

SEARCHING



Jl. DI Panjaitan 128 Purwokerto

* Untuk kalangan sendiri

Praktikum 9
Materi : Searching
Waktu : 100 menit

Tujuan

1. Mahasiswa dapat melakukan perancangan aplikasi menggunakan struktur Searching (Pencarian)
2. Mahasiswa mampu melakukan analisis pada algoritma Searching yang dibuat
3. Mahasiswa mampu mengimplementasikan algoritma Searching pada sebuah aplikasi secara tepat dan efisien
4. Mahasiswa mampu menjelaskan mengenai algoritma Searching
5. Mahasiswa mampu membuat dan mendeklarasikan struktur algoritma Searching
6. Mahasiswa mampu menerapkan dan mengimplementasikan algoritma Searching

Dasar Teori

Searching merupakan proses pencarian suatu data / nilai di dalam sebuah array dengan cara mengecek satu persatu pada setiap index baris atau setiap index kolomnya. Hampir sama dengan sorting, searching menggunakan teknik perulangan untuk melakukan pencarian data.

Terdapat 2 metode pada algoritma Searching, yaitu:

a. Sequential Search

Sequential search merupakan teknik pencarian data yang paling mudah, dimana data dalam array dibaca 1 demi satu, diurutkan dari index terkecil ke index terbesar, maupun sebaliknya

Array int
A[5] = {56, 98, 25, 100, 72}

0	1	2	3	4	→	indeks
56	98	25	100	72	→	data

Misalkan, dari data diatas angka yang akan dicari adalah angka 100 dalam array A, maka proses yang akan terjadi pada proses pencarian adalah sebagai berikut.

Pencarian dimulai pada index ke-0 yaitu angka 56, kemudian dicocokkan dengan angka yang akan dicari, jika tidak sama makapencarian akan dilanjutkan ke index selanjutnya.

Pada index ke-1, yaitu angka 98, juga bukan angka yang dicari, maka pencarian juga akan dilanjutkan pada index selanjutnya.

Pada index ke-2, yaitu angka 25, juga bukan angka yang dicari, maka pencarian juga akan dilanjutkan pada index selanjutnya

Pada index ke-3, yaitu angka 100, ternyata angka 100 merupakan angka yang dicari. Pencarian angka telah ditemukan, maka pencarian akan dihentikan dan keluar dari looping pencarian.

b. Binary Search

Metode pencarian yang kedua adalah binary search, pada metode pencarian ini, data harus diurutkan terlebih dahulu. Pada metode pencarian ini, data dibagi menjadi dua bagian (secara logika), untuk setiap tahap pencarian. Algoritma binary search :

1. Data diambil dari posisi 1 sampai posisi akhir N
2. Kemudian cari posisi data tengah dengan rumus: **(posisi awal + posisi akhir) / 2**
3. Kemudian data yang dicari dibandingkan dengan data yang di tengah, apakah sama atau lebih kecil, atau lebih besar?
4. Jika data terurut secara DESCENDING (Besar ke Kecil), Jika data di indeks lebih besar MAKA nilai awal adalah **posisi tengah + 1** atau Jika data di indeks lebih kecil ,MAKA nilai akhir adalah **posisi tengah - 1**
5. Jika data terurut secara ASCENDING (Kecil ke Besar), Jika data di indeks lebih besar MAKA nilai akhir adalah **posisi tengah - 1** atau Jika data di indeks lebih kecil ,MAKA nilai awal adalah **posisi tengah + 1**
6. Jika data sama, berarti ketemu.

GUIDED

1. Buatlah sebuah project dengan nama GD1_Kelas_NIM Dengan menggunakan sequential search sederhana lakukan pencarian untuk membuat, menampilkan sebuah program searching

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

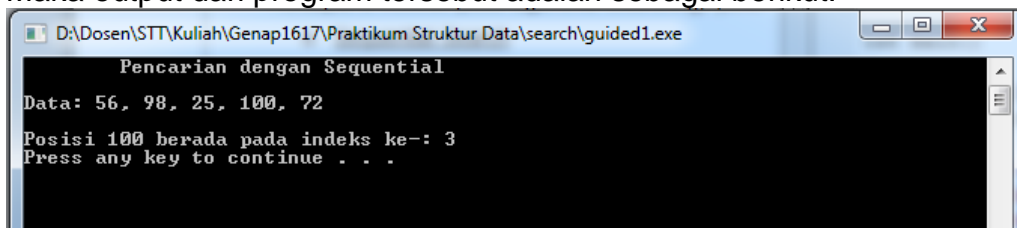
int cari(int data[], int n, int k)
{
    int posisi, i, ketemu;
    if(n<=0)
        posisi = -1;
    else
    {
        ketemu = 0;
        i=1;
        while((i<=n-1)&&(!ketemu))
            if(data[i]==k)
            {
                posisi = i;
                ketemu = 1;
            }
            else
                i++;
        if(!ketemu)
            posisi = -1;
    }

    return posisi;
}

int main()
{
    int data[5]={56,98,25,100,72};
    int dicari = 100;

    printf("\tPencarian dengan Sequential\n\n");
    printf("Data: 56, 98, 25, 100, 72\n\n");
    printf("Posisi %d berada pada indeks ke-: %d \n", dicari, cari(data,5,dicari));
    system("PAUSE");
    return 0;
}
```

Maka output dari program tersebut adalah sebagai berikut:



The screenshot shows a Windows command prompt window titled "D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\search\guided1.exe". The output of the program is displayed as follows:

