

# **TUGAS 4 PRAKTIKUM ANALISIS ALGORITMA**



Disusun Oleh :  
Bunga Azizha N  
140810180016

Asisten Praktikum:  
Faradilla Azranur, Felia Sri Indriyani, Agnes Hata

**PROGRAM STUDI S-1 TEKNIK INFORMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS PADJADJARAN  
2020**

## Studi Kasus

### Studi Kasus 1: MERGE SORT

 Source Code

---

```
/* Bunga Azizha N
140810180016 - Kelas B
Program Merge Sort */

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

void satu(int* in, int p, int q,int r);
void merge(int* in, int p, int r);
void input(int* a, int& n);

int main(){
    int in[100];
    int n;
    cout<<"\nPROGRAM MENGURUTKAN DENGAN MERGE SORT"<<endl;
    cout<<"-----"<<endl;
    input(in,n);
    auto start = chrono::steady_clock::now();
    merge(in,1,n);
    auto end = chrono::steady_clock::now();
    cout<<"-----"<<endl;
    cout << "\tHasil: ";
    for(int i=0; i<n; i++){
        cout << in[i] << " ";
    }

    cout<<endl;
    cout << "Elapsed time in nanoseconds : "
        << chrono::duration_cast<chrono::nanoseconds>(end - start).count()
        << " ns" << endl;

    return 0;
}

void satu(int* in, int p, int q,int r){
    int n1 = q-p+1;
    int n2 = r-q;
    int L[n1+1];
    int R[n2+1];
```

---

---

```
for (int i=1; i<=n1; i++){
    L[i-1] = in[(p-1)+i-1];
}

for (int j=1; j<=n2; j++){
    R[j-1] = in[(q-1)+j];
}

int i=0;
int j=0;
L[n1]=2147483647;
R[n2]=2147483647;

for (int k=(p-1); k<r; k++){
    if(L[i]<=R[j]){
        in[k]=L[i];
        i = i+1;
    }
    else{
        in[k]=R[j];
        j = j+1;
    }
}
}

void merge(int* in, int p, int r){
    int q;
    if(p<r){
        q = (p+r)/2;
        merge(in, p, q);
        merge(in, q+1, r);

        satu(in, p, q, r);
    }
}

void input(int* a, int& n){
    cout << "\tInput banyak data = "; cin >> n;
    cout<<endl;
    for (int i=0; i<n; i++){
        cout << "\tInput angka -> "; cin >> a[i];
    }
}
```

---

## Screenshoot Hasil Program

```
J:\- Semester 4\Analgo\Pertemuan 4\merge.exe

PROGRAM MENGURUTKAN DENGAN MERGE SORT
-----
      Input banyak data = 20

      Input angka -> 4
      Input angka -> 2
      Input angka -> 9
      Input angka -> 5
      Input angka -> 1
      Input angka -> 5
      Input angka -> 0
      Input angka -> 1
      Input angka -> 7
      Input angka -> 3
      Input angka -> 9
      Input angka -> 6
      Input angka -> 2
      Input angka -> 5
      Input angka -> 3
      Input angka -> 8
      Input angka -> 7
      Input angka -> 2
      Input angka -> 0
      Input angka -> 9
-----
Hasil: 0 0 1 1 2 2 2 3 3 4 5 5 5 6 7 7 8 9 9 9
Elapsed time in nanoseconds : 3200 ns
-----
Process exited after 40.49 seconds with return value 0
Press any key to continue . . .
```

Kompleksitas waktu algoritma merge sort adalah  $O(n \lg n)$ . Cari tahu kecepatan komputer Anda dalam memproses program. Hitung berapa running time yang dibutuhkan apabila input untuk merge sort-nya adalah 20?

