

# LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022

## Daftar Isi

<b>PERCOBAAN 1</b>	<b>4</b>
LANGKAH 1	4
LANGKAH 2	4
VERIFIKASI HASIL PERCOBAAN	5
<i>Pertanyaan Percobaan</i>	5
<i>Jawaban Percobaan</i>	5
<b>PERCOBAAN 2</b>	<b>7</b>
LANGKAH 1	7
LANGKAH 2	7
VERIFIKASI HASIL PERCOBAAN	8
<i>Pertanyaan percobaan</i>	8
<i>Jawaban percobaan</i>	8
<b>PERCOBAAN 3</b>	<b>11</b>
LANGKAH 1	11
LANGKAH 2	11
LANGKAH 3	11
LANGKAH 4	12
LANGKAH 5	12
VERIFIKASI HASIL PERCOBAAN	13
<i>Pertanyaan Percobaan</i>	13
<i>Jawaban Percobaan</i>	13
<b>TUGAS PRAKTIKUM</b>	<b>16</b>
NOMOR 1	16
JAWABAN NOMOR 1	17
<i>Source code Class Film_26</i>	17
<i>Source code Class MainTugas1_26</i>	18
<i>Output Menu Awal dan Tambah Data</i>	19
<i>Output Cetak Semua Data Film</i>	19
<i>Output Judul Film di Tumpukan Teratas</i>	19
<i>Output Hapus Film Teratas</i>	20
NOMOR 2	21
JAWABAN NOMOR 2	22



## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

<i>Source code Class Mahasiswa_26</i> .....	22
<i>Source code Class Nilai_26</i> .....	23
<i>Source code Class MataKuliah_26</i> .....	23
<i>Source code Class MainTugas2_26</i> .....	23
<i>Output Menu Awal dan Penambahan Data</i> .....	25
<i>Output Tampil Nilai</i> .....	26
<i>Output Pencarian Data Mahasiswa</i> .....	26
<i>Output Pengurutan Data Nilai</i> .....	26

## Percobaan 1

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

### Langkah 1

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

```
1  import java.util.ArrayList;
2  import java.util.List;
3
4  public class ContohList {
5      public static void main(String[] args) {
6          List l = new ArrayList();
7          l.add(1);
8          l.add(2);
9          l.add(3);
10         l.add("Cireng");
11         System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
12             l.get(0), l.size(), l.get(l.size() - 1));
13         l.add(4);
14         l.remove(index: 0);
15         System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %s\n",
16             l.get(0), l.size(), l.get(l.size() - 1));
17     }
18 }
```

### Langkah 2

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

```
19  List<String> names = new LinkedList<>();
20  names.add("Noureen");
21  names.add("Akhleema");
22  names.add("Shannum");
23  names.add("Uwais");
24  names.add("Al-Qarni");
25
26  System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
27      names.get(0), names.size(), names.get(names.size() - 1));
28  names.set(0, "My kid");
29  System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
30      names.get(0), names.size(), names.get(names.size() - 1));
31  System.out.println("Names: " + names.toString());
```

## Verifikasi Hasil Percobaan

```
Run: ContohList x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-ja
Elemen 0: 1 total elemen: 4 elemen terakhir: Cireng
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Nouredin 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]
```

## Pertanyaan Percobaan

1. Perhatikan baris kode 25-36, mengapa semua jenis data bisa ditampung ke dalam sebuah ArrayList?
2. Modifikasi baris kode 25-36 sehingga data yang ditampung hanya satu jenis atau spesifik tipe tertentu!
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());
```

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

## Jawaban Percobaan

1. Karena array list menggunakan type yang tidak disebutkan secara spesifik maka dapat di isi oleh semua tipe data.
2. Source code