```
Pencarian dengan Sequential
Data: 56, 98, 25, 100, 72
Posisi 100 berada pada indeks ke-: 3
Press any key to continue . . .
```

2. Buatlah sebuah project dengan nama GD2_Kelas_NIM dengan Binary Search untuk melakukan pencarian data.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main(){
    const int Ar[10] = {1,2,3,4,5,6,7,8,9,10}; // untuk proses ascending
    int tar,i;

    for (int i=0; i<10; i++)
    {
        cout<<Ar[i]<<" ";
    }
    cout<<"\n\nmasukan data yang akan dicari : ";
    cin>>tar;

    int awal=0, akhir=10, tengah;

    while (awal <= akhir)
    {
        tengah = (awal + akhir)/2;
        if (tar > Ar[tengah] ) // untuk data descending ubah tanda > menjadi <
        {
            awal = tengah + 1;
        }
        else if (tar < Ar[tengah]) // untuk data descending ubah tanda < menjadi >
        {
            akhir= tengah - 1;
        }
        else {awal = akhir +1;
        }
    }
    if (tar == Ar[tengah])
    {
        cout<<" Data ditemukan, Index ke- "<<tengah+1<<endl;
    }
    else
    {
        cout<<"data tidak ditemukan "<<endl;
    }
    getch();
}
```

3. Buatlah sebuah project dengan nama GD3_Kelas_NIM Untuk membuat sebuah program searching sebagai berikut

```

#include<stdlib.h>
#include<stdio.h>
#include<iostream.h>
#include<string.h>

int pencarian(char st[], char m)
{
    int i, posisi, panjang;

    i=0;
    posisi = -1;
    panjang = strlen(st);
    while((i<panjang-1)&&posisi == -1)
    {
        if(st[i]==m)
            posisi = i;
            i++;
    }
    return posisi;
}

```

```

int main()
{
    printf("\t Pencarian dengan metode Binary Search\n\n");

    char kalimat[]="Sisilia Thya Safitri"; //tuliskan nama kalian masing-masing
    char dicari = 't';
    printf("\nPosisi %c dalam string %s berada pada indeks ke- [%d]\n", dicari, kalimat, pencarian(kalimat,dicari));
    system("PAUSE");
    return 0;
}

```

Maka output dari program tersebut adalah sebagai berikut:

```

Pencarian dengan metode Binary Search

Posisi t dalam string Sisilia Thya Safitri berada pada indeks ke- [17]
Press any key to continue . . .

```

Unguided

1. Buat sebuah project dengan nama UGD1_NIM_Kelas yang merupakan sebuah program untuk menampung banyak nya data yang ingin dicari oleh user. Kemudian tampilkan data pencarian tersebut ada pada indeks beberapa. Dan apabila tidak diketemukan, maka akan muncul peringatan bahwa data yang dicari tidak ditemukan.
Sehingga output pada program tersebut adalah sebagai berikut:

```
D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\search\ugd1.exe
PENCARIAN ANGKA
Masukkan banyak Angka : 4
Masukan Angka ke - 1 = 67
Masukan Angka ke - 2 = 43
Masukan Angka ke - 3 = 90
Masukan Angka ke - 4 = 35

Data yang ingin dicari : 43
Angka 43 ditemukan pada posisi ke 2_
```

```
D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\search\ugd1.exe
PENCARIAN ANGKA
Masukkan banyak Angka : 3
Masukan Angka ke - 1 = 55
Masukan Angka ke - 2 = 28
Masukan Angka ke - 3 = 11

Data yang ingin dicari : 100
Maaf data yang dicari tidak ada_
```

2. Buatlah sebuah project dengan nama UGD2_NIM_Kelas yang merupakan sebuah program yang digunakan untuk melakukan penampungan sebuah string yang di inputkan oleh user, kemudian lakukan replace terhadap character tertentu pada string tersebut. Sehingga tampilan pada program tersebut adalah sebagai berikut

```
D:\Dosen\STT\Kuliah\Genap1617\Praktikum Struktur Data\search\Untitled1.exe
Masukkan sebuah kalimat: lingkaran kecil
> Masukkan karakter yang dicari: i
> Karakter pengganti: a

Hasil replace adalah sebagai berikut: langkaran kecal
Press any key to continue . . .
```

Resume

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

S1 TEKNIK INFORMATIKA

Hari/Tanggal Praktikum :

Modul :

NIM :

Nama Praktikan :

Nama Asistant : 1.....

2.....

Nilai dan Parat :

Hasil Analisa Praktikum

A large, empty rectangular frame with a thin black border, occupying the majority of the page below the header. The frame is intended for a drawing or illustration related to the text above.