Jawab : Untuk di program di atas, hasilnya adalah 3200 ns

Tapi jika sesuai dengan  $O \rightarrow T(20 \log_{10} 20) = 26$

## Studi Kasus 2: SELECTION SORT

Subproblem = 1

Masalah setiap subproblem =  $n-1$

Waktu proses pembagian =  $n$

Waktu proses penggabungan =  $n$

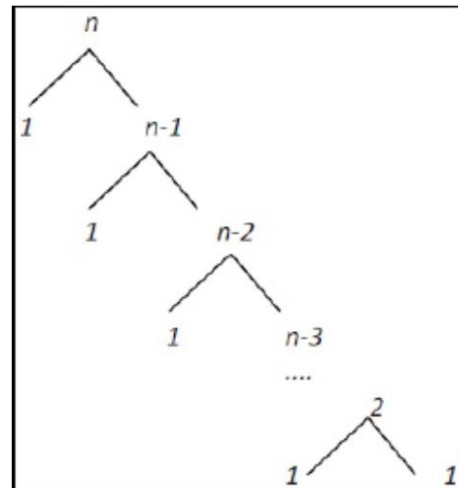
```
for i ← n downto 2 do {pass sebanyak n-1 kali}
  imaks ← 1
  for j ← 2 to i do
    if  $x_j > x_{imaks}$  then
      imaks ← j
    endif
  endfor
  {pertukarkan  $x_{imaks}$  dengan  $x_i$ }
  temp ←  $x_i$ 
   $x_i$  ←  $x_{imaks}$ 
   $x_{imaks}$  ← temp
endfor
```

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \\ &= c((n-1)(n-2)/2) + cn \\ &= c((n^2 - 3n + 2)/2) + cn \\ &= c(n^2/2) - (3n/2) + 1 + cn \\ &= O(n^2) \end{aligned}$$

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \\ &= c((n-1)(n-2)/2) + cn \\ &= c((n^2 - 3n + 2)/2) + cn \\ &= c(n^2/2) - (3n/2) + 1 + cn \\ &= \Omega(n^2) \end{aligned}$$

$$\begin{aligned} T(n) &= cn^2 \\ &= \Theta(n^2) \end{aligned}$$

$$T(n) = \{\Theta(1) T(n-1) + \Theta(n)\}$$



 Source Code

```
/* Bunga Azizha N
140810180016 - Kelas B
Program Selection Sort */
```

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

```
int data[100], data2[100];
int n;
```

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

```

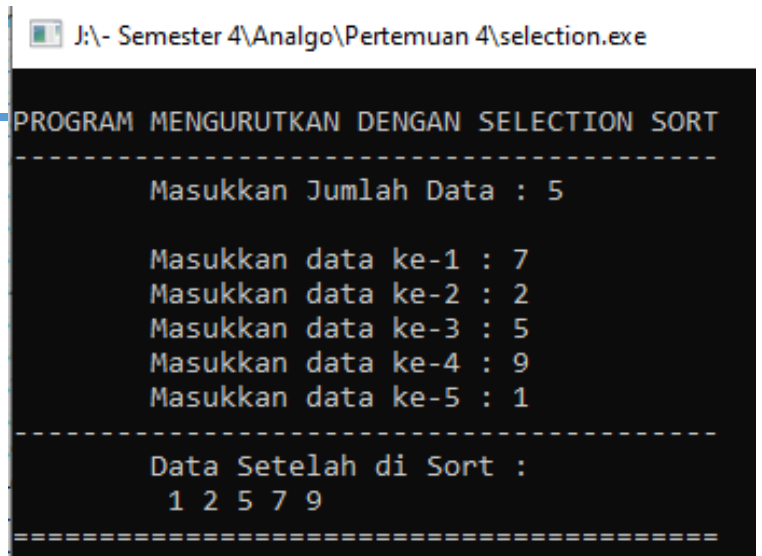
int main(){
    cout<<"\nPROGRAM MENGURUTKAN DENGAN SELECTION SORT"<<endl;
    cout<<"-----"<<endl;;
    cout<<"\tMasukkan Jumlah Data : ";cin>>n;
    cout<<endl;
    for(int i=1;i<=n;i++){
        cout<<"\tMasukkan data ke-"<<i<<" : ";
        cin>>data[i];
        data2[i]=data[i];
    }
    selectionSort();
    cout << "-----" << endl;
    cout<<"\tData Setelah di Sort : "<<endl;
    cout<<"\t";
    for(int i=1; i<=n; i++){
        cout<<" "<<data[i];
    }
    cout << "\n=====\\n";
    getch();
}

void tukar(int a, int b){
    int t;
    t = data[b];
    data[b] = data[a];
    data[a] = t;
}

void selectionSort(){
    int pos,i,j;
    for(i=1;i<=n-1;i++) {
        pos = i;
        for(j = i+1;j<=n;j++) {
            if(data[j] < data[pos]) pos = j;
        }
        if(pos != i) tukar(pos,i);
    }
}

