

# **Laporan Praktikum Algoritma dan Struktur Data**

## **Jobsheet 16 - Collection**

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



**Nama : Annisa**  
**NIM : 2341760032**  
**Kelas : SIB 1E**  
**Prodi : D-IV Sistem Informasi Bisnis**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

## 16.1. Tujuan Praktikum

## 16.2. Kegiatan Praktikum 1

### 16.2.1. Percobaan 1

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

```
1 import java.util.ArrayList;
2 import java.util.LinkedList;
3 import java.util.List;
4
5 public class ContohList04 {
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",
13            l.get(0), l.size(), l.get(l.size() - 1));
14
15        l.add(4);
16        l.remove(0);
17        System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
18            l.get(0), l.size(), l.get(l.size() - 1));
```

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

```
1 System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
2     names.get(0), names.size(), names.get(names.size() - 1));
3 names.set(0, "My Kid");
4 System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
5     names.get(0), names.size(), names.get(names.size() - 1));
6 System.out.println("Names: " + names.toString());
7 }
8 }
```

### 16.2.2. Verifikasi Hasil Percobaan

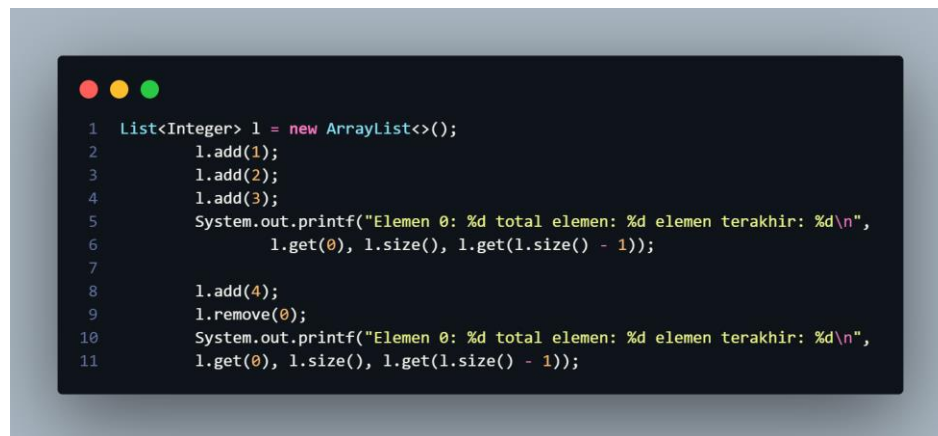
```
Elemen 0: 1 total elemen: 4 elemen terakhir: Cireng
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Nourreen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My Kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My Kid, Akhleema, Shannum, Uwais, Al-Qarni]
```

### 16.2.3. Pertanyaan Percobaan

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

Pada baris kode tersebut, semua jenis data bisa ditampung ke dalam sebuah `ArrayList` karena tidak ada pembatasan tipe yang ditetapkan pada `List`, Dimana pendeklarasian `List` dilakukan tanpa parameter generik, seperti `List l = new ArrayList();`, dan menggunakan versi "mentah" dari `List`, yang secara implisit memperbolehkan objek dari tipe apa pun.

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



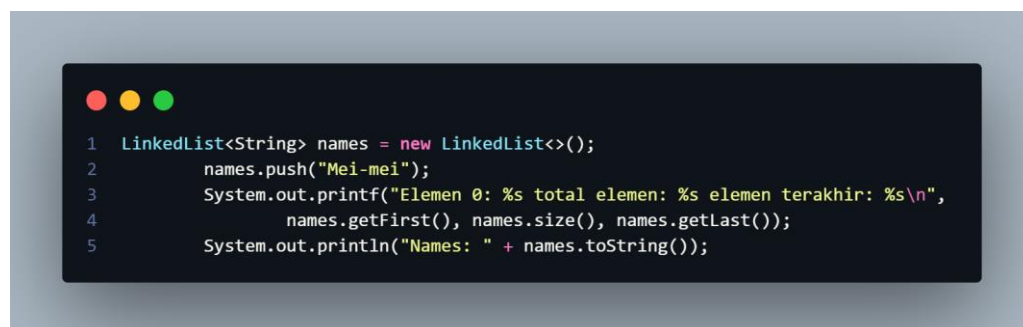
```
1 List<Integer> l = new ArrayList<>();
2     l.add(1);
3     l.add(2);
4     l.add(3);
5     System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %d\n",
6         l.get(0), l.size(), l.get(l.size() - 1));
7
8     l.add(4);
9     l.remove(0);
10    System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %d\n",
11        l.get(0), l.size(), l.get(l.size() - 1));
```

