**POLITEKNIK NEGERI MALANG**

**TEKNOLOGI INFORMASI**

**TEKNIK INFORMATIKA**

****

**Mohammad Ariq Baihaqi**

**244107020161**

**TI – 1A**

**16**

**5.2.1 Langkah Praktikum 1**

**a. SORTING – BUBBLE SORT**

package minggu6;

public class Sorting16 {

    int [] data;

    int jumData;

    Sorting16(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; i++){

                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();

}

}

package minggu6;

public class SortingMain16 {

    public static void main(String[] args) {

        int[] arr = { 5, 2, 8, 1, 9};

        int a[]= {20, 10, 2, 7, 12};

        Sorting16 dataurut1 = new Sorting16(a, 0);

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

        dataurut1.tampil();

        dataurut1.bubbleSort();

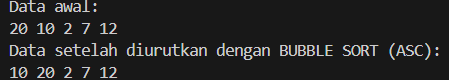
        System.out.println("Data sudah diurutkan dengan BUBBLE SORT (ASC)");

        dataurut1.tampil();

        }

}

**OUTPUT**

****

**b. SORTING – SELECTION SORT**

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;

        }

    }

Sorting16 dataurut2 = new Sorting16(b, 5);

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

        dataurut2.tampil();

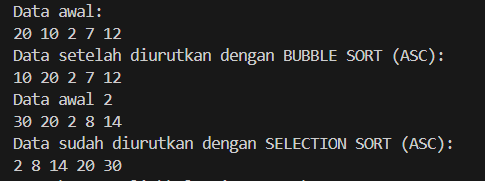
        dataurut2.SelectionSort();

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

        dataurut2.tampil();

    }

**OUTPUT**

****

**c. SORTING – INSERTION SORT**

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;

        }

    }

Sorting16 dataurut3 = new Sorting16(c, 5);

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

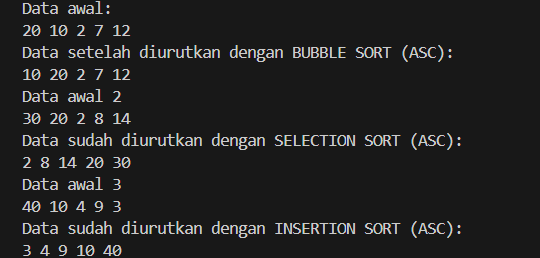
        dataurut3.tampil();

        dataurut3.insertionSort();

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

        dataurut3.tampil();

**OUTPUT**

****

**5.2.5 Pertanyaan!**

1. **Jelaskan fungsi kode program berikut**

**If (data[j-1]>data[j]){**

* 1. **Temp=data[j];**
  2. **Data[j]=data[j-1];**
  3. **Data[j-1]=temp;**

**}**

2. **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**

**While (j>=0 && data[i]>temp)**

* J >= 0 memastikan perulangan tidak keluar dari batas array
* Data[j] > temp memeriksa apakah elemen sebelumnya lebih besar dari temp

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

* Digunakan untuk menggeser elemen ke kanan untuk memberi ruang bagi elemen yang sedang di sisipkan

**5.3 Praktikum 2**

package minggu6;

public class Mahasiwa16 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    // konstruktor default

    Mahasiwa16() {

    }

    // konstruktor berparameter

    Mahasiwa16(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);

    }

}

package minggu6;

public class MahasiswaBerprestasi16 {

    Mahasiwa16 [] listMhs = new Mahasiwa16 [5];

    int idx;

    void tambah (Mahasiwa16 m){

        if (idx<listMhs.length){

            listMhs[idx] = m;

            idx++;

        }else {

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

        }

    }

    void tampilInformasi() {

        for(Mahasiwa16 m:listMhs){

            m.tampilInformasi();

            System.out.println("---------------------------");

        }

    }

    void bubbleSort() {

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

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

                if (listMhs[j].ipk>listMhs[j + 1].ipk) {

                    Mahasiwa16 temp = listMhs[j];

                    listMhs[j] = listMhs[j + 1];

                    listMhs[j + 1] = temp;

