Name : Azaria Cindy Sahasika
Number Id : 2341760169 / 06
Class : 1G – Business Information System
Lesson : Algorithm and Data Structure
Material : Material 13 – Collections
Github Link : https://github.com/azariacindy/algorithm-ds

JOBSHEET 16
Collection

16.1. Tujuan Praktikum

Setelah melakukan praktikum ini, mahasiswa mampu:

1. memahami bentuk-bentuk collection dan hierarkinya;
2. menerapkan collection sesuai dengan fungsi dan jenisnya;
3. menyelesaikan kasus menggunakan collection yang sesuai.

16.2. Kegiatan Praktikum 1

16.2.1. Percobaan 1

Pada percobaan 1 ini akan dicontohkan penggunaan collection untuk menambahkan sebuah elemen, mengakses elemen, dan menghapus sebuah elemen.

1. Buatlah sebuah class ContohList yang main methode berisi kode program seperti di bawah ini

```
25          List l = new ArrayList();
26          l.add(1);
27          l.add(2);
28          l.add(3);
29          l.add("Cireng");
30          System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
31                  l.get(0), l.size(), l.get(l.size() - 1));
32
33          l.add(4);
34          l.remove(0);
35          System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
36                  l.get(0), l.size(), l.get(l.size() - 1));
```

2. Tambahkan kode program untuk menggunakan collection dengan aturan penulisan kode program seperti berikut

```
38          List<String> names = new LinkedList<>();
39          names.add("Noureen");
40          names.add("Akhleema");
41          names.add("Shannum");
42          names.add("Uwais");
43          names.add("Al-Qarni");
44
45          System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
46                  names.get(0), names.size(), names.get(names.size() - 1));
47          names.set(0, "My kid");
48          System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
49                  names.get(0), names.size(), names.get(names.size() - 1));
50          System.out.println("Names: " + names.toString());
```

jobsheet13_collection > pract1 > ListExample06.java > ...

```
1   package jobsheet13_collection.pract1;
2   import java.util.ArrayList;
3   import java.util.LinkedList;
4   import java.util.List;
5
6   public class ListExample06 {
        Run | Debug
7       public static void main(String[] args) {
8           List<Object> l = new ArrayList<>();
9           l.add(e:1);
10          l.add(e:2);
11          l.add(e:3);
12          l.add(e:"Cireng");
13          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
14                  l.get(index:0), l.size(), l.get(l.size() - 1));
15
16          l.add(e:4);
17          l.remove(index:0);
18          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
19                  l.get(index:0), l.size(), l.get(l.size() - 1));
20
21          List<String> names = new LinkedList<>();
22          names.add(e:"Azaria");
23          names.add(e:"Cindy");
24          names.add(e:"Sahasika");
25          names.add(e:"Garcia");
26          names.add(e:"Catrine");
27
28          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
29                  names.get(index:0), names.size(), names.get(names.size() - 1));
30          names.set(index:0, element:"My kid");
31          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
32                  names.get(index:0), names.size(), names.get(names.size() - 1));
33          System.out.println("Names: " + names.toString());
34      }
35  }
```

```
Element 0: 1, total elements: 4, last element: Cireng
Element 0: 2, total elements: 4, last element: 4
Element 0: Azaria, total elements: 5, last element: Catrine
Element 0: My kid, total elements: 5, last element: Catrine
Names: [My kid, Cindy, Sahasika, Garcia, Catrine]
```

16.2.2. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

```
run:
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]
```

16.2.3. Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah Arraylist?
   ➔ Karena ArrayList tidak memmiliki kurung sudut yang menentukan data apa yang akan disimpan, ArrayList pada baris code 25-36 diatas digunakan debagai sintak umum yang dapat menyimpan data dari semua jenis data.
2. Modifikasi baris kode 25-36 seingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!

```
11          l.add(e:1);
12          l.add(e:2);
13          l.add(e:3);
14          // l.add("Cireng");
15          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
16                  l.get(index:0), l.size(), l.get(l.size() - 1));
17
18          l.add(e:4);
19          l.remove(index:0);
20          System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
21                  l.get(index:0), l.size(), l.get(l.size() - 1));
```

```
Element 0: 1, total elements: 3, last element: 3
Element 0: 2, total elements: 3, last element: 4
```