```
8 // MODIFIKASI SOAL NOMOR 2
9
10 List l = new ArrayList();
11 List<Integer> l = new ArrayList<>();
12
13 l.add(1);
14 l.add(2);
15 l.add(3);
16 // l.add("Cireng");
17 l.add(4);
18 System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %d\n", // ubah ke %d
19     l.get(0), l.size(), l.get(l.size() - 1));
20 l.add(4);
21 l.remove(index: 0);
22 System.out.printf("Elemen 0: %d total elemen: %d elemen terakhir: %d\n", // ubah ke %d
23     l.get(0), l.size(), l.get(l.size() - 1));
```

Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-jav
Elemen 0: 1 total elemen: 4 elemen terakhir: 4
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Noreen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]

Process finished with exit code 0

```

### 3. Source code modifikasi

```

25 // MODIFIKASI NOMOR 3
26
27 // List<String> names = new LinkedList<>();
28 LinkedList<String> names = new LinkedList<>();

```

### 4. Source code

```

42 // MODIFIKASI NOMOR 4
43
44 names.push(e: "Mei-mei");
45 System.out.printf("Elemen 0: %s total elemen: %s elemen terakhir: %s\n",
46                 names.getFirst(), names.size(), names.getLast());
47 System.out.println("Names: " + names.toString());

```

### 5. Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-jav
Elemen 0: 1 total elemen: 4 elemen terakhir: 4
Elemen 0: 2 total elemen: 4 elemen terakhir: 4
Elemen 0: Noreen total elemen: 5 elemen terakhir: Al-Qarni
Elemen 0: My kid total elemen: 5 elemen terakhir: Al-Qarni
Names: [My kid, Akhleema, Shannum, Uwais, Al-Qarni]
Elemen 0: Mei-mei total elemen: 6 elemen terakhir: Al-Qarni
Names: [Mei-mei, My kid, Akhleema, Shannum, Uwais, Al-Qarni]

Process finished with exit code 0

```

Pada source code tersebut melakukan penambahan data pada depan list dengan method push, kemudian menampilkan elemen pertama pada list menggunakan method `getFirst`, menampilkan jumlah elemen menggunakan method `size`, kemudian menampilkan elemen terakhir pada list menggunakan method `getLast`.

## Percobaan 2

Pada percobaan 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

### Langkah 1

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

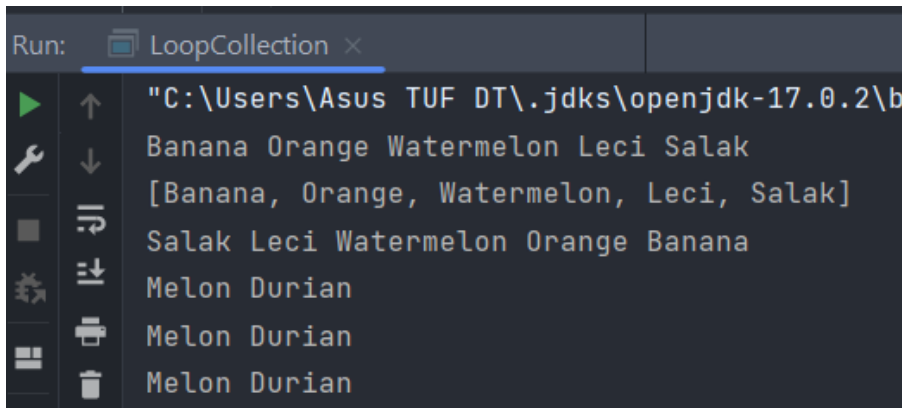
```
1  import java.util.Stack;
2
3  public class LoopCollection {
4      public static void main(String[] args) {
5          Stack<String> fruits = new Stack<>();
6          fruits.push(item: "Banana");
7          fruits.add("Orange");
8          fruits.add("Watermelon");
9          fruits.add("Leci");
10         fruits.add("Salak");
11
12         for (String fruit : fruits) {
13             System.out.printf("%s ", fruit);
14         }
15         System.out.println("\n" + fruits.toString());
16
17         while (!fruits.empty()) {
18             System.out.printf("%s ", fruits.pop());
19         }
20     }
21 }
```

### Langkah 2

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

