

**Tugas Praktikum Mata Kuliah  
Alogaritma dan Struktur Data**

**Jobsheet 16 : Collections**

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



**Eka Putri Natalya Kabelen**

**2341760107**

**SIB 1E**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**Tahun Ajaran 2024**

## 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 method berisi kode program seperti di bawah ini

```
J ContohList11.java > ContohList11 > main(String[])
1  import java.util.ArrayList;
2  import java.util.LinkedList;
3  import java.util.List;
4
5  public class ContohList11 {
6      public static void main(String[] args) {
7
8          List 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:"Elemen 0: %d total elemen: %d elemen terakhir: %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:"Elemen 0: %d total elemen: %d elemen terakhir: %s\n",l.get(index:0),
19             l.size(),l.get(l.size() - 1));
```

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

```
21 // Tipe data ditentukan (khusus)
22 List<String> names = new LinkedList<>();
23 names.add(e:"Noureen");
24 names.add(e:"Akhleema");
25 names.add(e:"Shannum");
26 names.add(e:"Uwais");
27 names.add(e:"Al-Qarni");
28
29 System.out.printf(format:"Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
30     names.get(index:0), names.size(), names.get(names.size() - 1));
31
32 names.set(index:0, element:"My kid");
33
34 System.out.printf(format:"Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
35     names.get(index:0), names.size(), names.get(names.size() - 1));
36
37 System.out.println("Names: " + names.toString());
38 }
39 }
```

3. Hasil 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]
PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma&StrukturData
```

### 16.2.3. Pertanyaan Percobaan

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

**Jawab :**

Karena arrayList tersebut tidak terdapat kurung sudut yang menyimpan secara spesifik data yang akan disimpan didalamnya. sehingga arraylist pada baris kode 25-36 diatas bersifat sintak umum yang dapat menyimpan data segala jenis tipe data.

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

**Jawab :**

Berikut adalah hasil modifikasinya, jika kita lihat pada baris ke-12, terdapat sebuah error yang dimana tipe data yang dimasukkan yaitu string tidak sesuai dengan tipe data arraylist nya yang sudah dibuat spesifik menjadi integer.

```
8      List<Integer> l = new ArrayList<>();
9      l.add(e:1);
10     l.add(e:2);
11     l.add(e:3);
12     l.add(e:"Cireng");
```

3. Ubah kode pada baris kode 38 menjadi seperti ini

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

**Jawab :**

```
21     // Tipe data ditentukan (khusus)
22     List<String> names = new LinkedList<>();
```

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 :**

```
39     ((LinkedList<String>) names).push(e:"Mei-mei");
40     System.out.printf(format:"Elemen 0: %s total elemen : %s elemen terakhir : %s\n",
41         ((LinkedList<String>) names).getFirst(), names.size(), ((LinkedList<String>) names).getLast());
42     System.out.println("Names: " + names.toString());
43 }
44 }
```

Hasil run

```
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]
Elemen 0: Mei-mei total elemen : 6 elemen terakhir : Al-Qarni
Names : [Mei-mei, My kid, Akhleema, Shannum, Uwais, Al-Qarni]
PS C:\Users\ASUS\Documents\SEMESTER 2\Algoritma&StrukturData\JOBSHEET 16>
```

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

**Jawab :**

yang dapat saya jelaskan adalah kita dapat membatasi tipe data yang akan dimasukkan dengan perintah dengan menggunakan class LinkedList kita dapat menggunakan method linked list seperti push(), getFirst(), getLast(), akan tetapi ketika menggunakan class List tidak dapat digunakan.

### 16.3. Kegiatan Praktikum 2

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

```
3 public class LoopCollection11 {  
    Run | Debug  
4     public static void main(String[] args) {  
5  
6         Stack<String> fruits = new Stack<>();  
7         fruits.push(item:"Banana");  
8         fruits.add(e:"Orange");  
9         fruits.add(e:"Watermelon");  
10        fruits.add(e:"Leci");  
11        fruits.push(item:"Salak");  
12  
13        for(String fruit : fruits){  
14            System.out.printf(format:"%s ", fruit);  
15        }  
16  
17        System.out.println("\n" + fruits.toString());  
18  
19        while(!fruits.empty()){  
20            System.out.printf(format:"%s ", fruits.pop());  
21        }  
    }  
}
```

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

```
26        fruits.push(item:"Melon");  
27        fruits.push(item:"Durian");  
28        System.out.println(x:"");  
29        for(Iterator<String> it = fruits.iterator(); it.hasNext();){  
30            String fruit = it.next();  
31            System.out.printf(format:"%s ", fruit);  
32        }  
33        System.out.println(x:"");  
34        fruits.stream().forEach(e ->{  
35            System.out.printf(format:"%s ", e);  
36        });  
37        System.out.println(x:"");  
38        for(int i=0; i<fruits.size(); i++){  
39            System.out.printf(format:"%s ", fruits.get(i));  
40        }  
41    }  
42 }
```

3. Hasil run code program

```
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?

