

LAPORAN PRAKTIKUM

STRUKTUR DATA LINIER

MODUL VI

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**

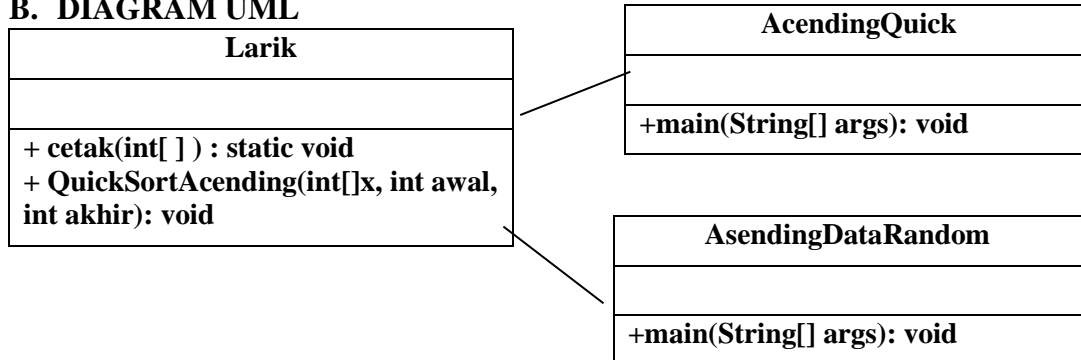
2024

No 1 (Asending)

A. TUJUAN PRAKTIKUM

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

B. DIAGRAM UML



C. SOURCE CODE

Class mainnya

```
package vscode.Modul6;

import vscode.Modul1.Larik;

public class AcendingQuick {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31, 7, 40, 23, 25};

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

        Larik.QuickSortAcending(data, awal:0, akhir:9);
        Larik.cetak(data);
    }
}
```

DataRandomnya

Data 1.000

```
> Modul6 > J AcendingDataRandom.java > ...
package vscode.Modul6;

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.QuickSortAcending(dataRandom.clone(),awal:0,akhir:999);
        long selesai = System.nanoTime() - mulai;

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

Data 10.000

```
package vscode.Modul6;

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.QuickSortAcending(dataRandom.clone(),awal:0,akhir:9999);
        long selesai = System.nanoTime() - mulai;

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

Data 100.000

```
package vscode.Modul6;

import vscode.Modul1.Larik;

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.QuickSortAcending(dataRandom.clone(),awal:0,akhir:99999);
        long selesai = System.nanoTime() - mulai;

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

Data 1.000.000

```
package vscode.Modul6;

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);
        }

        long mulai = System.nanoTime();
        Larik.QuickSortAcending(dataRandom.clone(),awal:0,akhir:999999);
        long selesai = System.nanoTime() - mulai;

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

Classnya

```
public static void QuickSortAcending(int[]x, int awal, int akhir){  
    if (awal < akhir) {  
        int i = awal + 1;  
        int j = akhir;  
        while (i <= akhir && x[i] <= x[awal]) {  
            i = i + 1;  
        }  
        while (j > awal && x[j] > x[awal]) {  
            j = j -1;  
        }  
        while (i < j) {  
            int tukar = x[i];  
            x[i] = x[j];  
            x[j] = tukar;  
        }  
        while (i <= akhir && x[i] <= x[awal]) {  
            i = i + 1;  
        }  
        while (j > awal && x[j] > x[awal]) {  
            j--;  
        }  
        int temp = x[awal];  
        x[awal] = x[j];  
        x[j] = temp;  
        QuickSortAcending(x, awal, j-1);  
        QuickSortAcending(x, j + 1, akhir);  
    }  
}
```

D. OUTPUT

Quick Sort

```
Data:  
5 8 26 15 11 31 7 40 23 25  
5 8 7 11 15 23 25 26 31 40
```

Data 1.000

```
Untuk = [I@1f32e575  
Waktu untuk Quick Sort: 11116000 nanoseconds
```

Data 10.000

```
Untuk = [I@1f32e575  
Waktu untuk Quick Sort: 6812200 nanoseconds
```

Data 100.000

```
Untuk = [I@1f32e575  
Waktu untuk Quick Sort: 63247200 nanoseconds
```

Data 1.000.000

```
Untuk = [I@1f32e575  
Waktu untuk Quick Sort: 594444000 nanoseconds
```

