

# JOBSHEET 16

## Collection

Nama: Rizqi Bagus Andrean

Absen: 25

Kelas: TI-1D

### 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 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

```
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));
```

```

5
6 class ContohList{
7     Run | Debug
8     public static void main(String[] args) {
9         List l = new ArrayList(); List is a raw type. References to generic
10        l.add(e:1); Type safety: The method add(Object) belongs to the
11        l.add(e:2); Type safety: The method add(Object) belongs to the
12        l.add(e:3); Type safety: The method add(Object) belongs to the
13        l.add(e:"Cireng"); Type safety: The method add(Object) belongs
14
15        System.out.println("isi List l : "+l.toString());
16        l.add(e:4); Type safety: The method add(Object) belongs to the
17        l.remove(index:0);
18        System.out.println("isi List l : "+l.toString());
19    }
20
21 }

```

PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL COMMENTS Run: ContohList + -

```

PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16> & 'C:\Program
lu-11\bin\java.exe' '-cp' 'C:\Users\Acer\AppData\Roaming\Code\User\workspaceStorage\38f
89fc6b8a26b7ae90\redhat.java\jdt_ws\jobsheet 16_887a23f3\bin' 'percobaan & 'C:\Program
lu-11\bin\java.exe' '-cp' 'C:\Users\Acer\AppData\Roaming\Code\User\workspaceStorage\38f
89fc6b8a26b7ae90\redhat.java\jdt_ws\jobsheet 16_887a23f3\bin' 'percobaan1.ContohList'
isi List l : [1, 2, 3, Cireng]
isi List l : [2, 3, Cireng, 4]
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16>

```

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());

```

```

lu-11\bin\java.exe' '-cp' 'C:\Users\Acer\AppData\Roaming\Code\User\workspaceStor
89fc6b8a26b7ae90\redhat.java\jdt_ws\jobsheet 16_887a23f3\bin' 'percobaan1.Contoh
isi List l : [1, 2, 3, Cireng]
isi List l : [2, 3, Cireng, 4]
isi List names : [Noureen, Akhleema, Shannum, Uwais, A;-Qarni]
Elemen 0: mu-Kids, total element: 5, element terakhir: A;-Qarni
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16>

```

### 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 di array list tidak diberi tipe data, jadi tipe datanya dynamic

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

```
public static void main(String[] args) {
    List<Integer> l = new ArrayList<Integer>();
    l.add(e:1);
    l.add(e:2);
    l.add(e:3);
}
```

3. Ubah kode pada baris kode 38 menjadi seperti ini

```
LinkedList<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());
```

```
LinkedList<String> names = new LinkedList<>();
names.add(e:"Nouren");
names.add(e:"Akhleena");
names.add(e:"Shannum");
names.add(e:"Uwais");
names.add(e:"A;-Qarni");
names.push(e:"mei mei");

System.out.println("isi List names : "+names.toString());
names.set(index:0, element:"mu-Kids");
System.out.printf(format:"Elemen 0: %s, total element: %s, ele

}

OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS  Run: ContohList + v

rs\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16> & 'C:\Progr
java.exe' '-cp' 'C:\Users\Acer\AppData\Roaming\Code\User\workspaceStorage\
5b7ae90\redhat.java\jdt_ws\jobsheet_16_887a23f3\bin' 'percobaan1.ContohList
l : [1, 2, 3]
l : [2, 3, 4]
names : [mei mei, Nouren, Akhleena, Shannum, Uwais, A;-Qarni]
mu-Kids, total element: 6, element terakhir: A;-Qarni
```

5. Dari penambahan kode tersebut, silakan dijalankan dan apakah yang dapat Anda jelaskan! Mengubah tipe data names menjadi linked list, dan berakhir ada method push

## 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.

