

**LAPORAN PRAKTIKUM STRUKTUR DATA  
DAN ALGORITMA MODUL 8  
“Searching”**



**DISUSUN OLEH:**  
**Farrell Edric Kelvianto**  
**2311102079**  
**S1 IF-11-B**

**DOSEN:**

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

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

## A. Dasar Teori

### a. Pengertian Searching

Searching merupakan pencarian yang berfungsi untuk menemukan suatu nilai tertentu pada sekumpulan data. Didalam searching ini ada banyak sekali jenisnya ada Binary Search, Sequential Search, Fibonacci Search, Interpolation search dan lain – lain.

### b. Jenis Searching (Pembahasan Dimodul)

#### a. Binary Search

Binary search ini merupakan metode pencarian dengan cara melakukan perbandingan pada kumpulan datanya data diambil dari posisi 1 sampai posisi terakhir yaitu N. kemudian mencari posisi data tengah dengan cara posisi awal + posisi akhir lalu dibagi dengan 2. Setelah itu data yang dicari dibandingkan dengan data yang ditengah apakah sama atau lebih kecil atau lebih besar jika lebih besar maka proses dicari dengan posisi awal adalah posisi tengah + 1, jika lebih kecil maka proses pencarian dicari dengan posisi akhir adlaah posisi tengah – 1. Jika data sama maka data telah ditemukan.

#### b. Sequential Search

Teknik pencarian data dari array yang paling mudah dengan cara sequential search, dan cara kerja dari sequential ini dibaca 1 demi 1,diurutkan dari index terkecil sampai index terbesar. Semisal diberikan sebuah data array memiliki  $A[4] = \{10,9,7,5\}$

Misalkan di dalam array ini yang dicari adalah angka 5 maka prosesnya demikian, pencarian dimulai dari angka 10 kemudian dicocokkan apakah angka yang akan dicari ini sama atau tidak sama, jika tidak sama maka lanjut lagi ke data berikutnya lalu dibandingkan kembali dengan index selanjutnya yaitu 9 ini datanya sama atau tidak jika tidak maka lanjut lagi ke index berikutnya yaitu 7 apakah ini sama dengan 5 jika tidak lagi maka dibandingkan lagi ke index berikutnya yaitu 5 apakah 5 sama dengan 5 jika iya maka dia ditemukan di index ke – 3 dari 0.

- Sebagai berikut adalah gambaran dari suatu Binary Search dengan Sequential Search yang berisikan sebuah data:



## Guided

### Guided 1

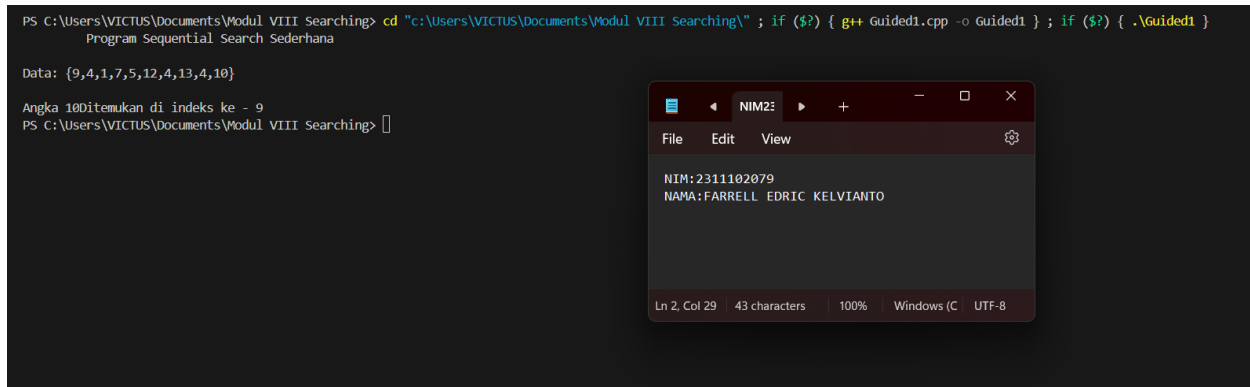
```
#include <iostream>

using namespace std;

int main()
{
    int n = 10;
    int data[10] = {9, 4, 1, 7, 5, 12, 4, 13, 4, 10};
    int cari = 10;
    bool ketemu = false;
    int i;

    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 << "\nAngka " << cari << "Ditemukan di indeks ke - " << i
    << endl;
    }
    else
    {
        cout << cari << " tidak dapat ditemukan pada data." << endl;
    }
}
```