```

 Screenshoot Hasil Program



```

J:\- Semester 4\Analgo\Pertemuan 4\selection.exe

PROGRAM MENGURUTKAN DENGAN SELECTION SORT
-----
Masukkan Jumlah Data : 5

Masukkan data ke-1 : 7
Masukkan data ke-2 : 2
Masukkan data ke-3 : 5
Masukkan data ke-4 : 9
Masukkan data ke-5 : 1
-----
Data Setelah di Sort :
1 2 5 7 9
=====

```

### Studi Kasus 3: INSERTION SORT

Subproblem = 1

Masalah setiap subproblem =  $n-1$

Waktu proses penggabungan =  $n$

Waktu proses pembagian =  $n$

#### Algoritma

```
for i ← 2 to n do
  insert ←  $x_i$ 
  j ← i
  while (j < i) and ( $x[j] > insert$ ) do
     $x[j] \leftarrow x[j-1]$ 
    j ← j-1
  endwhile
   $x[j] = insert$ 
endfor
```

$$T(n) = \{\theta(1) T(n-1) + \theta(n)\}$$

$$\begin{aligned} T(n) &= cn + cn - c + cn - 2c + \dots + 2c + cn \leq 2cn^2 + cn^2 \\ &= c((n-1)(n-2)/2) + cn \leq 2cn^2 + cn^2 \\ &= c((n^2 - 3n + 2)/2) + cn \leq 2cn^2 + cn^2 \\ &= c(n^2/2) - c(3n/2) + c + cn \leq 2cn^2 + cn^2 \\ &= O(n^2) \end{aligned}$$

$$\begin{aligned} T(n) &= cn \leq cn \\ &= \Omega(n) \end{aligned}$$

$$\begin{aligned} T(n) &= (cn + cn^2)/n \\ &= \Theta(n) \end{aligned}$$

 Source Code

---

```
/* Bunga Azizha N
140810180016 - Kelas B
Program Insertion Sort */

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

int data[100], data2[100], n;

void insertionSort();

int main(){
    cout<<"\nPROGRAM MENGURUTKAN DENGAN INSERTION SORT"<<endl;
    cout<<"-----"<<endl;;
    cout<<"\tMasukkan Jumlah Data : ";cin>>n;
    cout<<endl;
    for(int i=1;i<=n;i++){
        cout<<"\tMasukkan data ke-"<<i<<" : ";
        cin>>data[i];
        data2[i]=data[i];
```

---

---

```

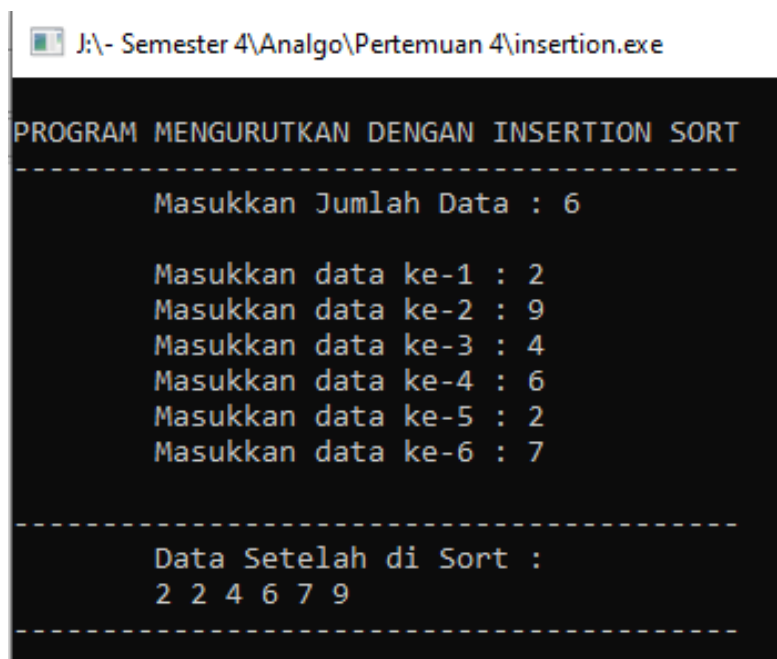
    }
    cout<<"\n-----" << endl;
    insertionSort();
    cout<<"\tData Setelah di Sort : "<<endl;
    cout<<"\t";
    for(int i=1; i<=n; i++){
        cout<<data[i]<<" ";
    }
    cout<<"\n-----"<<endl;;
    getch();
}

void insertionSort(){
    int temp,i,j;
    for(i=1;i<=n;i++){
        temp = data[i];
        j = i - 1;
        while(data[j]>temp && j>=0){
            data[j+1] = data[j];
            j--;
        }
        data[j+1] = temp;
    }
}
}