```
25 Stack<String> fruits = new Stack<>();
26 fruits.push("Banana");
27 fruits.add("Orange");
28 fruits.add("Watermelon");
29 fruits.add("Leci");
30 fruits.push("Salak");
31
32 for (String fruit : fruits) {
33     System.out.printf("%s ", fruit);
34 }
35
36 System.out.println("\n" + fruits.toString());
37
38 while (!fruits.empty()) {
39     System.out.printf("%s ", fruits.pop());
40 }
```

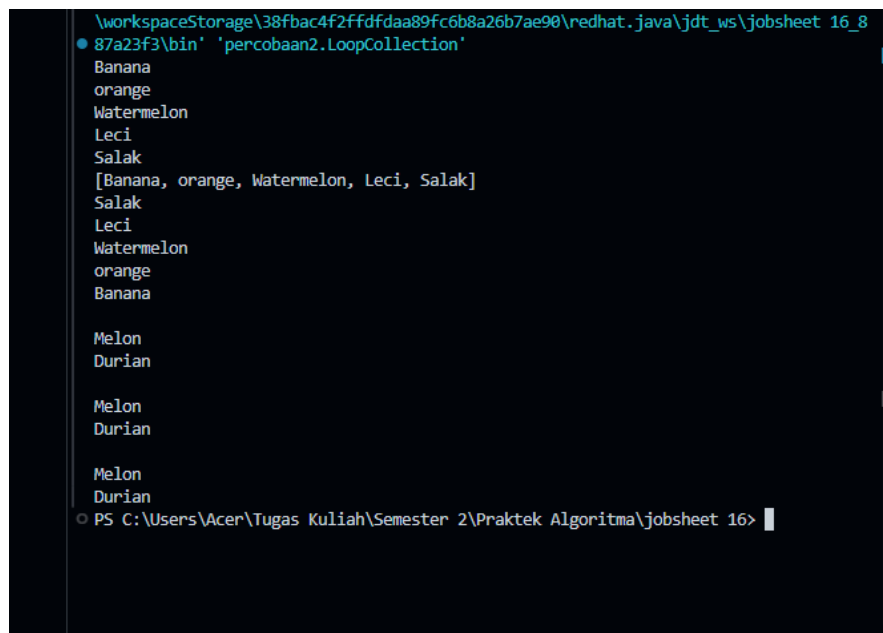
2. Tambahkan potongan kode berikut ini dari yang sebelumnya agar proses menampilkan elemen pada sebuah stack bervariasi.

```
43      fruits.push("Melon");
44      fruits.push("Durian");
45      System.out.println("");
46      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  }
```

### 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)
```



```
\workspaceStorage\38fbac4f2ffdfdaa89fc6b8a26b7ae90\redhat.java\jdt_ws\jobsheet_16_8
● 87a23f3\bin\ 'percobaan2.LoopCollection'
Banana
orange
Watermelon
Leci
Salak
[Banana, orange, Watermelon, Leci, Salak]
Salak
Leci
Watermelon
orange
Banana

Melon
Durian

Melon
Durian

Melon
Durian
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16> |
```

### 16.3.3. Pertanyaan Percobaan

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

**push():** Metode khusus dari kelas `Stack` untuk menambahkan elemen ke atas stack. Ini digunakan secara khusus untuk operasi pada stack.

**add():** Metode umum dari antarmuka `Collection` yang digunakan untuk menambahkan elemen ke

- koleksi. Ini tidak spesifik untuk stack dan dapat digunakan untuk berbagai jenis koleksi.
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian? Element melon dan durian hilang dan untuk looping dibawah method tersebut menampilkan list kosong, karena semua elemen telah dihapus sebelumnya
  3. Jelaskan fungsi dari baris 46-49?  
Sama aja kaya looping, tapi dia memeriksa apakah element setelah elemen saat ini ada, jika iya maka akan ditampilkan, jika tidak maka looping akan berhenti.
  4. Silakan ganti baris kode 25, *Stack<String>* menjadi *List<String>* dan apakah yang terjadi? Mengapa bisa demikian?  
Error karena type stack tidak sama dengan List
  5. Ganti elemen terakhir dari objek fruits menjadi "Strawberry"!
  6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!

```

    System.out.println(fruit);
});

// Ganti element terakhir dengan "Strawberry"
System.out.println("Asli: " + fruits.toString());
fruits.set(fruits.size()-1, element:"Strawberry");
System.out.println("Setelah ganti: "+fruits.toString());

fruits.add(e:"Mango");
fruits.add(e:"Guava");
fruits.add(e:"Avocado");

// Lakukan Sorting
fruits.sort((fruit1, fruit2) -> fruit1.compareTo(fruit2));
System.out.println("Setelah sorting: "+fruits.toString());
}

Asli: [Melon, Durian]
Setelah ganti: [Melon, Strawberry]
Setelah sorting: [Avocado, Guava, Mango, Melon, Strawberry]
PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16>