## Screenshot Output



```
PS C:\Users\VICTUS\Documents\Modul VIII Searching> cd "c:\Users\VICTUS\Documents\Modul VIII Searching\" ; if ($?) { g++ Guided1.cpp -o Guided1 } ; if ($?) { .\Guided1 }  
Program Sequential Search Sederhana  
  
Data: {9,4,1,7,5,12,4,13,4,10}  
  
Angka 10 ditemukan di indeks ke - 9  
PS C:\Users\VICTUS\Documents\Modul VIII Searching>
```

NIM:2311102079  
NAMA:FARRELL EDRIC KELVIAN TO

## Deskripsi Program

Program ini merupakan program sequential yang sederhana untuk mencari sebuah data. Di guided 1 ini membuat array dan dideklarasikan data [10] = {9,4,1,7,5,12,4,13,4,10} lalu yang dicari adalah angka 10. Lalu membuat sebuah perulangan untuk mengakses angka dari data yang diberikan tadi dan diberikan jika data = yang dicari maka ketemu dan program telah dibreak atau selesai dan data telah ditemukan.

## Guided 2

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

```

    }
    temp = data_array[i];
    data_array[i] = data_array[min];
    data_array[min] = temp;
}
}

void binary_search()
{
    int awal, akhir, tengah, b_flag = 0;
    awal = 0;
    akhir = 6;
    while (b_flag == 0 && awal <= akhir)
    {
        tengah = (awal + akhir) / 2;
        if (data_array[tengah] == cari)
        {
            b_flag = 1;
            break;
        }
        else if (data_array[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 : ";
    for (int x = 0; x < 7; x++)
        cout << setw(3) << data_array[x];
    cout << endl;
    cout << "\n Masukkan data yang ingin Anda cari: ";
    cin >> cari;
    cout << "\n Data diurutkan: ";

    selection_sort();
    for (int x = 0; x < 7; x++)
        cout << setw(3) << data_array[x];
}

```

```

    cout << endl;
    binary_search();
    _getche();
    return EXIT_SUCCESS;
}

```

## Screenshot Output

```

PS C:\Users\VICTUS\Documents\Modul VIII Searching> cd "c:\Users\VICTUS\Documents\Modul VIII Searching\" ; if ($?) { g++ Guided2.cpp -o Guided2 } ; if ($?) { .\Guided2
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

```

## Deskripsi Program

Didalam program ini ada dua operasi untuk mencari dan untuk mengurutkan sebuah data. Pencariannya ini menggunakan binary search dan untuk mengurutkannya ini menggunakan selection sort. Didalam program ini diberikan sebuah data array dengan 7 index. Untuk selection Sort ini mengurutkan data dengan cara index dimulai dari 0 dan sampai dengan jumlah index yang diberikan lalu membuat perulangan lagi untuk membuat perbandingan dengan index selanjutnya jika salah satu lebih kecil maka di geser ke sebelah kiri sampai program ini selesai mengurutkannya. Lalu disini ada operasi binary search. Dideklarasikan sebuah awal, akhir, tengah, b\_flag ini difungsikan untuk menguji benar atau salah. Di perulangan pertama terdapat sebuah perbandingan apakah tengah ini sama saja dengan yang dicari jika tidak maka lanjut ke iterasi ke 2 yang untuk mencari dengan cara data tengah ini apakah kurang dari dengan yang dicari jika iya maka akan mencari sampai ke titik tengah dan begitu juga sebaliknya jika besar maka dimulai dari yang terbesar sampai titik tengah tadi untuk melakukan pencariannya. Didalam fungsi utamanya ini seperti biasa panggil dari fungsi yang telah dibuat lalu cari data yang diinginkan karena ini menggunakan pengurutan maka diurutkan dahulu lalu panggil fungsi binary searchnya.

**B. Tugas**  
**Unguided 1**

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

void selectionSort(string& str) {
    int n = str.length();
    for (int i = 0; i < n - 1; ++i) {
        int minIndex = i;
        for (int j = i + 1; j < n; ++j) {
            if (str[j] < str[minIndex]) {
                minIndex = j;
            }
        }
        if (minIndex != i) {
            char temp = str[i];
            str[i] = str[minIndex];
            str[minIndex] = temp;
        }
    }
}

int binarySearch(const string& str, char target) {
    int left = 0;
    int right = str.length() - 1;

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

        if (str[mid] == target) {
            return mid;
        }
    }
}
```

```

        } else if (str[mid] < target) {
            left = mid + 1;
        } else {
            right = mid - 1;
        }
    }

    return -1;
}

bool isVowel(char ch) {
    ch = tolower(ch);
    return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u');
}

int countVowels(const string& str) {
    int count = 0;
    for (char ch : str) {
        if (isVowel(ch)) {
            count++;
        }
    }
    return count;
}

int main() {
    string kalimat;
    char target;

