

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM ALGORITMA STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 16 : COLLECTIONS



Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

PRAKTIKUM 1

Penggunaan collection untuk menambahkan sebuah elemen, mengakses elemen, dan menghapus sebuah elemen.

Kode program class contohList29 :

```
1 import java.util.ArrayList;
2 import java.util.LinkedList;
3 import java.util.List;
4
5 public class contohList29 {
6     public static void main(String[] args) {
7         List l = new ArrayList();
8         l.add(1);
9         l.add(2);
10        l.add(3);
11        l.add("Cireng");
12        System.out.printf("Elemen 0: %d total elemen : %d elemen terakhir : %s\n", l.get(0), l.size(), l.get(l.size()-1));
13
14        l.add(4);
15        l.remove(0);
16        System.out.printf("Elemen 0: %d total elemen : %d elemen terakhir : %s\n", l.get(0), l.size(), l.get(l.size()-1));
17
18        // Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode
19        List<String> names = new LinkedList<>();
20        names.add("Noureen");
21        names.add("AkhLeema");
22        names.add("Shannum");
23        names.add("Uwais");
24        names.add("Al-Qarni");
25
26        System.out.printf("Elemen 0: %s total elemen : %s elemen terakhir : %s\n", names.get(0), names.size(), names.get(names.size()-1));
27        names.set(0, "My kid");
28        System.out.printf("Elemen 0: %s total elemen : %s elemen terakhir : %s\n", names.get(0), names.size(), names.get(names.size()-1));
29        System.out.println("Names : " + names.toString());
30    }
31 }
```

Output :

```
Note: contohList29.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
Elemen 0: 1 total elemen : 4 elemen terakhir : Cireng
Elemen 0: 2 total elemen : 4 elemen terakhir : 4
Elemen 0: Noureen total elemen : 5 elemen terakhir : Al-Qarni
Elemen 0: My kid total elemen : 5 elemen terakhir : Al-Qarni
Names : [My kid, AkhLeema, Shannum, Uwais, Al-Qarni]
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16> 
```

Pertanyaan :

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?

Jawab : Karena, ArrayList di Java Collection Framework dapat menampung semua jenis tipe data seperti fungsi array tetapi lebih dinamis dalam hal ukurannya. Dan juga ArrayList adalah implementasi dari Interface List yang memungkinkan untuk menyimpan dan mengakses elemen secara sequensial. Perlu diketahui, objek list tidak bisa diinstance dari List, tetapi dari implementasi class yaitu ArrayList

2. Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!

Jawab : Berikut adalah hasil modifikasinya :

- Kode program :

```
// hasil modifikasi soal no 1
List<Integer> l = new ArrayList<>();
l.add(e:1);
l.add(e:2);
l.add(e:3);
// l.add("Cireng"); // Tline ini dapat menyebabkan compile error karena typenya string bukan int
System.out.printf(format:"Elemen 0: %d total elemen : %d elemen terakhir : %d\n", l.get(index:0), l.size(), l.get(l.size() - 1));

l.add(e:4);
l.remove(index:0);
System.out.printf(format:"Elemen 0: %d total elemen : %d elemen terakhir : %d\n", l.get(index:0), l.size(), l.get(l.size() - 1));
```

- Output :

```
Elemen 0: 1 total elemen : 3 elemen terakhir : 3
Elemen 0: 2 total elemen : 3 elemen terakhir : 4
Elemen 0: Noreen total elemen : 5 elemen terakhir : Al-Qarni
Elemen 0: My kid total elemen : 5 elemen terakhir : Al-Qarni
Names : [My kid, AkhLeema, Shannum, Uwais, Al-Qarni]
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

Jadi, dengan cara tersebut, kita memastikan bahwa hanya data yang bertipe int yang dapat ditambahkan ke dalam list l. jika kita mencoba menambahkan data bertipe lain, maka akan terjadi error saat compile.

3. Ubah kode pada baris kode 38 menjadi seperti ini

```
LinkedList<String> names = new LinkedList<>();
```

Jawab : Berikut adalah hasil modifikasinya :