Kode yang dimodifikasi yaitu deklarasi daftaryang awalnya `List l = new ArrayList();` menjadi `List<Integer> l = new ArrayList<>();`, memastikan bahwa daftar hanya dapat menampung elemen bertipe `Integer`. Baris kode `l.add("Cireng");` dihapus karena elemen bertipe `String` tidak sesuai dengan tipe `Integer`. Penggunaan format specifier pada `printf` juga diubah dari `%s` menjadi `%d` untuk mencocokkan tipe `Integer`. Perubahan ini dilakukan untuk memastikan keamanan tipe data, sehingga hanya `Integer` yang dapat ditambahkan ke daftar, menghindari potensi kesalahan runtime dan meningkatkan konsistensi serta keterbacaan kode. Dengan demikian, kesalahan tipe dapat dideteksi saat kompilasi, menjadikan kode lebih aman dan mudah dikelola.

#### 3. Ubah kode pada baris kode 38 menjadi seperti ini

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

Berikut merupakan modifikasi dan penambahan kode program pada soal nomor 3 & 4



```
1 LinkedList<String> names = new LinkedList<>();
2     names.push("Mei-mei");
3     System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
4         names.getFirst(), names.size(), names.getLast());
5     System.out.println("Names: " + names.toString());
```

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

```
Elemen 0: 1 total elemen: 3 elemen terakhir: 3
Elemen 0: 2 total elemen: 3 elemen terakhir: 4
Elemen 0: Mei-mei total elemen: 1 elemen terakhir: Mei-mei
Names: [Mei-mei]
Elemen 0: Mei-mei total elemen: 6 elemen terakhir: Al-Qarni
Elemen 0: My Kid total elemen: 6 elemen terakhir: Al-Qarni
Names: [My Kid, Noreen, Akhleema, Shannum, Uwais, Al-Qarni]
```

Dengan mengganti deklarasi `names` menjadi `LinkedList<String> names = new LinkedList<>();` dan menambahkan baris `names.push("Mei-mei");`, secara eksplisit memilih `LinkedList` sebagai jenis implementasi untuk daftar. Penggunaan metode `push()` menambahkan elemen ke awal daftar, yang berbeda dengan `add()` yang menambahkan elemen di akhir daftar. Meskipun demikian, penggunaan metode `getFirst()` dan `getLast()` dalam perintah `printf` tetap sama, mengambil elemen pertama dan terakhir dari daftar, secara berurutan. Oleh karena itu, output masih akan menampilkan elemen pertama, jumlah total elemen, elemen terakhir, dan seluruh isi daftar sesuai dengan implementasi `LinkedList`. Dengan demikian, perubahan ini mempengaruhi perilaku dan cara kerja daftar, tetapi tidak mengubah cara output hasil yang ditampilkan.

## 16.3. Kegiatan Praktikum 2

### 16.3.1. Tahapan Percobaan

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

```
1  import java.util.Iterator;
2  import java.util.Stack;
3
4  public class LoopCollection04 {
5      public static void main(String[] args) {
6          Stack<String> fruits = new Stack<>();
7          fruits.push("Banana");
8          fruits.add("Orange");
9          fruits.add("Watermelon");
10         fruits.add("Leci");
11         fruits.push("Salak");
12
13         for (String fruit : fruits) {
14             System.out.printf("%s ", fruit);
15         }
16         System.out.println("\n" + fruits.toString());
17
18         while (!fruits.empty()) {
19             System.out.printf("%s ", fruits.pop());
20         }
```

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

```
1  fruits.push("Melon");
2      fruits.push("Durian");
3      System.out.println(" ");
4      for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
5          String fruit = it.next();
6          System.out.printf("%s ", fruit);
7      }
8      System.out.println("");
9      fruits.stream().forEach(e -> {
10         System.out.printf("%s ", e);
11     });
12     System.out.println("");
13     for (int i = 0; i < fruits.size(); i++) {
14         System.out.printf("%s ", fruits.get(i));
15     }
16
17 }
18 }
```

### 16.3.2. Verifikasi Hasil Percobaan

```
Banana Orange Watermelon Leci Salak  
[Banana, Orange, Watermelon, Leci, Salak]  
Salak Leci Watermelon Orange Banana  
Melon Durian  
Melon Durian  
Melon Durian  
D:\Jobsheet 16>
```

