

Alvina Vania Kirana
140810180010
Praktikum Analgo

1. Studi kasus 1

Program

```
/*  
Alvina Vania Kirana  
140810180010  
Praktikum Analisis Algoritma  
Tugas 2 studi kasus 1  
*/  
  
#include<iostream>  
using namespace std;  
  
int main(){  
    int x[50];  
    int n,i,nmaks;  
  
    cout<<"Masukkan banyak angka: ";cin>>n;  
    for(int i=0;i<n;i++){  
        cout<<"Bilangan ke-" <<i+1<< "="; cin>>x[i];  
    }  
  
    nmaks = 0;  
  
    for(int i=0;i<n;i++){  
        if(x[i]>nmaks)  
            nmaks=x[i];  
    }  
    cout<<"Maka nilai maksimumnya adalah "<<nmaks<<endl;  
}
```

```
alvinas-MacBook-Pro:~ alvinavania$  
StudiKasus1  
Masukkan banyak angka: 3  
Bilangan ke-1=2  
Bilangan ke-2=5  
Bilangan ke-3=6  
Maka nilai maksimumnya adalah 6  
alvinas-MacBook-Pro:~ alvinavania$
```

Kompleksitas waktu

- Assignment

maks<- xi	1
i<-2	1
x<-x1	n-1
i<-i+1	n-1

- Perbandingan

If xi>maks	n-1
------------	-----

- Penjumlahan

i<-i+1	n-1
--------	-----

Maka $T(n)=2+4(n-1)$

2. Studi kasus 2

Program

```
/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Tugas 2 studi kasus 2
*/
#include <iostream>
using namespace std;

int main()
{
    int n, c, x[100], index;
    bool found = false;

    cout << "\nMasukan banyak data = "; cin >> n;
    cout << endl;

    for(int i=0; i<n; i++)
    {
        cout << "Data ke-" << i+1 << " : ";
        cin >> x[i];
    }
```

```
cout << "Apa yang kamu cari?"; cin >> c;
cout << endl;
```

```
for(int i=0; i<n; i++){
    if(x[i] == c){
        found = true;
        index = i;
        i = n;
    }
}
if(found == true){
    cout << "Data ditemukan pada data ke-" << index+1<<endl;
}
else{
    cout << "Data tidak ditemukan"<<endl;
}
}
```

```
alvinas-MacBook-Pro:~ alvinavania$ ./StudiKasus2

Masukan banyak data = 3

Data ke-1 : 1
Data ke-2 : 2
Data ke-3 : 3
Apa yang kamu cari?2

Data ditemukan pada data ke-2
alvinas-MacBook-Pro:~ alvinavania$
```

Kompleksitas waktu

- Assignment

i<-1	1
found<-false	1
i<-i+1	n
found<-true	n
idx<-i	1

idx<-o 1

- Perbandingan

If xi=y n

If found 1

- Penjumlahan

i<-i+1 n

Maka $T(n)=5+4n$

Kasus Terbaik:

$T_{\text{minimal}}(n)=1$

Kasus Terburuk:

Bila $a(n)=x$ atau tidak ditemukan maka $T_{\text{maximal}}(n)=n$

Rata-rata:

Jika x ditemukan samapai di posisi ke- n maka operasi $a(k)=x$ akan dieksekusi sampai n kali.

$T_{\text{average}}(n)=(1+2+3+..+n)/n=(1/2n(1+n))/n=(n+1)/2$

3. Studi kasus 3

Program

```
/*
Alvina Vania Kirana
140810180010
Praktikum Analisis Algoritma
Tugas 2 studi kasus 3
*/
#include<iostream>

using namespace std;
int main()
{
    int n, i, x[100], c;
    int a, b, t;

    cout<<"Masukkan banyak data: ";cin>>n;
    cout<< endl;

    for (i=0; i<n; i++)
    {
        cout<<"Data ke-"<<i+1<<" :";
```

```

        cin>>x[i];
    }
    cout<<"\nMasukkan data yang akan di cari :"; cin>>c;
    a = 0;
    b = n-1;
    cout << endl;

```

```

while (a <= b)
{
    t = (a+b)/2;
    if(x[t]<c)
    {
        a=t+1;

```

```

    }
    else if(x[t] == c)
    {
        cout<<c<<" ditemukan pada data ke-"<<t+1<<"\n";
        break;
    }
    else
    {
        b=t-1;
    }
    t=(a+b)/2;
}

```

```

if(a>b)
{
    cout<<c<<" Tidak ada";
}
cout <<endl;
}

```

```

alvinas-MacBook-Pro:~ alvinavania$ 70
StudiKasus3
Masukkan banyak data: 3

Data ke-1 :1
Data ke-2 :2
Data ke-3 :3

Masukkan data yang akan di cari :4

4Tidak ada
alvinas-MacBook-Pro:~ alvinavania$

```

Kompleksitas waktu

- Assignment

i<-1	1
j<-n	1
found<-false	1
mid<-(i+j)div1	n
Found<- true	n
i<-mid+1	n
i<-mid-1	n
idx<- mid	1
idx<-o	1

- Pembagian

mid<-(i+j)div1	n
----------------	---

- Perbandingan

If xmid= y	n
If xmid< y then	n
If found	1

- Penjumlahan

mid<-(i+j)div1	n
i<-mid+1	n

- Pengurangan

i<-mid-1	n
----------	---

Maka $T(n) = 6 + 10n$

Kasus terbaik:

Kasus Terburuk:

4. Studi kasus 4

Program

```
/*  
Alvina Vania Kirana  
140810180010  
Praktikum Analisis Algoritma  
Tugas 2 studi kasus 4  
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int x[100],x2[100],n;
```

```
void insertion_sort()
```

```
{
```

```
    int temp,i,j;
```

```
    for(i=1;i<=n;i++){
```

```
        temp = x[i];
```

```
        j = i -1;
```

```
        while(x[j]>temp && j>=0){
```

```
            x[j+1] = x[j];
```

```
            j--;
```

```
        }
```

```
        x[j+1] = temp;
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    cout<<"Masukkan Jumlah Data : "; cin>>n;
```

```
    cout<<endl;
```

```
    cout<< endl;
```

```
    for(int i=1;i<=n;i++)
```

```
    {
```

```
        cout<<"Masukkan data ke-"<<i<<" : ";
```

```
        cin>>x[i];
```

```
        x2[i]=x[i];
```

```
    }
```

```
    cout<< endl;
```

```
    insertion_sort();
```

```
    cout<<"\nData Setelah di Sort : "<<endl;
```

```
    for(int i=1; i<=n; i++)
```

```
    {
```

```
        cout<<x[i]<<" ";
```

```
    }
```

```
cout<<endl;
```

```
}
```

```
alvinas-MacBook-Pro:~ alvinavania$ /U
StudiKasus4
Masukkan Jumlah Data : 5

Masukkan data ke-1 : 10
Masukkan data ke-2 : 23
Masukkan data ke-3 : 22
Masukkan data ke-4 : 31
Masukkan data ke-5 : 1

Data Setelah di Sort :
1 10 22 23 31
alvinas-MacBook-Pro:~ alvinavania$
```

Kompleksitas waktu

- Assignment

```
insert<-xi      n-1
j<-i            n-1
x[j]<- x[j-1]    n-1
j<-j-1         n-1
x[j]=insert
```

- Pengurangan

```
x[j]<- x[j-1]    n-1
j<-j-1         n-1
```

Maka $T(n)=6(n-1)$

Loop akan dijalankan jika $i > j$ dan $x[i] < x[j]$. Jumlah total iterasi sementara sama dengan inversi. Kompleksitas waktu keseluruhan dari jenis penyisipan adalah $O(n+f(n))$ dimana $f(n)$ adalah jumlah inversi. Jika jumlah inversi adalah $O(n)$, maka kompleksitas waktu dari jenis penyisipan adalah $O(n)$.

Kasus terburuknya bisa ada inversi $n*(n-1)/2$. Kasus terburuk terjadi ketika array diurutkan dalam urutan terbalik. Jika kompleksitas waktu kasus terburuk dari penyisipan adalah $O(n)$.

5. Studi kasus 5

Program

```
/*
Alvina Vania Kirana
140810180010
```


Praktikum Analisis Algoritma

Tugas 2 studi kasus 5

```
*/
```

```
#include <iostream>
```

```
using namespace std;
```

```
int x[100],x2[100];
```

```
int n;
```

```
void tukar(int a, int b)
```

```
{
```

```
    int t;
```

```
    t = x[b];
```

```
    x[b] = x[a];
```

```
    x[a] = t;
```

```
}
```

```
void selection_sort()
```

```
{
```

```
    int pos,i,j;
```

```
    for(i=1;i<=n-1;i++)
```

```
    {
```

```
        pos = i;
```

```
        for(j = i+1;j<=n;j++)
```

```
        {
```

```
            if(x[j] < x[pos]) pos = j;
```

```
        }
```

```
        if(pos != i) tukar(pos,i);
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    cout<<"\nMasukkan Jumlah Data : ";cin>>n;
```

```
    cout<< endl;
```

```
    for(int i=1;i<=n;i++)
```

```
    {
```

```
        cout<<"Masukkan data ke-"<<i<<" : ";
```

```
        cin>>x[i];
```

```
        x2[i]=x[i];
```

```
}
```

```
selection_sort();  
cout<< endl;  
cout<<"Data Setelah di Sort :"<<endl;  
for(int i=1; i<=n; i++)  
{  
    cout<<" "<<x[i];  
}  
cout<< endl;  
}
```

```
alvinas-MacBook-Pro:~ alvinavania$ /  
StudiKasus5  
  
Masukkan Jumlah Data : 5  
  
Masukkan data ke-1 : 133  
Masukkan data ke-2 : 22  
Masukkan data ke-3 : 456  
Masukkan data ke-4 : 1  
Masukkan data ke-5 : 14  
  
Data Setelah di Sort :  
1 14 22 133 456  
alvinas-MacBook-Pro:~ alvinavania$
```

Kompleksitas waktu

- Assignment

imaks<-1	n
imaks<-j	n
temp<-xi	n
xi<-Ximaks	n
ximaks<-temp	n

- Perbandingan

If xj>ximaks	n
--------------	---

Maka $T(n)=6n$

Operasi perbandingan untuk setiap pass ke-i

i=1 ->n-1

i=2 ->n-2

$i=k \rightarrow n-k$

$i=n-1 \rightarrow 1$

Algoritma urut tidak tergantung pada batasan apakah data masuknya sudah terurut atau acak membuat ini kompleksitas waktu terbaik dan terburuk

Operasi pertukaran

Untuk setiap i dari 1 Sampai $n-1$, terjadi satu kali pertukaran elemen sehingga jumlah operasi pertukaran seluruhnya $T(n)=n-1$

jadi, algoritma pengurutan maksimum membutuhkan $n(n-1)/2$ buah operasi perbandingan elemen dan $n-1$ buah operasi pertukaran