

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN
ALGORITMA**

**MODUL VIII
ALGORITMA SEARCHING**



Disusun Oleh :

Muhammad Rusdiyanto Asatman
2311102053

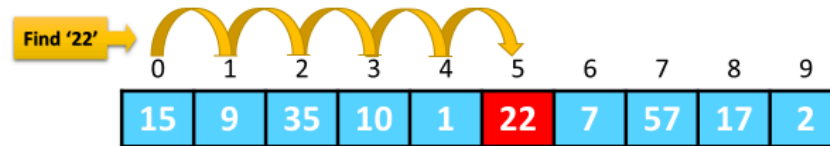
Dosen :

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

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

A. Dasar Teori

1. Sequential Search

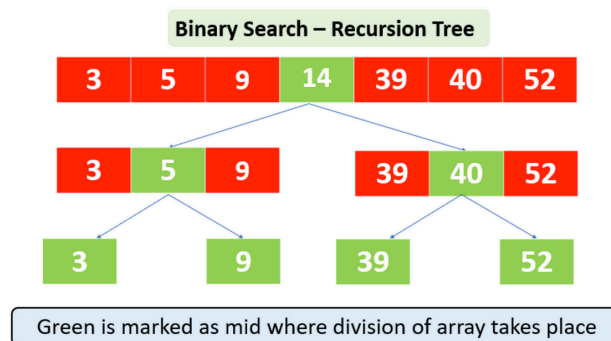


Gambar 1.0, Ilustrasi Algoritma Sequential Search

Sumber : <https://rohanjpatil63.medium.com/>

Sequential Search adalah teknik pencarian data yang dilakukan dengan cara membandingkan setiap elemen data satu per satu, mulai dari elemen pertama hingga elemen yang dicari ditemukan. Proses ini terus berlanjut hingga data ditemukan atau seluruh elemen telah diperiksa. Jadi, jika ada data yang belum teratur atau berpola acak, Sequential Search dapat digunakan untuk mencari elemen yang dibutuhkan. Algoritma ini akan membandingkan setiap elemen dengan elemen yang dicari hingga menemukan hasil yang diinginkan. Jika elemen yang dicari terletak di awal daftar, pencarian akan lebih efisien, tetapi jika terletak di akhir daftar, proses pencarian akan memakan waktu lebih lama.

2. Binary Search



Gambar 1.1. Ilustrasi Algoritma Binary Search

Sumber : <https://medium.com/@imanshu822/>

Binary Search adalah algoritma pencarian yang efisien untuk menemukan nilai target dalam suatu himpunan data yang telah diurutkan. Algoritma ini bekerja dengan membagi sehimpunan data menjadi dua bagian secara berulang hingga nilai target ditemukan atau interval pencarian menjadi kosong. Keunggulan Binary Search terletak pada penggunaan informasi bahwa himpunan data telah diurutkan, sehingga

kompleksitas waktu dapat dikurangi menjadi $O(\log N)$.

Berikut langkah-langkah kerja Binary Search :

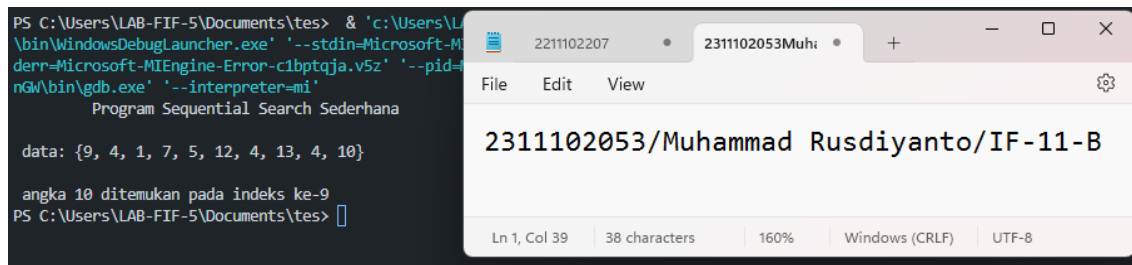
- Bagi ruang pencarian menjadi dua bagian dengan menemukan indeks tengah “mid”.
- Bandingkan elemen tengah ruang pencarian dengan kunci yang dicari.
- Jika kunci ditemukan pada elemen tengah, proses berhenti.
- Jika kunci tidak ditemukan pada elemen tengah, tentukan bagian mana yang akan digunakan sebagai ruang pencarian berikutnya:
- Jika kunci lebih kecil dari elemen tengah, gunakan sisi kiri sebagai ruang pencarian berikutnya.
- Jika kunci lebih besar dari elemen tengah, gunakan sisi kanan sebagai ruang pencarian berikutnya.
- Proses ini berlanjut hingga kunci ditemukan atau seluruh ruang pencarian habis.