### 16.3.3. Pertanyaan Percobaan

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

Perbedaan antara `push()` dan `add()` pada objek `fruits` terletak pada fungsionalitas masing-masing dalam konteks sebuah `Stack`. Saat menggunakan `push()`, elemen yang ditambahkan akan diletakkan di atas tumpukan (`stack`), sehingga menjadi elemen teratas. Hal ini khusus untuk `Stack`, dan sering digunakan Ketika menambahkan elemen baru ke dalam tumpukan, sehingga elemen tersebut menjadi yang paling atas. Di sisi lain, `add()` adalah metode umum yang ada dalam antarmuka `List`, yang juga diterapkan oleh `Stack`. Saat menggunakan `add()`, elemen akan ditambahkan di akhir tumpukan, yang dalam kasus ini sama dengan menambahkan elemen baru di atas tumpukan karena `Stack` adalah tipe koleksi terurut. Jadi, penggunaan `push()` lebih khusus untuk menambahkan elemen ke tumpukan, sementara `add()` digunakan untuk menambahkan elemen ke koleksi di akhir.

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

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

Jika baris program `fruits.push("Melon");` dan `fruits.push("Durian");` dihilangkan, maka tidak akan ada penambahan elemen "Melon" dan "Durian" ke dalam `Stack` namun elemen lainnya yang telah ditambah sebelumnya ("Banana", "Orange", "Watermelon", "Leci", dan "Salak") masih tetap berada di dalam `Stack`. Karena LIFO dari `Stack`, elemen-elemen ini akan tetap diproses dan dicetak terlebih dahulu, sesuai dengan urutan mereka dalam `Stack`.

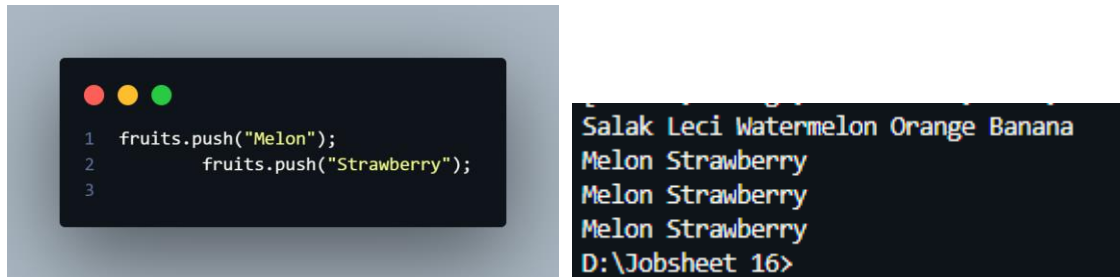
#### 3. Jelaskan fungsi dari baris 46-49?

Baris kode tersebut melakukan iterasi melalui setiap elemen dalam objek `fruits` yang merupakan sebuah `Stack` dan bertipe `String`. Untuk melakukan iterasi, kode menggunakan iterator yang disediakan oleh objek `fruits`. Pada setiap iterasi, kode memeriksa apakah masih ada elemen yang tersisa menggunakan `it.hasNext()`. Jika masih ada, kode mengambil elemen berikutnya dari `fruits` menggunakan `it.next()` dan menyimpannya dalam variabel `fruit`. Kemudian, kode mencetak nilai `fruit` ke konsol dengan diakhiri spasi menggunakan `System.out.printf()`. Proses ini dilakukan untuk setiap elemen dalam `fruits`, sehingga semua elemen dicetak secara berurutan. Dengan demikian, baris kode tersebut bertanggung jawab untuk mencetak semua elemen dalam `fruits` ke konsol dengan menggunakan iterator.

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

Terdapat kesalahan kompilasi karena perbedaan tipe data. Ketika `fruits` dideklarasikan sebagai `List`, kompiler menganggapnya sebagai objek yang mengikuti antarmuka `List`, bukan sebagai `Stack`. Karena itu, hanya metode yang didefinisikan dalam antarmuka `List` yang dapat diakses langsung pada `fruits`. Namun, metode khusus `push()`, `pop()`, dan `peek()` tidak ada dalam antarmuka `List`, sehingga kompiler menghasilkan kesalahan kompilasi karena tidak dapat menemukan metode tersebut dalam konteks `List`.

