Laporan Hasil Pratikum

Algoritma Dan Struktur Data

Jobsheet 6



|  |  |
| --- | --- |
| Nama : | Zacky Rio Orlando |
| NIM : | 244107020086 |
| Kelas : | 1E |

Program Studi D-IV Teknik Informatika

Jurusan Teknologi Informasi

Praktikum

2025

6.6 Praktikum 1 - Mengimplementasikan Sorting menggunakan object

6.2.1 Langkah Praktikum 1a. SORTING – BUBBLE SORT

Class Sorting27

public class Sorting27 {

int [] data;

int jumData;

Sorting27 (int Data[], int jmlDat) {

jumData = jmlDat;

data = new int[jmlDat];

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();

}

}

Class SortingMain27

public class SortingMain27 {

public static void main(String[] args) {

int a[] = {20, 10, 2, 7, 12};

Sorting27 dataurut1 = new Sorting27 (a, a.length);

System.out.println("Data awal 1");

dataurut1.tampil();

dataurut1.bubbleSort();

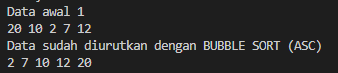
System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");

dataurut1.tampil();

}

}

Hasil Output dari kode program diatas



b. SORTING – SELECTION SORT

Class Sorting27

public class Sorting27 {

int [] data;

int jumData;

Sorting27 (int Data[], int jmlDat) {

jumData = jmlDat;

data = new int[jmlDat];

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; 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;

}

}

}

Class SortingMain27

public class SortingMain27 {

public static void main(String[] args) {

int a[] = {20, 10, 2, 7, 12};

int b[] = {30, 20, 2, 8, 14};

Sorting27 dataurut1 = new Sorting27 (a, a.length);

Sorting27 dataurut2 = new Sorting27 (b, b.length);

System.out.println("Data awal 1");

dataurut1.tampil();

dataurut1.bubbleSort();

System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");

dataurut1.tampil();

System.out.println("Data awal 2");

dataurut2.tampil();

dataurut2.SelectionSort();

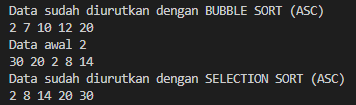
System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");

dataurut2.tampil();

}

}

Hasil Output kode program diatas



c. SORTING – INSERTION SORT

Class Sorting27

public class Sorting27 {

int [] data;

int jumData;

Sorting27 (int Data[], int jmlDat) {

jumData = jmlDat;

data = new int[jmlDat];

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; 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;

}

}

}

Class SortingMain27

public class SortingMain27 {

public static void main(String[] args) {

int a[] = {20, 10, 2, 7, 12};

int b[] = {30, 20, 2, 8, 14};

int c[] = {40, 10, 4, 9, 3};

Sorting27 dataurut1 = new Sorting27 (a, a.length);

Sorting27 dataurut2 = new Sorting27 (b, b.length);

Sorting27 dataurut3 = new Sorting27 (c, c.length);

System.out.println("Data awal 1");

dataurut1.tampil();

dataurut1.bubbleSort();

System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");

dataurut1.tampil();

System.out.println("Data awal 2");

dataurut2.tampil();

dataurut2.SelectionSort();

System.out.println("Data sudah diurutkan dengan SELECTION SORT (ASC)");

dataurut2.tampil();

System.out.println("Data awal 3");

dataurut3.tampil();

dataurut3.insertionSort();

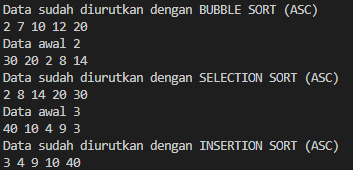
System.out.println("Data sudah diurutkan dengan INSERTION SORT (ASC)");

dataurut3.tampil();

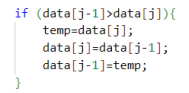
}

}

Hasil Output kode program diatas



6.2.5 Pertanyaan!1. Jelaskan fungsi kode program berikut



Jawaban : Kode diatas digunakan untuk **Bubble Sort** buat menukar posisi 2 angka jika urutannya salah. Jika angka di kiri lebih besar daripada angka di kanan, maka kedua angka itu ditukar agar angka yang lebih kecil berpindah ke depan. Proses ini dilakukan berulang kali hingga semua angka tersusun dari yang terkecil hingga terbesar.

2. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum padaselection sort!

int min=i;

for (int j=i+1; j<jumData; j++) {

if (data[j] < data[min]) {

min = j;

}

}

3. Pada Insertion sort , jelaskan maksud dari kondisi pada perulangan

****

Jawaban :

Potongan kode diatas berfungsi untuk menggeser angka yang lebih besar ke kanan agar ada tempat untuk menyisipkan angka baru. Syarat j > 0 untuk tidak keluar dari batas array, sedangkan data[j] > temp untuk mengecek apakah angka sebelumnya lebih besar dari temp atau tidak. Jika iya, angka itu akan digeser ke kanan. Proses ini terus berjalan sampai ditemukan angka yang lebih kecil atau sampai j sudah tidak ada lagi, lalu temp dimasukkan ke posisi yang tepat sehingga datanya tetap urut.

4. Pada Insertion sort, apakah tujuan dari perintah 

Jawaban :

Perintah kode diatas digunakan untuk **menggeser angka ke kanan** agar ada tempat untuk menyisipkan angka baru. Jika ada angka yang lebih besar dari temp, angka tersebut dipindahkan ke posisi setelahnya (j+1). Proses ini terus berlanjut sampai ditemukan angka yang lebih kecil atau sampai tidak ada lagi angka yang digeser dan Angka baru bisa dimasukkan ke posisi yang tepat tanpa merusak urutan data.

6.7 Praktikum 2- (Sorting Menggunakan Array of Object)

6.3.1 Langkah Praktikum 2 - Mengurutkan Data Mahasiswa Berdasarkan IPK (Bubble Sort)

Class Mahasiswa27

public class Mahasiswa27 {

String nim;

String nama;

String kelas;

double ipk;

Mahasiswa27() {

}

Mahasiswa27(String nm, String name, String kls, double ip) {

nim = nm;

nama = name;

ipk = ip;

kelas = kls;

}

void tampilInformasi() {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("Kelas: " + kelas);

System.out.println("IPK: " + ipk);

}

}

Class MahasiswaBerprestasi27

public class MahasiswaBerprestasi27 {

Mahasiswa27 [] listMhs = new Mahasiswa27[5];

int idx;

void tambah (Mahasiswa27 m) {

if (idx < listMhs.length) {

listMhs[idx] = m;

idx++;

} else {

System.out.println("data sudah penuh");

}

}

void tampil () {

for (Mahasiswa27 m : listMhs) {

m.tampilInformasi();

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) {

Mahasiswa27 tmp = listMhs[j];

listMhs[j] = listMhs[j-1];

listMhs[j-1] = tmp;

}

}

}

}

}

Class MahasiswaDemo27

public class MahasiswaDemo27 {

public static void main(String[] args) {

MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27();

Mahasiswa27 m1 = new Mahasiswa27("123", "Zidan", "2A", 3.2);

Mahasiswa27 m2 = new Mahasiswa27("124", "Ayu", "2A", 3.5);

Mahasiswa27 m3 = new Mahasiswa27("125", "Sofi", "2A", 3.1);

Mahasiswa27 m4 = new Mahasiswa27("126", "Sita", "2A", 3.9);

Mahasiswa27 m5 = new Mahasiswa27("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 (DECS) : ");

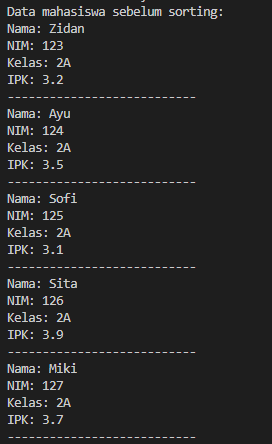
list.bubbleSort();

