

PERTEMUAN 6
TUGAS TERSTRUKTUR STURKTUR DATA
KELAS 04TPLP017

Nama : Nurul Cessy Zulma

Nim : 191011402706

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Searching ?
2. Jelaskan perbedaan dari Sequential Search, Binary Search, dan Interpolation Search sebagai metode-metode searching ?
3. Jelaskan apa saja yang mempengaruhi kecepatan proses pencarian data didalam penyimpanan data ?
4. Buatlah contoh program sederhana menggunakan Searching

Jawab :

1. **Searching (pencarian)** merupakan proses yang mendasar di dalam pemrograman. Pencarian (Searching) merupakan tindakan untuk mendapatkan suatu data dalam kumpulan data berdasarkan suatu kunci (key) atau acuan data. Dalam kehidupan sehari-hari seringkali kita berurusan dengan pencarian; misalnya menemukan nomor telepon seseorang pada buku telepon atau mencari istilah dalam kamus, dan masih banyak lagi. Pada aplikasi computer pencarian kerap kali dilakukan. Misalnya untuk proses penghapusan data/record atau mengubah data/record tertentu didalam suatu tabel atau file.
2. Perbedaan dari Sequential Search, Binary Search, dan Interpolation Search sebagai metode-metode searching :

- **Sequential Search** (pencarian beruntun) adalah metode pencarian yang paling mudah. Pencarian berurutan adalah proses membandingkan setiap elemen array satu per satu secara berurutan yang dimulai dari elemen pertama hingga elemen yang dicari ditemukan atau hingga elemen terakhir dari array. Pencarian beruntun dapat dilakukan terhadap elemen array yang belum terurut atau terhadap elemen array yang belum terurut atau terhadap elemen array yang terurut.
- **Binary Search** adalah metode pencarian suatu data atau elemen di dalam suatu array dengan kondisi data dalam keadaan terurut.
- **Interpolation Search** adalah sebuah algoritma atau metode untuk mencari nilai key yang diberikan dalam array diindeks yang telah diperintahkan oleh nilai-nilai kunci.

3. Faktor yang mempengaruhi kecepatan proses pencarian data didalam penyimpanan data :

- a. Register ; menyimpan instruksi selama proses berjalan, ukuran sesuai dengan jumlah data yang tepat diproses dalam satu waktu.
- b. RAM ; semakin besar RAM maka semakin banyak data yang disimpan sehingga akses ke data lebih cepat (dibandingkan dengan mengakses ke secondary storage).
- c. CPU clock ; semakin tinggi clock CPU maka semakin cepat/semakin banyak perintah yang dapat dieksekusi.
- d. Cache memory ; semakin cepat, dan besar cache maka proses akan menjadi lebih cepat.
- e. Bus/datapath ; menentukan besarnya data yang dapat ditransmisikan antara CPU dan device yang lain.

4. Contoh program sederhana menggunakan Searching

```
#include<stdio.h>
#include<iostream>
#include<conio.h>
#include<iomanip>
using namespace std;
int main()
{
    //Deklarasi Variabel
    int Nilai[20];
    int i, j, N;
    int temp, awal, akhir, tengah, Bilangan;

    //Proses Penginputan data
    cout<<"Banyak Bilangan : ";
    cin>>N;
    for(i=0; i<N; i++)
    {
        cout<<"Elemen ke-"<<i<<" = ";
        cin>>Nilai[i];
    }
    cout<<"\nElemen Sebelum diurut = ";
    for(i=0; i<N; i++)
        cout<<setw(3)<<Nilai[i];

    //Proses Pengurutan Data
    for(i=0; i<N-1; i++)
    {
        for(j=i+1; j<N; j++)
```

```

{
    if (Nilai[i] > Nilai[j])
    {
        temp = Nilai[i];
        Nilai[i] = Nilai[j];
        Nilai[j] = temp;
    }
}
}

cout<<"\nElemen Setelah diurut = ";
for(i=0; i<N; i++)
    cout<<setw(3)<<Nilai[i];
    cout<<"\nindeks Elemen = ";
for(i=0; i<N; i++)
    cout<<setw(3)<<i;
    cout<<"\nMasukkan data yang akan anda cari: ";
    cin>>Bilangan;