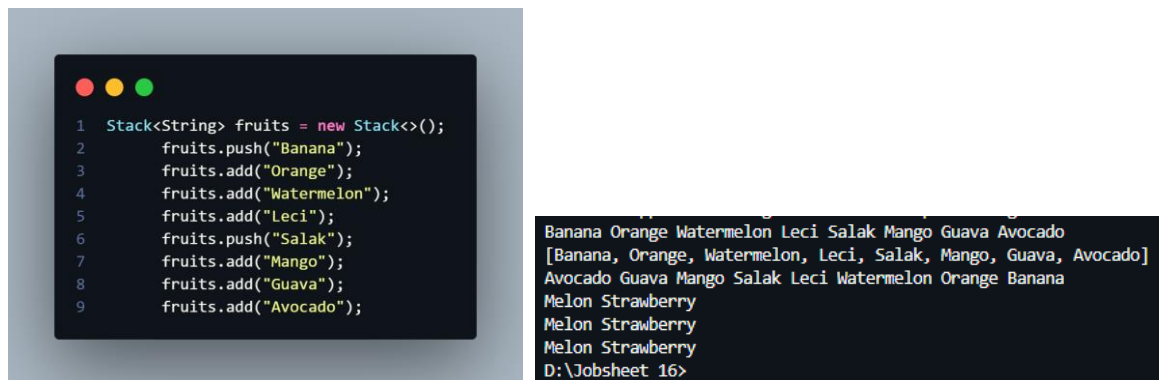
**5. Ganti elemen terakhir dari objek `fruits` menjadi "Strawberry"!**



```
1 fruits.push("Melon");
2     fruits.push("Strawberry");
3
```

```
Salak Leci Watermelon Orange Banana
Melon Strawberry
Melon Strawberry
Melon Strawberry
D:\Jobsheet 16>
```

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



```
1 Stack<String> fruits = new Stack<>();
2     fruits.push("Banana");
3     fruits.add("Orange");
4     fruits.add("Watermelon");
5     fruits.add("Leci");
6     fruits.push("Salak");
7     fruits.add("Mango");
8     fruits.add("Guava");
9     fruits.add("Avocado");
```

```
Banana Orange Watermelon Leci Salak Mango Guava Avocado
[Banana, Orange, Watermelon, Leci, Salak, Mango, Guava, Avocado]
Avocado Guava Mango Salak Leci Watermelon Orange Banana
Melon Strawberry
Melon Strawberry
Melon Strawberry
D:\Jobsheet 16>
```

## 16.4. Kegiatan Praktikum 3

### 16.4.1. Tahapan Percobaan

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

```
1 public class Mahasiswa04 {
2
3     String nim;
4     String nama;
5     String notelp;
6
7     public Mahasiswa04() {
8     }
9
10    public Mahasiswa04(String nim, String nama, String notelp) {
11        this.nim = nim;
12        this.nama = nama;
13        this.notelp = notelp;
14    }
15
16    @Override
17    public String toString() {
18        return "Mahasiswa{" + "nim=" + nim + ", nama=" + nama + ", notelp=" + notelp + '}';
19    }
20 }
```

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

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

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

```
1 import java.util.ArrayList;
2 import java.util.Arrays;
3 import java.util.List;
4
5 public class ListMahasiswa04 {
6
7     List<Mahasiswa04> mahasiswas = new ArrayList<>();
8     public void tambah(Mahasiswa04... mahasiswa) {
9         mahasiswas.addAll(Arrays.asList(mahasiswa));
10    }
11
12    public void hapus(int index) {
13        mahasiswas.remove(index);
14    }
15
16    public void update(int index, Mahasiswa04 mhs) {
17        mahasiswas.set(index, mhs);
18    }
19
20    public void tampil() {
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 }
```



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

```
1 public static void main(String[] args) {
2     ListMahasiswa04 lm = new ListMahasiswa04();
3     Mahasiswa04 m = new Mahasiswa04("201234", "Noureen", "02xx1");
4     Mahasiswa04 m1 = new Mahasiswa04("201235", "Akhleema", "021xx2");
5     Mahasiswa04 m2 = new Mahasiswa04("201236", "Shannum", "021xx3");
6
7     // Menambahkan objek mahasiswa
8     lm.tambah(m, m1, m2);
9
10    // Menampilkan list mahasiswa
11    lm.tampil();
12
13    // Update mahasiswa
14    lm.update(lm.linearSearch("201235"), new Mahasiswa04("201235", "Akhleema Lela", "021xx2"));
15    System.out.println("");
16    lm.tampil();
17 }
18 }
```