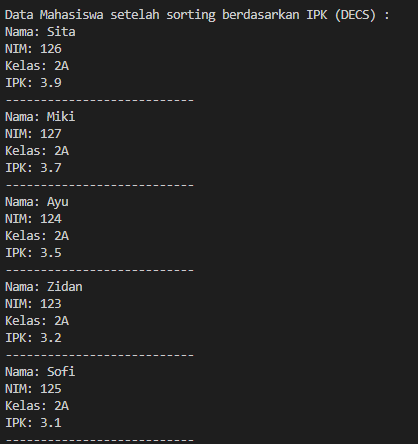
list.tampil();

}

}

Hasil dari output kode program diatas





6.3.4 Pertanyaan1. Perhatikan perulangan di dalam bubbleSort() di bawah ini:

  
a. Mengapa syarat dari perulangan i adalah i<listMhs.length-1 ?

Jawaban :

Karena dalam **bubble sort**, kita hanya perlu melakukan **(n - 1) kali** perulangan untuk mengurutkan semua data. Setiap perulangan akan menempatkan satu elemen terbesar ke posisi akhir, jadi tidak perlu melakukan perulangan hingga listMhs.length, karena saat elemen kedua terakhir sudah berada di posisi yang benar, elemen terakhir otomatis sudah urut.

b. Mengapa syarat dari perulangan j adalah j<listMhs.length-i ?

Jawaban :

Agar elemen yang sudah berada di posisi yang benar tidak ikut dibandingkan lagi. Setiap perulangan i, satu elemen terbesar sudah ditempatkan di posisi akhirnya, jadi jumlah elemen yang perlu dibandingkan semakin berkurang. Dengan cara ini, proses bubble sort menjadi lebih efisien karena tidak membuang waktu untuk mengecek elemen yang sudah urut.

c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan  
berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?

Jawaban :

Maka perulangan i akan berlangsung sebanyak 49 kali (i = 0 sampai i = 48) karena bubble sort membutuhkan **(n - 1) tahap** untuk mengurutkan seluruh data. Jadi, ada **49 tahap bubble sort**, di mana setiap tahap memastikan satu elemen terbesar sudah berada di posisi yang benar sampai akhirnya seluruh data urut.

2. Modifikasi program diatas dimana data mahasiswa bersifat dinamis (input dari keyborad)  
yang terdiri dari nim, nama, kelas, dan ipk!

Jawaban :

Class Mahasiswa27

public class Mahasiswa27 {

String nim;

String nama;

String kelas;

double ipk;

Mahasiswa27() {

}

Mahasiswa27(String nm, String name, String kls, double ip) {

nim = nm;

nama = name;

ipk = ip;

kelas = kls;

}

void tampilInformasi() {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("Kelas: " + kelas);

System.out.println("IPK: " + ipk);

}

}

Class MahasiswaBerprestasi27

public class MahasiswaBerprestasi27 {

Mahasiswa27 [] listMhs = new Mahasiswa27[5];

int idx;

void tambah (Mahasiswa27 m) {

if (idx < listMhs.length) {

listMhs[idx] = m;

idx++;

} else {

System.out.println("data sudah penuh");

}

}

void tampil () {

for (Mahasiswa27 m : listMhs) {

m.tampilInformasi();

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) {

Mahasiswa27 tmp = listMhs[j];

listMhs[j] = listMhs[j-1];

listMhs[j-1] = tmp;

}

}

}

}

MahasiswaBerprestasi27(int jumlah) {

listMhs = new Mahasiswa27[jumlah];

idx = 0;

}

}

Class MahasiswaDemo27

import java.util.Scanner;

public class MahasiswaDemo27 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Masukkan jumlah mahasiswa: ");

int jumlah = scanner.nextInt();

scanner.nextLine();

MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27(jumlah);

for (int i = 0; i < jumlah; i++) {

System.out.println("Masukkan data mahasiswa ke-" + (i + 1) + ":");

System.out.print("NIM: ");

String nim = scanner.nextLine();

System.out.print("Nama: ");

String nama = scanner.nextLine();

System.out.print("Kelas: ");

String kelas = scanner.nextLine();

System.out.print("IPK: ");

double ipk = scanner.nextDouble();

scanner.nextLine();

Mahasiswa27 m = new Mahasiswa27(nim, nama, kelas, ipk);

list.tambah(m);

