

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VIII
SEARCHING**



Disusun Oleh :

Anisah Syifa Mustika Riyanto
2311102080

Dosen

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

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

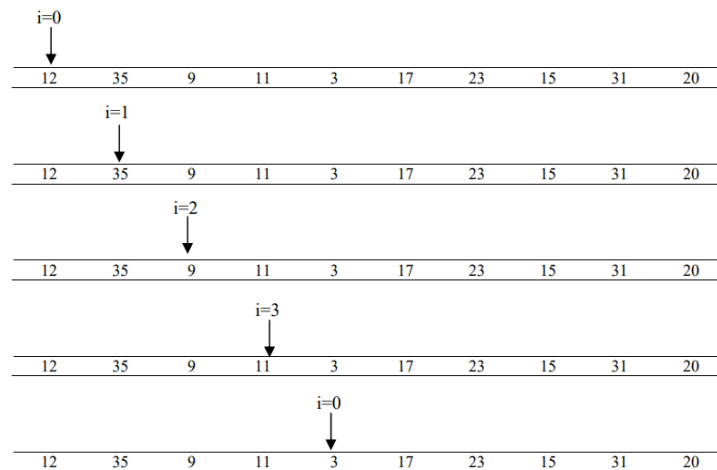
A. Dasar Teori

Algoritma *searching* merupakan algoritma yang menerima sebuah argumen kunci dan dengan langkah-langkah tertentu akan mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (successful) atau tidak ditemukan (unsuccessful). Ada dua macam teknik pencarian yaitu pencarian sekuensial (sequential search) dan pencarian biner (binary search). Perbedaan dari dua teknik ini terletak pada keadaan data. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak teratur. Sebaliknya, pencarian biner digunakan pada data yang sudah dalam keadaan urut.

1. Pencarian Berurutan (Sequential Search)

Algoritma pencarian dapat dijelaskan sebagai berikut : pencarian dimulai dari data paling awal, kemudian ditelusuri dengan menaikkan indeks data, apabila data sama dengan kunci pencarian dihentikan dan diberikan nilai pengembalian true, apabila sampai indeks terakhir data tidak ditemukan maka diberikan nilai pengembalian false. Ilustrasi dari algoritma pencarian biner adalah sebagai berikut :

Kunci=3



Data[4]=3 sama dengan kunci=3 maka data ditemukan dan diberikan nilai pengembalian i (posisi) dan proses dihentikan. Apabila data tidak ditemukan, maka fungsi akan mengembalikan nilai -1. Contoh menggunakan Sequential Search:

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

using namespace std;

int main()
{
    int a[11] = {12, 17, 10, 5, 15, 25, 11, 7, 25, 16, 19};
    int i, n, flag;

    cout << " ===== Contoh Sequential Search =====" << endl <<
    endl;
    cout << " Masukkan Bilangan Integer : ";
    cin >> n;
```

```

    cout << endl;

    for (i=0; i<11; i++){
        cout << setw (5) << a[i];
    }
    cout << endl;

    i = 0;
    flag = 0;
    while (i<=10){
        if (a[i] == n){
            flag++;
        }
        i++;
    }
    if (flag > 0){
        cout << "\n Bilangan Ditemukan ";
        cout << "\n Terdapat " << flag << " Bilangan yang Sama";
    } else {
        cout << "\n Bilangan Tidak Ditemukan ";
    }

