
**JURNAL PRAKTIKUM
(LAB. ACTIVITY)
STRUKTUR DATA
SI025**

**Materi 3:
SEARCHING
(PENCARIAN)**

Dosen:

**Agung Nugroho, M.Kom
Prof. Dr. Ema Utami, S.Si, M.Kom
Ikmah, M.Kom
Ninik Tri H, M.Kom
Lilis Dwi Farida, M.Kom
Windha Mega PD, M.Kom**

**S1 - SISTEM INFORMASI
UNIVERSITAS AMIKOM YOGYAKARTA
2018**

Searching(Pencarian)

Pendahuluan

A. Tujuan

Setelah praktikum ini, praktikan diharapkan dapat:

1. Memahami apa itu Searching
2. Menunjukkan beberapa Algoritma yang berbeda
3. Menerapkan Searching menggunakan bahasa C++

B. Peralatan

1. PC Desktop
2. Windows 7
3. MinGW

C. Teori

SEARCHING (PENCARIAN)

Dalam kehidupan sehari-hari sebenarnya kita sering melakukan pencarian data. Pencarian data sering juga disebut *table look-up* atau *storage and retrieval information* adalah suatu proses untuk mengumpulkan sejumlah informasi di dalam memori komputer dan kemudian mencari kembali informasi yang diperlukan secepat mungkin.

Algoritma pencarian (*searching algorithm*) adalah 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 beberapa pencarian yang akan kita uraikan disini:

- Pencarian Beruntun (Sekuensial Search)
- Pencarian Bagi dua (Binary Search)

I. **PENCARIAN BERURUTAN (SEKUENSIAL SEARCH)**

Pencarian berurutan sering disebut pencarian linear merupakan metode pencarian yang paling sederhana. Pencarian berurutan menggunakan prinsip sebagai berikut : data yang ada dibandingkan satu per satu secara berurutan dengan yang dicari sampai data tersebut ditemukan atau tidak ditemukan.

Pada dasarnya, pencarian ini hanya melakukan pengulangan dari 1 sampai dengan jumlah data. Pada setiap pengulangan, dibandingkan data ke-i dengan yang dicari. Apabila sama, berarti data telah ditemukan. Sebaliknya apabila sampai akhir pengulangan tidak ada data yang sama, berarti data tidak ada. Pada kasus yang paling buruk, untuk N elemen data harus dilakukan pencarian sebanyak N kali pula.

Berikut ini adalah contoh pencarian sekuensial:

<i>elemen</i>	<i>13</i>	<i>16</i>	<i>14</i>	<i>21</i>	<i>76</i>	<i>15</i>
<i>index</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>

Dari data diatas, larik L mempunyai 6 elemen. Pencarian akan dimulai dari indeks ke-0 yaitu posisi pertama. Misalkan elemen yang dicari: $X = 21$. Urutan elemen yang dibandingkan:

```
13 <> 21
16 <> 21
14 <> 21
21 = 21 (ditemukan idx = 4)
```

Misalkan elemen yang dicari: $X = 13$. Urutan elemen yang dibandingkan:

```
13 = 13 (ditemukan idx = 1)
```

Indeks array yang dikembalikan adalah 0.



D.I Praktikum Sequential Search

Kode.5.1. contoh sequential search

```
#include <iostream>
using namespace std;
int main() {
    int jml;
    int i ;
    int cari;
    int arr[50];
    int tanda=-1;
    cout<<"Masukkan banyaknya bilangan = " ;
    cin>>jml;
    for(i=0;i<jml;i++)
    {
        cout<<"Masukkan bilangan ke-"<<i+1<<" : ";
        cin>>arr[i];
    }
    cout<<"Isi dari array : "<<endl;
    for (i=0;i<jml;i++)
        cout<<" "<<arr[i];
    cout<<"\n\nMasukkan data yang dicari: ";
    cin>>cari;
    for (i=0;i<jml;i++)
    {
        if (cari ==arr[i])
        {
            tanda=i;break;
        }
    }
    if (tanda!=-1)
        cout<<"\n\n Data ditemukan" ;
    else
        cout<<"\n\n Data tidak ditemukan";
    return 0;
}
```