```

## 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, konstruktor, 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> mahasiswa = 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) {
    mahasiswa.addAll(Arrays.asList(mahasiswa));
}

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

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

public void tampil() {
    mahasiswa.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

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

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

```
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();
```

```
\workspaceStorage\38fbac4f2ffdfdaa89fc6b8a26b7ae90\redhat.java\jdt_ws\jobsheet 16_8
87a23f3\bin' 'percobaan3.ListMahasiswa'
Mahasiswa{nim=201234, nama=Nouren, notelp=08123456789}
Mahasiswa{nim=201235, nama=Akhleena, notelp=08123456789}
Mahasiswa{nim=201236, nama=Shannum, notelp=08123456789}

Mahasiswa{nim=201234, nama=Nouren, notelp=08123456789}
Mahasiswa{nim=201235, nama=Akhleena Lela, notelp=08123456789}
Mahasiswa{nim=201236, nama=Shannum, notelp=08123456789}
○ PS C:\Users\Acer\Tugas Kuliah\Semester 2\Praktek Algoritma\jobsheet 16>
```

#### 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)
```

#### 16.4.3. Pertanyaan Percobaan

1. Pada fungsi tambah() yang menggunakan unlimited argument itu menggunakan konsep apa? Dan kelebihan apa?  
menggunakan konsep varargs dalam Java, memungkinkan penggunaan jumlah argumen yang tidak terbatas. Kelebihannya adalah memudahkan penambahan beberapa elemen sekaligus ke dalam koleksi.
2. Pada fungsi linearSearch() di atas, silakan diganti dengan fungsi binarySearch() dari collection!



```

    int binarySearch(String nim) {
        // Urutkan dulu array list
        Collections.sort(mahasiswas, (mhs1, mhs2) -> mhs1.nim.compareTo(mhs2.nim));

        // Lakukan binary search
        int index = Collections.binarySearch(mahasiswas, new Mahasiswa(nim, nama:"", notelp:"")
        (mhs1, mhs2) -> mhs1.nim.compareTo(mhs2.nim));

        return index >= 0 ? index : -1;
    }

```

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

```

17     void sortAscending() {
18         Collections.sort(mahasiswas, Comparator.comparing(mhs -> mhs.nim));
19     }
20
21     void sortDescending() {
22         Collections.sort(mahasiswas, (mhs1, mhs2) ->
23         mhs2.nim.compareTo(mhs1.nim));
24     }

```

## 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!

--- \*\*\* ---

Matakuliah.java

```
package tugas;  
  
public class Matakuliah {  
    String nama, kode;  
    double nilai;  
    int sks;  
  
    public Matakuliah(String nama, String kode, double nilai, int sks){  
        this.nama = nama;  
        this.kode = kode;  
        this.nilai = nilai;  
        this.sks = sks;  
    }  
  
    public Matakuliah(String nama, String kode, int sks){  
        this.nama = nama;  
        this.kode = kode;  
        this.sks = sks;  
    }  
}
```

Mahasiswa.java

```
package tugas;  
  
import java.util.ArrayList;
```

```

public class Mahasiswa {
    String nim, nama, notelp;
    ArrayList<MataKuliah> mataKuliahs = new ArrayList<>();

    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 +
    '}'';
    }
}

```

ListMahasiwa.java

```

package tugas;

import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.Queue;
import java.util.Scanner;

public class ListMahasiswa {
    ArrayList<Mahasiswa> mahasiswas = new ArrayList<>();
    ArrayList<MataKuliah> mataKuliahs = new ArrayList<>();
    private static Scanner scanner = new Scanner(System.in);
    Queue<Integer> queue = new LinkedList<>();