3. Ubah kode pada baris kode 38 menjadi seperti ini

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

```
23          LinkedList<String> names = new LinkedList<>();
24          names.add(e:"Azaria");
25          names.add(e:"Cindy");
26          names.add(e:"Sahasika");
27          names.add(e:"Garcia");
28          names.add(e:"Catrine");
```

```
Element 0: Azaria, total elements: 5, last element: Catrine
Element 0: My kid, total elements: 5, last element: Catrine
Names: [My kid, Cindy, Sahasika, Garcia, Catrine]
```

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());
```

```
        LinkedList<String> names = new LinkedList<>();
        names.add(e:"Azaria");
        names.add(e:"Cindy");
        names.add(e:"Sahasika");
        names.add(e:"Garcia");
        names.add(e:"Catrine");

        System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
                names.get(index:0), names.size(), names.get(names.size() - 1));
        names.set(index:0, element:"My kid");
        System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
                names.get(index:0), names.size(), names.get(names.size() - 1));
        System.out.println("Names: " + names.toString());

        names.push(e:"Mei-mei");
        System.out.printf(format:"Element 0: %s, total elements: %d, last element: %s\n",
                names.getFirst(), names.size(), names.getLast());
        System.out.println("Names: " + names.toString());
```

```
Element 0: Azaria, total elements: 5, last element: Catrine
Element 0: My kid, total elements: 5, last element: Catrine
Names: [My kid, Cindy, Sahasika, Garcia, Catrine]
Element 0: Mei-mei, total elements: 6, last element: Catrine
Names: [Mei-mei, My kid, Cindy, Sahasika, Garcia, Catrine]
```

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

➔ Dengan menggunakan class LinkedList, kita dapat menggunakan metode daftar terkait seperti push(), getFirst(), dan getLast(), tetapi ketika menggunakan class List tidak dapat melakukannya.

16.3. Kegiatan Praktikum 2

16.3.1. Tahapan Percobaan

Pada praktikum 2 ini akan dibuat beberapa method untuk menampilkan beberapa cara yang dapat dilakukan untuk mengambil/menampilkan elemen pada sebuah collection. Silakan ikutilah Langkah-langkah di bawah ini

1. Buatlah class dengan nama LoopCollection serta tambahkan method main yang isinya adalah sebagai berikut.

```
        Stack<String> fruits = new Stack<>();
        fruits.push("Banana");
        fruits.add("Orange");
        fruits.add("Watermelon");
        fruits.add("Leci");
        fruits.push("Salak");

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

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

        while (!fruits.empty()) {
            System.out.printf("%s ", fruits.pop());
        }
```

2. Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.

```
43            fruits.push("Melon");
44            fruits.push("Durian");
45            System.out.println("");
              for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
47                String fruit = it.next();
48                System.out.printf("%s ", fruit);
49            }
50            System.out.println("");
51            fruits.stream().forEach(e -> {
52                System.out.printf("%s ", e);
53            });
54            System.out.println("");
55            for (int i = 0; i < fruits.size(); i++) {
56                System.out.printf("%s ", fruits.get(i));
57
58            }
```

jobsheet13_collection > pract1 > LoopCollection06.java > ...

```
1   package jobsheet13_collection.pract1;
2
3   import java.util.Iterator;
4   import java.util.Stack;
5
6   public class LoopCollection06 {
        Run | Debug
7       public static void main(String[] args) {
8           Stack<String> fruits = new Stack<>();
9           fruits.push(item:"Banana");
10          fruits.push(item:"Orange");
11          fruits.push(item:"Watermelon");
12          fruits.push(item:"Leci");
13          fruits.push(item:"Salak");
14
15          for (String fruit : fruits) {
16              System.out.printf(format:"%s ", fruit);
17          }
18
19          System.out.println("\n" + fruits.toString());
20
21          while (!fruits.empty()) {
22              System.out.printf(format:"%s ", fruits.pop());
23          }
24
25          fruits.push(item:"Melon");
26          fruits.push(item:"Durian");
27          System.out.println(x:"");
28
29          for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
30              String fruit = it.next();
31              System.out.printf(format:"%s ", fruit);
32          }
33
34          System.out.println(x:"");
35          fruits.stream().forEach(e -> {
36              System.out.printf(format:"%s ", e);
37          });
38
39          System.out.println(x:"");
40          for (int i = 0; i < fruits.size(); i++) {
41              System.out.printf(format:"%s ", fruits.get(i));
42          }
43      }
44  }
```

16.3.2. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

