

LAPORAN HASIL PRATIKUM
ANALISIS DAN STRUKTUR DATA
JOBSHEET 6



Raihan Akbar Putra Prasetyo/244107020087

Kelas: TI-1E

D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
PRAKTIKUM 25

Praktikum 1

- **Code program sorting24.java (bubble sort)**

```
package jobsheet6;
public class Sorting24 {
    int [] data;
    int jumData;

    Sorting24 (int Data [],int jmlDat ) {
        jumData = jmlDat;
        data = new int [jumData];
        for (int i = 0; i < jumData; i++) {
            data [i] = Data [i];
        }
    }
    void bubbleSort() {
        int temp = 0;
        for (int i = 0; i < jumData - 1; i++) {
            for (int j = 1; j < jumData - i; j++) {
                if (data[j - 1] > data[j]) {
                    temp = data[j];
                    data[j] = data[j - 1];
                    data[j - 1] = temp;
                }
            }
        }
    }
    void tampil(){
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i]+ " ");
        }
        System.out.println();
    }
}
```

- **Code program sortingMain24.java (bubble sort)**

```
package jobsheet6;

public class sortingMain24 {
    public static void main(String[] args) {
        int a[]={20,10,2,7,12};
        Sorting24 dataurut1 = new Sorting24(a,a.length);
        System.out.println("Data Awal 1 ");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("data sudah diurutkan dengan
BUBBLE SORT (ASC )");
        dataurut1.tampil();

    }
}
```

- **Output bubble sort**

```
Data Awal 1
20 10 2 7 12
data sudah diurutkan dengan BUBBLE SORT (ASC )
2 7 10 12 20
PS D:\kuliah\PRAKTIKUM-ASD>
```

- **Code program sorting24.java (SELECTION SORT)**

```
package jobsheet6;

public class Sorting24 {
    int [] data;
    int jumData;

    Sorting24 (int Data [],int jmlDat ) {
        jumData = jmlDat;
        data = new int [jumData];
        for (int i = 0; i < jumData; i++) {
            data [i] = Data [i];
        }
    }

    void bubbleSort() {
        int temp = 0;
        for (int i = 0; i < jumData - 1; i++) {
            for (int j = 1; j < jumData - i; j++) {
                if (data[j - 1] > data[j]) {
                    temp = data[j];
                    data[j] = data[j - 1];
                    data[j - 1] = temp;
                }
            }
        }
    }

    void tampil(){
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i]+ " ");
        }
        System.out.println();
    }

    void SelectionSort(){
        for (int i=0; i<jumData-1;i++){
            int min=i;
            for (int j=i+1;j<jumData;j++){
                if(data[j]<data[min]){
                    min=j;
                }
            }
            int temp=data[i];
            data[i]=data[min];
            data[min]=temp;
        }
    }
}
```

- **Code program SortingMain24.java (selection sort)**

```
package jobsheet6;
public class sortingMain24 {
    public static void main(String[] args) {
        int a[]={20,10,2,7,12};
        Sorting24 dataurut1 = new Sorting24(a,a.length);
        System.out.println("Data Awal 1 ");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("data sudah diurutkan dengan
BUBBLE SORT (ASC )");
        dataurut1.tampil();

        int b[]={30,20,2,8,14};
        Sorting24 dataurut2 = new Sorting24(a, b.length);
        System.out.println("Data Awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println("Data sudah diurutkan dengan
SELECTION SORT (ASC)");
        dataurut2.tampil();
    }
}
```

- **Output selection sort**

```
Data Awal 2
30 20 2 8 14
Data sudah diurutkan dengan SELECTION SORT (ASC)
2 8 14 20 30
PS D:\kuliah\PRAKTIKUM-ASD> █
```