Kode 5.2.

```
#include <stdio.h>
#include <conio.h>
#include <iostream>
using namespace std;

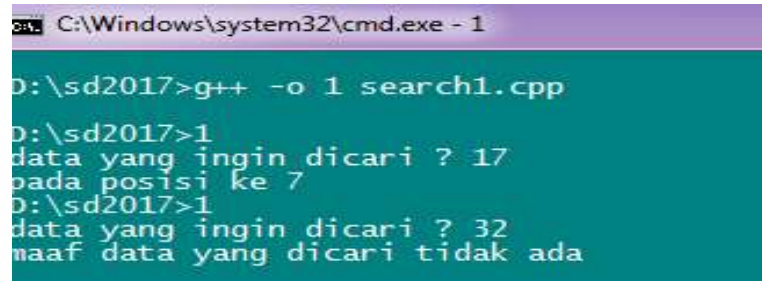
main()
{
    int larik[10]={15,12,3,7,10,6,17,11,9,14}, i,n, x, posisi;

    cout<<"data yang ingin dicari ? ";cin>>x;
    i=0;
    posisi=0;

    while(i<9 && larik[i]!=x)
    {
        i++;
    }
    if (larik[i]!=x)
        cout<<"maaf data yang dicari tidak ada";
    else if (larik[i]==x)
    {
        posisi=i+1;

        cout<<"pada posisi ke "<<posisi;
    }
    getch();
}
```

Output kode 5.2.



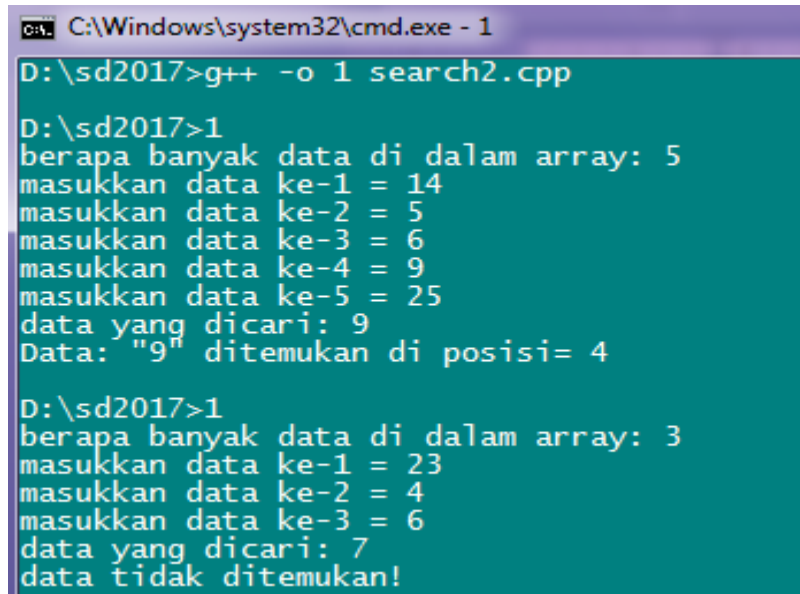
```
C:\Windows\system32\cmd.exe - 1

D:\sd2017>g++ -o 1 search1.cpp

D:\sd2017>1
data yang ingin dicari ? 17
pada posisi ke 7
D:\sd2017>1
data yang ingin dicari ? 32
maaf data yang dicari tidak ada
```

Penugasan!! Sequential Search.

Berdasarkan contoh program diatas, modifikasi sehingga output program seperti berikut ini.



```
C:\Windows\system32\cmd.exe - 1
D:\sd2017>g++ -o 1 search2.cpp
D:\sd2017>1
berapa banyak data di dalam array: 5
masukkan data ke-1 = 14
masukkan data ke-2 = 5
masukkan data ke-3 = 6
masukkan data ke-4 = 9
masukkan data ke-5 = 25
data yang dicari: 9
Data: "9" ditemukan di posisi= 4

D:\sd2017>1
berapa banyak data di dalam array: 3
masukkan data ke-1 = 23
masukkan data ke-2 = 4
masukkan data ke-3 = 6
data yang dicari: 7
data tidak ditemukan!
```

II. PENCARIAN BINER (BINARY SEARCH)

Pencarian biner adalah proses mencari data dengan membagi data atas dua bagian secara terus menerus sampai elemen yang dicari sudah ditemukan, atau indeks kiri lebih besar dari indeks kanan.

Algoritma ini lebih efisien daripada algoritma pencarian sekuensial, tetapi pencarian ini mempunyai syarat yaitu bahwa kumpulan data yang harus dilakukan pencarian harus sudah terurut terlebih dahulu. Karena data sudah terurut, algoritma dapat menentukan apakah nilai data yang dicari berada sebelum atau sesudah elemen larik yang sedang dibandingkan pada suatu saat. Dengan cara ini, algoritma dapat lebih menghemat waktu pencarian.