B. Guided

Guided 1

```
#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 C:\Users\LAB-FIF-5\Documents\tes> & 'c:\Users\LAB-FIF-5\Documents\tes\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-Error-c1bptqja.v5z' '--pid=1000'
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\LAB-FIF-5\Documents\tes>
```

2311102053/Muhammad Rusdiyanto/IF-11-B

Deskripsi:

Program di atas adalah program yang mendemonstrasikan penggunaan sequential search. Demonstrasi ini dilakukan dengan mencari salah satu data dalam array berukuran 10 dengan tipe data integer (array berisi angka). Untuk proses pencarian dalam program ini dimulai dengan looping ke setiap index dalam array. Lalu, nilai array dalam indeks tersebut akan dibandingkan dengan nilai yang dicari. Jika nilai dalam indeks tersebut sama dengan nilai yang dicari (dalam konteks ini, nilai yang dicari adalah 10), maka program akan mengubah nilai variabel `ketemu` menjadi true dan looping akan diberhentikan. Selain itu, maka program akan melanjutkan looping selagi menambahkan nilai `i` dengan 1 ($i + 1$). Setelah looping selesai, program akan menampilkan seluruh elemen array dan pesan sesuai dengan kondisi apakah nilai ditemukan atau tidak (dari nilai variabel `ketemu`). Jika nilai ditemukan, maka program akan menampilkan pesan bahwa nilai ditemukan. Jika tidak, maka program akan menampilkan pesan bahwa nilai tidak ditemukan dalam array.

Guided 2

```
#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;
}

```

Screenshots Output

```

PS C:\Users\LAB-FIF-5\Documents\tes> & 'c:\Users\LAB-FIF-5\.vscode\extensions\ms-vscode.cpptools-1.20.2-win32-x64\debugAdapters
\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-e2kedzxc.usr' '--stdout=Microsoft-MIEngine-Out-4rvxkn10.kf5' '--st
derr=Microsoft-MIEngine-Error-rpz2wk1g.3yv' '--pid=Microsoft-MIEngine-Pid-aqb5lfco.kqj' '--dbgExe=C:\Program Files\CodeBlocks\Mi
nGW\bin\gdb.exe' '--interpreter=mi'
BINARY SEARCH

Data :  1  8  2  5  4  9  7

Masukkan data yang ingin Anda cari :4

Data diurutkan :  1  2  4  5  7  8  9

Data ditemukan pada index ke-2
PS C:\Users\LAB-FIF-5\Documents\tes>

```

Deskripsi:

Program di atas adalah program yang mendemonstrasikan penggunaan algoritma binary search. Dalam program ini, data disimpan dalam array dengan ukuran 7 elemen yang menyimpan data integer. Untuk proses running program dimulai dengan

menampilkan semua data dalam array. Setelah semua data ditampilkan, program akan meminta input dari pengguna berupa angka yang akan dikembalikan/tampilkan indeksinya. Selanjutnya, program akan mengurutkan data, lalu data baru akan dicari menggunakan binary search. Setelah itu, data ditemukan, maka program akan menampilkan pesan disertai indeks dari elemen yang dicari. Di samping itu, program ini memiliki 2 fungsi selain fungsi main, yaitu :

1. `selection_sort` : sebuah fungsi yang digunakan untuk menyortir/mengurutkan elemen array. Proses pengurutan dilakukan dengan cara memilih elemen dari array dengan nilai terkecil, lalu menempatkannya di indeks yang sesuai (semakin kecil nilai elemen, semakin kecil indeksinya, begitu pula sebaliknya).
2. `binarysearch` : berfungsi untuk mencari nilai dari array yang telah diurutkan menggunakan algoritma binary search.