```
// hasil modifikasi no 3
LinkedList<String> names = new LinkedList<>();
names.add(e:"Noreen");
names.add(e:"AkhLeema");
names.add(e:"Shannum");
names.add(e:"Uwais");
names.add(e:"Al-Qarni");
```

4. Tambahkan juga baris berikut ini, untuk memberikan perbedaan dari tampilan yang sebelumnya

```
names.push("Mei-mei");
System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
    names.getFirst(), names.size(), names.getLast());
System.out.println("Names: " + names.toString());
```

Jawab : Berikut adalah hasil modifikasinya :

```
// hasil modifikasi no 3
LinkedList<String> names = new LinkedList<>();
names.add(e:"Noureen");
names.add(e:"AkhLeema");
names.add(e:"Shannum");
names.add(e:"Uwais");
names.add(e:"Al-Qarni");

names.push(e:"Mei - mei");
System.out.printf(format:"Elemen 0: %s total elemen : %s elemen", names.getFirst(), names.size(), names.getLast());
```

5. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan!

Jawab : Berikut adalah output dari penambahan kode tersebut :

```
Elemen 0: 1 total elemen : 3 elemen terakhir : 3
Elemen 0: 2 total elemen : 3 elemen terakhir : 4
Elemen 0: Mei - mei total elemen : 6 elemenElemen 0: Mei - mei total elemen : 6 elemen terakhir : Al-Qarni
Elemen 0: My kid total elemen : 6 elemen terakhir : Al-Qarni
Names : [My kid, Noureen, AkhLeema, Shannum, Uwais, Al-Qarni]
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16> █
```

Perbedaannya adalah dari elemen yang ditambahkan ke dalam list nama. Sebelumnya elemen 0 berisi Noureen, sekarang menjadi Mei-mei, karena names.push ("Mei-mei") menambahkan names Mei – mei ke elemen list paling depan. Karena penambahan elemen tersebut, jumlah total elemen pun ikut bertambah dari semula 5 menjadi 6.

PRAKTIKUM 2

Akan dibuat beberapa method untuk menampilkan beberapa cara yang dapat dilakukan untuk mengambil/menampilkan elemen pada sebuah collection

Kode program class LoopCollection29 :

```
1  import java.util.ArrayList;
2  import java.util.List;
3  import java.util.Stack;
4  import java.util.Iterator;
5
6  public class loopCollection29 {
7      public static void main(String[] args) {
8          Stack<String> fruits = new Stack<>();
9          fruits.push("Banana");
10         fruits.push("Orange");
11         fruits.push("Watermelon");
12         fruits.push("Leci");
13         fruits.push("Salak");
14
15         for (String fruit : fruits) {
16             System.out.printf("%s ", fruit);
17         }
18
19         System.out.println("\n" + fruits.toString());
20
21         while (!fruits.empty()) {
22             System.out.printf("%s ", fruits.pop());
23         }
24
25         fruits.push("Melon");
26         fruits.push("Durian");
27         System.out.println("");
28         for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
29             String fruit = it.next();
30             System.out.printf("%s ", fruit);
31         }
32         System.out.println("");
33         fruits.stream().forEach(e -> {
34             System.out.printf("%s ", e);
35         });
36         System.out.println("");
37         for (int i = 0; i < fruits.size(); i++) {
38             System.out.printf("%s ", fruits.get(i));
39         }
40     }
41 }
42
```

Output :

```
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Melon Durian
Melon Durian
Melon Durian
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

Pertanyaan :

1. Apakah perbedaan fungsi push() dan add() pada objek fruits?

Jawab : push() dan add() adalah method yang digunakan untuk menambahkan elemen ke dalam list, tetapi mereka memiliki perbedaan tergantung pada jenis list yang digunakan. Fruits adalah objek Stack, yang merupakan subclass dari Vector dan mengikuti prinsip LIFO (Last In First Out).

- a) push(): Method ini digunakan dalam Stack dan menambahkan item ke stack paling atas.
- b) Sementara itu, add() adalah method yang umumnya digunakan dalam List dan Set.

Jadi, dalam konteks Stack, push() adalah method yang tepat untuk menambahkan elemen. Jika kita menggunakan List atau Set, kita akan menggunakan add().