}

System.out.println("\nData mahasiswa sebelum sorting:");

list.tampil();

System.out.println("Data mahasiswa setelah sorting berdasarkan IPK (DESC):");

list.bubbleSort();

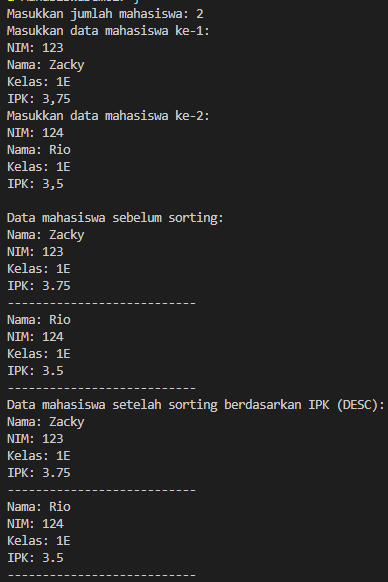
list.tampil();

scanner.close();

}

}

Hasil Output kode program diatas



6.4 Mengurutkan Data Mahasiswa Berdasarkan IPK (Selection Sort)

Class Mahasiswa27

public class Mahasiswa27 {

String nim;

String nama;

String kelas;

double ipk;

Mahasiswa27() {

}

Mahasiswa27(String nm, String name, String kls, double ip) {

nim = nm;

nama = name;

ipk = ip;

kelas = kls;

}

void tampilInformasi() {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("Kelas: " + kelas);

System.out.println("IPK: " + ipk);

}

}

Class MahasiswaBeprestasi27

public class MahasiswaBerprestasi27 {

Mahasiswa27 [] listMhs = new Mahasiswa27[5];

int idx;

void tambah (Mahasiswa27 m) {

if (idx < listMhs.length) {

listMhs[idx] = m;

idx++;

} else {

System.out.println("data sudah penuh");

}

}

void tampil () {

for (Mahasiswa27 m : listMhs) {

m.tampilInformasi();

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) {

Mahasiswa27 tmp = listMhs[j];

listMhs[j] = listMhs[j-1];

listMhs[j-1] = tmp;

}

}

}

}

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;

}

}

Mahasiswa27 tmp = listMhs[idxMin];

listMhs[idxMin] = listMhs[i];

listMhs[i] = tmp;

}

}

}

Class MahasiswaDemo27

public class MahasiswaDemo27 {

public static void main(String[] args) {

MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27();

Mahasiswa27 m1 = new Mahasiswa27("123", "Ali", "2B", 3.9);

Mahasiswa27 m2 = new Mahasiswa27("124", "ila", "2B", 3.1);

Mahasiswa27 m3 = new Mahasiswa27("125", "agus", "2B", 3.6);

Mahasiswa27 m4 = new Mahasiswa27("126", "tika", "2B", 3.3);

Mahasiswa27 m5 = new Mahasiswa27("127", "udin", "2B", 3.2);

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 (DECS) : ");

list.bubbleSort();

list.tampil();

System.out.println("Data yang sudah terurut menggunakan SELECTION SORT (ASC)");

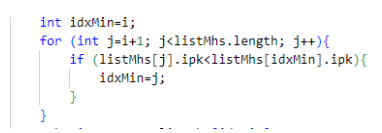
list.selectionSort();

list.tampil();

}

}

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

  
Untuk apakah proses tersebut, jelaskan!

Jawaban :  
Untuk mencari mahasiswa dengan **ipk terkecil** dalam bagian data yang belum urut. Pertama, idxMin menyimpan posisi awal sebagai nilai terkecil sementara. Lalu, perulangan for memeriksa semua elemen setelahnya untuk melihat apakah ada ipk yang lebih kecil. Jika ada, posisi idxMin diperbarui ke indeks baru yang memiliki ipk lebih kecil. Setelah perulangan selesai, idxMin akan menunjukkan posisi mahasiswa dengan ipk terkecil, yang kemudian akan ditukar dengan posisi awal agar proses **selection sort** bisa berjalan dengan benar.