C. Unguided

Unguided 1

```
#include <iostream>
#include <conio.h>
#include <iomanip>

using namespace std;

string sentence;
char c;

void toLower() {
    string temp;
    for (int i = 0; i < sentence.length(); i++) {
        temp += tolower(sentence[i]);
    }
    sentence = temp;
}

void selection_sort()
{
    int min, i, j;
    char temp;
    toLower();
    for (i = 0; i < sentence.length(); i++)
    {
        min = i;
        for (j = i + 1; j < sentence.length(); j++)
        {
            if (sentence[j] < sentence[min])
            {
                min = j;
            }
        }
        temp = sentence[i];
        sentence[i] = sentence[min];
        sentence[min] = temp;
    }
}

void binarysearch()
{
    int awal, akhir, tengah, b_flag = 0;
```

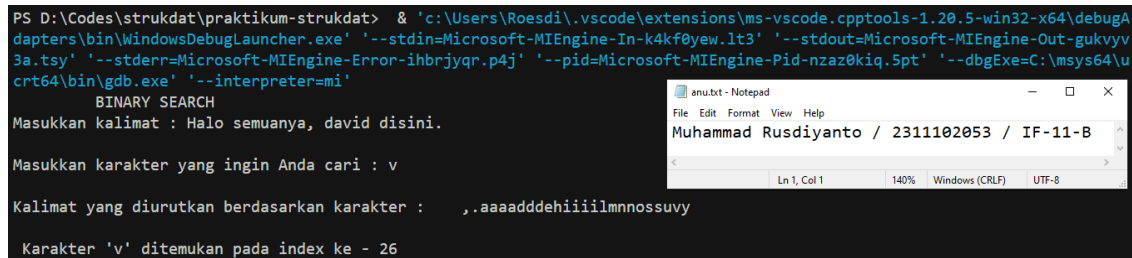
```

    awal = 0;
    akhir = sentence.length();
    while (b_flag == 0 && awal <= akhir)
    {
        tengah = (awal + akhir) / 2;
        if (sentence[tengah] == c)
        {
            b_flag = 1;
            break;
        }
        else if (sentence[tengah] < c)
            awal = tengah + 1;
        else
            akhir = tengah - 1;
    }
    if (b_flag == 1)
        cout << "\n Karakter '" << c << "' ditemukan pada index ke
- " << tengah << endl;
    else
        cout << "\nData tidak ditemukan\n";
}

int main()
{
    cout << "\t BINARY SEARCH " << endl;
    cout << "Masukkan kalimat : ";
    getline(cin, sentence);
    cout << "\nMasukkan karakter yang ingin Anda cari : ";
    cin >> c;
    c = tolower(c);
    cout << "\nKalimat yang diurutkan berdasarkan karakter : ";
    selection_sort();
    for (int x = 0; x < sentence.length(); x++)
        cout << sentence[x];
    cout << endl;
    binarysearch();
    return EXIT_SUCCESS;
}

```

Screenshots Output



```
PS D:\Codes\strukdat\praktikum-strukdat> & 'c:\Users\Roesdi\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-k4kf0yew.lt3' '--stdout=Microsoft-MIEngine-Out-gukvyv3a.tsy' '--stderr=Microsoft-MIEngine-Error-ihbrjyqr.p4j' '--pid=Microsoft-MIEngine-Pid-nzaz0kiq.5pt' '--dbgExe=C:\msys64\ucrtd64\bin\gdb.exe' '--interpreter=mi'
BINARY SEARCH
Masukkan kalimat : Halo semuanya, david disini.

Masukkan karakter yang ingin Anda cari : v

Kalimat yang diurutkan berdasarkan karakter : ,.aaaadddehiilmmnossuvy

Karakter 'v' ditemukan pada index ke - 26
```

The screenshot shows a terminal window with a Windows Debug Launcher command and its output. The output includes a prompt for a sentence, a prompt for a character, and the sorted sentence with the character's index. A Notepad window is also visible, showing the text 'Muhammad Rusdiyanto / 2311102053 / IF-11-B'.