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

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

    // Prosedur hapus menggunakan Queue
    void hapusMahasiswaDenganQueue() {
        System.out.print("Masukkan NIM mahasiswa yang ingin dihapus: ");
        String nim = scanner.nextLine();
    }
}

```

```

        int index = linearSearch(nim);
        if (index == -1) {
            System.out.println("Mahasiswa tidak ditemukan");
            return;
        }

        queue.add(index);
        hapus(queue.poll());
        System.out.println("Mahasiswa dengan NIM " + nim + " telah dihapus.");
    }

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

    void tampil() {
        mahasiswas.stream().forEach((mhs) -> {
            System.out.println(mhs);
        });
    }

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

    int binarySearch(String nim) {
        // Urutkan dulu array list
        Collections.sort(mahasiswas, (mhs1, mhs2) -> mhs1.nim.compareTo(mhs2.nim));

        // Lakukan binary search
        int index = Collections.binarySearch(mahasiswas, new Mahasiswa(nim, "", ""),
            (mhs1, mhs2) ->
mhs1.nim.compareTo(mhs2.nim));

        return index >= 0 ? index : -1;
    }

    MataKuliah searchByKode(String kode) {
        for (MataKuliah mk : mataKuliahs) {
            if (mk.kode.equals(kode)) {
                return mk;
            }
        }
    }
}

```

```

        return null;
    }

    void sortAscending() {
        Collections.sort(mahasiswas, Comparator.comparing(mhs -> mhs.nim));
    }

    void sortDescending() {
        Collections.sort(mahasiswas, (mhs1, mhs2) -> mhs2.nim.compareTo(mhs1.nim));
    }

    void menu() {
        System.out.println("Menu: ");
        System.out.println("1. Input Nilai");
        System.out.println("2. Tampil Nilai");
        System.out.println("3. Mencari Nilai Mahasiswa");
        System.out.println("4. Urut Data Nilai");
        System.out.println("5. Hapus Mahasiswa");
        System.out.println("6. Exit");
    }

    void inputNilai() {
        System.out.print("Masukkan NIM: ");
        String nim = scanner.nextLine();

        int index = linearSearch(nim);
        if (index == -1) {
            System.out.println("Mahasiswa tidak ditemukan");
            return;
        }

        Mahasiswa m = mahasiswas.get(index);
        System.out.print("Masukkan Kode Mk: ");
        String kode = scanner.nextLine();
        System.out.print("Masukkan Nilai: ");
        double nilai = scanner.nextDouble();
        scanner.nextLine();

        // Cari mata kuliah
        MataKuliah mk = searchByKode(kode);
        if (mk == null) {
            System.out.println("Mata kuliah tidak ditemukan");
            return;
        }

        mk.nilai = nilai;
        m.mataKuliahs.add(mk);
        System.out.println("Nilai berhasil dimasukkan ke matakuliah: " + mk.nama);
    }
}

```

```

void tampilSemuaNilaiMahasiswa() {
    mahasiswa.stream().forEach((mhs) -> {
        System.out.println("NIM: " + mhs.nim + ", Nama: " + mhs.nama);
        mhs.mataKuliahs.stream().forEach((mk) -> {
            System.out.println("- Mata Kuliah: " + mk.nama + ", Nilai: " + mk.nilai);
        });
        System.out.println();
    });
}

void cariNilaiByNim() {
    System.out.print("Masukkan nim: ");
    String nim = scanner.nextLine();

    int index = linearSearch(nim);
    if (index == -1) {
        System.out.println("Mahasiswa tidak ditemukan");
        return;
    }

    Mahasiswa m = mahasiswa.get(index);
    System.out.println("NIM: " + m.nim + ", Nama: " + m.nama);
    m.mataKuliahs.stream().forEach((mk) -> {
        System.out.println("- Mata Kuliah: " + mk.nama + ", Nilai: " + mk.nilai);
    });
    System.out.println();
}

void urutNilaiByNilaiTerbesar() {
    mahasiswa.stream().forEach((mhs) -> {
        mhs.mataKuliahs.sort((mk1, mk2) -> Double.compare(mk2.nilai, mk1.nilai));
    });
}

public static void main(String[] args) {
    ListMahasiswa lm = new ListMahasiswa();