```
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Melon Durian
Melon Durian
Melon Durian BUILD SUCCESSFUL (total time: 0 seconds)
```

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

16.3.3. Pertanyaan Percobaan

1. Apakah perbedaan fungsi push() dan add() pada objek *fruits*?
   - Push(): fungsi dari interface stack
   - Add(): bukan fungsi dari interface stack

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

```
// fruits.push("Melon");
// fruits.push("Durian");
System.out.println(x:"");
```

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

   - Jika tidak push "Melon" dan "Durian" kembali ke dalam stack, stack akan tetap kosong setelah perulangan while, sehingga tidak ada keluaran lebih lanjut dari perulangan berikutnya dan pemrosesan stream. Hal ini terjadi karena bagian kode tersebut mengulang di atas tumpukan, yang pada saat itu, semua elemennya telah dihapus.

3. Jelaskan fungsi dari baris 46-49?

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

   - For-each loop: untuk menginisialisasi dan mengatur perulangan untuk iterasi melalui setiap elemen dalam stack fruits. ‘fruit’ untuk mencetak setiap elemen dalam stack.

```
while (!fruits.empty()) {
    System.out.printf(format:"%s ", fruits.pop());
}
```

   - While loop: untuk menginisialisadi while loop yang akan terus berjalan selama stack fruits tidak kosong. ‘fruits.pop()’ untuk mengeluarkan dan mencetak setiap elemen dari stack hingga stack tidak kosong.

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

   - Iterator loop: ‘Iterator<String> it’ untuk mengatur perulangan menggunakan iterator pada stack fruits. ‘it.next()’ untuk mengatur variabel ‘fruit’ dengan elemen berikutnya dalam iterasi.

```
System.out.println(x:"");
fruits.stream().forEach(e -> {
    System.out.printf(format:"%s ", e);
});
```

   - Stream forEach: menginisialisasi perulangan menggunakan stream dan method ‘forEach’ untek setiap elemen ‘e’ dalam stack ‘fruits’.

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

```java
package jobsheet13_collection.pract1;

import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class LoopCollection06 {
    Run | Debug
    public static void main(String[] args) {
        // Use ArrayList instead of Stack
        List<String> fruits = new ArrayList<>();
        fruits.add(e:"Banana");
        fruits.add(e:"Orange");
        fruits.add(e:"Watermelon");
        fruits.add(e:"Leci");
        fruits.add(e:"Salak");

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

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

        // Simulating Stack pop operation with List
        while (!fruits.isEmpty()) {
            System.out.printf(format:"%s ", fruits.remove(fruits.size() - 1));
        }

        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:"");
        fruits.stream().forEach(e -> {
            System.out.printf(format:"%s ", e);
        });

        System.out.println(x:"");
        for (int i = 0; i < fruits.size(); i++) {
            System.out.printf(format:"%s ", fruits.get(i));
        }
    }
}
```

- push untuk Stack, sedangkan metode add digunakan untuk List
- pop untuk Stack, sedangkan metode hapus dapat digunakan untuk List

5. Ganti elemen terakhir dari dari objek fruits menjadi “Strawberry”!

```java
// Ganti elemen terakhir dari dari objek fruits menjadi "Strawberry"
fruits.set(fruits.size() - 1, element:"Strawberry");
```

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

```java
// Tambahkan 3 buah seperti "Mango","Guava", dan "Avocado"
fruits.push(item:"Mango");
fruits.push(item:"Guava");
fruits.push(item:"Avocado");

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

```
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Avocado Banana Guava Leci Mango Orange Strawberry Watermelon
[Avocado, Banana, Guava, Leci, Mango, Orange, Strawberry, Watermelon]
Avocado Banana Guava Leci Mango Orange Strawberry Watermelon
Avocado Banana Guava Leci Mango Orange Strawberry Watermelon
Avocado Banana Guava Leci Mango Orange Strawberry Watermelon
```

16.4. Kegiatan Praktikum 3

16.4.1. Tahapan Percobaan

Pada praktikum 3 ini dilakukan uji coba untuk mengimplementasikan sebuah collection untuk menampung objek yang dibuat sesuai kebutuhan. Objek tersebut adalah sebuah objek mahasiswa dengan fungsi-fungsi umum seperti menambahkan, menghapus, mengubah, dan mencari.