**Jawab :**

fungsi push() dan add() diatas sama-sama menambahkan elemen, yang membedakannya adalah push() merupakan fungsi dari interface stack sedangkan add() tidak atau bukan interface dari stack.

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

**Jawab :**

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

Yang terjadi adalah data "melon" dan "durian" tidak muncul, karena saat kita mengakses fungsi push() artinya kita akan menambahkan data "melon" dan "durian".

3. Jelaskan fungsi dari baris 46-49?

**Jawab :**

- pada baris 46 berfungsi untuk proses inisialisasi dan mengatur aturan perulangan sesuai perintah yang di instruksikan
- pada baris 47 adalah untuk pembuatan variabel fruit menjadi it.next()
- pada baris 48 berfungsi untuk mencetak variabel fruit

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

**Jawab :**

Yang terjadi adalah eror pada bagian push, empty, pop karena tidak dapat dijalankan pada interface list, yang mana perintah yang diatas adalah fungsi dari interface stack

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

**Jawab :**

```
22     }  
23     fruits.push(item:"Melon");  
24     fruits.push(item:"Strawberry");  
25     System.out.println(x:"");
```

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

**Jawab :**

```
38     //tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!  
39     fruits.push(item:"Manggo");  
40     fruits.push(item:"Guava");  
41     fruits.push(item:"Avocado");  
42  
43     System.out.println(x:"");  
44     System.out.println(x:"\nData sesudah disorting ");  
45     Collections.sort(fruits);  
46     Iterator i= fruits.iterator();
```

## 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.

```
J Mahasiswa11.java > ...
1  import java.util.ArrayList;
2  import java.util.Arrays;
3  import java.util.LinkedList;
4  import java.util.List;
5
6  public class Mahasiswa11 {
7      String nim;
8      String nama;
9      String notelp;
10
11     public Mahasiswa11 () {
12     }
13     public Mahasiswa11(String nim, String nama, String notelp){
14         this.nim = nim;
15         this.nama = nama;
16         this.notelp = notelp;
17     }
18
19     @Override
20     public String toString(){
21         return "Mahasiswa{" + "nim= " + nim + ", nama=" + nama + ", notelp=" + notelp + '}';
22     }
23 }
```

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

```
5  public class ListMahasiswa11 {
6      List<Mahasiswa11> mahasiswas = new ArrayList<>();
7  }
```

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

```
8      public void tambah(Mahasiswa11... 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, Mahasiswa11 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     }
```

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

```

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         }

```

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

```

35 public static void main(String[] args) {
36     ListMahasiswa11 lm = new ListMahasiswa11();
37     Mahasiswa11 m = new Mahasiswa11(nim:"201234", nama:"Noureen", notelp:"021xx1");
38     Mahasiswa11 m1 = new Mahasiswa11(nim:"201235", nama:"Akhleema", notelp:"021xx2");
39     Mahasiswa11 m2 = new Mahasiswa11(nim:"201236", nama:"Shannum", notelp:"021xx3");
40
41     // Menambahkan Objek Mahasiswa
42     lm.tambah(m, m1, m2);
43     // Menampilkan List Mahasiswa
44     lm.tampil();
45     // Update Mahasiswa
46     lm.update(lm.LinearSearch(nim:"201235"), new Mahasiswa11(nim:"201235", nama:"Akhleema Lela",
47     notelp:"021xx2"));
48     System.err.println(x:"");
49     lm.tampil();

```

6. Hasil run code program

```

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}

```

### 16.4.3. Pertanyaan Percobaan

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

**Jawab :**

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!

**Jawab :**

```

51 //tugas no 2
52 Mahasiswa11 key = new Mahasiswa11(nim:"201235", nama:null, notelp:null);
53 lm.update(Collections.binarySearch(lm.mahasiswas, key, new MhsComparator11()),
54     new Mahasiswa11(nim:"201235", nama:"Akhleema", notelp:"021xx2"));
55 System.out.println(x:"");
56 lm.tampil();
57 }
58

```

```

J MhsComparator11.java > MhsComparator11 > compare(Mahasiswa11, Mahasiswa11)
1  import java.util.Comparator;
2  public class MhsComparator11 implements Comparator<Mahasiswa11>{
3
4      public int compare(Mahasiswa11 mhs1, Mahasiswa11 mhs2) {
5          if (mhs1.nim == mhs2.nim) {
6              return 0;
7          } else {
8
9              return -1;
10         }
11     }
12
13     @Override
14     public Comparator<Mahasiswa11> reversed() {
15         return Comparator.super.reversed();
16     }
17 }