                }

            }

        }

    }

}

package minggu6;

import minggu2.Mahasiswa16;

public class MahasiswaDemo16 {

    public static void main(String[] args) {

        MahasiswaBerprestasi16 list = new MahasiswaBerprestasi16();

        Mahasiwa16 m1 = new Mahasiwa16("123", "Zidan", "2A", 3.5);

        Mahasiwa16 m2 = new Mahasiwa16("245", "Ilham", "3A", 3.9);

        Mahasiwa16 m3 = new Mahasiwa16("345", "Hanif", "1A", 3.6);

        Mahasiwa16 m4 = new Mahasiwa16("543", "Fika", "3B", 3.7);

        Mahasiwa16 m5 = new Mahasiwa16("666", "Abid", "1H", 2.7);

        list.tambah(m1);

        list.tambah(m2);

        list.tambah(m3);

        list.tambah(m4);

        list.tambah(m5);

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

        list.tampilInformasi();

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

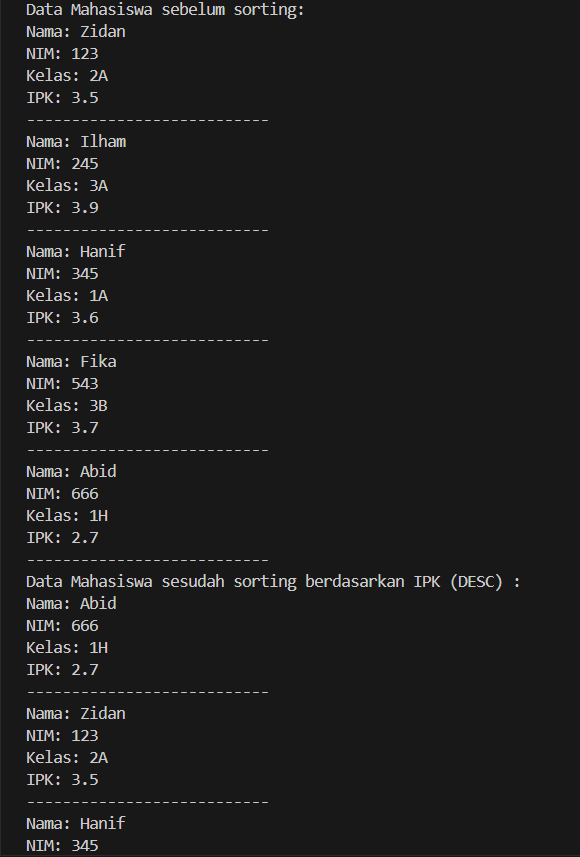
        list.bubbleSort();

        list.tampilInformasi();

    }

}

**OUTPUT**

****

**5.3.4 Pertanyaan**

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<listMhs.length-1 ?**

* Karena setelah n-1 iterasi semua elemen sudah diurutkan

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

* Karena untuk menghindari perbandingan yang tidak perlu, iterasi j dikurangi degan I, karena elemen terakhir sudah terurut

**c. Jika banyak data di dalam listMhs adalah 50, maka berapakali perulangan i akan**

**berlangsung? Dan ada berapa Tahap bubble sort yang ditempuh?**

* N(n-1)/2 = 50(49)/2 = 1225, jadi ada 1225 kali perbandingan selama proses sorting

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

package minggu6;

class Mahasiwa16 {

    String nim;

    String nama;

    String kelas;

    double ipk;

    // Konstruktor berparameter

    Mahasiwa16(String nim, String nama, String kelas, double ipk) {

        this.nim = nim;

        this.nama = nama;

        this.kelas = kelas;

        this.ipk = ipk;

    }

    void tampilInformasi() {

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

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

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

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

        System.out.println("---------------------------");

    }

}

package minggu6;

public class MahasiswaBerprestasi16modif {

    Mahasiwa16[] listMhs;

    int idx = 0;

    MahasiswaBerprestasi16modif(int jumlahMahasiswa) {

        listMhs = new Mahasiwa16[jumlahMahasiswa];

    }

    void tambah(Mahasiwa16 m) {

        if (idx < listMhs.length) {

            listMhs[idx] = m;

            idx++;