    // isi mahasiswa
    Mahasiswa m = new Mahasiswa("201234", "Nouren", "08123456789");
    Mahasiswa m1 = new Mahasiswa("201235", "Akhleena", "08123456789");
    Mahasiswa m2 = new Mahasiswa("201236", "Shannum", "08123456789");
    lm.tambah(m, m1, m2);
    lm.tambah(
        new Mahasiswa("201237", "Uwais", "08123456789"),
        new Mahasiswa("201238", "Al-Qarni", "08123456789"),
        new Mahasiswa("201239", "mei mei", "08123456789"));

    // isi list mk

```

```

lm.mataKuliahs.add(new MataKuliah("Internet Of Things", "00001", 3));
lm.mataKuliahs.add(new MataKuliah("Algoritma Struktur data", "00002", 2));
lm.mataKuliahs.add(new MataKuliah("Algoritma dan Pemrograman", "00003", 2));
lm.mataKuliahs.add(new MataKuliah("Praktikum Algoritma dan Struktur Data",
"00004", 3));
lm.mataKuliahs.add(new MataKuliah("Praktikum Algoritma dan Pemrograman", "00005",
3));

boolean isExit = false;

while (!isExit) {
    lm.menu();

    System.out.print("Pilih menu: ");
    int menu = scanner.nextInt();
    scanner.nextLine();

    switch (menu) {
        case 1:
            lm.inputNilai();
            break;
        case 2:
            lm.tampilSemuaNilaiMahasiswa();
            break;
        case 3:
            lm.cariNilaiByNim();
            break;
        case 4:
            lm.urutNilaiByNilaiTerbesar();
            break;
        case 5:
            lm.hapusMahasiswaDenganQueue();
            break;
        case 6:
            isExit = true;
            break;
        default:
            System.out.println("Pilihan tidak valid");
            break;
    }
}

scanner.close();
}
}

```

Tambah Nilai



```

nat: java\jdk_ws\jobsheet_16_887a23f3\bin\tugas.ListMahasiswa
oll... Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 1
Masukkan NIM: 201237
Masukkan Kode Mk: 00001
Masukkan Nilai: 90
Nilai berhasil dimasukkan ke matakuliah: Internet Of Things
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 1
Masukkan NIM: 201237
Masukkan Kode Mk: 00003
Masukkan Nilai: 75
Nilai berhasil dimasukkan ke matakuliah: Algoritma dan Pemrograman
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 1
Masukkan NIM: 201238
Masukkan Kode Mk: 00005
Masukkan Nilai: 98
Nilai berhasil dimasukkan ke matakuliah: Praktikum Algoritma dan Pemrograman

```

### Tampil Nilai

```

Nilai berhasil dimasukkan ke matakuliah: Praktikum Algoritma dan Pemrograman
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 2
NIM: 201234, Nama: Nouren

NIM: 201235, Nama: Akhleena

NIM: 201236, Nama: Shannum

NIM: 201237, Nama: Uwais
- Mata Kuliah: Internet Of Things, Nilai: 90.0
- Mata Kuliah: Algoritma dan Pemrograman, Nilai: 75.0

NIM: 201238, Nama: Al-Qarni
- Mata Kuliah: Praktikum Algoritma dan Pemrograman, Nilai: 98.0

NIM: 201239, Nama: mei mei

```

Tampil nilai by nim

```
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 3
Masukkan nim: 201238
NIM: 201238, Nama: Al-Qarni
- Mata Kuliah: Praktikum Algoritma dan Pemrograman, Nilai: 98.0
```

Hapus Mahasiswa

```
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 5
Masukkan NIM mahasiswa yang ingin dihapus: 201238
Mahasiswa dengan NIM 201238 telah dihapus.
Menu:
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Hapus Mahasiswa
6. Exit
Pilih menu: 2
NIM: 201234, Nama: Nouran

NIM: 201235, Nama: Akhleena

NIM: 201236, Nama: Shannum

NIM: 201237, Nama: Uwais
- Mata Kuliah: Internet Of Things, Nilai: 90.0
- Mata Kuliah: Algoritma dan Pemrograman, Nilai: 75.0

NIM: 201239, Nama: mei mei

Menu:
```