Deskripsi:

Program di atas adalah sebuah program yang digunakan untuk mencari karakter di dalam suatu kalimat menggunakan binary search. Dalam program ini, pengguna akan memasukkan kalimat mereka, dilanjutkan dengan memasukkan karakter yang akan mereka cari indeksnya. Sebelum dicari, kalimat akan di-convert menjadi huruf kecil (lowercase). Setelah di-convert, program akan menyortir kalimat berdasarkan karakter terkecil ke terbesar (ascending) menggunakan algoritma selection sort (seperti program Guided 2). Program akan menampilkan kalimat yang telah di-convert dan diurutkan. Baru setelah itu, program akan mencari karakter yang diinginkan dan menampilkan posisi/indeksnya. Indeks dari karakter tadi ditentukan berdasarkan kalimat yang telah disortir (bukan kalimat asli, karena menggunakan binary search).

Unguided 2

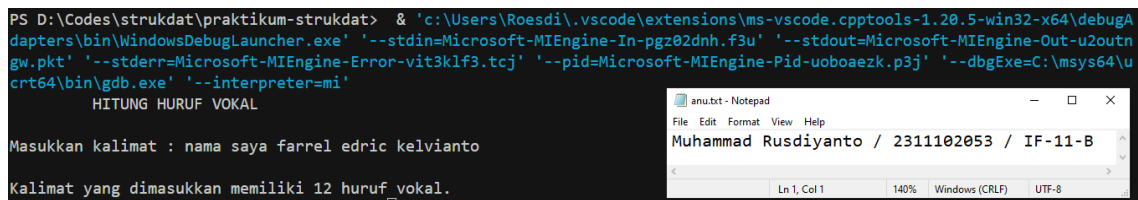
```
#include <iostream>
using namespace std;
int main()
{
    string s;
    int count = 0;

    cout << "\t HITUNG HURUF VOKAL" << endl;
    cout << "\nMasukkan kalimat : ";
    getline(cin, s);

    for (int i = 0; i < s.length(); i++) {
        char c = tolower(s[i]);
        if (c == 'a' || c == 'i' || c == 'u' || c == 'e' || c ==
'o') count++;
    }

    cout << "\nKalimat yang dimasukkan memiliki " << count << "
huruf vokal.";
    return 0;
}
```

Screenshots Output



```
PS D:\Codes\strukdat\praktikum-strukdat> & 'c:\Users\Roesdi\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugA
dapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-pgz02dnh.f3u' '--stdout=Microsoft-MIEngine-Out-u2outn
gw.pkt' '--stderr=Microsoft-MIEngine-Error-vit3klf3.tcj' '--pid=Microsoft-MIEngine-Pid-uoboaekz.p3j' '--dbgExe=C:\msys64\u
crt64\bin\gdb.exe' '--interpreter=mi'
HITUNG HURUF VOKAL

Masukkan kalimat : nama saya farrel edric kelvianto

Kalimat yang dimasukkan memiliki 12 huruf vokal.
```

Deskripsi:

Program tersebut merupakan sebuah program untuk menghitung banyaknya huruf vokal dalam suatu kalimat. Dalam program ini, pengguna perlu memasukkan kalimat mereka. Lalu, program akan memulai proses dengan mengonversi setiap karakter dari awal sampai akhir secara berurutan menjadi lowercase dan dicek apakah karakter tersebut termasuk dari salah satu huruf vokal (a, i, u, e, o). Jika iya, maka program akan menambahkan jumlah count dengan 1. Jika tidak, maka program akan lanjut ke karakter selanjutnya. Setelah itu, program akan menampilkan pesan yang disertai dengan jumlah huruf vokal dalam kalimat yang dimasukkan tadi.

Unguided 3

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

int arr[] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
int arrLength = sizeof(arr)/sizeof(arr[0]);
int cari;

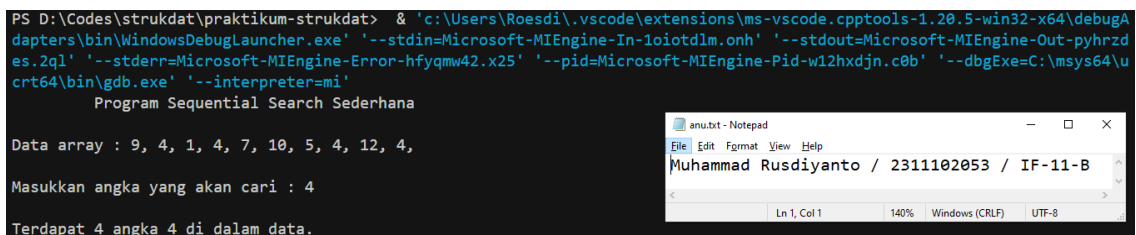
int seqSearch() {
    int count = 0;
    for (int i = 0; i < arrLength; i++)
    {
        if (arr[i] == cari)
        {
            count++;
        }
    }
    return count;
}