2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?

Jawab : Output yang terjadi adalah sebagai berikut :

```
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana

PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

Jika kita menghapus baris kode tersebut, maka "Melon" dan "Durian" tidak akan ditambahkan ke dalam stack fruits. Method push() digunakan untuk menambahkan elemen ke puncak stack. Jika kita menghapus baris ini, "Melon" dan "Durian" tidak akan ditambahkan ke stack, dan oleh karena itu, mereka tidak akan muncul dalam output saat kita mencetak atau mengulangi elemen dalam stack.

3. Jelaskan fungsi dari baris 46-49?

Jawab : Iterator digunakan untuk mengulangi setiap elemen dalam koleksi fruits dan mencetaknya. Jika kita menghapus bagian ini, program tidak akan melakukan iterasi dan mencetak elemen menggunakan Iterator

4. Silakan ganti baris kode 25, Stack menjadi List dan apakah yang terjadi? Mengapa bisa demikian?

Jawab : Yang terjadi adalah error karena beberapa method stack tidak sesuai dengan method pada List.

```
import java.util.Stack;
import java.util.Iterator;
import java.util.List;

public class loopCollection29 {
    Run | Debug
    public static void main(String[] args) {
        List<String> fruits = new Stack<>();
        fruits.push("Banana");    The method push(String) is undefined for the type List<String>
        fruits.push("Orange");    The method push(String) is undefined for the type List<String>
        fruits.push("Watermelon"); The method push(String) is undefined for the type List<String>
        fruits.push("Leci");      The method push(String) is undefined for the type List<String>
        fruits.push("Salak");     The method push(String) is undefined for the type List<String>

        for (String fruit : fruits) {
            System.out.printf(format:"%s ", fruit);
        }

        System.out.println("\n" + fruits.toString());

        while (!fruits.empty()) {    The method empty() is undefined for the type List<String>
            System.out.printf(format:"%s ", fruits.pop());    The method pop() is undefined for the type List<String>
        }

        fruits.push("Melon");    The method push(String) is undefined for the type List<String>
        fruits.push("Durian");    The method push(String) is undefined for the type List<String>
        System.out.println(x:"");
        for (Iterator<String> it = fruits.iterator(); it.hasNext(); ) {
            String fruit = it.next();
            System.out.printf(format:"%s ", fruit);
        }
    }
}
```

Jika kita mengganti Stack <String> menjadi List<String>, kita harus mengganti beberapa method untuk Stack yang sesuai dengan operasi pada List agar tidak error. Perubahan yang perlu dilakukan antara lain :

- a) Stack<String> diganti dengan List<String> fruits = new ArrayList<>();
- b) fruits.empty() diganti dengan fruits.isEmpty().
- c) fruits.pop() diganti dengan fruits.remove(fruits.size() - 1). Ini karena pop() menghapus dan mengembalikan elemen terakhir dari stack, yang setara dengan menghapus elemen terakhir dari list.
- d) fruits.push("Melon") dan fruits.push("Durian") diganti dengan fruits.add("Melon") dan fruits.add("Durian"). Ini karena push() menambahkan elemen ke puncak stack, yang setara dengan menambahkan elemen ke akhir list.

Berikut adalah hasil perubahannya :

```

1 // hasil modifikasi no 4 praktikum 2
2 import java.util.ArrayList;
3 import java.util.Iterator;
4 import java.util.List;
5
6 public class Main {
7     public static void main(String[] args) {
8         List<String> fruits = new ArrayList<>();
9         fruits.add("Apple");
10        fruits.add("Banana");
11        fruits.add("Cherry");
12
13        for (String fruit : fruits) {
14            System.out.printf("%s ", fruit);
15        }
16
17        System.out.println("\n" + fruits.toString());
18
19        while (!fruits.isEmpty()) {
20            System.out.printf("%s ", fruits.remove(fruits.size() - 1));
21        }
22
23        fruits.add("Melon");
24        fruits.add("Durian");
25        System.out.println("");
26        for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
27            String fruit = it.next();
28            System.out.printf("%s ", fruit);
29        }
30        System.out.println("");
31        fruits.stream().forEach(e -> {
32            System.out.printf("%s ", e);
33        });
34        System.out.println("");
35        for (int i = 0; i < fruits.size(); i++) {
36            System.out.printf("%s ", fruits.get(i));
37        }
38    }
39 }
40

```

5. Ganti elemen terakhir dari objek fruits menjadi "Strawberry"!

Jawab : Berikut adalah kode programnya :


