

LAPORAN PRAKTIKUM

STRUKTUR DATA LINIER

MODUL IV

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**

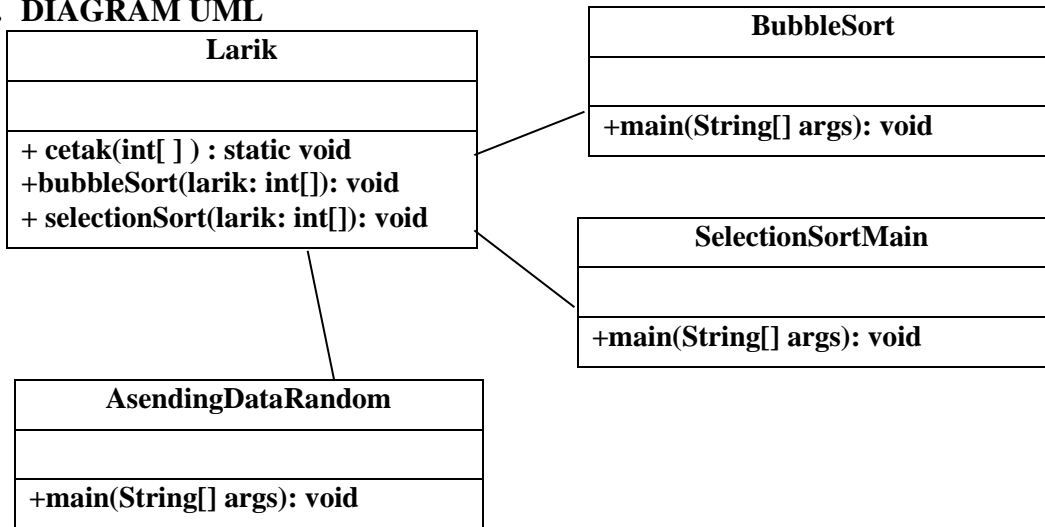
2024

No 1 (Asending)

A. TUJUAN PRAKTIKUM

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

B. DIAGRAM UML



C. SOURCE CODE

Class utamanya

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class BubleSort {
    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.BubleSort(data);
        Larik.cetak(data);
    }
}
```

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class SelectionSortMain {
    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.SelectionSort(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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubleSort(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSort(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```

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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubleSort(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSort(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```

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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubleSort(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSort(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```

Classnya

```
public static void BubleSort (int[] larik){
    for (int iterasi = 1; iterasi <= larik.length -1; iterasi++) {
        for (int elemet = 0; elemet <= larik.length -1 -iterasi; elemet++) {
            if (larik[elemet]> larik[elemet+1]) {
                int tukar = larik[elemet];
                larik[elemet]= larik[elemet+1];
                larik[elemet+1]= tukar;
            }
        }
    }
}
```

```

public static void SelectionSort (int[] larik){
    for (int iterasi = 0; iterasi < larik.length-2; iterasi++) {
        int minIndek = iterasi;
        for (int elemen = iterasi + 1; elemen < larik.length; elemen++) {
            if (larik[elemen] < larik[minIndek]) {
                minIndek = elemen;
                int tukar = larik[iterasi];
                larik[iterasi] = larik[minIndek];
                larik[minIndek] = tukar;
            }
        }
    }
}

```

D. OUTPUT

Bubble Sort

```

013.BubbleSort
Data:
5 8 26 15 11 31
5 8 11 15 26 31

```

Selection Sort

```

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

```

Data 1.000

```

Waktu untuk BubbleSort: 10636100
Waktu untuk SelectionSort: 5913600

```

Data 10.000

```

Waktu untuk BubbleSort: 178671701
Waktu untuk SelectionSort: 33534099

```

Data 1.000.000