#### 16.4.2. Verifikasi Hasil Percobaan

```
Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
Mahasiswa{nim= 201235, nama=Akhleema Lela, 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 kelebihan apa?

Pada fungsi tambah() menggunakan konsep atau method addAll() dari collections Dimana konsep ini Memiliki kelebihan yaitu dapat menambahkan element tanpa ada batas jumlah elemen yang ditambahkan.

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

```
1 // Update mahasiswa
2 lm.update(lm.binarySearch("201235"), new Mahasiswa04("201235", "Akhleema Lela", "021xx2"));
3 System.out.println("");
4 lm.tampil();
5 }
```

Pada perubahan tersebut tidak mempengaruhi hasil output yang dihasilkan oleh kode lainnya, seperti penambahan, penghapusan, pembaruan mahasiswa, serta tampilan daftar mahasiswa. Jadi, output dari kode akan tetap sama seperti sebelumnya, hanya dengan perubahan cara pencarian data mahasiswa tertentu.

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

Berikut kode program yang telah dimodifikasi dengan penambahan fungsi sorting ascending & descending

```
1 import java.util.ArrayList;
2 import java.util.Arrays;
3 import java.util.Collections;
4 import java.util.Comparator;
5 import java.util.List;
6
7 public class ListMahasiswa04 {
8
9     List<Mahasiswa04> mahasiswa = new ArrayList<>();
10
11     //fungsi sorting ascending berdasarkan nim
12     public void sortAscending() {
13         Collections.sort(mahasiswa, Comparator.comparing(mhs -> mhs.nim));
14     }
15
16     //fungsi sorting descending berdasarkan nim
17     public void sortDescending() {
18         Collections.sort(mahasiswa, (mhs1, mhs2) -> mhs2.nim.compareTo(mhs1.nim));
19     }
20
21     public void tambah(Mahasiswa04... mahasiswa) {
22         mahasiswa.addAll(Arrays.asList(mahasiswa));
23     }
24
25     public void hapus(int index) {
26         mahasiswa.remove(index);
27     }
28
29     public void update(int index, Mahasiswa04 mhs) {
30         mahasiswa.set(index, mhs);
31     }
32
33     public void tampil() {
34         mahasiswa.forEach(mhs -> {
35             System.out.println(mhs.toString());
36         });
37     }
38
39     int binarySearch(String nim) {
40         for (int i = 0; i < mahasiswa.size(); i++) {
41             if (nim.equals(mahasiswa.get(i).nim)) {
42                 return i;
43             }
44         }
45         return -1;
46     }
47
48     public static void main(String[] args) {
49         ListMahasiswa04 lm = new ListMahasiswa04();
50         Mahasiswa04 m = new Mahasiswa04("201234", "Noureen", "02xx1");
51         Mahasiswa04 m1 = new Mahasiswa04("201235", "Akhleema", "021xx2");
52         Mahasiswa04 m2 = new Mahasiswa04("201236", "Shannum", "021xx3");
53
54         // Menambahkan objek mahasiswa
55         lm.tambah(m, m1, m2);
56
57         // Menampilkan list mahasiswa
58         System.out.println("Daftar Mahasiswa:");
59         lm.tampil();
60
61         // Update mahasiswa
62         lm.update(lm.binarySearch("201235"), new Mahasiswa04("201235", "Akhleema Lela", "021xx2"));
63         System.out.println("\nSetelah Update:");
64         lm.tampil();
65
66         // Sorting Ascending
67         System.out.println("\nSorting Ascending:");
68         lm.sortAscending();
69         lm.tampil();
70
71         // Sorting Descending
72         System.out.println("\nSorting Descending:");
73         lm.sortDescending();
74         lm.tampil();
75     }
76 }
```

Berikut merupakan output dari kode program di atas

```
Daftar Mahasiswa:
Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Setelah Update:
Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
Mahasiswa{nim= 201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Sorting Ascending:
Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
Mahasiswa{nim= 201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}

Sorting Descending:
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}
Mahasiswa{nim= 201235, nama=Akhleema Lela, notelp=021xx2}
Mahasiswa{nim= 201234, nama=Noureen, notelp=02xx1}
```

## 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.

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

### Jawab

Berikut merupakan kode program dari tugas nomor 1 dan 2, disini menggunakan 5 class yaitu Mahasiswa, Nilai, MataKuliah, ListNilai, dan MahasiswaMain

#### Class Mahasiswa