1. Buatlah sebuah class Mahasiswa dengan attribute, kontruktor, dan fungsi sebagai berikut.

```
String nim;
String nama;
String notelp;

public Mahasiswa() {
}

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

@Override
public String toString() {
    return "Mahasiswa{" + "nim=" + nim + ", nama=" + nama + ", notelp=" + notelp + '}';
}
```

2. Selanjutnya, buatlah sebuah class ListMahasiswa yang memiliki attribute seperti di bawah ini

```
List<Mahasiswa> mahasiswas = new ArrayList<>();
```

3. Method **tambah(), hapus(), update(), dan tampil()** secara berurut dibuat agar bisa melakukan operasi-operasi seperti yang telah disebutkan.

```
public void tambah(Mahasiswa... mahasiswa) {
    mahasiswas.addAll(Arrays.asList(mahasiswa));
}

public void hapus(int index) {
    mahasiswas.remove(index);
}

public void update(int index, Mahasiswa mhs) {
    mahasiswas.set(index, mhs);
}

public void tampil() {
    mahasiswas.stream().forEach(mhs -> {
        System.out.println("" + mhs.toString());
    });
}
```

4. Untuk proses hapus, update membutuhkan fungsi pencarian terlebih dahulu yang potongan kode programnya adalah sebagai berikut

```java
int linearSearch(String nim) {
    for (int i = 0; i < mahasiswas.size(); i++) {
        if (nim.equals(mahasiswas.get(i).nim)) {
            return i;
        }
    }
    return -1;
}
```

5. Pada class yang sama, tambahkan main method seperti potongan program berikut dan amati hasilnya!

```java
ListMahasiswa lm = new ListMahasiswa();
Mahasiswa m = new Mahasiswa("201234", "Noureen", "021xx1");
Mahasiswa m1 = new Mahasiswa("201235", "Akhleema", "021xx2");
Mahasiswa m2 = new Mahasiswa("201236", "Shannum", "021xx3");
  menambahkan objek mahasiswa
lm.tambah(m, m1, m2);
  menampilkan list mahasiswa
lm.tampil();
  update mahasiswa
lm.update(lm.linearSearch("201235"), new Mahasiswa("201235", "Akhleema Lela", "021xx2"));
System.out.println("");
lm.tampil();
```

jobsheet13_collection > pract1 > Mahasiswa06.java > ...

```java
package jobsheet13_collection.pract1;

public class Mahasiswa06 {
    String nim, nama, notelp;

    public Mahasiswa06() {}

    public Mahasiswa06(String nim, String nama, String notelp) {
        this.nim = nim;
        this.nama = nama;
        this.notelp = notelp;
    }

    @Override
    public String toString() {
        return "Mahasiswa{" + "nim:" + nim + ", nama: " + nama + ", notelp: " + notelp + '}';
    }
}
```

jobsheet13_collection > pract1 > MahasiswaMain06.java > ...

```java
package jobsheet13_collection.pract1;

public class MahasiswaMain06 {
    Run | Debug
    public static void main(String[] args) {
        ListMahasiswa06 lm = new ListMahasiswa06();
        Mahasiswa06 mhs1 = new Mahasiswa06(nim:"201234", nama:"Noureen", notelp:"021xx1");
        Mahasiswa06 mhs2 = new Mahasiswa06(nim:"201235", nama:"Akhleema", notelp:"021xx2");
        Mahasiswa06 mhs3 = new Mahasiswa06(nim:"201236", nama:"Shannum", notelp:"021xx3");

        lm.add(mhs1, mhs2, mhs3);
        lm.print();

        // Update mahasiswa
        lm.update(lm.linearSearch(nim:"201235"), new Mahasiswa06(nim:"201235", nama:"Akhleema Updated", notelp:"021xx2"));
        System.out.println(x:"");
        lm.print();
    }
}
```

jobsheet13_collection > pract1 > ListMahasiswa06.java > ...

```java
package jobsheet13_collection.pract1;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class ListMahasiswa06 {
    List<Mahasiswa06> mahasiswas = new ArrayList<>();

    public void add(Mahasiswa06... mahasiswas) {
        this.mahasiswas.addAll(Arrays.asList(mahasiswas));
    }

    public void remove(int index) {
        mahasiswas.remove(index);
    }

    public void update(int index, Mahasiswa06 mhs) {
        mahasiswas.set(index, mhs);
    }

    public void print() {
        mahasiswas.stream().forEach(mhs -> {
            System.out.println("" + mhs.toString());
        });
    }

    public int linearSearch(String nim) {
        for (int i = 0; i < mahasiswas.size(); i++) {
            if (nim.equals(mahasiswas.get(i).nim)) {
                return i;
            }
        }
        return -1;
    }
}
```