```

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}

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

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

**Jawab :**

```

26 // no 3 //
27 void ascendingSort() {
28     this.mahasiswas.sort((Mahasiswa11 d1, Mahasiswa11 d2) -> d1.nama.compareTo(d2.nama));
29 }
30
31 void descending() {
32     this.mahasiswas.sort((Mahasiswa11 d1, Mahasiswa11 d2) -> d2.nama.compareTo(d1.nama));
33 }
34
68     System.out.println(x:"Ascending : ");
69     lm.ascendingSort();
70     lm.tampil();
71
72     System.out.println(x:"\n");
73
74     System.out.println(x:"Descending : ");
75     lm.descending();
76     lm.tampil();
77 }
78 }

```



Hasil run

```
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}

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

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

Descending :
Mahasiswa{nim= 201236, nama=Shannum, notelp=021xx3}
Mahasiswa{nim= 201234, nama=Noureen, notelp=021xx1}
Mahasiswa{nim= 201235, nama=Akhleema, notelp=021xx2}
PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma&StrukturD
```

## 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.

Code program class mahasiswa

```
TUGAS > J Mahasiswa.java > ...
1  package TUGAS;
2  public class Mahasiswa {
3      String nim, nama, telf;
4
5      public Mahasiswa(String nim, String nama, String telf) {
6          this.nim = nim;
7          this.nama = nama;
8          this.telf = telf;
9      }
10     void tampil() {
11         System.out.printf(format:"%-10s %-15s %s", nim, nama, telf);
12         System.out.println(x:"");
13     }
14 }
```

## Code program class matakuliah

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

## Code program class nilai

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

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

## Code program class Main

```
1 package TUGAS;
2 import java.util.Scanner;
3 public class Main {
4     public static void main(String[] args) {
5         ListNilai ln = new ListNilai();
6
7         // Input data mahasiswa
8         Mahasiswa m1 = new Mahasiswa("20001", "Thalhah", "021xxx");
9         Mahasiswa m2 = new Mahasiswa("20002", "Zubair", "021xxx");
10        Mahasiswa m3 = new Mahasiswa("20003", "Abdur-Rahman", "021xxx");
11        Mahasiswa m4 = new Mahasiswa("20004", "Sa'ad", "021xxx");
12        Mahasiswa m5 = new Mahasiswa("20005", "Sa'ad", "021xxx");
13        Mahasiswa m6 = new Mahasiswa("20006", "Ubaidah", "021xxx");
14        ln.inputDataMahasiswa(m1, m2, m3, m4, m5, m6);
15
16        // Input data mata kuliah
17        MataKuliah mt11 = new MataKuliah("00001", "Internet " + "of Things", 3);
18        MataKuliah mt12 = new MataKuliah("00002", "Algoritma " + "dan Struktur Data", 2);
19        MataKuliah mt13 = new MataKuliah("00003", "Algoritma " + "dan Struktur Data", 2);
20        MataKuliah mt14 = new MataKuliah("00004", "Praktikum " + "Algoritma dan Struktur Data", 3);
21        MataKuliah mt15 = new MataKuliah("00005", "Praktikum " + "Algoritma dan Pemrograman", 3);
22        ln.inputDataMataKuliah(mt11, mt12, mt13, mt14, mt15);
23
24        Scanner sc = new Scanner(System.in);
25        System.out.println("=====");
26        System.out.println("SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER");
27        System.out.println("=====");
28        boolean isTrue = true;
29        while (isTrue){
30
31            System.out.println("\n1. Input Nilai");
32            System.out.println("2. Tampil Nilai");
33            System.out.println("3. Mencari Nilai Mahasiswa");
34            System.out.println("4. Urut Data Nilai");
35            System.out.println("5. Keluar");
36            System.out.println("=====");
37            System.out.print("Pilih : ");
38            int choice = sc.nextInt();
39            switch (choice) {
```

```
1     case 1:
2         // menu input nilai
3         System.out.println("Masukkan data");
4         System.out.print("Kode : ");
5         String kd = sc.next();
6         System.out.print("Nilai : ");
7         double inputNilai = sc.nextDouble();
8
9         System.out.println("DAFTAR MAHASISWA");
10        System.out.println("=====");
11        System.out.printf("%-10s %-15s %s\n", "NIM", "Nama", "Telp");
12        ln.tampilDataMahasiswa();
13        System.out.println("Pilih mahasiswa by nim: ");
14        String tempNim = sc.next();
15
16        System.out.println("DAFTAR MATA KULIAH");
17        System.out.println("=====");
18        System.out.printf("%-10s %-40s %s\n", "Kode", "Mata Kuliah", "SKS");
19
20        ln.tampilDataMataKuliah();
21
22        System.out.print("Pilih MK by kode: ");
23        String tempKd = sc.next();
24        Nilai temp = new Nilai(kd, inputNilai,
25                                tempKd, tempNim);
26        ln.inputNilaiMahasiswa(temp);
27        break;
28        case 2:
29            // menu menampilkan nilai
30            System.out.println("DAFTAR NILAI MAHASISWA");
31            System.out.println("=====");
32            System.out.printf("%-10s %-15s %-40s %-10s %s\n", "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
33            ln.tampilListNilai();
34            break;
```

```

35
36     case 3:
37         // menu untuk mencari nilai mahasiswa
38         System.out.println("DAFTAR NILAI MAHASISWA");
39         System.out.println("*****");
40         System.out.printf("%-10s %-15s %-40s %-10s %s\n", "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
41         ln.tampilListNilai();
42         System.out.print("Masukkan data mahasiswa[nim]: ");
43         String tempNIM = sc.next();
44         int indexSearch = ln.linearSearchNilaiMahasiswa(tempNIM);
45         int indexMatkul = 0;
46         for(int i = 0; i < ln.matkul.size(); i++){
47             if(ln.listNilai.get(indexSearch).
48                 kdMatkul.equals(ln.matkul.get(i).kode)){
49                 indexMatkul = i;
50                 break;
51             }
52         }
53         System.out.printf("%-10s %-15s %-40s %-10s %s\n", "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
54         System.out.printf("%-10s %-15s %-40s %-10d %.2f\n",
55             ln.mhs.get(indexSearch).nim,
56             ln.mhs.get(indexSearch).nama,
57             ln.matkul.get(indexMatkul).namaMatkul,
58             ln.matkul.get(indexMatkul).sks,
59             ln.listNilai.get(indexSearch).nilai);
60         System.out.println("Total SKS "+ ln.matkul.get(indexMatkul).sks+" telah diambil");
61         break;
62
63     case 4:
64         // menu untuk menyorting nilai mahasiswa rekil->besar
65         System.out.println("DAFTAR NILAI MAHASISWA");
66         System.out.println("*****");
67         System.out.printf("%-10s %-15s %-40s %-10s %s\n", "NIM", "Nama", "Mata Kuliah", "SKS", "Nilai");
68         ln.sortingData();
69         ln.tampilListNilai();
70         break;
71
72     case 5:
73         // menu untuk keluar
74         isTrue = false;
75         break;
76     default:
77         System.out.println("Inputan salah");
78     }
79 }
80 }
81 }