- **Code program sorting24.java (insertion sort)**

```
package jobsheet6;
public class Sorting24 {
    int [] data;
    int jumData;
    Sorting24 (int Data [],int jmlDat ) {
        jumData = jmlDat;
        data = new int [jumData];
        for (int i = 0; i < jumData; i++) {
            data [i] = Data [i];
        }
    }
    void bubbleSort() {
        int temp = 0;
        for (int i = 0; i < jumData - 1; i++) {
            for (int j = 1; j < jumData - i; j++) {
                if (data[j - 1] > data[j]) {
                    temp = data[j];
                    data[j] = data[j - 1];
                    data[j - 1] = temp;
                }
            }
        }
    }
    void tampil(){
        for (int i = 0; i < jumData; i++) {
            System.out.print(data[i]+ " ");
        }
        System.out.println();
    }

    void SelectionSort(){
        for (int i=0; i<jumData-1;i++){
            int min=i;
            for (int j=i+1;j<jumData;j++){
                if(data[j]<data[min]){
                    min=j;
                }
            }
            int temp=data[i];
            data[i]=data[min];
            data[min]=temp;
        }
    }
    void insertionSort(){
        for ( int i=1; i<=data.length-1;i++){
            int temp=data[i];
            int j=i-1 ;
            while(j>=0 &&data[j]>temp){
                data[j+1]=data[j];
                j--;
            }
            data [j+1]=temp;
        }
    }
}
```

- **Code SortingMain24.java (insertion sort)**

```
package jobsheet6;

public class sortingMain24 {
    public static void main(String[] args) {
        int a[] = {20,10,2,7,12};
        Sorting24 dataurut1 = new Sorting24(a,a.length);
        System.out.println("Data Awal 1 ");
        dataurut1.tampil();
        dataurut1.bubbleSort();
        System.out.println("data sudah diurutkan dengan BUBBLE SORT
(ASC )");
        dataurut1.tampil();

        int b[] = {30,20,2,8,14};
        Sorting24 dataurut2 = new Sorting24(b, b.length);
        System.out.println("Data Awal 2");
        dataurut2.tampil();
        dataurut2.SelectionSort();
        System.out.println("Data sudah diurutkan dengan SELECTION
SORT (ASC)");
        dataurut2.tampil();

        int c[] = {40,19,4,9,3};
        Sorting24 dataurut3 = new Sorting24(c,c.length);
        System.out.println("Data Awal 3 ");
        dataurut3.tampil();
        dataurut3.insertionSort();
        System.out.println("Data sudah diurutkan dengan INSERTION
SORT (ASC)");
        dataurut3.tampil();
    }
}
```

- **Output insertion sort**

```
Data Awal 3
40 19 4 9 3
Data sudah diurutkan dengan INSERTION SORT (ASC)
3 4 9 19 40
PS D:\kuliah\PRAKTIKUM-ASD>
```

Pertanyaan:

1. Jelaskan fungsi kode program berikut

```
if (data[j-1]>data[j]){  
    temp=data[j];  
    data[j]=data[j-1];  
    data[j-1]=temp;  
}
```

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!
3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

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

4. Pada Insertion sort, apakah tujuan dari perintah `data[j+1]= data[j];`

1. Berfungsi menukar dua elemen jika elemen sebelumnya lebih besar, sehingga angka terbesar secara bertahap berpindah ke posisi akhir.