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

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

    selectionSort(kalimat);

    int result = binarySearch(kalimat, target);

    if (result != -1) {
        cout << "Huruf '" << target << "' ditemukan pada indeks: " << result << endl;
    } else {
        cout << "Huruf '" << target << "' tidak ditemukan dalam kalimat." << endl;
    }
}

```



```

int vowelCount = countVowels(kalimat);
cout << "Jumlah huruf vokal dalam kalimat tersebut adalah: " << vowelCount << endl;

return 0;
}

```

## Screenshot Output

Soal no 1. Mencari huruf dan soal no. 2 menghitung jumlah vokal dari kalimat.

```

PS C:\Users\VICTUS\Documents\Modul VIII Searching> cd "c:\Users\VICTUS\Documents\Modul VIII Searching\" ; if ($?) { g++ Unguided1.cpp -o Unguided1 } ; if ($?) { .\Unguided1 }
Masukkan sebuah kalimat: Farrell Edric Kelvianto
Masukkan huruf yang ingin dicari: e
Huruf 'e' ditemukan pada indeks: 9
Jumlah huruf vokal dalam kalimat tersebut adalah: 8
PS C:\Users\VICTUS\Documents\Modul VIII Searching>

```

```

NIM:2311102079
NAMA:FARRELL EDRIC KELVANTO

```

Ln 2, Col 29 | 43 characters | 100% | Windows (C | UTF-8

## Deskripsi Program

Cara kerja program ini adalah dimulai dari prosedur selection sort dideklarasikan oleh alamat string. Lalu membuat perbandingan dan bertukar posisi dibagian percabangan setelah perulangan. Untuk binary searchnya ini menggunakan berapa banyak huruf didalam satu kalimat lalu seperti biasa namanya binary search dibagi menjadi dua bagian, bagian kiri, kanan dan tengah jika ditengah sudah ditemukan adalah yang dicari maka langsung return dia adalah yang dicari, jika tidak dia mencari melakukan perbandingan kembali jika lebih besar maka di ruas kanan, jika lebih kecil maka perbandingan di ruas kiri. Dan ketika huruf telah ditemukan maka akan return. Dan untuk operasi programnya di bagian fungsi utamanya ini diurutkan terlebih dahulu, setelah diurutkan maka urutan sebelumnya yang telah dideklarasikan menjadi deklarasi baru dan untuk pencariannya menjadi sesuai dengan setelah diurutkan. Dan untuk tambahan lagi di unguided 2 ini menghitung sebuah huruf vokal dengan mendeklarasikan boolean isChar untuk menyatakan jika ketemu huruf vocal maka dia akan dipanggil di dalam countChar untuk melakukan pencarian vocal dalam kalimat dan dihitung.

## Unguided 2

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

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

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

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

    return 0;
}
```

## Screenshot Output

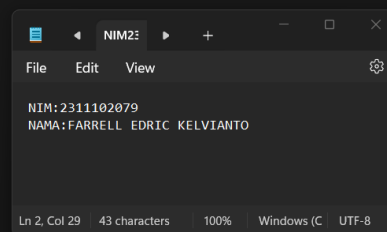
```
PS C:\Users\VICTUS\Documents\Modul VIII Searching> cd "c:\Users\VICTUS\Documents\Modul VIII Searching\" ; if ($?) { g++ Unguided2.cpp -o Unguided2 } ; if ($?) { .\Unguided2 }
```

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\VICTUS\Documents\Modul VIII Searching> █



## Deskripsi Program

Program ini sangat simpel disini diminta untuk menampilkan berapa banyak angka 4 dalam data hanya mengubah sedikit dari guided 1 dan ditambahkan jumlahnya untuk menghitung jumlah dari angka yang sama didalam datanya itu sendiri.

**Kesimpulan**

Kesimpulannya adalah didalam searching ini seperti pada pencarian yang telah dipelajari didalam algoritma pemrograman. Disini sequential yang berarti mencari satu per satu untuk mencari didalam suatu data. Dan untuk Binary Search ini mencari dengan cara membagi 2 terlebih dahulu, lalu menyamakan dengan yang dicari dibagian titik tengah apakah sama atau tidak jika sama return jika tidak lanjut ke iterasi selanjutnya sampai ketemu.

**Refrensi**

Asisten Praktikum. "Modul 8 Searching ". Learning Management System. 2024