POLITEKNIK NEGERI MALANG

TEKNOLOGI INFORMASI

TEKNIK INFORMATIKA



#### Nama : Muhammad Nuril Huda

#### Kelas : TI-1A

#### No : 19

#### Mata Kuliah : Algoritma dan Struktur Data

## 5.2 Praktikum 1 - Mengimplementasikan Sorting menggunakan object

##### 5.2.1 Kode Program

* Kode Program Sorting19

package Pertemuan6;

public class Sorting19 {

    int [] data;

    int jumData;

    Sorting19 (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-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;

        }

    }

}

* Kode Program SortingMain19

package Pertemuan6;

public class SortingMain19 {

    public static void main(String[] args) {

        int a[] = {20, 10, 2, 7, 12};

        Sorting19 dataurut1 = new Sorting19(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};

        Sorting19 dataurut2 = new Sorting19(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, 10, 4, 9, 3};

        Sorting19 dataurut3 = new Sorting19(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();

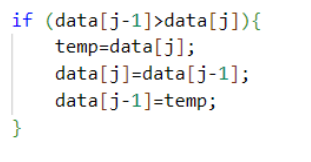
    }

}

##### 5.2.2 Hasil Percobaan

##### 5.2.3 Pertanyaan

1. Jelaskan Fungsi kode program berikut



* Maksud dari kode program tersebut adalah menukar nilai data[j] dengan data[j-1] menggunakan bantuan variable temp untuk menyimpan data sementara.

1. Tunjukkan kode program yang merupakan algoritma pencarian nilai minimum pada selection sort!

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;

        }

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



* Kondisi ini digunakan untuk menggeser elemen yang lebih besar dari temp ke kanan agar temp dapat dimasukkan ke posisi yang benar. j >= 0 memastikan bahwa perulangan tidak keluar dari batas kiri array, sementara data[j] > temp memastikan elemen yang lebih besar dari temp digeser. Proses ini berlanjut hingga ditemukan posisi yang tepat untuk menempatkan temp, sehingga array tetap dalam keadaan terurut selama iterasi.

1. Pada Insertion sort, apakah tujuan dari perintah



* Perintah ini digunakan untuk menggeser angka ke kanan agar ada tempat bagi angka yang akan dimasukkan. Jika angka sebelumnya lebih besar dari angka yang sedang diproses, angka tersebut dipindahkan ke kanan. Proses ini terus dilakukan sampai ditemukan posisi yang tepat untuk angka yang akan disisipkan.

## 5.3 Praktikum 2- (Sorting Menggunakan Array of Object)

##### 5.3.1 Kode Program (Bubble Sort)

* Kode Program Mahasiswa19

package Pertemuan6;

public class Mahasiswa19 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    // Konstruktor default

    Mahasiswa19 () {

    }

    // Konstruktor berparameter (dibuat ada yang nama var parameter inputnya sama ada yang tidak)

    Mahasiswa19(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);

    }

}

* Kode Program MahasiswaBerprestasi19

package Pertemuan6;

public class MahasiswaBerprestasi19 {

    Mahasiswa19 [] listMhs = new Mahasiswa19[5];

    int idx;

    void tambah (Mahasiswa19 m){

        if (idx<listMhs.length){

            listMhs[idx]=m;

            idx++;

        }else {

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

        }

    }

    void tampil (){

        for (Mahasiswa19 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){

                    Mahasiswa19 tmp = listMhs[j];

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

                    listMhs[j-1]=tmp;

                }

            }

        }

    }

}

* Kode Program MahasiswaDemo19

package Pertemuan6;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19();

        Mahasiswa19 m1 = new Mahasiswa19 ("123", "Zidan", "2A",3.2);

        Mahasiswa19 m2 = new Mahasiswa19 ("124", "Ayu", "2A",3.5);

        Mahasiswa19 m3 = new Mahasiswa19 ("125", "Sofi", "2A",3.1);

        Mahasiswa19 m4 = new Mahasiswa19 ("126", "Sita", "2A",3.9);

        Mahasiswa19 m5 = new Mahasiswa19 ("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();

    }

}

##### 5.3.2 Hasil Percobaan (Bubble Sort)

##### 5.3.3 Pertanyaan (Bubble Sort)

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



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

* Karena setiap Langkah BubleSort decending akan menempatkan mahasiswa dengan IPK terbesar ke posisi indeks paling kiri. Setalah listMhs.length-1 langkah, semua mahasiswa sudah terurut berdasarkan IPK, sehingga tidak perlu melakukan perulangan sebeanyak listMhs.length.