```
// Mengganti elemen terakhir dengan "Strawberry"
if (!fruits.isEmpty()) {
    fruits.set(fruits.size() - 1, element:"Strawberry");
}

System.out.println(x:"");
for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
    String fruit = it.next();
    System.out.printf(format:"%s ", fruit);
}
System.out.println(x:"");
```

Output :

```
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Melon Strawberry
Melon Strawberry
Melon Strawberry
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

6. Tambahkan 3 buah seperti "Mango","guava", dan "avocado" kemudian dilakukan sorting!

Jawab :

Berikut adalah kode programnya :

```
// Menambahkan "Mango", "Guava", dan "Avocado"
fruits.add(e:"Mango");
fruits.add(e:"Guava");
fruits.add(e:"Avocado");

// Melakukan sorting
Collections.sort(fruits);
```

Output :

```
Avocado Banana Guava Leci Mango Orange Salak Watermelon
[Avocado, Banana, Guava, Leci, Mango, Orange, Salak, Watermelon]
Watermelon Salak Orange Mango Leci Guava Banana Avocado
Melon Strawberry
Melon Strawberry
Melon Strawberry
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

PRAKTIKUM 3

Kode program class Mahasiswa29 :

```
public class Mahasiswa29 {  
    String nim, nama, noTelp;  
  
    public Mahasiswa29() {  
  
    }  
  
    public Mahasiswa29(String nim, String nama, String noTelp) {  
        this.nim = nim;  
        this.nama = nama;  
        this.noTelp = noTelp;  
    }  
  
    @Override  
    public String toString() {  
        return "Mahasiswa29{" + "nim= " + nim + ", nama= " + nama + ", noTelp= " + noTelp + '}';  
    }  
}
```

Kode program class ListMahasiswa29 :

```

1  import java.util.ArrayList;
2  import java.util.Arrays;
3  import java.util.List;
4
5  public class ListMahasiswa29 {
6      List<Mahasiswa29> mahasiswas = new ArrayList<>();
7
8      public void add(Mahasiswa29... mahasiswa) {
9          mahasiswas.addAll(Arrays.asList(mahasiswa));
10     }
11
12     public void remove(int index) {
13         mahasiswas.remove(index);
14     }
15
16     public void update(int index, Mahasiswa29 mhs) {
17         mahasiswas.set(index, mhs);
18     }
19
20     public void display() {
21         mahasiswas.stream().forEach(mhs -> {
22             System.out.println(" " + mhs.toString());
23         });
24     }
25
26     int linearSearch(String nim) {
27         for (int i = 0; i < mahasiswas.size(); i++) {
28             if (nim.equals(mahasiswas.get(i).nim)) {
29                 return i;
30             }
31         }
32         return -1;
33     }
34
35     public static void main(String[] args) {
36         ListMahasiswa29 lm = new ListMahasiswa29();
37         Mahasiswa29 m = new Mahasiswa29("201234", "Noureen", "021xx1");
38         Mahasiswa29 m1 = new Mahasiswa29("201235", "Akhleema", "021xx2");
39         Mahasiswa29 m2 = new Mahasiswa29("201236", "Shannum", "021xx3");
40
41         // menambahkan objek mahasiswa
42         lm.add(m, m1, m2);
43         // menampilkan list mahasiswa
44         lm.display();
45         // update mahasiswa
46         lm.update(lm.linearSearch("201235"), new Mahasiswa29("201235", "Akhleema Iela", "021xx2"));
47         System.out.println("");
48         lm.display();
49     }
50 }
51

```

Output :

```
Mahasiswa29{nim= 201234, nama= Nouredin, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}

Mahasiswa29{nim= 201234, nama= Nouredin, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema Lela, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```

Pertanyaan :

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?