16.4.2. Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

```
Mahasiswa{nim=201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim=201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim=201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim=201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim=201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim=201236, nama=Shannum, notelp=021xx3}
BUILD SUCCESSFUL (total time: 0 seconds)
```

```
Mahasiswa{nim:201234, nama: Noureen, notelp: 021xx1}
Mahasiswa{nim:201235, nama: Akhleema, notelp: 021xx2}
Mahasiswa{nim:201236, nama: Shannum, notelp: 021xx3}

Mahasiswa{nim:201234, nama: Noureen, notelp: 021xx1}
Mahasiswa{nim:201235, nama: Akhleema lelas, notelp: 021xx2}
Mahasiswa{nim:201236, nama: Shannum, notelp: 021xx3}
```

16.4.3. Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihannya apa?
   - Fungsi add() dengan menggunakan varargs (Mahasiswa06... mahasiswas) menggunakan konsep varargs (variable-length arguments). Kelebihannya adalah memungkinkan kita untuk memasukkan jumlah argumen yang variatif saat memanggil metode add(), tanpa harus membuat array terlebih dahulu.
2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!

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

16.5. Tugas Praktikum

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.

   Ilustrasi Program

   Menu Awal dan Penambahan Data

```
************************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
************************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
************************************************
Pilih     :
```

```
Pilih     : 1
Masukan data
Kode      : 0001
Nilai     : 80.75

DAFTAR MAHASISWA
**************************************************
NIM             Nama            Telf
20001           Thalhah         021xxx
20002           Zubair          021xxx
20003           Abdur-Rahman    021xxx
20004           Sa'ad           021xxx
20005           Sa'id           021xxx
20006           Ubaidah         021xxx
Pilih mahasiswa by nim: 20001

DAFTAR MATA KULIAH
**********************************************
Kode      Mata Kuliah                           SKS
00001     Internet of Things                    3
00002     Algoritma dan Struktur Data           2
00003     Algoritma dan Pemrograman             2
00004     Praktikum Algoritma dan Struktur Data 3
00005     Praktikum Algoritma dan Pemrograman   3
Pilih MK by kode: 00001
```

Tampil Nilai

```
**********************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
**********************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
**********************************************
Pilih     : 2

DAFTAR NILAI MAHASISWA
**********************************************
Nim             Nama            Mata Kuliah                     SKS       Nilai
20001           Thalhah         Internet of Things              3         80.75
```

Pencarian Data Mahasiswa

```
**********************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
**********************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
**********************************************
Pilih     : 3

DAFTAR NILAI MAHASISWA
**********************************************
Nim            Nama           Mata Kuliah                          SKS       Nilai
20001          Thalhah        Internet of Things                   3         90.00
20002          Zubair         Praktikum Algoritma dan Pemrograman  3         80.75
Masukkan data mahasiswa[nim] :20002
Nim            Nama           Mata Kuliah                          SKS       Nilai
20002          Zubair         Praktikum Algoritma dan Pemrograman  3         80.75
Total SKS 3 telah diambil.
```

Pengurutan Data Nilai

```
**********************************************
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
**********************************************

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
**********************************************
Pilih     : 4

DAFTAR NILAI MAHASISWA
**********************************************
Nim            Nama           Mata Kuliah                          SKS       Nilai
20002          Zubair         Praktikum Algoritma dan Pemrograman  3         80.75
20001          Thalhah        Internet of Things                   3         90.00
```

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

jobsheet13_collection > assignment > Mahasiswa06.java > ...

```java
package jobsheet13_collection.assignment;

public class Mahasiswa06 {
    String nim, nama, noTelp;

    public Mahasiswa06(String nim, String nama, String noTelp) {
        this.nim = nim;
        this.nama = nama;
        this.noTelp = noTelp;
    }

    @Override
    public String toString() {
        return nim + " " + nama + " " + noTelp;
    }
}
```

jobsheet13_collection > assignment > MataKuliah06.java > ...

```java
package jobsheet13_collection.assignment;

public class MataKuliah06 {
    String kodeMK, namaMk;
    int sks;

    public MataKuliah06(String kodeMK, String namaMk, int sks) {
        this.kodeMK = kodeMK;
        this.namaMk = namaMk;
        this.sks = sks;
    }

    @Override
    public String toString() {
        return kodeMK + " " + namaMk + " " + sks;
    }
}
```

jobsheet13_collection > assignment > J NilaiMain06.java > ...