1. Mengapa syarat dari perulangan j adalah j<Mhs.length-i?

* Karena setiap kali perulangan i berjalan, mahasiswa dengan IPK tertinggi akan berada diposisi yang sudah benar. Oleh karena itu, jumlah perbandingan berkurang setiap kali i bertambah agar tidak memeriksa IPK yang sudah benar posisinya.

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

* Perulangan i akan berlangsung sebanyak 49 kali, karena i < 50 - 1.
* Bubble sort akan menempuh 49 tahap, karena setiap tahap menempatkan satu mahasiswa dengan IPK tertinggi ke posisi akhirnya hingga seluruh daftar mahasiswa terurut.

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

package Pertemuan6;

public class MahasiswaBerprestasi19 {

    Mahasiswa19[] listMhs;

    int idx;

    MahasiswaBerprestasi19(int jumlah) {

        listMhs = new Mahasiswa19[jumlah];

        idx = 0;

    }

package Pertemuan6;

import java.util.Scanner;

public class MahasiswaDemo19 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int jumlah = sc.nextInt();

        sc.nextLine();

        MahasiswaBerprestasi19 list = new MahasiswaBerprestasi19(jumlah);

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

            System.out.println("\nMasukkan data mahasiswa ke-" + (i + 1));

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

            String nama = sc.nextLine();

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

            String nim = sc.nextLine();

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

            String kelas = sc.nextLine();

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

            double ipk = sc.nextDouble();

            sc.nextLine();

            list.tambah(new Mahasiswa19(nama, nim, kelas, ipk));

        }

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

        list.tampil();

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

        list.bubbleSort();

        list.tampil();

    }

}

##### 5.3.4 Kode Program (Selection Sort)

* Kode Program MahasiswaBerprestasi19

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;

                }

            }

            Mahasiswa19 tmp = listMhs[idxMin];

            listMhs[idxMin]=listMhs[i];

            listMhs[i]=tmp;

        }

    }

* Kode Program MahasiswaDemo19

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

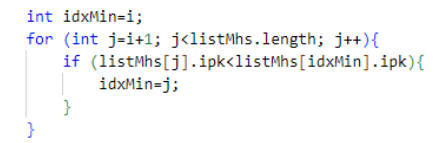
        list.selectionSort();

        list.tampil();

##### 5.3.5 Hasil Percobaan (Selection Sort)

##### 5.3.6 Pertanyaan (Selection Sort)

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



Untuk apakah proses tersebut, jelaskan!

* Kode tersebut digunakan untuk mencari mahasiswa dengan IPK terkecil dalam daftar yang belum diurutkan. Pertama, indeks awal (idxMin) diset sebagai nilai terkecil sementara. Kemudian, perulangan dilakukan untuk membandingkan IPK mahasiswa lainnya yang berada setelah posisi tersebut. Jika ditemukan mahasiswa dengan IPK lebih kecil, maka indeksnya disimpan ke dalam idxMin. Setelah proses pencarian selesai, idxMin akan berisi indeks mahasiswa dengan IPK terkecil, yang nantinya akan ditukar dengan posisi awal. Proses ini dilakukan berulang hingga seluruh data tersusun dari IPK terkecil hingga terbesar.

## 5.4 Mengurutkan Data Mahasiswa Berdasarkan IPK Menggunakan Insertion Sort Waktu

##### 5.4.1 Kode Program

* Kode Program MahasiswaBerprestasi19

void insertionSort(){

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

            Mahasiswa19 temp = listMhs[i];

            int j=i;

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

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

                j--;

            }

            listMhs[j]=temp;

        }

    }

* Kode Program MahasiswaDemo19