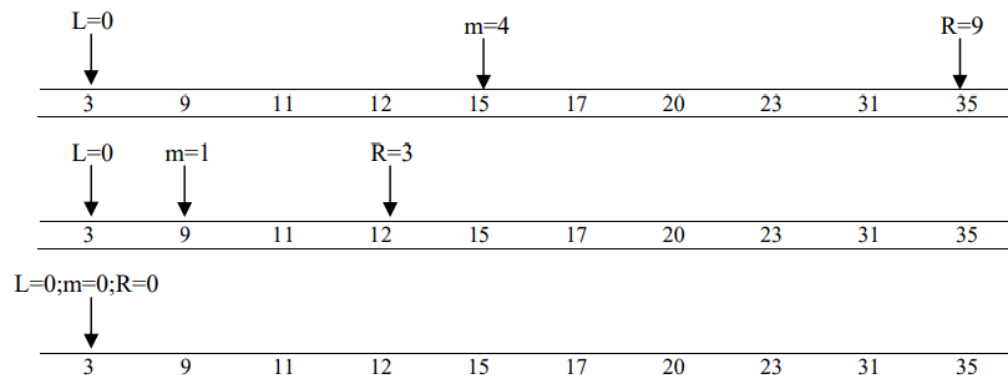
    return 0;
}

```

2. Pencarian Biner (Binary Search)

Salah satu syarat agar pencarian biner dapat dilakukan adalah data sudah dalam keadaan urut. Dengan kata lain, apabila data belum dalam keadaan urut, pencarian biner tidak dapat dilakukan. Ilustrasi dari algoritma pencarian biner adalah sebagai berikut :

Kunci=3



Data[1]=3 sama dengan kunci=3 maka data ditemukan dan diberikan nilai pengembalian i (posisi) dan proses dihentikan. Apabila data tidak ditemukan, maka fungsi akan mengembalikan nilai -1. Contoh program menggunakan binary search:

```

#include <iostream>
using namespace std;
int binary_search(int a[], int l, int r, int key) {
    while (l <= r) {
        int m = l + (r - l) / 2;
        if (key == a[m])
            return m;
        else if (key < a[m])
            r = m - 1;
        else
            l = m + 1;
    }
    return -1;
}

int *bubble_sort(int a[],int n){
    int param;
    int y = n-2;
    while(y >= 0){
        int index = 0;
        while(index <= y){
            if(a[index] > a[index+1]){
                param = a[index];
                a[index] = a[index+1];
                a[index+1] = param;
            }
            index++;
        }
        y--;
    }
    return a;
}

int main(int argc, char const* argv[]) {
    int n, key;
    cout << "Masukan banyak array: ";
    cin >> n;
    int* a = new int[n];
    for (int i = 0; i < n; i++) {
        cout<<endl<<"Masukan value array ke "<<i<<" :";
        cin >> a[i];
    }
    cout <<endl<< "Masukan angka yang ingin dicari : ";
    cin >> key;

    a = bubble_sort(a,n);
    int res = binary_search(a, 0, n - 1, key);
    if (res != -1)
        std::cout << key << "ditemukan " << endl;
    else
        std::cout << key << "tidak ditemukan" << endl;
    return 0;
}

```

B. Guided

Guided 1

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

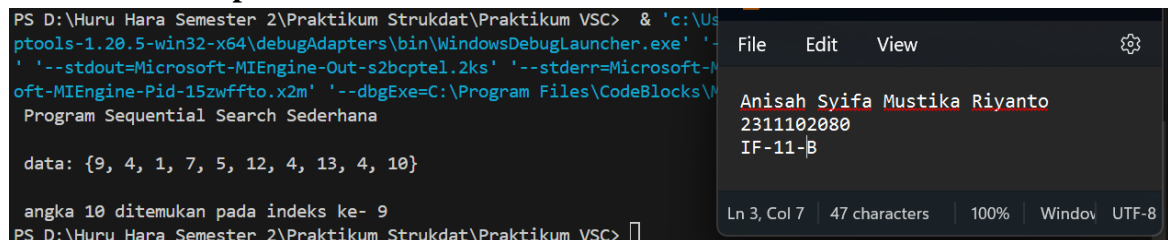
Source code:

```
#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 << " 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 << "\n angka " << cari << " tidak dapat ditemukan pada
        data." << endl;
    }
    return 0;
}
```

Screenshots Output:



```
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Us
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-
' '--stdout=Microsoft-MIEngine-Out-s2bcptel.2ks' '--stderr=Microsoft-M
oft-MIEngine-Pid-15zwffto.x2m' '--dbgExe=C:\Program Files\CodeBlocks\
Program Sequential Search Sederhana