6.5 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort

Class Mahasiswa27

public class Mahasiswa27 {

String nim;

String nama;

String kelas;

double ipk;

Mahasiswa27() {

}

Mahasiswa27(String nm, String name, String kls, double ip) {

nim = nm;

nama = name;

ipk = ip;

kelas = kls;

}

void tampilInformasi() {

System.out.println("Nama: " + nama);

System.out.println("NIM: " + nim);

System.out.println("Kelas: " + kelas);

System.out.println("IPK: " + ipk);

}

}

Class MahasiswaBerprestasi27

public class MahasiswaBerprestasi27 {

Mahasiswa27 [] listMhs = new Mahasiswa27[5];

int idx;

void tambah (Mahasiswa27 m) {

if (idx < listMhs.length) {

listMhs[idx] = m;

idx++;

} else {

System.out.println("data sudah penuh");

}

}

void tampil () {

for (Mahasiswa27 m : listMhs) {

m.tampilInformasi();

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) {

Mahasiswa27 tmp = listMhs[j];

listMhs[j] = listMhs[j-1];

listMhs[j-1] = tmp;

}

}

}

}

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;

}

}

Mahasiswa27 tmp = listMhs[idxMin];

listMhs[idxMin] = listMhs[i];

listMhs[i] = tmp;

}

}

void insertionSort() {

for (int i=1; i<listMhs.length; i++) {

Mahasiswa27 temp = listMhs[i];

int j=i;

while (j>0 && listMhs[j-1].ipk>temp.ipk) {

listMhs[j] = listMhs[j-1];

j--;

}

listMhs[j] = temp;

}

}

}

Class MahasiswaDemo27

public class MahasiswaDemo27 {

public static void main(String[] args) {

MahasiswaBerprestasi27 list = new MahasiswaBerprestasi27();

Mahasiswa27 m1 = new Mahasiswa27("111", "ayu", "2c", 3.7);

Mahasiswa27 m2 = new Mahasiswa27("222", "dika", "2c", 3.0);

Mahasiswa27 m3 = new Mahasiswa27("333", "ila", "2c", 3.8);

Mahasiswa27 m4 = new Mahasiswa27("444", "susi", "2c", 3.1);

Mahasiswa27 m5 = new Mahasiswa27("555", "yayuk", "2c", 3.4);

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 (DECS) : ");

list.bubbleSort();

list.tampil();

System.out.println("Data yang sudah terurut menggunakan SELECTION SORT (ASC)");

list.selectionSort();

list.tampil();

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

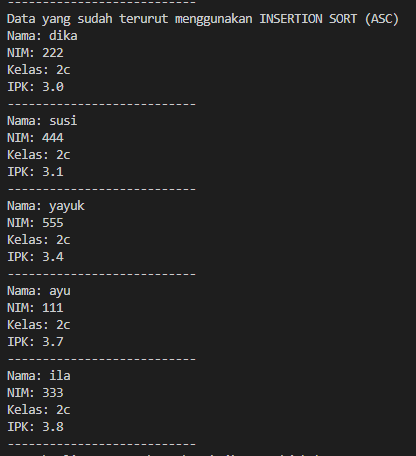
list.insertionSort();

list.tampil();

}

}

Hasil Output dari kode program diatas

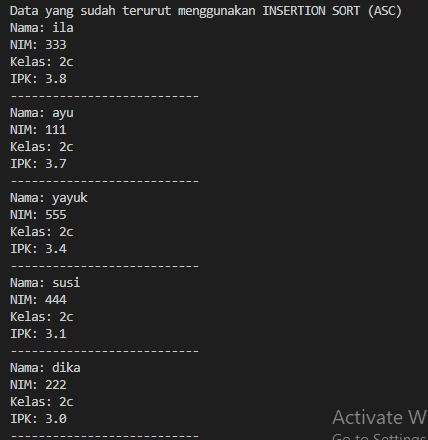


6.5.3 PertanyaanUbahlah fungsi pada InsertionSort sehingga fungsi ini dapat melaksanakan proses sorting  
dengan cara descending

Jawaban :