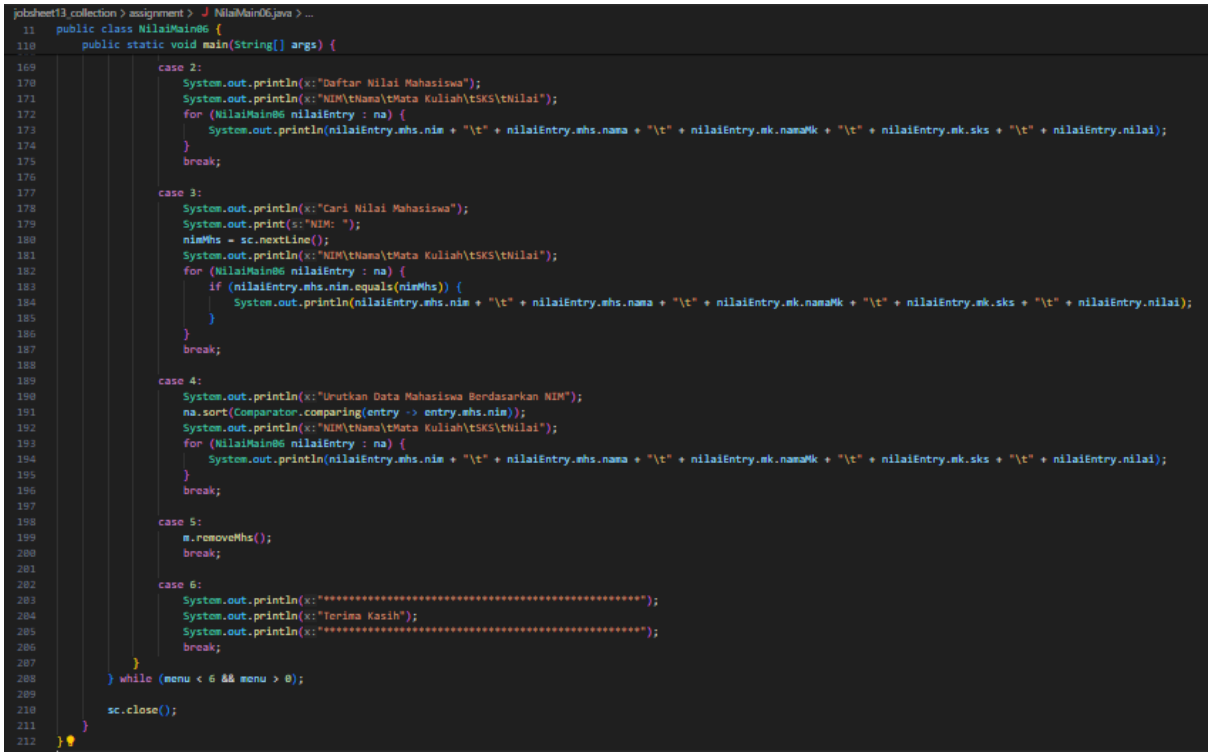
```java
package jobsheet13_collection.assignment;

import java.util.Scanner;
import java.util.List;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Comparator;
import java.util.Queue;
import java.util.LinkedList;

public class NilaiMain06 {
    String kodeN;
    double nilai;
    Mahasiswa06 mhs;
    MataKuliah06 mk;

    public NilaiMain06(String kodeN, double nilai, Mahasiswa06 mhs, MataKuliah06 mk) {
        this.kodeN = kodeN;
        this.nilai = nilai;
        this.mhs = mhs;
        this.mk = mk;
    }

    List<Mahasiswa06> mhsA = new ArrayList<>();
    List<MataKuliah06> mkA = new ArrayList<>();
    Queue<Mahasiswa06> mhsQueue = new LinkedList<>();

    public void addMhs(Mahasiswa06... mahasiswa) {
        mhsA.addAll(Arrays.asList(mahasiswa));
        mhsQueue.addAll(Arrays.asList(mahasiswa));
    }

    public void addMk(MataKuliah06... matkul) {
        mkA.addAll(Arrays.asList(matkul));
    }

    public void printMhs() {
        mhsA.stream().forEach(mhs -> {
            System.out.println("" + mhs.toString());
        });
    }

    public void printMk() {
        mkA.stream().forEach(mk -> {
            System.out.println("" + mk.toString());
        });
    }

    public int linearSearchMhs(String nim) {
        for (int i = 0; i < mhsA.size(); i++) {
            if (nim.equals(mhsA.get(i).nim)) {
                return i;
            }
        }
        return -1;
    }
```