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

angka 10 ditemukan pada indeks ke- 9
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> |
```

File Edit View

Anisah Syifa Mustika Riyanto
2311102080
IF-11-B

Ln 3, Col 7 | 47 characters | 100% | Window UTF-8

Deskripsi:

Program ini menggunakan pencarian sequential untuk mencari angka dalam array. Array berukuran sepuluh yang berisi sejumlah angka dan sebuah variabel "cari" yang menyimpan angka yang akan dicari, yaitu sepuluh. Untuk mengetahui apakah angka yang dicari ditemukan, fungsi boolean, juga dikenal sebagai "bool found", diaktifkan ke "false". Kemudian algoritma pencarian berurutan melakukan iterasi melalui array untuk melihat apakah setiap elemen sesuai dengan angka yang dicari. Jika ditemukan, bool diubah menjadi "benar" dan "break" digunakan untuk menghentikan iterasi. Program mencetak hasil pencarian setelah pencarian selesai dan menampilkan pesan yang menunjukkan apakah angka tersebut ditemukan dalam array dan pada indeks mana. Jika angka tersebut tidak ditemukan, program menampilkan pesan "tidak 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 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;
        }
    }
}
```

```

        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 << " 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 D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Users\hp151\.vscode\extensions\ms-vscode.cp
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-afvun13q.f5s
' '--stdout=Microsoft-MIEngine-Out-zk1denns.vrr' '--stderr=Microsoft-MIEngine-Error-yrfppou1.zux' '--pid=Micros
oft-MIEngine-Pid-bnrjji25.gxv' '--dbgExe=C:\Program Files\CodeBlocks\W
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

```

Deskripsi:

Selection sort digunakan dalam program untuk mengurutkan array dan kemudian melakukan pencarian biner untuk menemukan elemen tertentu. Program dimulai dengan menciptakan array data berukuran tujuh angka dan meminta pengguna memasukkan angka yang mereka cari. Untuk mengurutkan array, fungsi "selection_sort" menukar elemen terkecil yang ditemukan di bagian luar array dengan elemen saat ini. Fungsi pencarian biner melakukan pencarian biner pada array yang telah diurutkan setelah pengurutan. Jika angka yang dicari ditemukan, fungsi akan mencetak indeksinya atau pesan bahwa data tidak ditemukan. Selain menampilkan hasil pencarian, program menampilkan array sebelum dan sesudah pengurutan dan menunggu input sebelum mengakhiri eksekusi.

C. Unguided

Unguided 1

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

Source code:

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

#include <iostream>

using namespace std;
void selectionSort(string &huruf, int n)
{
    int a, b, min;
    for (a = 0; a < n - 1; a++)
    {
        min = a;
        for (b = a + 1; b < n; b++)
            if (huruf[b] < huruf[min])
                min = b;
        char temp = huruf[a];
        huruf[a] = huruf[min];
        huruf[min] = temp;
    }
}
int binarySearch(string huruf, int kiri, int kanan, char target)
{
    while (kiri <= kanan)
    {
        int mid = kiri + (kanan - kiri) / 2;
        if (huruf[mid] == target)
            return mid;
        if (huruf[mid] < target)
            kiri = mid + 1;
        else
            kanan = mid - 1;
    }
    return -1;
}
int main()
```



```

{
    string kalimat;
    char input;
    cout << "-----" << endl;
    cout << "          Pencarian Huruf pada Kalimat          " << endl;
    cout << "-----" << endl;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);
    cout << "Masukkan huruf yang ingin dicari: ";
    cin >> input;
    cout << endl;
    selectionSort(kalimat, kalimat.size());
    int result = binarySearch(kalimat, 0, kalimat.size() - 1, input);
    if (result == -1)
    {
        cout << "Huruf yang Anda cari tidak ditemukan!" << endl;
    }
    else
    {
        cout << "Huruf setelah diurutkan: " << kalimat << endl;
        cout << "Huruf ditemukan pada indeks ke- " << result << endl;
    }
    return 0;
}

```

Screenshoot Output:

```

PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Users\hp151\.vscode\extensions\ms-vscode.cp
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '
' '--stdout=Microsoft-MIEngine-Out-4mhktp2k.rfh' '--stderr=Microsoft-M
oft-MIEngine-Pid-n2vdbktr.hlk' '--dbgExe=C:\Program Files\CodeBlocks\M
-----
          Pencarian Huruf pada Kalimat
-----
Masukkan kalimat: anisahsyifa
Masukkan huruf yang ingin dicari: h

