

LAPORAN PRAKTIKUM

STRUKTUR DATA LINIER

MODUL V

Dosen Pengampu

JB. Budi Darmawan S.T., M.Sc.



DISUSUN OLEH :

AGUSTINUS KEVIN YUDIPRATAMA

235314029

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA**

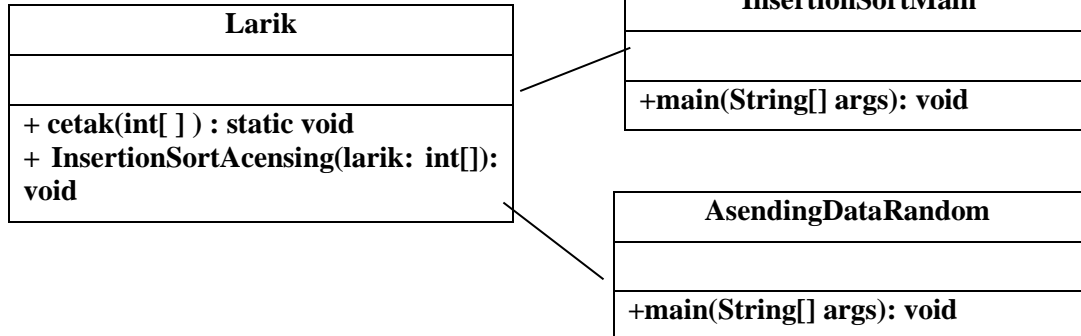
2024

No 1 (Asending)

A. TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami algoritma pengurutan Insertion sort
- Mampu mengimplementasikan dan membandingkan program pengurutan Buble sort, Selection sort dan Insertion sort dengan tipe data primitive
- Mampu membandingkan program pengurutan Buble sort, Selection sort dan Insertion sort.

B. DIAGRAM UML



C. SOURCE CODE

Class mainnya

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class InsertionSortMain {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31};

        System.out.println(x:"Data: ");
        Larik.cetak(data);
        System.out.println(x:"");

        Larik.InsertionSortAcending(data);
        Larik.cetak(data);
    }
}
```

DataRandomnya

Data 1.000

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class AcendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[1000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortAcending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Data 10.000

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class AcendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[10000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortAcending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Data 100.000

```
public class AcendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[100000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortAcending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Data 1.000.000

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class AcendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[1000000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortAcending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Classnya

```
public static void InsertionSortAcending (int []larik){  
    for (int iterasi = 1; iterasi < larik.length; iterasi++) {  
        int kunci = larik[iterasi];  
        int elemen = iterasi - 1;  
        while (elemen >= 0 && larik[elemen ]> kunci) {  
            larik[elemen +1] = larik[elemen];  
            elemen = elemen - 1;  
        }  
        larik[elemen + 1] = kunci;  
    }  
}
```

D. OUTPUT

Insertion Sort

Data:

```
5 8 26 15 11 31  
5 8 11 15 26 31
```

Data 1.000

```
Untuk = [I@1f32e575  
Waktu untuk Insertion Sort: 4359600 nanoseconds
```

Data 10.000

```
Untuk = [I@1f32e575  
Waktu untuk Insertion Sort: 40689800 nanoseconds
```

Data 100.000

```
Untuk = [I@1f32e575  
Waktu untuk Insertion Sort: 2574344900 nanoseconds
```

Data 1.000.000

```
Untuk = [I@1f32e575  
Waktu untuk Insertion Sort: 306900777100 nanoseconds
```