jobsheet13_collection > assignment > NilaiMain06.java > ...

```
11  public class NilaiMain06 {
58      public int linearSearchMk(String matkul) {
59          for (int i = 0; i < mkA.size(); i++) {
60              if (matkul.equals(mkA.get(i).kodeMK)) {
61                  return i;
62              }
63          }
64          return -1;
65      }
66
67      Mahasiswa06 searchMhs(String nim) {
68          for (int i = 0; i < mhsA.size(); i++) {
69              if (nim.equals(mhsA.get(i).nim)) {
70                  return mhsA.get(i);
71              }
72          }
73          return null;
74      }
75
76      MataKuliah06 searchMk(String matkul) {
77          for (int i = 0; i < mkA.size(); i++) {
78              if (matkul.equals(mkA.get(i).kodeMK)) {
79                  return mkA.get(i);
80              }
81          }
82          return null;
83      }
84
85      public void removeMhs() {
86          if (!mhsQueue.isEmpty()) {
87              Mahasiswa06 removedMhs = mhsQueue.poll();
88              mhsA.remove(removedMhs);
89              System.out.println("Mahasiswa removed: " + removedMhs);
90          } else {
91              System.out.println(x:"Queue is empty. No mahasiswa to remove.");
92          }
93      }
94
95      public String toString(int idx1, int idx2) {
96          return mhsA.get(idx1).nim + " - " + mhsA.get(idx1).nama + "\n" + mkA.get(idx2).namaMk + "\n" + mkA.get(idx2).namaMk + "\n" +
97                  mkA.get(idx2).toString();
98      }
99
100     static Comparator<NilaiMain06> CompNilai = new Comparator<NilaiMain06>() {
101         public int compare(NilaiMain06 n1, NilaiMain06 n2) {
102             if (n1.nilai < n2.nilai) {
103                 return 1;
104             } else {
105                 return -1;
106             }
107         }
108     };
```