```

## Hasil Run

```

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

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih : 1
Masukkan data
Kode : 20001
Nilai : 90.00
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

```

```

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
Thalhah  Internet of Things 90.00
Zubair   Praktikum Algoritma dan Pemrograman 90.00
Masukkan data mahasiswa[nim]: 20001
NIM      Nama      Mata Kuliah      SKS      Nilai
20001    Thalhah    Internet of Things 3        90.00
Total SKS 3 telah diambil

1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
*****
Pilih : 1
Masukkan data
Kode : 0003
Nilai : 85
DAFTAR MAHASISWA
*****

```

```

Pilih : 2
DAFTAR NILAI MAHASISWA
*****
NIM      Nama      Mata Kuliah
Thalhah  Internet of Things 90.00
Zubair   Praktikum Algoritma dan Pemrograman 90.00
Abdur-Rahman Algoritma dan Struktur Data 85.00

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
Abdur-Rahman Algoritma dan Struktur Data 85.00
Thalhah  Internet of Things 90.00
Zubair   Praktikum Algoritma dan Pemrograman 90.00

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

```

○ PS C:\Users\ASUS\Documents\SEMESTER 2\Alogaritma&Stru

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

Menambahkan code di kelas listnilai

```
87 //menghapus data pertama sesuai dengan fifo
88 public void removeList (){
89     listNilai.remove(index:0);
90 }
91 }
```

Clas main ditambhkan satu case

```
12 case 5:
13     // menu untuk menghapus data pertama nilai mahasiswa
14     System.out.println(x:"Data berhasil dihapus");
15     ln.removeList();
16     break;
17
18 case 6:
19     // menu untuk keluar
20     isTrue = false;
21     break;
22 default:
23     System.out.println(x:"Inputan salah");
24 }
```

Hasil run

```
Pilih MK by kode: 00005

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
Thalhah  Internet of Things 90.00
Zubair   Praktikum Algoritma dan Pemrograman 80.75

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