {

        char hurufKecil = tolower(huruf);

        if (hurufKecil == 'a' || hurufKecil == 'e' ||
hurufKecil == 'i' ||
        hurufKecil == 'o' || hurufKecil == 'u') {
            jumlahVokal++;

            if (vokal.find(hurufKecil) == string::npos) {
                if (!vokal.empty())
                    vokal += ' ';
                vokal += hurufKecil;
            }
        }
    }
    return jumlahVokal;
}

int main() {
    string kalimat;
    string hurufVokal;

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

    int jumlahVokal = hitungVokal(kalimat, hurufVokal);

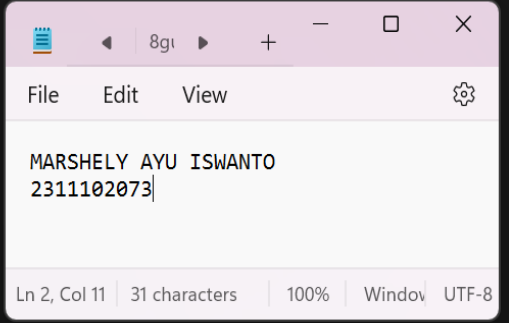
    cout << "Jumlah huruf vokal dalam kalimat adalah: " <<
jumlahVokal << endl;
    cout << "Huruf vokal yang muncul dalam kalimat: " <<
hurufVokal << endl;

    return 0;
}

```

Screenshots Output

```
PS E:\modul 8> cd "e:\modul 8\" ; if ($?) { g++ unguided2_modul8.cpp -o unguided2_modul8 } ; if ($?) { .\unguided2_modul8 }
Masukkan kalimat: arya bagus divariyanthio
Jumlah huruf vokal dalam kalimat adalah: 10
Huruf vokal yang muncul dalam kalimat: a u i o
PS E:\modul 8>
```



Deskripsi Program

Program ini merupakan program c++ yaitu program menghitung jumlah huruf vokal dan memunculkan huruf vokal apa saja yang muncul pada kalimat yang diinputkan pengguna. Setelah pengguna memasukkan kalimat, program menggunakan fungsi 'hitungVokal' untuk memprosesnya. Dalam fungsi ini, setiap karakter diubah menjadi huruf kecil, dan jika merupakan huruf 'vokal' dan belum ada dalam string 'vokal', maka huruf tersebut ditambahkan ke vokal.

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>
#include <iomanip>

using namespace std;

void bubbleSort(int data[], int size) {
    for (int i = 0; i < size-1; i++) {
        for (int j = 0; j < size-i-1; j++) {
            if (data[j] > data[j+1]) {
                int temp = data[j];
                data[j] = data[j+1];
                data[j+1] = temp;
            }
        }
    }
}

int sequentialSearch(const int data[], int size, int target) {
    int count = 0;
    for (int i = 0; i < size; i++) {
        if (data[i] == target) {
            count++;
        }
    }
}
```

```

    }
    return count;
}

int main() {
    int originalData[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int dataSize = sizeof(originalData) /
sizeof(originalData[0]);
    int data[dataSize];
    for (int i = 0; i < dataSize; ++i) {
        data[i] = originalData[i];
    }

    int target = 4;

    cout << "Data sebelum diurutkan: ";
    for (int i = 0; i < dataSize; ++i) {
        cout << setw(2) << data[i] << " ";
    }
    cout << endl;

    bubbleSort(data, dataSize);

    cout << "Data yang sudah terurut: ";
    for (int i = 0; i < dataSize; ++i) {
        cout << setw(2) << data[i] << " ";
    }
    cout << endl;

    int jumlah_angka_4 = sequentialSearch(data, dataSize,
target);

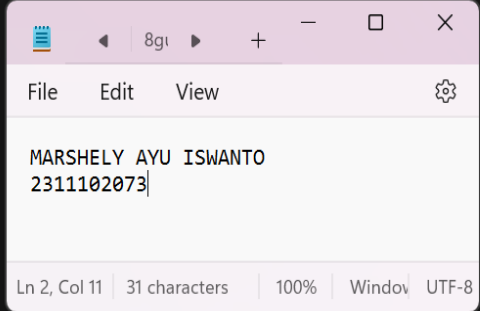
    cout << "Jumlah angka 4 dalam data adalah: " <<
jumlah_angka_4 << endl;

    return 0;
}

```

Screenshoots Output

```
PS E:\modul 8> cd "e:\modul 8\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Data sebelum diurutkan: 9 4 1 4 7 10 5 4 12 4
Data yang sudah terurut: 1 4 4 4 4 5 7 9 10 12
Jumlah angka 4 dalam data adalah: 4
PS E:\modul 8>
```



Deskripsi Program

Program tersebut merupakan program c++ yaitu mengimplementasikan algoritma bubble sort untuk mengurutkan array dari terkecil ke terbesar, dan pencarian sekuensial untuk menghitung jumlah kemunculan suatu nilai target dalam array yang telah diurutkan. Array asli diinisialisasi dengan nilai-nilai tertentu, kemudian diurutkan menggunakan bubble sort. Setelah diurutkan, program menghitung berapa kali nilai target muncul dalam array tersebut dan mencetak jumlahnya.

Kesimpulan

Alogaritma Searching digunakan untuk menemukan elemen tertentu dalam kumpulan data. Pemilihan algoritma tergantung pada ukuran data, urutan data, frekuensi pencarian, dan informasi tambahan tentang data.

D. Referensi

- Abba, I. V. (2023, Maret 17). *Pencarian Biner di C++ – Contoh Algoritma*. From <https://www.freecodecamp.org>: <https://www.freecodecamp.org/news/binary-search-in-c-algorithm-example/>
- Andre. (2021, oktober 4). *Latihan Kode Program C++: Pencarian Data Array (Searching)*. From <https://www.duniailkom.com>: https://www.duniailkom.com/latihan-kode-program-cpp-pencarian-data-array-searching/#google_vignette
- Dimas, S. (2020, Januari 17). *Contoh Program Searching C++ Dengan Array*. From <https://kelasprogrammer.com>: <https://kelasprogrammer.com/contoh-program-searching-c-array/>