jobsheet13_collection > assignment > J NilaiMain06.java > ...

```java
public class NilaiMain06 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        List<NilaiMain06> na = new ArrayList<>();
        NilaiMain06 m = new NilaiMain06(kodeN:null, nilai:0, mhs:null, mk:null);

        Mahasiswa06[] mhs = new Mahasiswa06[6];
        mhs[0] = new Mahasiswa06(nim:"200001", nama:"Thalhah", noTelp:"021xxx");
        mhs[1] = new Mahasiswa06(nim:"200002", nama:"Zubair", noTelp:"021xxx");
        mhs[2] = new Mahasiswa06(nim:"200003", nama:"Abdur-Rahman", noTelp:"021xxx");
        mhs[3] = new Mahasiswa06(nim:"200004", nama:"Sa'ad", noTelp:"021xxx");
        mhs[4] = new Mahasiswa06(nim:"200005", nama:"Sa'id", noTelp:"021xxx");
        mhs[5] = new Mahasiswa06(nim:"200006", nama:"Ubaidah", noTelp:"021xxx");

        m.addMhs(mhs);

        MataKuliah06[] mk = new MataKuliah06[5];
        mk[0] = new MataKuliah06(kodeMK:"0001", namaMk:"Internet of Things", sks:3);
        mk[1] = new MataKuliah06(kodeMK:"0002", namaMk:"Algoritma dan Struktur Data", sks:2);
        mk[2] = new MataKuliah06(kodeMK:"0003", namaMk:"Algoritma dan Pemrograman", sks:2);
        mk[3] = new MataKuliah06(kodeMK:"0004", namaMk:"Praktikum Algoritma dan Struktur Data", sks:3);
        mk[4] = new MataKuliah06(kodeMK:"0005", namaMk:"Praktikum Algoritma dan Pemrograman", sks:3);

        m.addMk(mk);

        int menu = 0;
        do {
            System.out.println(x:"************************************************");
            System.out.println(x:" Sistem Pengelolaan Data Nilai Mahasiswa Semester ");
            System.out.println(x:"************************************************");
            System.out.println(x:"1. Input Nilai");
            System.out.println(x:"2. Tampil Nilai");
            System.out.println(x:"3. Mencari Nilai Mahasiswa");
            System.out.println(x:"4. Urut Data Nilai");
            System.out.println(x:"5. Hapus Mahasiswa");
            System.out.println(x:"6. Keluar");
            System.out.println(x:"************************************************");
            System.out.print(s:"Pilih Menu: ");
            menu = sc.nextInt();
            sc.nextLine();

            switch (menu) {
                case 1:
                    System.out.print(s:"Kode MK: ");
                    String kd = sc.nextLine();
                    System.out.print(s:"NIM: ");
                    String nimMhs = sc.nextLine();
                    System.out.print(s:"Nilai: ");
                    double nilai = sc.nextDouble();

                    Mahasiswa06 mahaSiswas = m.searchMhs(nimMhs);
                    MataKuliah06 mataKuliahs = m.searchMk(kd);
                    if (mahaSiswas != null && mataKuliahs != null) {
                        NilaiMain06 nil = new NilaiMain06(kd, nilai, mahaSiswas, mataKuliahs);
                        na.add(nil);
                    } else {
                        System.out.println(x:"Mahasiswa atau Mata Kuliah tidak ditemukan.");
                    }
                    break;
```



```
jobsheet13_collection > assignment > J NilaiMain06.java > ...
11  public class NilaiMain06 {
110     public static void main(String[] args) {
169                 case 2:
170                     System.out.println(x:"Daftar Nilai Mahasiswa");
171                     System.out.println(x:"NIM\tNama\tMata Kuliah\tSKS\tNilai");
172                     for (NilaiMain06 nilaiEntry : na) {
173                         System.out.println(nilaiEntry.mhs.nim + "\t" + nilaiEntry.mhs.nama + "\t" + nilaiEntry.mk.namaMk + "\t" + nilaiEntry.mk.sks + "\t" + nilaiEntry.nilai);
174                     }
175                     break;
176
177                 case 3:
178                     System.out.println(x:"Cari Nilai Mahasiswa");
179                     System.out.print(s:"NIM: ");
180                     nimMhs = sc.nextLine();
181                     System.out.println(x:"NIM\tNama\tMata Kuliah\tSKS\tNilai");
182                     for (NilaiMain06 nilaiEntry : na) {
183                         if (nilaiEntry.mhs.nim.equals(nimMhs)) {
184                             System.out.println(nilaiEntry.mhs.nim + "\t" + nilaiEntry.mhs.nama + "\t" + nilaiEntry.mk.namaMk + "\t" + nilaiEntry.mk.sks + "\t" + nilaiEntry.nilai);
185                         }
186                     }
187                     break;
188
189                 case 4:
190                     System.out.println(x:"Urutkan Data Mahasiswa Berdasarkan NIM");
191                     na.sort(Comparator.comparing(entry -> entry.mhs.nim));
192                     System.out.println(x:"NIM\tNama\tMata Kuliah\tSKS\tNilai");
193                     for (NilaiMain06 nilaiEntry : na) {
194                         System.out.println(nilaiEntry.mhs.nim + "\t" + nilaiEntry.mhs.nama + "\t" + nilaiEntry.mk.namaMk + "\t" + nilaiEntry.mk.sks + "\t" + nilaiEntry.nilai);
195                     }
196                     break;
197
198                 case 5:
199                     m.removeMhs();
200                     break;
201
202                 case 6:
203                     System.out.println(x:"**************************************************");
204                     System.out.println(x:"Terima Kasih");
205                     System.out.println(x:"**************************************************");
206                     break;
207             }
208         } while (menu < 6 && menu > 0);
209
210         sc.close();
211     }
212 }
```

--- *** ---