```

---

 Screenshoot Hasil Program



```

J:\- Semester 4\Analgo\Pertemuan 4\insertion.exe

PROGRAM MENGURUTKAN DENGAN INSERTION SORT
-----
Masukkan Jumlah Data : 6

Masukkan data ke-1 : 2
Masukkan data ke-2 : 9
Masukkan data ke-3 : 4
Masukkan data ke-4 : 6
Masukkan data ke-5 : 2
Masukkan data ke-6 : 7

-----
Data Setelah di Sort :
2 2 4 6 7 9
-----

```



## Studi Kasus 4: BUBBLE SORT

Subproblem = 1

Masalah setiap subproblem =  $n-1$

Waktu proses pembagian =  $n$

Waktu proses penggabungan =  $n$

$$\begin{aligned}T(n) &= \{\theta(1) T(n-1) + \theta(n)\} \\T(n) &= cn + cn - c + cn - 2c + \dots + 2c + c \leq 2cn^2 + cn^2 \\&= c((n-1)(n-2)/2) + c \leq 2cn^2 + cn^2 \\&= c((n^2 - 3n + 2)/2) + c \leq 2cn^2 + cn^2 \\&= c(n^2/2) - c(3n/2) + 2c \leq 2cn^2 + cn^2 \\&= O(n^2)\end{aligned}$$

$$\begin{aligned}T(n) &= cn + cn - c + cn - 2c + \dots + 2c + c \leq 2cn^2 + cn^2 \\&= c((n-1)(n-2)/2) + c \leq 2cn^2 + cn^2 \\&= c((n^2 - 3n + 2)/2) + c \leq 2cn^2 + cn^2 \\&= c(n^2/2) - c(3n/2) + 2c \leq 2cn^2 + cn^2 \\&= \Omega(n^2)\end{aligned}$$

$$\begin{aligned}T(n) &= cn^2 + cn^2 \\&= \Theta(n^2)\end{aligned}$$

 Source Code

---

```
/* Bunga Azizha N
140810180016 - Kelas B
Program Bubble Sort */

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

int main(){
    int arr[100],n,temp;
    cout<<"\n\tPROGRAM MENGURUTKAN DENGAN BUBBLE SORT"<<endl;
    cout<<"-----"<<endl;;
    cout<<"Masukkan banyak elemen yang akan diinputkan : ";cin>>n;
    cout<<endl;
    for(int i=0;i<n;++i){
        cout<<"\tMasukkan Elemen ke-"<<i+1<<" : ";cin>>arr[i];
    }

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

---

---


```

        arr[j]=arr[j+1];
        arr[j+1]=temp;
    }
}
cout<<"-----" << endl;
cout<<"\tHasil dari Bubble Sort : "<<endl;
cout<<"\t";
for(int i=0;i<n;i++){
    cout<<" "<<arr[i];
}
cout<<"\n=====
===== "<<endl;
}

```

---

#### Screenshoot Hasil Program

 J:\- Semester 4\Analgo\Pertemuan 4\bubble.exe

```

PROGRAM MENGURUTKAN DENGAN BUBBLE SORT
-----
Masukkan banyak elemen yang akan diinputkan : 8

Masukkan Elemen ke-1 : 2
Masukkan Elemen ke-2 : 5
Masukkan Elemen ke-3 : 1
Masukkan Elemen ke-4 : 8
Masukkan Elemen ke-5 : 6
Masukkan Elemen ke-6 : 9
Masukkan Elemen ke-7 : 0
Masukkan Elemen ke-8 : 7
-----
Hasil dari Bubble Sort :
0 1 2 5 6 7 8 9
=====

```