

Nama : Andri Firman Saputra
NIM : 201011402125

Praktikum Algoritma
Laporan Awal - Perkenalan 13

Teori Dasar

a) Pendahuluan

Proses pencarian suatu elemen di dalam array disebut searching, ada 2 macam pencarian yaitu pencarian sekuensial (sequential searching) dan pencarian biner (binary searching). Perbedaannya terletak pada keadaan suatu elemen atau data yg berada pada array. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak urut. Sedangkan pencarian biner digunakan pada data yg sudah dalam keadaan urut.

b) Pencarian Sekuensial

Pencarian sekuensial menggunakan prinsip sebagai berikut: data yg ada pada suatu array dibandingkan satu persatu dgn data yg dicari.

Pencarian ini dilakukan dgn melakukan suatu pengulangan dari 1 sampai semua data yg ada. Pada setiap kali pengulangan, dibandingkan data yg posisinya ke - i dengan data yg dicari atau dimausud. Apabila sama, maka data tersebut telah ditemukan dan proses pengulangan dihentikan.

Sebaliknya, kalau sampai pengulangan selesai dan data yg dicari tidak ditemukan, maka tsb tidak ada.

Tugas Pendahuluan

1. Apa yg dimaksud dgn Searching?

Searching adalah proses pencarian suatu elemen di dalam array.

2. Jelaskan kelebihan dan kekurangan Searching pada bahasa C/C++!

Kelebihan: apabila data yg dicari letaknya pada data-data awal sehingga prosesnya berjalan cepat.

Kekurangan: apabila data yg dicari letaknya pada data terakhir maka dalam penggunaan waktu, proses ini berjalan lama.

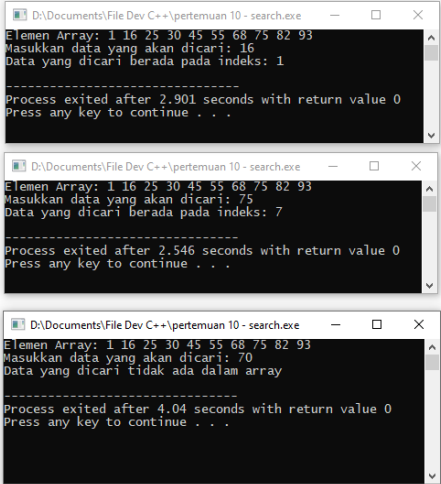
3. Sebagai Programmer, mengapa anda menggunakan Searching? Karena dapat memudahkan pencarian data.



4. Buatlah contoh algoritma dan program sederhana menggunakan Searching dan memandu flowchart!

Jawaban No. 4

```
search.cpp
1 #include<iostream>
2 using namespace std;
3
4 void BinSearch(int data[], int n, int x, int *idx);
5
6 int main()
7 {
8     int data[] = {1, 16, 25, 30, 45, 55, 68, 75, 82, 93};
9     int idx, x, i, jmlData = 10;
10
11     cout<<"Elemen Array: ";
12     for(i = 0; i < jmlData; i++)
13     {
14         cout<<data[i]<<" ";
15     }
16     cout<<endl;
17
18     cout<<"Masukkan data yang akan dicari: ";
19     cin>>x;
20
21     BinSearch(data, jmlData, x, &idx);
22     if(idx != -1)
23     {
24         cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
25     }
26     else
27     {
28         cout<<"Data yang dicari tidak ada dalam array"<<endl;
29     }
30 }
31
32 void BinSearch(int data[], int n, int x, int *idx)
33 {
34     bool ketemu = false;
35     int top = n-1, bottom = 0, mid;
36     int i = 0;
37     while(bottom <= top && !ketemu)
38     {
39         mid = (top + bottom) / 2;
40
41         if(data[mid] == x)
42         {
43             ketemu = true;
44         }
45         else
46         {
47             if(data[mid] > x)
48             {
49                 top = mid - 1;
50             }
51             else
52             {
53                 bottom = mid + 1;
54             }
55         }
56     }
57
58     if(ketemu)
59     {
60         *idx = mid;
61     }
62     else
63     {
64         *idx = -1;
65     }
66 }
67 }
```



The image shows three screenshots of the search.exe application output, demonstrating the binary search process for different input values.

First Screenshot: The application displays the array elements: 1 16 25 30 45 55 68 75 82 93. It prompts the user to enter the data to be searched, and the user enters 16. The output shows that the data is found at index 1.

Second Screenshot: The application displays the array elements: 1 16 25 30 45 55 68 75 82 93. It prompts the user to enter the data to be searched, and the user enters 75. The output shows that the data is found at index 7.

Third Screenshot: The application displays the array elements: 1 16 25 30 45 55 68 75 82 93. It prompts the user to enter the data to be searched, and the user enters 70. The output shows that the data is not found in the array.

Source Code:

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

void BinSearch(int data[], int n, int x, int *idx);

int main()
{
    int data[] = {1, 16, 25, 30, 45, 55, 68, 75, 82, 93};
    int idx, x, i, jmlData = 10;

    cout<<"Elemen Array: ";
    for(i = 0; i < jmlData; i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<endl;

    cout<<"Masukkan data yang akan dicari: ";
    cin>>x;

    BinSearch(data, jmlData, x, &idx);
    if(idx != -1)
    {
        cout<<"Data yang dicari berada pada indeks: "<<idx<<endl;
    }
    else
    {
        cout<<"Data yang dicari tidak ada dalam array"<<endl;
    }
}

void BinSearch(int data[], int n, int x, int *idx)
{
```

```

bool ketemu = false;
int top = n-1, bottom = 0, mid;
int i = 0;
while(bottom <= top && !ketemu)
{
    mid = (top + bottom) / 2;

    if(data[mid] == x)
    {
        ketemu = true;
    }
    else
    {
        if(data[mid] > x)
        {
            top = mid - 1;
        }
        else
        {
            bottom = mid + 1;
        }
    }
}

if(ketemu)
{
    *idx = mid;
}
else
{
    *idx = -1;
}
}

```