```
1 package Tugas;
2
3 /**
4  * Mahasiswa04
5  */
6 public class Mahasiswa04 {
7
8     String nim, nama, telf;
9
10    public Mahasiswa04(String nim, String nama, String telf) {
11        this.nim = nim;
12        this.nama = nama;
13        this.telf = telf;
14    }
15    void tampil() {
16        System.out.printf("%-10s %-15s %s", nim, nama, telf);
17        System.out.println("");
18    }
19 }
```

#### Class Nilai

```
1 package Tugas;
2
3 public class Nilai04 {
4
5     double nilai;
6     String kode, kdMatkul, nimMhs;
7
8     public Nilai04(String kode, double nilai, String kdMatkul, String nimMhs) {
9         this.nilai = nilai;
10        this.kode = kode;
11        this.kdMatkul = kdMatkul;
12        this.nimMhs = nimMhs;
13    }
14
15    void tampil() {
16        System.out.printf("%d", nilai);
17    }
18 }
```

## Class MataKuliah

```
1 package Tugas;
2
3 public class MataKuliah04 {
4
5     String kode, namaMatkul;
6     int sks;
7
8     public MataKuliah04(String kode, String namaMatkul, int sks) {
9         this.kode = kode;
10        this.namaMatkul = namaMatkul;
11        this.sks = sks;
12    }
13
14    void tampil() {
15        System.out.printf("%-10s %-40s %d", kode, namaMatkul, sks);
16        System.out.println("");
17    }
18 }
```

## Class ListNilai

```
1 package Tugas;
2 import java.util.ArrayList;
3 import java.util.Arrays;
4 import java.util.Collections;
5 import java.util.Comparator;
6 import java.util.List;
7
8
9 public class ListNilai04 {
10
11     List<Mahasiswa04> mhs = new ArrayList<>();
12     List<MataKuliah04> matkul = new ArrayList<>();
13     List<Nilai04> listNilai = new ArrayList<>();
14
15     // input data Mahasiswa
16     public void inputDataMahasiswa(Mahasiswa04... mahasiswa) {
17         mhs.addAll(Arrays.asList(mahasiswa));
18     }
19
20     // input list nilai mahasiswa
21     public void inputNilaiMahasiswa(Nilai04 n) {
22         listNilai.add(n);
23     }
24
25     // input data Mata Kuliah
26     public void inputDataMataKuliah(MataKuliah04... mk) {
27         matkul.addAll(Arrays.asList(mk));
28     }
29     // tampil data Mahasiswa
30     public void tampilDataMahasiswa() {
31         mhs.stream().forEach(mhs -> {
32             mhs.tampil();
33         });
34     }
35
36     // tampil data Mata Kuliah
37     public void tampilDataMataKuliah() {
38         matkul.stream().forEach(matkul -> {
39             matkul.tampil();
40         });
41     }
42
43     // tampil list nilai mahasiswa
44     public void tampilListNilai() {
45         for (int i = 0; i < listNilai.size(); i++) {
46             int indexMhs = linearSearchMHS(listNilai.get(i).nimMhs);
47             int indexMatkul = linearSearchMatkul(listNilai.get(i).kdMatkul);
48             System.out.printf("%-10s %-12s %-10.2f\n",
49                 mhs.get(indexMhs).nama, matkul.get(indexMatkul).namaMatkul,
50                 listNilai.get(i).nilai);
51         }
52     }
53
54     // mencari nim mahasiswa
55     int linearSearchMHS(String nim) {
56         for (int i = 0; i < mhs.size(); i++) {
57             if (nim.equals(mhs.get(i).nim)) {
58                 return i;
59             }
60         }
61         return -1;
62     }
63
64     // mencari kode mata kuliah
65     int linearSearchMatkul(String kode) {
66         for (int i = 0; i < matkul.size(); i++) {
67             if (kode.equals(matkul.get(i).kode)) {
68                 return i;
69             }
70         }
71         return -1;
72     }
73
74     // mencari nilai mahasiswa
75     int linearSearchNilaiMahasiswa(String nimMahasiswa) {
76         for (int i = 0; i < listNilai.size(); i++) {
77             if (nimMahasiswa.equals(listNilai.get(i).nimMhs)) {
78                 return i;
79             }
80         }
81         return -1;
82     }
83
84     // sorting data nilai mahasiswa
85     public void sortingData() {
86         Collections.sort(listNilai, new Comparator<Nilai04>() {
87             public int compare(Nilai04 n1, Nilai04 n2) {
88                 return Double.compare(n1.nilai, n2.nilai);
89             }
90         });
91     }
92
93     //menghapus data pertama sesuai dengan fifo
94     public void removeList () {
95         listNilai.remove(0);
96     }
97 }
```

## Class MahasiswaMain