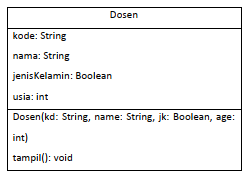
Mengubah kondisi dalam while dari listMhs[j-1].ipk > temp.ipk menjadi listMhs[j-1].ipk < temp.ipk

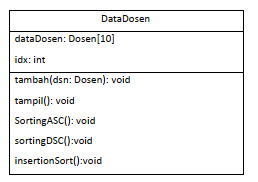
Hasil Outputnya



6.6 Latihan Praktikum

Perhatikan class diagram dibawah ini:





Berdasarkan class diagram diatas buatlah menu dikelas main dengan pilihan menu:1. Tambah data digunakan untuk menambahkan data dosen2. Tampil data digunakan untuk menampilkan data seluruh dosen3. Sorting ASC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai daridosen termuda ke dosen tertua menggunakan bublle Sort.4. Sorting DSC digunakan untuk mengurutkan data dosen berdasarkan usia dimulai daritertua ke dosen termuda dapat menggunakan algoritma selection sort atau insertionsort

Jawaban :  
Class Dosen27

public class Dosen27 {

String kode;

String nama;

boolean jenisKelamin;

int usia;

Dosen27(){

}

Dosen27(String kd, String name, boolean jk, int age) {

kode = kd;

nama = name;

usia = age;

this.jenisKelamin = jk;

}

void tampil() {

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("------------------------");

}

}

Class DataDosen27

public class DataDosen27 {

Dosen27[] dataDosen27 = new Dosen27[10];

int idx = 0;

void tambah(Dosen27 dsn) {

if (idx < dataDosen27.length) {

dataDosen27[idx] = dsn;

idx++;

} else {

System.out.println("Data penuh!");

}

}

void tampil() {

for (int i = 0; i < idx; i++) {

dataDosen27[i].tampil();

}

}

void sortingASC() {

for (int i = 0; i < idx - 1; i++) {

for (int j = 0; j < idx - i - 1; j++) {

if (dataDosen27[j].usia > dataDosen27[j + 1].usia) {

Dosen27 temp = dataDosen27[j];

dataDosen27[j] = dataDosen27[j + 1];

dataDosen27[j + 1] = temp;

}

}

}

}

void sortingDSC() {

for (int i = 0; i < idx - 1; i++) {

int maxIdx = i;

for (int j = i + 1; j < idx; j++) {

if (dataDosen27[j].usia > dataDosen27[maxIdx].usia) {

maxIdx = j;

}

}

Dosen27 temp = dataDosen27[maxIdx];

dataDosen27[maxIdx] = dataDosen27[i];

dataDosen27[i] = temp;

}

}

}

Class DosenMain27

import java.util.Scanner;

public class DosenMain27 {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

DataDosen27 data27 = new DataDosen27();

int pilihan;

do {

System.out.println("\nMENU");

System.out.println("1. Tambah Data Dosen");

System.out.println("2. Tampilkan Data Dosen");

System.out.println("3. Sorting Usia (ASC)");

System.out.println("4. Sorting Usia (DSC)");

System.out.println("5. Keluar");

System.out.print("Pilih menu: ");

pilihan = sc.nextInt();

sc.nextLine();

switch (pilihan) {

case 1:

System.out.print("Masukkan Kode Dosen: ");

String kode = sc.nextLine();

System.out.print("Masukkan Nama Dosen: ");

String nama = sc.nextLine();

System.out.print("Masukkan Jenis Kelamin: ");

String inputJK = sc.nextLine();

boolean jk = inputJK.equalsIgnoreCase("laki-laki");

System.out.print("Masukkan Usia: ");

int usia = sc.nextInt();

data27.tambah(new Dosen27(kode, nama, jk, usia));

break;

case 2:

data27.tampil();

break;

case 3:

data27.sortingASC();

System.out.println("Data telah diurutkan dari termuda ke tertua.");

break;

case 4:

data27.sortingDSC();

System.out.println("Data telah diurutkan dari tertua ke termuda.");

break;

case 5:

System.out.println("Keluar dari program.");

break;

default:

System.out.println("Pilihan tidak valid!");

}

} while (pilihan != 5);

}

}

Hasil Output kode program diatas