        } else {

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

        }

    }

    void tampilInformasi() {

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

            listMhs[i].tampilInformasi();

        }

    }

    void bubbleSort() {

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

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

                if (listMhs[j].ipk < listMhs[j + 1].ipk) {

                    Mahasiwa16 temp = listMhs[j];

                    listMhs[j] = listMhs[j + 1];

                    listMhs[j + 1] = temp;

                }

            }

        }

    }

}

package minggu6;

import java.util.Scanner;

public class MahasiswaDemo16modif {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int jumlahMahasiswa = sc.nextInt();

        sc.nextLine();

        MahasiswaBerprestasi16modif list = new MahasiswaBerprestasi16modif(jumlahMahasiswa);

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

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

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

            String nim = sc.nextLine();

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

            String nama = sc.nextLine();

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

            String kelas = sc.nextLine();

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

            double ipk = sc.nextDouble();

            sc.nextLine();

            Mahasiwa16 mhs = new Mahasiwa16(nim, nama, kelas, ipk);

            list.tambah(mhs);

        }

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

        list.tampilInformasi();

        list.bubbleSort();

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

        list.tampilInformasi();

    }

}

**5.3.1. Langkah-langkah Percobaan**

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;

                }

            }

            Mahasiwa16 temp = listMhs[i];

            listMhs[idxMin] = listMhs[i];

            listMhs[i] = temp;

        }

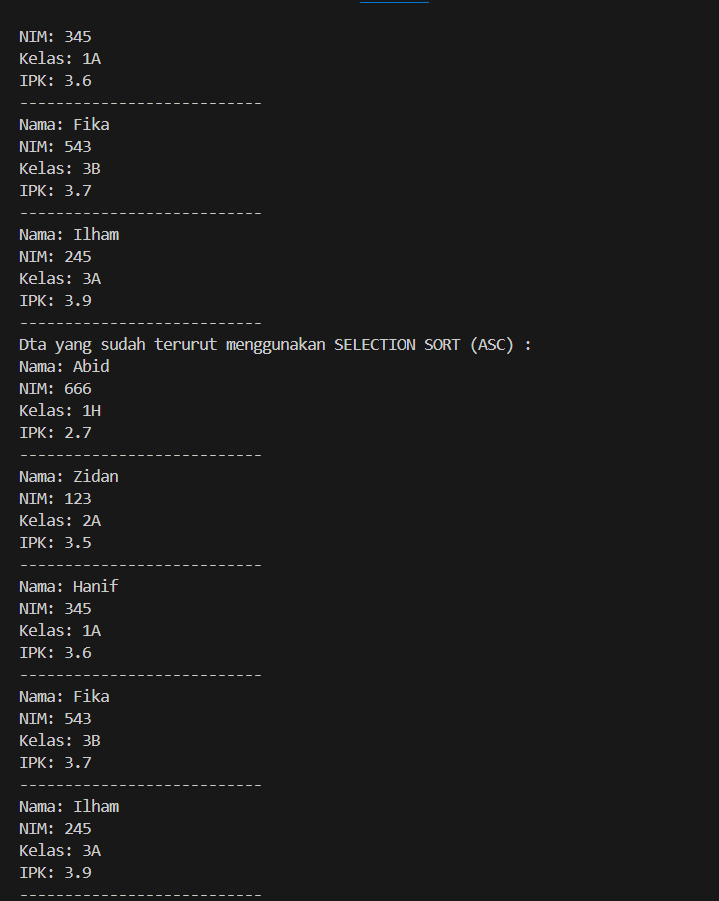
    }

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

        list.selectionSort();

        list.tampilInformasi();

**OUTPUT**

****

**5.3.7 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!**

* idxMin = i; menginisialisasi indeks ke elemen minimum sementara dengan indkes i
* for(int j = i + 1; j < listMhs.length; j++); mencari elemen dengan nilai ipk terkceil di antara elemn-elemen yang belum di urutkan
* if(listMhs[j].ipk < listMhs[idxMin].ipk); elemen pada indeks j adalah kandidat baru unuk nilai minimum