Jawab : Fungsi add() menggunakan konsep "varargs" atau "variable arguments" dalam Java. Varargs memungkinkan kita untuk meneruskan jumlah argumen yang tidak terbatas ke metode atau fungsi. Kelebihan menggunakan varargs adalah fleksibilitas. Kita dapat memanggil method dengan jumlah argumen yang berbeda-beda tanpa perlu mendefinisikan metode baru untuk setiap jumlah argumen. Ini sangat berguna jika kita tidak tahu berapa banyak argumen yang akan diperlukan.

2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!

Jawab :

Berikut adalah kode programnya :

```

1 // modifikasi linearSearch menjadi binarySearch
2
3 import java.util.*;
4 import java.util.Comparator;
5
6 public class ListMahasiswa29 {
7     List<Mahasiswa29> mahasiswas = new ArrayList<>();
8
9     public void add(Mahasiswa29... mahasiswa) {
10         mahasiswas.addAll(Arrays.asList(mahasiswa));
11     }
12
13     public void remove(int index) {
14         mahasiswas.remove(index);
15     }
16
17     public void update(int index, Mahasiswa29 mhs) {
18         mahasiswas.set(index, mhs);
19     }
20
21     public void display() {
22         mahasiswas.stream().forEach(mhs -> {
23             System.out.println(" " + mhs.toString());
24         });
25     }
26
27     int binarySearch(String nim) {
28         Comparator<Mahasiswa29> comparator = Comparator.comparing(Mahasiswa29::getNim);
29         Collections.sort(mahasiswas, comparator);
30         return Collections.binarySearch(mahasiswas, new Mahasiswa29(nim, "", ""), comparator);
31     }
32
33     public static void main(String[] args) {
34         ListMahasiswa29 lm = new ListMahasiswa29();
35         Mahasiswa29 m = new Mahasiswa29("201234", "Noureen", "021xx1");
36         Mahasiswa29 m1 = new Mahasiswa29("201235", "AkhLeema", "021xx2");
37         Mahasiswa29 m2 = new Mahasiswa29("201236", "Shannum", "021xx3");
38
39         lm.add(m, m1, m2);
40         lm.display();
41         int index = lm.binarySearch("201235");
42         if (index >= 0) {
43             lm.update(index, new Mahasiswa29("201235", "AkhLeema Lela", "021xx2"));
44         }
45         System.out.println("");
46         lm.display();
47     }
48 }
49

```

Output :

```

Mahasiswa29{nim= 201234, nama= Noureen, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}

Mahasiswa29{nim= 201234, nama= Noureen, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema Lela, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>

```

3. Tambahkan fungsi sorting baik secara ascending ataupun descending pada class tersebut!

Jawab :

Berikut adalah kode programnya :

```
public void sortAscending() {
    Comparator<Mahasiswa29> comparator = Comparator.comparing(Mahasiswa29::getNim);
    Collections.sort(mahasiswas, comparator);
}

public void sortDescending() {
    Comparator<Mahasiswa29> comparator = Comparator.comparing(Mahasiswa29::getNim).reversed();
    Collections.sort(mahasiswas, comparator);
}
```

Main :

```
// sort ascending
lm.sortAscending();
System.out.println();
System.out.println(x:"After sorting ascending:");
lm.display();
// sort descending
lm.sortDescending();
System.out.println();
System.out.println(x:"After sorting descending:");
lm.display();
```

Output :

```
Mahasiswa29{nim= 201234, nama= Noreen, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}

Mahasiswa29{nim= 201234, nama= Noreen, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema Lela, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}

After sorting ascending:
Mahasiswa29{nim= 201234, nama= Noreen, noTelp= 021xx1}
Mahasiswa29{nim= 201235, nama= AkhLeema Lela, noTelp= 021xx2}
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}

After sorting descending:
Mahasiswa29{nim= 201236, nama= Shannum, noTelp= 021xx3}
Mahasiswa29{nim= 201235, nama= AkhLeema Lela, noTelp= 021xx2}
Mahasiswa29{nim= 201234, nama= Noreen, noTelp= 021xx1}
PS D:\KULIAH\college\smt 2\29_yonanda_asd\P16>
```


TUGAS

1. Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Data Mahasiswa dan Mata Kuliah perlu melalui penginputan data terlebih dahulu.