E. ANALISIS

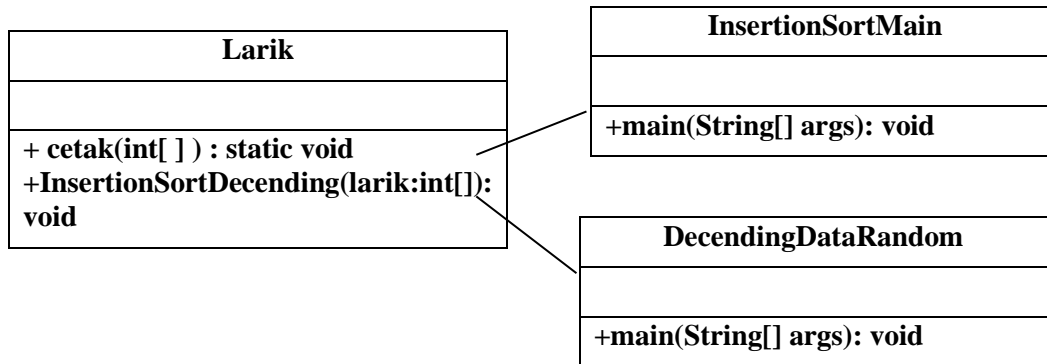
| | | | | | | |
|-----------|---|---|----|----|----|----|
| sebelum | 5 | 8 | 26 | 15 | 11 | 31 |
| Iterasi 1 | 5 | 8 | 26 | 15 | 11 | 31 |
| Iterasi 2 | 5 | 8 | 26 | 15 | 11 | 31 |
| Iterasi 3 | 5 | 8 | 15 | 26 | 11 | 31 |
| Iterasi 4 | 5 | 8 | 11 | 15 | 26 | 31 |
| Iterasi 5 | 5 | 8 | 11 | 15 | 26 | 31 |
| Sesudah | 5 | 8 | 11 | 15 | 26 | 31 |

No 2 (Desending)

A. TUJUAN PRAKTIKUM

- Mahasiswa mampu memahami algoritma pengurutan Insertion sort
- Mampu mengimplementasikan dan membandingkan program pengurutan Buble sort, Selection sort dan Insertion sort dengan tipe data primitive
- Mampu membandingkan program pengurutan Buble sort, Selection sort dan Insertion sort.

B. DIAGRAM UML



C. SOURCE CODE

InsertionSort

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class InsertionSortMain {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31};

        System.out.println(x:"Data: ");
        Larik.cetak(data);
        System.out.println(x:"");

        Larik.InsertionSortDecending(data);
        Larik.cetak(data);
    }
}
```

Data 1.000

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class DecendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[1000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortDecending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Data 10.000

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class DecendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[10000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortDecending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```


Data 100.000

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class DecendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[100000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortDecending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

Data 1.000.000

```
package vscode.Modul5;

import vscode.Modul1.Larik;

public class DecendingDataRandom {
    Run | Debug
    public static void main(String[] args) {
        int [] dataRandom = new int[1000000];
        for (int i = 0; i < dataRandom.length; i++) {
            dataRandom[i] = (int) (Math.random()* 1000);
        }
        //Insertion Sort
        long mulai = System.nanoTime();
        Larik.InsertionSortDecending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk Insertion Sort: " + selesai + " nanoseconds ");
    }
}
```

CLASNYA

```
public static void InsertionSortDecending(int[]larik){
    for (int iterasi = 1; iterasi < larik.length; iterasi++) {
        int kunci = larik[iterasi];
        int elemen = iterasi -1;
        while (elemen >= 0 && larik[elemen ]< kunci) {
            larik[elemen +1] = larik[elemen];
            elemen = elemen - 1;
        }
        larik[elemen + 1] = kunci;
    }
}
```

D. OUTPUT

Insertion Sort

```
Data:
5 8 26 15 11 31
31 26 15 11 8 5
```

Data 1.000

```
Untuk = [I@1f32e575
Waktu untuk Insertion Sort: 3612900 nanoseconds
```

Data 10.000

```
Untuk = [I@1f32e575
Waktu untuk Insertion Sort: 40113600 nanoseconds
```

Data 100.000

```
Untuk = [I@1f32e575
Waktu untuk Insertion Sort: 2637940800 nanoseconds
```

Data 1.000.000

```
Untuk = [I@1f32e575
Waktu untuk Insertion Sort: 320337168100 nanoseconds
```

E. ANALISIS

| | | | | | | |
|-----------|----|----|----|----|----|----|
| Sebelum | 5 | 8 | 26 | 15 | 11 | 31 |
| Iterasi 1 | 5 | 8 | 26 | 15 | 11 | 31 |
| Iterasi 2 | 8 | 5 | 26 | 15 | 11 | 31 |
| Iterasi 3 | 26 | 15 | 8 | 5 | 11 | 31 |
| Iterasi 4 | 26 | 15 | 11 | 8 | 5 | 31 |
| Iterasi 5 | 31 | 26 | 15 | 11 | 8 | 5 |
| Sesudah | 31 | 26 | 15 | 11 | 8 | 5 |

Insertion sort ini pengurutan yang dengan menyisipkan nilai yang ada baik ascending maupun descending.