```
1 package tugas;
2 import java.util.Scanner;
3
4 public class MahasiswaMain {
5     public static void main(String[] args) {
6         ListNilai04 ln = new ListNilai04();
7
8         // Input data mahasiswa
9         Mahasiswa04 m1 = new Mahasiswa04("00001", "Thalhah", "021xxx");
10        Mahasiswa04 m2 = new Mahasiswa04("20002", "Zubair", "021xxx");
11        Mahasiswa04 m3 = new Mahasiswa04("20003", "Abdur-Rahman", "021xxx");
12        Mahasiswa04 m4 = new Mahasiswa04("20004", "Sa'ad", "021xxx");
13        Mahasiswa04 m5 = new Mahasiswa04("20005", "Sa'ad", "021xxx");
14        Mahasiswa04 m6 = new Mahasiswa04("20006", "Ibaidah", "021xxx");
15        ln.addDataMahasiswa(m1, m2, m3, m4, m5, m6);
16
17        // Input data mata kuliah
18        MataKuliah04 mt04 = new MataKuliah04("00001", "Internet - 'of Things");
19        MataKuliah04 mt05 = new MataKuliah04("00002", "Algoritma - 'dan Struktur Data", 2);
20        MataKuliah04 mt06 = new MataKuliah04("00003", "Algoritma - 'dan Struktur Data", 2);
21        MataKuliah04 mt07 = new MataKuliah04("00004", "Praktikum - 'Algoritma dan Struktur Data", 3);
22        MataKuliah04 mt08 = new MataKuliah04("00005", "Praktikum - 'Algoritma dan Pemrograman", 3);
23        ln.addDataMataKuliah(mt04, mt05, mt06, mt07, mt08);
24
25        Scanner sc = new Scanner(System.in);
26        System.out.println("=====");
27        System.out.println("SISTEM PENGOLAHAN DATA NILAI MAHASISMA SEMESTER");
28        System.out.println("=====");
29        boolean isTrue = true;
30        while (isTrue){
31
32            System.out.println("\n1. Input Nilai");
33            System.out.println("2. Tampil Nilai");
34            System.out.println("3. Mencari Nilai Mahasiswa");
35            System.out.println("4. Urut Data Nilai");
36            System.out.println("5. Menghapus data pertama");
37            System.out.println("6. Keluar");
38            System.out.println("=====");
39            System.out.print("Pilih : ");
40            int choice = sc.nextInt();
41            switch (choice) {
42                case 1:
43                    // menu input nilai
44                    System.out.println("Masukkan data");
45                    System.out.print("Kode : ");
46                    String kd = sc.next();
47                    System.out.print("Nilai : ");
48                    double inputNilai = sc.nextDouble();
49
50                    System.out.println("DAFTAR MAHASISMA");
51                    System.out.println("=====");
52                    System.out.printf("%-10s %-15s %s\n",
53                        "NIM", "Nama", "Temp");
54                    ln.tampilDataMahasiswa();
55                    System.out.print("Pilih mahasiswa by nim: ");
56                    String tempNim = sc.next();
57
58                    System.out.println("DAFTAR MATA KULIAH");
59                    System.out.println("=====");
60                    System.out.printf("%-10s %-40s %s\n",
61                        "Kode", "Mata Kuliah", "SKS");
62
63                    ln.tampilDataMataKuliah();
64                    System.out.print("Pilih MK by kode: ");
65                    String tempKd = sc.next();
66
67                    Nilai04 temp = new Nilai04(kd, inputNilai,
68                        tempKd, tempNim);
69                    ln.inputNilaiMahasiswa(temp);
70
71                    break;
72                case 2:
73                    // menu menampilkan nilai
74                    System.out.println("DAFTAR NILAI MAHASISMA");
75                    System.out.println("=====");
76                    System.out.printf("%-10s %-15s %-40s %-10s %s\n",
77                        "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
78                    ln.tampilListNilai();
79                    break;
80                case 3:
81                    // menu untuk mencari nilai mahasiswa
82                    System.out.println("DAFTAR NILAI MAHASISMA");
83                    System.out.println("=====");
84                    System.out.printf("%-10s %-15s %-40s %-10s %s\n",
85                        "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
86                    ln.tampilListNilai();
87                    System.out.print("Masukkan data mahasiswa(nim): ");
88                    String tempNIM = sc.next();
89
90                    int indexSearch = ln.linearSearchNilaiMahasiswa(tempNIM);
91                    int indexMatkul = 0;
92                    for(int i = 0; i < ln.matkul.size(); i++){
93                        if(ln.listNilai.get(indexSearch).
94                            kdMatkul.equals(ln.matkul.get(i).kode)){
95                            indexMatkul = i;
96                            break;
97                        }
98                    }
99                    System.out.printf("%-10s %-15s %-40s %-10s %s\n",
100                        "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
101                    System.out.printf("%-10s %-15s %-40s %-10s %-10s %s\n",
102                        ln.mhs.get(indexSearch).nim,
103                        ln.mhs.get(indexSearch).nama,
104                        ln.matkul.get(indexMatkul).namaMatkul,
105                        ln.matkul.get(indexMatkul).sks,
106                        ln.listNilai.get(indexSearch).nilai);
107                    System.out.println("Total SKS = " + ln.matkul.get(indexMatkul).sks + " telah diambil");
108                    break;
109                case 4:
110                    // menu untuk mengurutkan nilai mahasiswa kecil->besar
111                    System.out.println("DAFTAR NILAI MAHASISMA");
112                    System.out.println("=====");
113                    System.out.printf("%-10s %-15s %-40s %-10s %s\n",
114                        "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
115                    ln.sortingData();
116                    ln.tampilListNilai();
117                    break;
118                case 5:
119                    // menu untuk menghapus data pertama nilai mahasiswa
120                    System.out.println("Data berhasil dihapus");
121                    ln.removeList();
122                    break;
123                case 6:
124                    // menu untuk keluar
125                    isTrue = false;
126                    break;
127                default:
128                    System.out.println("Inputan salah");
129            }
130        }
131    }
132 }
133 }
```

Berikut merupakan output dari kode program