**5.4.1 Langkah-langkah Percobaan**

void insertionSort() {

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

            Mahasiswa16 temp = listMhs[i];

            int j = i;

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

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

                j--;

            }

            listMhs[j] = temp;

        }

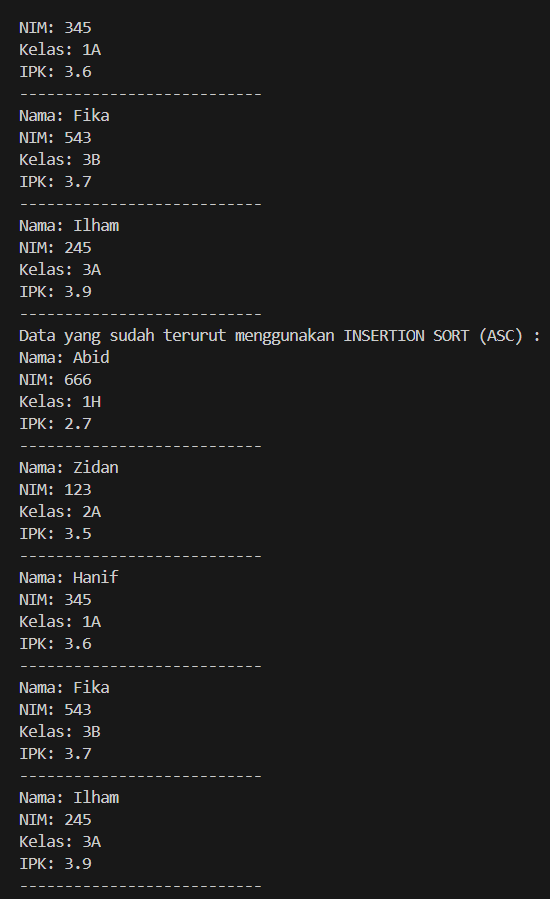
    }

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

        list.insertionSort();

        list.tampilInformasi();

**OUTPUT**

****

**5.4.3 Pertanyaan**

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

void insertionSortDescending() {

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

            Mahasiswa16 temp = listMhs[i];

            int j = i;

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

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

                j--;

            }

            listMhs[j] = temp;

        }

    }

**5.2.5 Pertanyaan! (Lanjutan dari yang atas)**

1. **Jelaskan fungsi kode program berikut**

**If (data[j-1]>data[j]){**

* 1. **Temp=data[j];**
* Menyimpan nilai data[j] dalam variabel sementara (temp) agar tidak hilang saat ditukar.
  1. **Data[j]=data[j-1];**
* Menggeser nilai data[j-1] ke posisi data[j].
  1. **Data[j-1]=temp;**

**}**

* Memasukkan nilai awal data[j] yang tadi disimpan di temp ke posisi data[j-1].

**5.5 Latihan Praktikum**

package minggu6;

public class Dosen16 {

    String kode;

    String nama;

    boolean jenisKelamin;

    int usia;

    public Dosen16(String kd, String name, boolean jk, int age) {

        this.kode = kd;

        this.nama = name;

        this.jenisKelamin = jk;

        this.usia = age;

    }

    public void tampil() {

        System.out.println("Kode: " + kode + ", Nama: " + nama + ", Jenis Kelamin: " + (jenisKelamin ? "Laki-laki" : "Perempuan") + ", Usia: " + usia);

    }

}

class DataDosen {

    Dosen16[] dataDosen = new Dosen16[10];

    int idx = 0;

    public void tambah(Dosen16 dsn) {

        if (idx < dataDosen.length) {

            dataDosen[idx] = dsn;

            idx++;

        } else {

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

        }

    }

    public void tampil() {

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

            dataDosen[i].tampil();

        }

    }

    public void SortingASC() { // Bubble Sort

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

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

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

                    Dosen16 temp = dataDosen[j];

                    dataDosen[j] = dataDosen[j + 1];

                    dataDosen[j + 1] = temp;

                }

            }

        }

    }

    public void sortingDSC() { // Selection Sort

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

            int maxIdx = i;

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

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

                    maxIdx = j;