2. Mencari nilai terkecil dalam array dengan membandingkan setiap elemen, lalu menyimpan indeksnya dalam variabel min.

3. Menggeser elemen yang lebih besar ke kanan agar ada tempat kosong untuk menyisipkan angka yang sedang diurutkan.

4. Menggeser elemen ke kanan agar elemen baru bisa dimasukkan ke posisi yang tepat

Praktikum 2

1. Bubble sort

- Code program

```
package jobsheet6;
public class Mahasiswa24 {
    String nim,nama,kelas;
    double ipk;
    Mahasiswa24 () {

    }
    Mahasiswa24(String nm, String name, String kls, double
ip ) {
        nim=nm;
        nama=name;
        ipk=ip;
        kelas=kls;
    }
    void tampilkaninformasi(){
        System.out.println("Nama : " + nama);
        System.out.println("NIM : " + nim);
        System.out.println("kelas : " + kelas);
        System.out.println("IPK : " + ipk);
    }
}
```

```
package jobsheet6;
public class MahasiswaBerprestasi24 {
    Mahasiswa24 [] listMhs = new Mahasiswa24[5];
    int idx;
    void tambah(Mahasiswa24 m) {
        if ( idx<listMhs.length){
            listMhs[idx] = m;
            idx++;
        }else {
            System.out.println("Data sudah penuh" );
        }
    }
    void tampil() {
        for(Mahasiswa24 m : listMhs){
            m.tampilkaninformasi();
            System.out.println("-----");
        }
    }
    void bubblesort() {
        for (int i=0;i<listMhs.length-1;i++){
            for (int j=1; j<listMhs.length-i;j++){
                if (listMhs[j].ipk>listMhs[j-1].ipk){
                    Mahasiswa24 tmp = listMhs[j];
                    listMhs[j] = listMhs[j-1];
                    listMhs[j-1] = tmp;
                }
            }
        }
    }
}
```



```

package jobsheet6;
public class MahasiswaDemo {
    public static void main(String[] args) {
        MahasiswaBerprestasi24 list = new MahasiswaBerprestasi24();
        Mahasiswa24 m1 = new Mahasiswa24("123", "Zidan","2A",3.2) ;
        Mahasiswa24 m2 = new Mahasiswa24("124", "Ayu","2A",3.5) ;
        Mahasiswa24 m3 = new Mahasiswa24("125", "Sofi","2A",3.1) ;
        Mahasiswa24 m4 = new Mahasiswa24("126", "Sita","2A",3.9) ;
        Mahasiswa24 m5 = new Mahasiswa24("127", "Miki","2A",3.7) ;

        list.tambah(m1);
        list.tambah(m2);
        list.tambah(m3);
        list.tambah(m4);
        list.tambah(m5);
        System.out.println("Data Mahasiswa Sebelum sorting");
        list.tampil();
        System.out.println("Data Mahasiswa setelah sorting berdasarkan
ipk (DESC) : ");
        list.bubblesort();
        list.tampil();
    }
}

```

- **Output**

```

Data Mahasiswa Sebelum sorting
Nama : Zidan
NIM : 123
kelas : 2A
IPK : 3.2
-----
Nama : Ayu
NIM : 124
kelas : 2A
IPK : 3.5
-----
Nama : Sofi
NIM : 125
kelas : 2A
IPK : 3.1
-----
Nama : Sita
NIM : 126
kelas : 2A
IPK : 3.9
-----
Nama : Miki
NIM : 127
kelas : 2A
IPK : 3.7
-----
Data Mahasiswa setelah sorting berdasarkan ipk (DESC) :
Nama : Sita
NIM : 126
kelas : 2A
IPK : 3.9
-----
Nama : Miki
NIM : 127
kelas : 2A
IPK : 3.7
-----

```

Pertanyaan bubble sort

1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

```
for (int i=0; i<listMhs.length-1; i++){  
    for (int j=1; j<listMhs.length-i; j++){
```

- a. Mengapa syarat dari perulangan i adalah $i < \text{listMhs.length} - 1$?
 - b. Mengapa syarat dari perulangan j adalah $j < \text{listMhs.length} - i$?
 - c. Jika banyak data di dalam listMhs adalah 50, maka berapakah perulangan i akan berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?
2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyboard) yang terdiri dari nim, nama, kelas, dan ipk!

1.

a. Syarat $i < \text{listMhs.length} - 1$ digunakan karena Bubble Sort melakukan $n - 1$ tahap iterasi untuk memastikan seluruh data terurut dengan benar.

b. Syarat $j < \text{listMhs.length} - i$ digunakan agar elemen terbesar dari setiap iterasi tidak ikut dibandingkan lagi, sehingga jumlah perbandingan berkurang setiap tahapnya.

c. Jika ada 50 data, maka perulangan i terjadi 49 kali dan total perulangan j adalah 1225 kali, sehingga Bubble Sort membutuhkan 49 tahap untuk menyelesaikan proses pengurutan.

2. perubahan kode program menjadi dinamis

```
package jobsheet6;  
import java.util.Scanner;  
public class MahasiswaDemo {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        MahasiswaBerprestasi24 list = new MahasiswaBerprestasi24();  
        System.out.print("Masukkan jumlah mahasiswa: ");  
        int jumlah = input.nextInt();  
        input.nextLine();  
        for (int i = 0; i < jumlah; i++) {  
            System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");  
            System.out.print("NIM: ");  
            String nim = input.nextLine();  
            System.out.print("Nama: ");  
            String nama = input.nextLine();  
            System.out.print("Kelas: ");  
            String kelas = input.nextLine();  
            System.out.print("IPK: ");  
            double ipk = input.nextDouble();  
            input.nextLine();  
  
            Mahasiswa24 m = new Mahasiswa24(nim, nama, kelas, ipk);  
            list.tambah(m);  
        }  
        System.out.println("\nData Mahasiswa Sebelum sorting:");  
        list.tampil();  
  
        System.out.println("\nData Mahasiswa setelah sorting berdasarkan IPK (DESC):");  
        list.bubblesort();  
        list.tampil();  
    }  
}
```