```
=====
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
=====

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

```
Pilih : 1
Masukkan data
Kode : 0001
Nilai : 80.75
DAFTAR MAHASISWA
*****
NIM      Nama      Telp
20001    Thalhah    021xxx
20002    Zubair      021xxx
20003    Abdur-Rahman 021xxx
20004    Sa'ad       021xxx
20005    Sa'ad       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 Struktur Data 2
00004     Praktikum Algoritma dan Struktur Data 3
00005     Praktikum Algoritma dan Pemrograman 3
Pilih MK by kode: 00001
```

## Tampil Nilai

```
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Menghapus data pertama
6. Keluar
*****
Pilih : 2
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah      SKS      Nilai
Thalhah  Internet of Things 80.75
```

## Pencarian Data Mahasiswa

```
Pilih : 3
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah      SKS      Nilai
Abdur-Rahman Algoritma dan Struktur Data 72.50
Ubaidah     Praktikum Algoritma dan Pemrograman 88.75
Thalhah     Internet of Things 80.75
Sa'ad       Praktikum Algoritma dan Struktur Data 88.00
Zubair      Algoritma dan Struktur Data 90.50
Sa'ad       Praktikum Algoritma dan Pemrograman 95.50
Masukkan data mahasiswa(nim):
```

## Pengurutan Data Nilai

```
Pilih : 4
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah      SKS      Nilai
Abdur-Rahman Algoritma dan Struktur Data 72.50
Ubaidah     Praktikum Algoritma dan Pemrograman 88.75
Thalhah     Internet of Things 80.75
Sa'ad       Praktikum Algoritma dan Struktur Data 88.00
Zubair      Algoritma dan Struktur Data 90.50
Sa'ad       Praktikum Algoritma dan Pemrograman 95.50
```

## Penghapusan Data Pertama

```
Pilih : 5
Data berhasil dihapus

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Menghapus data pertama
6. Keluar
*****
Pilih : 2
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah      SKS      Nilai
Ubaidah     Praktikum Algoritma dan Pemrograman 88.75
Thalhah     Internet of Things 80.75
Sa'ad       Praktikum Algoritma dan Struktur Data 88.00
Zubair      Algoritma dan Struktur Data 90.50
Sa'ad       Praktikum Algoritma dan Pemrograman 95.50
```

NB: mohon maaf bu atas keterlambatan pengumpulan dikarenakan sudah selesai tapi hilang jadi buat dari awal lagi