Huruf setelah diurutkan: aaafhiinssy
Huruf ditemukan pada indeks ke- 4
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC>

```

Deskripsi:

Dalam program ini, pengguna diminta untuk memasukkan kalimat dan huruf yang ingin dicari, lalu algoritma pengurutan pilihan mengurutkan huruf-huruf dalam kalimat tersebut. Setelah pengurutan, algoritma pencarian binar digunakan untuk mencari huruf yang telah dimasukkan pengguna dalam kalimat yang telah diurutkan. Program menampilkan kalimat dengan indeks dan kalimat yang sudah diurutkan jika huruf ditemukan; jika tidak ditemukan, program menampilkan pesan bahwa huruf tersebut tidak ada dalam kalimat.

Unguided 2

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

Source code:

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

#include <iostream>
#include <string>

using namespace std;

int main() {
    string kalimat;
    int jumlah = 0;

    cout << "-----" << endl;
    cout << "        Program Menghitung Huruf Vokal        " << endl;
    cout << "-----" << endl;

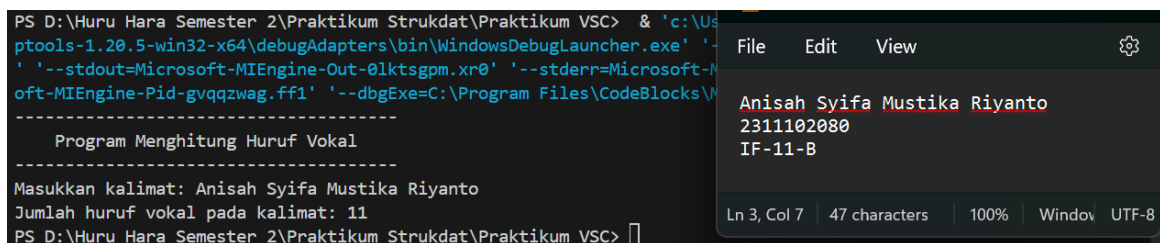
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);

    for (int i = 0; i < kalimat.length(); i++) {
        char a = kalimat[i];
        if (a == 'a' || a == 'i' || a == 'u' || a == 'e' || a == 'o'
||
            a == 'A' || a == 'I' || a == 'U' || a == 'E' || a == 'O')
        {
            jumlah++;
        }
    }

    cout << "Jumlah huruf vokal pada kalimat: " << jumlah << endl;

    return 0;
}
```

Screenshots Output:



```
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Us
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-
' '--stdout=Microsoft-MIEngine-Out-0lktsgpm.xr0' '--stderr=Microsoft-M
oft-MIEngine-Pid-gvqqzwag.fff1' '--dbgExe=C:\Program Files\CodeBlocks\
-----
        Program Menghitung Huruf Vokal
-----
Masukkan kalimat: Anisah Syifa Mustika Riyanto
Jumlah huruf vokal pada kalimat: 11
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> |
```

File Edit View

Anisah Syifa Mustika Riyanto
2311102080
IF-11-B

Ln 3, Col 7 | 47 characters | 100% | Window UTF-8

Deskripsi:

Program ini menghitung banyak huruf vokal dalam kalimat yang dimasukkan pengguna. Program menggunakan "getline(cin, kalimat)" untuk menerima input kalimat. Kemudian, program memeriksa setiap karakter dalam kalimat untuk memastikan apakah itu adalah huruf vokal ('a', 'i', 'u', 'e', 'o', dan versi huruf besarnya). Jika iya, variabel "jumlah" akan ditambahkan 1. Program mencetak jumlah huruf vokal yang ada dalam kalimat.

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:

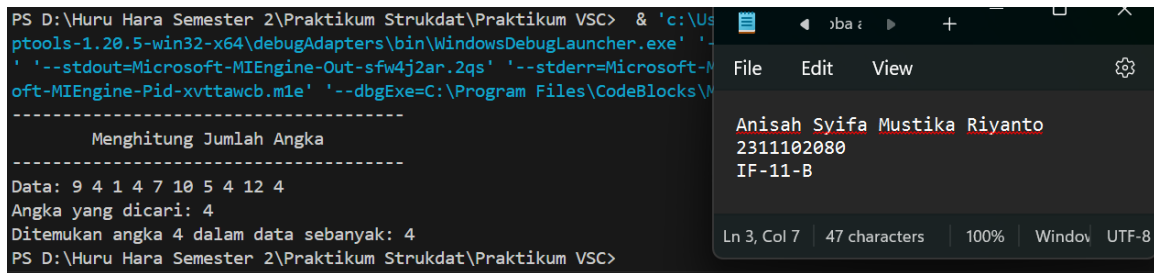
```
// Diketahui data = 9, 4, 1, 4, 7, 10, 5, 4, 12, 4.
// Hitunglah berapa banyak angka 4 dengan menggunakan algoritma
Sequential Search!

#include <iostream>

using namespace std;
int hitungAngka(const int array[], int size, int target)
{
    int jumlah = 0;
    for (int i = 0; i < size; i++)
    {
        if (array[i] == target)
        {
            jumlah++;
        }
    }
    return jumlah;
}

int main()
{
    const int size = 10;
    int array[size] = {9, 4, 1, 4, 7, 10, 5, 4, 12, 4};
    int target = 4;
    int jumlah = hitungAngka(array, size, target);
    cout << "-----" << endl;
    cout << "          Menghitung Jumlah Angka          " << endl;
    cout << "-----" << endl;
    cout << "Data: ";
    for (int element : array)
    {
        cout << element << " ";
    }
    cout << "\nAngka yang dicari: " << target << endl;
    cout << "Ditemukan angka " << target << " dalam data sebanyak: "
    << jumlah << endl;
    return 0;
}
```

Screenshots Output:



```
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Us
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-
' '--stdout=Microsoft-MIEngine-Out-sfw4j2ar.2qs' '--stderr=Microsoft-M
oft-MIEngine-Pid-xvttawcb.m1e' '--dbgExe=C:\Program Files\CodeBlocks\N
-----
Menghitung Jumlah Angka
-----
Data: 9 4 1 4 7 10 5 4 12 4
Angka yang dicari: 4
Ditemukan angka 4 dalam data sebanyak: 4
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC>
```

Anisah Syifa Mustika Riyanto
2311102080
IF-11-B

Ln 3, Col 7 | 47 characters | 100% | Window UTF-8

Deskripsi:

Algoritma sequential search digunakan untuk menghitung jumlah angka 4 dalam sebuah array. Dimulai dengan mendefinisikan sebuah fungsi "hitung angka" yang menerima array, ukurannya, dan target angka. Fungsi ini mengiterasi setiap elemen dalam array dan menambah jumlah jika elemen tersebut sama dengan target. Selanjutnya, fungsi "main" menginisialisasi sebuah array berisi sepuluh angka, dan target angka yang akan dihitung ditetapkan ke 4. Jumlah angka 4 dalam array adalah hasil akhir dari program ini.

D. Kesimpulan

Berikut adalah kesimpulan mengenai materi searching:

1. Sequential search adalah metode pencarian yang sederhana di mana elemen dicari dengan memeriksa setiap elemen dalam array satu per satu mulai dari awal hingga akhir. Kelebihannya adalah implementasinya yang mudah dan tidak memerlukan data terurut.
2. Binary Search adalah metode pencarian yang lebih efisien yang digunakan pada array yang sudah diurutkan. Pencarian dilakukan dengan membagi array menjadi dua bagian secara berulang-ulang.
3. Sequential search cocok untuk dataset kecil atau dataset yang tidak diurutkan di mana kecepatan pencarian tidak kritis.
4. Binary search sangat berguna dalam aplikasi yang memerlukan pencarian cepat dalam dataset yang besar dan sudah diurutkan, seperti pencarian dalam database atau pengindeksan.
5. Memilih antara sequential search dan binary search bergantung pada ukuran dan sifat dataset serta kebutuhan kinerja aplikasi.

E. Referensi

Anita Sindar, R. M. S. (2019). *Struktur Data Dan Algoritma Dengan C++* (Vol. 1). CV. AA. RIZKY.

Goodrich, M. T., Tamassia, R., & Mount, D. M. (2021). *Data structures and algorithms in C++*. John Wiley & Sons.

Kruse, R. L., & Ryba, A. J. (2019). *Data structures and program design in C++*. Prentice Hall.

Langsam, Y., Augenstein, M. J., & Tenenbaum, A. M. (2020). *Data Structures using C and C++*. Prentice Hall Press.

Sedgewick, R. (2022). *Algorithms in c++, parts 1-4: fundamentals, data structure, sorting, searching*. Pearson Education.