2. Selection sort

- **Code program di mahasiswa berprestasi**

```
void selectionSort(){
    for (int i = 0; i < listMhs.length - 1; i++){
        int idxMin = i;
        for (int j = i + 1; j < listMhs.length; j++){
            if (listMhs[j].ipk <
listMhs[idxMin].ipk){
                idxMin = j;
            }
        }
        Mahasiswa24 tmp = listMhs[idxMin];
        listMhs[idxMin] = listMhs[i];
        listMhs[i] = tmp;
    }
}
```

- **Code program di mahasiswademo**

```
System.out.println("Data yang sudah terurut menggunakan
selection sort (ASC)");
list.selectionSort();
list.tampil();
```

- **Output**

```
Data yang sudah terurut menggunakan selection sort (ASC)
Nama : Ila
NIM : 124
kelas : 2B
IPK : 3.1
-----
Nama : Udin
NIM : 127
kelas : 2B
IPK : 3.2
-----
Nama : Tika
NIM : 126
kelas : 2B
IPK : 3.3
-----
Nama : Agus
NIM : 125
kelas : 2B
IPK : 3.6
-----
Nama : Ali
NIM : 123
kelas : 2B
IPK : 3.9
-----
PS D:\kuliah\PRAKTIKUM-ASD> █
```

Pertanyaan:

Di dalam method selection sort, terdapat baris program seperti di bawah ini:

```
int idxMin=i;
for (int j=i+1; j<listMhs.length; j++){
    if (listMhs[j].ipk<listMhs[idxMin].ipk){
        idxMin=j;
    }
}
```

Untuk apakah proses tersebut, jelaskan!

Kode tersebut mencari mahasiswa dengan IPK terkecil dalam sisa daftar menggunakan Selection Sort. Indeks `idxMin` menyimpan posisi IPK terkecil yang ditemukan, lalu elemen tersebut akan ditukar dengan elemen di posisi saat ini agar data terurut.

3. Insertion sort

- **Code program**

```
void insertionSort(){
    for ( int i=1; i<listMhs.length;i++){
        Mahasiswa24 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk>temp.ipk){
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

- **Code program di mahasiswademmo**

```
System.out.println("Data yang sudah terurut menggunakan
INSERTION SORT (ASC)");
list.insertionSort();
list.tampil();
```

- **Output**

```
Data yang sudah terurut menggunakan INSERTION SORT (ASC)
Nama : dika
NIM : 222
kelas : 2c
IPK : 3.0
-----
Nama : susi
NIM : 444
kelas : 2c
IPK : 3.1
-----
Nama : yayuk
NIM : 555
kelas : 2c
IPK : 3.4
-----
Nama : ayu
NIM : 111
kelas : 2c
IPK : 3.7
-----
Nama : ila
NIM : 333
kelas : 2c
IPK : 3.8
-----
PS D:\kuliah\PRAKTIKUM-ASD> █
```

Pertanyaan

Ubahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting dengan cara descending.

```
void insertionSort(){
    for ( int i=1; i<listMhs.length;i++){
        Mahasiswa24 temp = listMhs[i];
        int j=i;
        while (j>0 && listMhs[j-1].ipk<temp.ipk){
            listMhs[j] = listMhs[j-1];
            j--;
        }
        listMhs[j] = temp;
    }
}
```

Latihan Praktikum

- Dosen24.java

```
package jobsheet6;
public class Dosen24 {
    String kode;
    String nama;
    boolean jenisKelamin;
    int usia;

    public Dosen24(String kode, String nama, boolean jk,
int usia) {
        this.kode = kode;
        this.nama = nama;
        this.jenisKelamin = jk;
        this.usia = usia;
    }

    public void tampil() {
        System.out.println();
        System.out.println("Kode : " + kode);
        System.out.println("Nama : " + nama);
        System.out.println("Jenis Kelamin : " +
(jenisKelamin ? "Laki-laki" : "Perempuan"));
        System.out.println("Usia : " + usia);
        System.out.println("-----");
    }
}
```

- DataDosen24.java