int main()
{
    // algoritma Sequential Search

    cout << "\t Program Sequential Search Sederhana\n" << endl;
    cout << "Data array : ";
    for (int x : arr) {
        cout << x << ", ";
    }
    cout << "\n\nMasukkan angka yang akan cari : ";
    cin >> cari;

    cout << "\nTerdapat " << seqSearch() << " angka " << cari << "
di dalam data.";
    return 0;
}
```

Screenshots Output



The screenshot shows the output of a C++ program in a terminal window. The program is titled "Program Sequential Search Sederhana". It displays the data array: 9, 4, 1, 4, 7, 10, 5, 4, 12, 4. It prompts the user to enter a number to search for, and the user enters 4. The output shows that there are 4 occurrences of the number 4 in the array.

```
PS D:\Codes\strukdat\praktikum-strukdat> & 'c:\Users\Roesdi\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-1oioetdml.onh' '--stdout=Microsoft-MIEngine-Out-pyhrzdes.2ql' '--stderr=Microsoft-MIEngine-Error-hfyqmw42.x25' '--pid=Microsoft-MIEngine-Pid-w12hxdjn.c0b' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Program Sequential Search Sederhana

Data array : 9, 4, 1, 4, 7, 10, 5, 4, 12, 4,

Masukkan angka yang akan cari : 4

Terdapat 4 angka 4 di dalam data.
```

In the background, a Notepad window titled "anu.txt - Notepad" is visible, showing the name "Muhammad Rusdiyanto" and the ID "2311102053 / IF-11-B".

Deskripsi:

Program di atas adalah sebuah program sequential search sederhana yang digunakan untuk menghitung banyaknya angka yang dicari dalam array. Dalam program ini, data sudah di-set atau ditentukan di dalam kode program, sehingga pengguna hanya bisa memasukkan angka yang akan dicari. Setelah program menampilkan semua data array dan mendapatkan input dari pengguna, program akan looping dari awal sampai akhir array untuk mencari angka yang dicari sembari menghitung banyaknya angka tersebut. Setelah looping selesai, program akan menampilkan pesan berupa banyaknya angka yang dicari dalam data array.

D. Kesimpulan

Sequential Search adalah salah satu algoritma pencarian data yang biasa digunakan untuk data yang berpola acak atau belum terurut. Algoritma ini akan mencari data sesuai kata kunci yang diberikan mulai dari elemen awal pada array hingga elemen akhir array. Proses pencarian dimulai dari indeks pertama hingga indeks terakhir, dan berhenti apabila data ditemukan. Jika hingga akhir array data masih juga tidak ditemukan, maka proses pencarian tetap akan dihentikan. Sedangkan Binary Search membagi array menjadi dua bagian dan mencari data dengan membandingkan kata kunci dengan elemen tengah array. Algoritma ini memerlukan data yang sudah terurut dengan baik, baik secara ascending maupun descending. Jadi, metode binary search membutuhkan metode sorting.

E. Referensi

Asisten Praktikum, “Modul 8 Algoritma Searching”

Annisa, 2023. Algoritma Sequential Search: Pengertian, Fungsi dan Cara Kerjanya. FIKTI UMSU. Tersedia di : <https://fikti.umsu.ac.id/algoritma-sequential-search-pengertian-fungsi-dan-cara-kerjanya/> [Diakses 29 Mei 2024].

Wafa, M., 2021. Sequential Search - Algoritma Pencarian. MikirinKode. Tersedia di : <https://mikirinkode.com/sequential-search/> [Diakses 29 Mei 2024].

Anshu, 2021. Binary Search and Its Powerful Applications. Medium. Tersedia di : <https://medium.com/@imanshu822/binary-search-and-its-powerful-applications-39ae7d7bca69> [Diakses 29 Mei 2024].