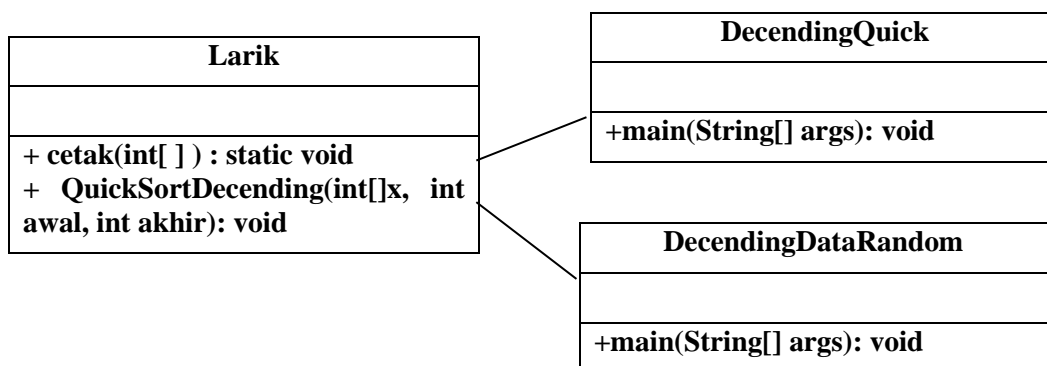
E. ANALISIS

No 2 (Desending)

A. TUJUAN PRAKTIKUM

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

B. DIAGRAM UML



C. SOURCE CODE

InsertionSort

```
package vscode.Modul16;

import vscode.Modul1.Larik;

public class DecendingQuick {
    Run | Debug
    public static void main(String[] args) {
        int data [] = {5, 8, 26, 15, 11, 31, 7, 40, 23, 25};

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

        Larik.QuickSortDecending(data, awal:0, akhir:9);
        Larik.cetak(data);
    }
}
```

Data 1.000

```
package vscode.Modul6;

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);
        }

        long mulai = System.nanoTime();
        Larik.QuickSortDecending(dataRandom.clone(),awal:0,akhir:99);
        long selesai = System.nanoTime() - mulai;

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

Data 10.000

```
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);
        }

        long mulai = System.nanoTime();
        Larik.QuickSortDecending(dataRandom.clone(),awal:0,akhir:999);
        long selesai = System.nanoTime() - mulai;

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


Data 100.000

```
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);  
        }  
  
        long mulai = System.nanoTime();  
        Larik.QuickSortDecending(dataRandom.clone(),awal:0,akhir:99999);  
        long selesai = System.nanoTime() - mulai;  
  
        System.out.println("Untuk = " + dataRandom);  
        System.out.println("Waktu untuk Quick Sort: " + selesai + " nanoseconds ");  
    }  
}
```

Data 1.000.000

```
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);  
        }  
  
        long mulai = System.nanoTime();  
        Larik.QuickSortDecending(dataRandom.clone(),awal:0,akhir:999999);  
        long selesai = System.nanoTime() - mulai;  
  
        System.out.println("Untuk = " + dataRandom);  
        System.out.println("Waktu untuk Quick Sort: " + selesai + " nanoseconds ");  
    }  
}
```

CLASNYA

```

public static void QuickSortDecending(int[]x, int awal, int akhir){
    if (awal < akhir) {
        int i = awal + 1;
        int j = akhir;
        while (i <= j) {
            while (i <= akhir && x[i] >= x[awal]) {
                i++;
            }
            while (j > awal && x[j] < x[awal]) {
                j--;
            }
            if (i < j) {
                int tukar = x[i];
                x[i] = x[j];
                x[j] = tukar;
            }
        }
        int temp = x[awal];
        x[awal] = x[j];
        x[j] = temp;

        QuickSortDecending(x, awal, j - 1);
        QuickSortDecending(x, j + 1, akhir);
    }
}

```

D. OUTPUT

Insertion Sort

Data:
5 8 26 15 11 31 7 40 23 25
40 31 26 25 23 15 11 8 7 5

Data 1.000

Untuk = [I@1f32e575
Waktu untuk Quick Sort: 1294300 nanoseconds

Data 10.000

Untuk = [I@1f32e575
Waktu untuk Quick Sort: 1802000 nanoseconds

Data 100.000

Untuk = [I@1f32e575
Waktu untuk Quick Sort: 5028800 nanoseconds

Data 1.000.000

Untuk = [I@1f32e575
Waktu untuk Quick Sort: 69715500 nanoseconds

E. ANALYSIS