                }

            }

            Dosen16 temp = dataDosen[maxIdx];

            dataDosen[maxIdx] = dataDosen[i];

            dataDosen[i] = temp;

        }

    }

}

public void SortingASC() { // Bubble Sort

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

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

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

                    Dosen16 temp = dataDosen[j];

                    dataDosen[j] = dataDosen[j + 1];

                    dataDosen[j + 1] = temp;

                }

            }

        }

    }

    public void sortingDSC() { // Selection Sort

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

            int maxIdx = i;

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

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

                    maxIdx = j;

                }

            }

            Dosen16 temp = dataDosen[maxIdx];

            dataDosen[maxIdx] = dataDosen[i];

            dataDosen[i] = temp;

        }

    }

}

package minggu6;

import java.util.Scanner;

public class DosenMain16 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        DataDosen data = new DataDosen();

        int pilihan;

        do {

            System.out.println("Menu:");

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

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

            System.out.println("3. Urutkan ASC (Usia Muda ke Tua) - Bubble Sort");

            System.out.println("4. Urutkan DSC (Usia Tua ke Muda) - Selection Sort");

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

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

            pilihan = sc.nextInt();

            sc.nextLine();

            switch (pilihan) {

                case 1:

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

                    String kode = sc.nextLine();

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

                    String nama = sc.nextLine();

                    System.out.print("Masukkan jenis kelamin (true = Laki-laki, false = Perempuan): ");

                    boolean jk = sc.nextBoolean();

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

                    int usia = sc.nextInt();

                    data.tambah(new Dosen16(kode, nama, jk, usia));

                    break;

                case 2:

                    data.tampil();

                    break;

                case 3:

                    data.SortingASC();

                    System.out.println("Data telah diurutkan ASC.");

                    break;

                case 4:

                    data.sortingDSC();

                    System.out.println("Data telah diurutkan DSC.");

                    break;

                case 5:

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

                    break;

                default:

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

            }

        } while (pilihan != 5);

    }

}

case 3:

                    data.SortingASC();

                    System.out.println("Data telah diurutkan ASC.");

                    break;

                case 4:

                    data.sortingDSC();

                    System.out.println("Data telah diurutkan DSC.");

                    break;

                case 5:

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

                    break;

                default:

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

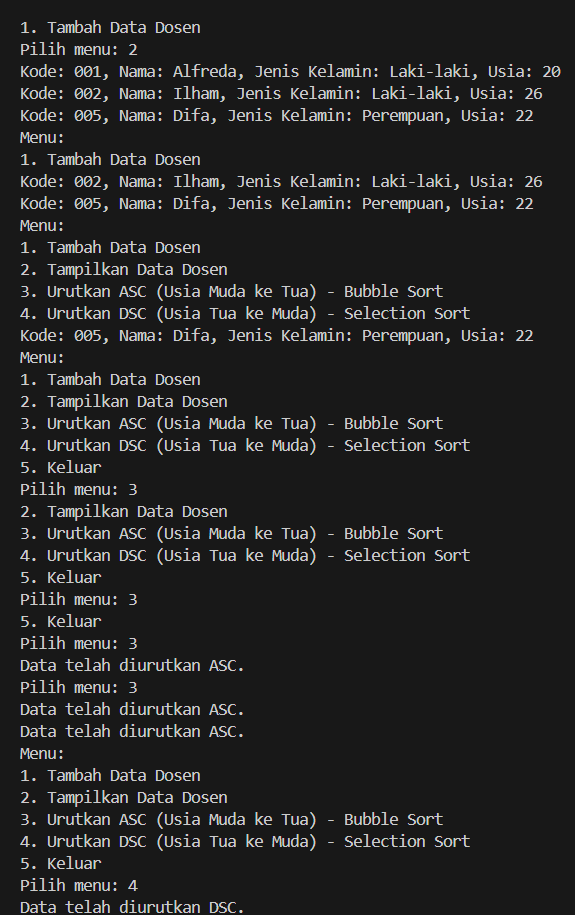
            }

        } while (pilihan != 5);

    }

}

**OUTPUT**

****

**Link Github :** [**https://github.com/Ariqq16/semester2**](https://github.com/Ariqq16/semester2)