```

Waktu untuk BubbleSort: 2284533241901
Waktu untuk SelectionSort: 306341677500

```

E. ANALISIS

Bubble Sort

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

Selection Sort

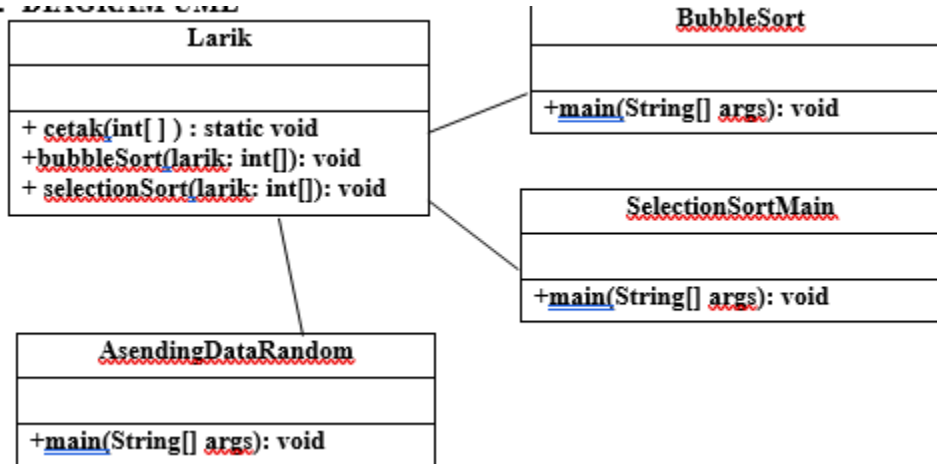
Sebelum	5	8	26	15	11	31
	5	8	26	15	11	31
Iterasi 1	5	8	26	15	11	31
Iterasi 2	5	8	11	15	26	31
Iterasi 3	5	8	11	15	26	31
Iterasi 4	5	8	11	15	26	31
Iterasi 5	5	8	11	15	26	31

No 2 (Desending)

A. TUJUAN PRAKTIKUM

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

B. DIAGRAM UML



C. SOURCE CODE

BubbleSort

```
package vscode.Modul13;

import vscode.Modul11.Larik;

public class BubbleSortDecending {
    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.BubbleSortDesending(data);
        Larik.cetak(data);
    }
}
```

SelectionSort

```
package vscode.Modul3;

import vscode.Modul1.Larik;

public class SelectionSortDecending {
    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.SelectionSortDecending(data);
        Larik.cetak(data);
    }
}
```

Data 1.000

```
package vscode.Modul3;

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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubbleSortDesending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSortDecending(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```


Data 10.000

```
package vscode.Modul3;

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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubbleSortDesending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSortDecending(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```

Data 1.000.000

```
package vscode.Modul3;

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);
        }
        //bubbleSort
        long mulai = System.nanoTime();
        Larik.BubbleSortDesending(dataRandom.clone());
        long selesai = System.nanoTime() - mulai;

        //SelectionSort
        long mulai2 = System.nanoTime();
        Larik.SelectionSortDecending(dataRandom.clone());
        long selesai2 = System.nanoTime() - mulai2;

        System.out.println("Untuk = " + dataRandom);
        System.out.println("Waktu untuk BubbleSort: " + selesai);
        System.out.println("Waktu untuk SelectionSort: " + selesai2);
    }
}
```

CLASNYA

```
public static void BubbleSortDesending (int [] larik){  
    for (int iterasi = 1; iterasi <= larik.length -1; iterasi++) {  
        for (int elemet = 0; elemet <= larik.length -1 -iterasi; elemet++) {  
            if (larik[elemet]< larik[elemet+1]) {  
                int tukar = larik[elemet];  
                larik[elemet]= larik[elemet+1];  
                larik[elemet+1]= tukar;  
            }  
        }  
    }  
}
```

```
public static void SelectionSortDesending(int []larik){  
    for (int iterasi = 0; iterasi < larik.length-2; iterasi++) {  
        int minIndek = iterasi;  
        for (int elemen = iterasi + 1; elemen < larik.length; elemen++) {  
            if (larik[elemen]> larik[minIndek]) {  
                minIndek = elemen;  
                int tukar = larik[iterasi];  
                larik[iterasi] = larik[minIndek];  
                larik[minIndek]= tukar;  
            }  
        }  
    }  
}
```

D. OUTPUT

BubbleSort

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

SelectionSort

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

Data 1.000

```
Waktu untuk BubbleSort: 7010100  
Waktu untuk SelectionSort: 3567400
```

Data 10.000

```
Waktu untuk BubbleSort: 146547700  
Waktu untuk SelectionSort: 33670199
```

Data 1.000.000

```
Waktu untuk BubbleSort: 2058014638501  
Waktu untuk SelectionSort: 291986318400
```

E. ANALISIS

Bubble Sort

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

Selection Sort

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

Perbedaannya

Bubble Sort

Membandingkan elemen secara berulang

Selection Sort

Memilih elemen dengan nilai dari sisa array yang belum diurutkan