```

//Proses Pencarian Data

```

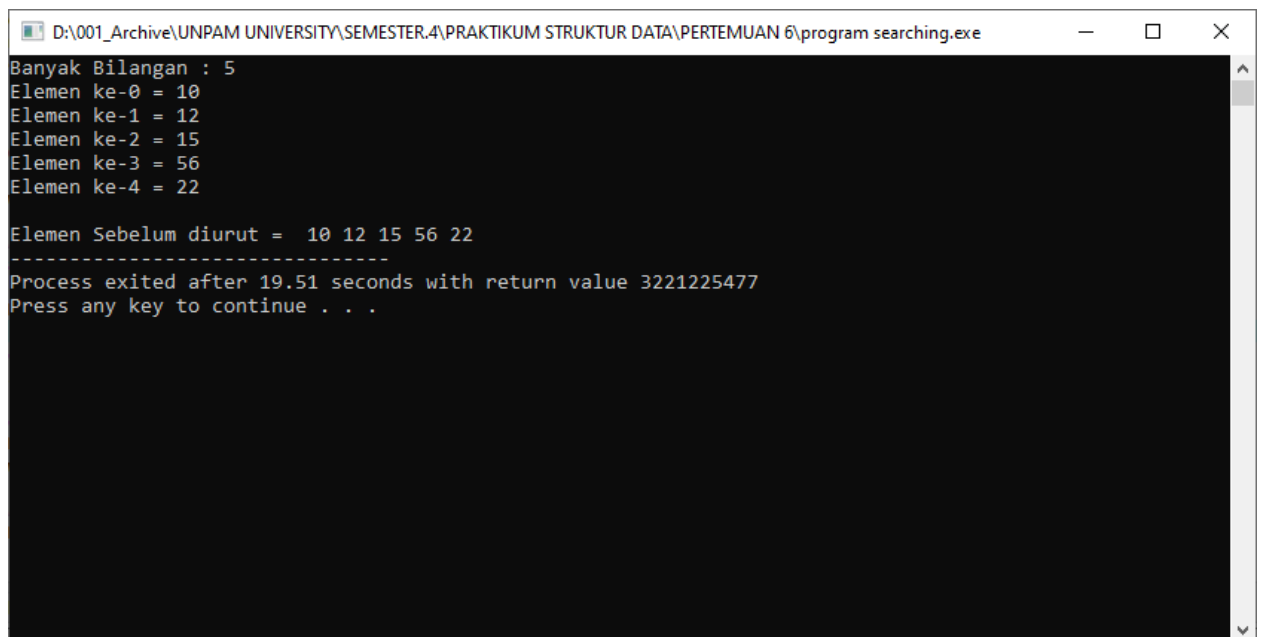
awal = 0;
akhir = N-1;
do
{
    tengah = (akhir + awal)/2;
    if(Bilangan < Nilai[tengah])
        akhir = tengah - 1;
    else
        awal = tengah + 1;
}

```

```

while((akhir >= awal) && (Nilai[tengah] !=Bilangan));
if(Nilai[tengah] == Bilangan)
{
    cout<<"\nData "<<Bilangan<<" ada dalam array";
    cout<<" pada posisi "<<tengah;
}
else
    cout<<"\nData "<<Bilangan<<" tidak ada dalam array\n";
getch();
}

```



```

D:\001_Archive\UNPAM UNIVERSITY\SEMESTER.4\PRAKTIKUM STRUKTUR DATA\PERTEMUAN 6\program searching.exe
Banyak Bilangan : 5
Elemen ke-0 = 10
Elemen ke-1 = 12
Elemen ke-2 = 15
Elemen ke-3 = 56
Elemen ke-4 = 22

Elemen Sebelum diurut = 10 12 15 56 22
-----
Process exited after 19.51 seconds with return value 3221225477
Press any key to continue . . .

```

TUGAS AKHIR

1. Buatlah program untuk melakukan pencarian terhadap data 12, 15 dan 37 dari sederetan data 34, 8, 50, 74, 87, 90, 12, 25, 20, 30, 35, 45, 40, 22, 29, 72, 60, 55, 53, 12, 32, 33, 12, 41, 12 ! Jika data yang dicari terdapat lebih dari satu, tentukan banyaknya dan sebutkan berada pada posisi berapa saja data yang dicari berada pada sederetan data !

```
#include <iostream>
#include <conio.h>
using namespace std;
main()
{
    char nama[40];
    char nim[20];
    bool ketemu;
    int posisi[25];
    int c,i,banyak=0;
    int
    data[25]={ 34,8,50,74,87,90,12,25,20,30,35,45,40,22,29,72,60,55,53,12,32,
33,12,41,12};
    cout<<"Nama : "; gets(nama);
    cout<<"NIM : "; cin>>nim;
    cout<<"\nData : ";
    for(i=0;i<25;i++)
    {
        cout<<data[i]<<" ";
    }
    cout<<"\n\nData yang ingin di cari : "; cin>>c;
```

```

        for (i=0;i<25;i++){
            if (data[i]==c){
                ketemu=true;
                posisi[banyak]=i+1;
                banyak++;
            }
        }

        if (ketemu)
        {
            cout<<"Data : "<<c;
            cout<<"\nDitemukan sebanyak "<<banyak;
            cout<<"\nPada posisi ke ";
            for(i=0;i<banyak;i++){
                cout<<posisi[i]<<" ";
            }
        }

        else
        {
            cout<<"Data "<<c<<" tidak di temukan";
        }

        getch();
    }

```

```

D:\001_Archive\UNPAM UNIVERSITY\SEMESTER.4\PRAKTIKUM STRUKTUR DATA\PERTEMUAN 6\tugas akhir.exe
Nama : Nurul C
NIM : 191011402706

Data : 34 8 50 74 87 90 12 25 20 30 35 45 40 22 29 72 60 55 53 12 32 33 12 41 12

Data yang ingin di cari : 25
Data : 25
Ditemukan sebanyak 1
Pada posisi ke 8
-----
Process exited after 23.02 seconds with return value 0
Press any key to continue . . .

```