Jawab :

Berikut adalah kode programnya :

- Class Mahasiswa :

```
class Mahasiswa {
    String nim;
    String nama;
    String telfon;

    Mahasiswa(String nim, String nama, String telfon) {
        this.nim = nim;
        this.nama = nama;
        this.telfon = telfon;
    }

    String getNim() {
        return nim;
    }

    String getNama() {
        return nama;
    }

    String getTelfon() {
        return telfon;
    }
}
```

- Class MataKuliah :

```
class MataKuliah {
    String kode;
    String nama;
    int sks;

    MataKuliah(String kode, String nama, int sks) {
        this.kode = kode;
        this.nama = nama;
        this.sks = sks;
    }

    String getKode() {
        return kode;
    }

    String getNama() {
        return nama;
    }

    int getSks() {
        return sks;
    }
}
```

- Class Nilai :

```
class Nilai {  
    Mahasiswa mahasiswa;  
    MataKuliah mataKuliah;  
    double nilai;  
  
    Nilai(Mahasiswa mahasiswa, MataKuliah mataKuliah, double nilai) {  
        this.mahasiswa = mahasiswa;  
        this.mataKuliah = mataKuliah;  
        this.nilai = nilai;  
    }  
  
    Mahasiswa getMahasiswa() {  
        return mahasiswa;  
    }  
  
    MataKuliah getMataKuliah() {  
        return mataKuliah;  
    }  
  
    double getNilai() {  
        return nilai;  
    }  
}
```

- Class Main MahasiswaTugas29 :

- Output :
 - Input nilai mahasiswa

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
Pilih menu: 1
Masukkan NIM: 20001
Masukkan Nama: Thalhah
Masukkan No Telfon: 021xxx
Masukkan Kode Mata Kuliah: 0001
Masukkan Nama Mata Kuliah: IoT
Masukkan SKS: 3
Masukkan Nilai: 80.75
```

- Tampil nilai mahasiswa

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
Pilih menu: 2
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
NIM      Nama      Telfon      Mata Kuliah      SKS      Nilai
20001    Thalhah    021xxx     IoT              3        80.75
20002    Zubair     021xxx     ASD              3        90.00
```

- Mencari nilai mahasiswa

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
Pilih menu: 3
Masukkan NIM: 20002
NIM      Nama      Telfon      Mata Kuliah      SKS      Nilai
20002    Zubair     021xxx     ASD              3        90.00
Total SKS 3 yang telah diambil
```

- Urut data nilai dari terkecil ke terbesar (asc)

```

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
Pilih menu: 4

```

NIM	Nama	Telfon	Mata Kuliah	SKS	Nilai
20001	Thalhah	021xxx	IoT	3	80.75
20002	Zubair	021xxx	ASD	3	90.00

2. Tambahkan prosedur hapus data mahasiswa melalui implementasi Queue pada collections Tugas nomor 1!

Jawab :

```

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
6. Hapus Data Mahasiswa
Pilih menu: 1
Masukkan NIM: 20001
Masukkan Nama: Thalhah
Masukkan No Telfon: 021xxx
Masukkan Kode Mata Kuliah: 0001
Masukkan Nama Mata Kuliah: IoT
Masukkan SKS: 3
Masukkan Nilai: 87.55

```

```

*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
6. Hapus Data Mahasiswa
Pilih menu: 1
Masukkan NIM: 20002
Masukkan Nama: Zubair
Masukkan No Telfon: 021xxx
Masukkan Kode Mata Kuliah: 0002
Masukkan Nama Mata Kuliah: ASD
Masukkan SKS: 3
Masukkan Nilai: 90.00

```

```
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
6. Hapus Data Mahasiswa
Pilih menu: 6
Masukkan NIM mahasiswa yang akan dihapus: 20001
Data mahasiswa dengan NIM 20001 telah dihapus.
Data yang tersisa:
NIM: 20002, Nama: Zubair, Telfon: 021xxx, Mata Kuliah: ASD, SKS: 3, Nilai: 90.0
```