Prinsip dari pencarian biner dapat dijelaskan sebagai berikut:

1. mula-mula diambil posisi awal 0 dan posisi akhir = $N - 1$,
2. kemudian dicari posisi data tengah dengan rumus $(\text{posisi awal} + \text{posisi akhir}) / 2$.
3. Kemudian data yang dicari dibandingkan dengan data tengah.
4. Jika lebih kecil, proses dilakukan kembali tetapi posisi akhir dianggap sama dengan posisi tengah -1 .
5. Jika lebih besar, proses dilakukan kembali tetapi posisi awal dianggap sama dengan posisi tengah $+1$. Demikian seterusnya sampai data tengah sama dengan yang dicari

D.II.Praktikum Binary Search

Latihan

Kode 5.3. ketikkan program pengurutan secara Ascending seperti di bawah ini:

```
#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;

    cout<<"masukan data yang dicari : ";
    cin>>tar;
    int awal=0, akhir=10, tengah;

    while (awal <= akhir)
    {
        tengah = (awal + akhir)/2;
        if (tar > Ar[tengah] )
            { awal = tengah + 1; }
        else if (tar < Ar[tengah])
            {akhir= tengah - 1;}
        else {awal = akhir +1;
        }
        if (tar == Ar[tengah])
        {cout<<" Data ditemukan, Ke- "<<tengah+1<<endl;
        }
        else {
            cout<<"target tidak ditemukan "<<endl;
        }
        getch();
    }
}
```

Penugasan!

- Dari program Kode 5.3. Modifikasi sehingga inputan menjadi dinamis, gunakan pengurutan secara descending.

Kode5.4.

```
#include<iostream>

using namespace std;

int main() {

    int n, angka[10], kiri , kanan, tengah , temp, key;

    bool ketemu = false;

    cout<<"Masukkan jumlah data: ";
    cin>>n;

    for(int i = 0; i < n; i++)
    {
        cout<<"Angka ke-["<<i<<"] : ";
        cin>>angka[i];
    }

    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n-i-1; j++)
        {
            if(angka[j] > angka[j+1])
            {
                temp = angka[j];
                angka[j] = angka[j+1];
                angka[j+1] = temp;
            }
        }
    }

    cout<<"Data yang telah diurutkan adalah : ";

    for(int i = 0; i < n; i++)
    {
        cout<<angka[i]<<" ";
    }

    cout<<"\n Masukkan angka yang dicari: ";

    cin>>key;
```

```

kiri = 0;
kanan = n-1;

while(kiri<=kanan)
{
    tengah = (kiri + kanan)/2;

    if(key == angka[tengah])
    {
        ketemu = true;
        break;
    }
    else if(key < angka[tengah])
    {
        kanan = tengah - 1;
    }
    else
    {
        kiri = tengah + 1;
    }
}

if(ketemu==true)
    cout<<"Angka ditemukan!";
else
    cout<<"Angka tidak ditemukan";

return 0;
}

```

E. Tugas

1. Buatlah sebuah program dengan pencarian karakter dalam sebuah nama.

Langkah tugas :

1. Contoh program sebelumnya
2. Data yang diinputkan berupa nama siswa
3. Lakukan input array secara dinamis
4. Jika pencarian tidak ditemukan maka beri keterangan

Berikut contoh output program:

```
C:\Windows\system32\cmd.exe
D:\sd2017>g++ -o 1 tugas1search.cpp
D:\sd2017>1
Masukan Nama = andika huda
Masukan Karakter yang dicari = a
Data Ditemukan!!!3
```

2. Buatlah sebuah program dengan pencarian data nama dalam sebuah kelas.

Langkah tugas :

1. Contoh program sebelumnya
2. Lakukan input array secara dinamis
3. Pencarian dilakukan dengan menggunakan nama siswa

```
C:\Windows\system32\cmd.exe - 1
D:\sd2017>g++ -o 1 tugas2.cpp
D:\sd2017>1
Masukan Jumlah Nama = 3
1. yuni
2. gunawan
3. fahri
Masukan Nama yang dicari = yuni
Data Ditemukan
D:\sd2017>1
Masukan Jumlah Nama = 3
1. yuni
2. gunawan
3. fahri
Masukan Nama yang dicari = tika
Data Tidak Ditemukan
```

“Dikumpulkan di Google Classroom”