```
package jobsheet6;
public class DataDosen24 {
    Dosen24[] dataDosen = new Dosen24[10];
    int idx = 0;

    void tambah(Dosen24 dsn) {
        if (idx < dataDosen.length) {
            dataDosen[idx] = dsn;
            idx++;
        } else {
            System.out.println("Data Dosen sudah penuh!");
        }
    }

    void tampil() {
        if (idx == 0) {
            System.out.println("Belum ada data.");
        } else {
            for (int i = 0; i < idx; i++) {
                dataDosen[i].tampil();
            }
        }
    }

    void sortingASC() {
        for (int i = 0; i < idx - 1; i++) {
            for (int j = 0; j < idx - 1 - i; j++) {
                if (dataDosen[j].usia > dataDosen[j + 1].usia) {
                    Dosen24 temp = dataDosen[j];
                    dataDosen[j] = dataDosen[j + 1];
                    dataDosen[j + 1] = temp;
                }
            }
        }
    }
}
```

```

void sortingDSC() {
    for (int i = 0; i < idx - 1; i++) {
        int maxIndex = i;
        for (int j = i + 1; j < idx; j++) {
            if (dataDosen[j].usia > dataDosen[maxIndex].usia) {
                maxIndex = j;
            }
        }
        Dosen24 temp = dataDosen[maxIndex];
        dataDosen[maxIndex] = dataDosen[i];
        dataDosen[i] = temp;
    }
}

void insertionSort() {
    for (int i = 1; i < idx; i++) {
        Dosen24 temp = dataDosen[i];
        int j = i;
        while (j > 0 && dataDosen[j - 1].usia < temp.usia) {
            dataDosen[j] = dataDosen[j - 1];
            j--;
        }
        dataDosen[j] = temp;
    }
}
}
}

```

- **Main**

```

package jobsheet6;
import java.util.Scanner;
public class DosenMain24 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        DataDosen24 listDosen = new DataDosen24();

        while (true) {
            System.out.println("===== Menu =====");
            System.out.println("1. Tambahkan Data Dosen");
            System.out.println("2. Tampilkan Data Dosen");
            System.out.println("3. Urutkan Data Bubble Sort (ASC)");
            System.out.println("4. Urutkan Data Selection Sort (DSC)");
            System.out.println("5. Urutkan Data Insertion Sort (DSC)");
            System.out.println("0. Keluar");
            System.out.print("Pilih menu: ");
            int pilihan = sc.nextInt();
            sc.nextLine();

            switch (pilihan) {
                case 1:
                    System.out.println();
                    for (int i = 0; i < 10; i++) {
                        System.out.print("Kode : ");
                        String kode = sc.nextLine();
                        System.out.print("Nama : ");
                        String nama = sc.nextLine();
                        System.out.print("Jenis Kelamin (true = Laki-laki, false =
Perempuan) : ");
                        ();
                    }
                }
            }
        }
    }
}

```

```

boolean jk = sc.nextBoolean();
        System.out.print("Usia : ");
        int usia = sc.nextInt();
        sc.nextLine();
        System.out.println();
        Dosen24 dsn = new Dosen24(kode, nama, jk, usia);
        listDosen.tambah(dsn);
    }
    break;
case 2:
    System.out.println();
    System.out.println("=== Data Keseluruhan ===");
    listDosen.tampil();
    break;
case 3:
    System.out.println();
    System.out.println("Data telah diurutkan (ASC - Bubble Sort).");
    listDosen.sortingASC();
    listDosen.tampil();
    break;
case 4:
    System.out.println();
    System.out.println("Data telah diurutkan (DSC - Selection
Sort).");

    listDosen.sortingDSC();
    listDosen.tampil();
    break;
case 5:
    System.out.println();
    System.out.println("Data telah diurutkan (DSC - Insertion
Sort).");

    listDosen.insertionSort();
    listDosen.tampil();
    break;
case 0:
    System.out.println("Keluar dari program.");
    sc.close();
    return;
default:
    System.out.println("Pilihan tidak valid.");
}
    }
}
}

```