```
22     fruits.push( item: "MeLon");
23     fruits.push( item: "Durian");
24     System.out.println("");
25     for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
26         String fruit = it.next();
27         System.out.printf("%s ", fruit);
28     }
29     System.out.println("");
30     fruits.stream().forEach(e -> {
31         System.out.printf("%s ", e);
32     });
33     System.out.println("");
34     for (int i = 0; i < fruits.size(); i++) {
35         System.out.printf("%s ", fruits.get(i));
36     }
```

### Verifikasi Hasil Percobaan



```
Run: LoopCollection x
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Melon Durian
Melon Durian
Melon Durian
```

### Pertanyaan percobaan

1. Apakah perbedaan fungsi push() dan add() pada objek fruits?
2. Silakan hilangkan baris 43 dan 44, apakah yang akan terjadi? Mengapa bisa demikian?
3. Jelaskan fungsi dari baris 46-49?
4. Silakan ganti baris kode 25, Stack menjadi List dan apakah yang terjadi? Mengapa bisa demikian?
5. Ganti elemen terakhir dari objek fruits menjadi "Strawberry"!
6. Tambahkan 3 buah seperti "Mango", "guava", dan "avocado" kemudian dilakukan sorting!

### Jawaban percobaan

1. Push berfungsi untuk memasukkan data ke top of stack, sedangkan add merupakan method inherited dari class java.util.Vector yang berfungsi untuk memasukkan data ke akhir Vector atau disini stack.
2. Source code



```

22 // MODIFIKASI NOMOR 2
23
24 //      fruits.push("Melon");
25 //      fruits.push("Durian");
26 System.out.println("");
27 for (Iterator<String> it = fruits.iterator(); it.hasNext();) {
28     String fruit = it.next();
29     System.out.printf("%s ", fruit);
30 }
31 System.out.println("");
32 fruits.stream().forEach(e -> {
33     System.out.printf("%s ", e);
34 });
35 System.out.println("");
36 for (int i = 0; i < fruits.size(); i++) {
37     System.out.printf("%s ", fruits.get(i));
38 }

```

Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana

```

Data melon dan durian tidak ditampilkan karena data tersebut tidak pernah dimasukkan.

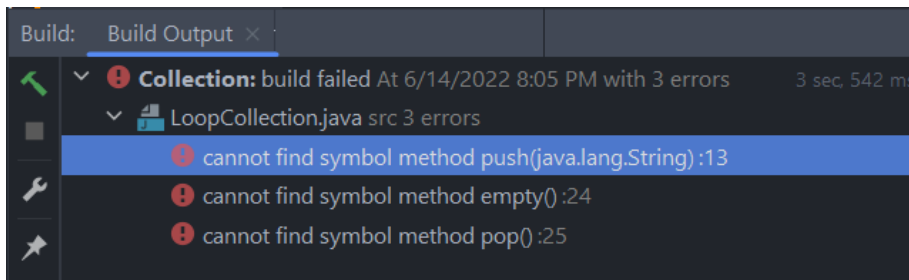
- Baris kode tersebut digunakan untuk melakukan perulangan menggunakan Interface iterator dan menggunakan hasNext yang akan mengembalikan nilai true jika iteration masih mempunyai elemen.
- Source code

```

9 // MODIFIKASI NOMOR 4
10
11 //      Stack<String> fruits = new Stack<>();
12 List<String> fruits = new Stack<>();
13 fruits.push("Banana");
14 fruits.add("Orange");
15 fruits.add("Watermelon");
16 fruits.add("Leci");
17 fruits.add("Salak");

```

Output



Hasilnya adalah error karena pada interface List tidak memiliki method push, empty dan pop.

##### 5. Source code

```

33 // MODIFIKASI NOMOR 5
34
35 fruits.push( item: "MeLon");
36 fruits.push( item: "Durian");
37 fruits.push( item: "Strawberry");

```

##### Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Melon Durian Strawberry
Melon Durian Strawberry
Melon Durian Strawberry
Process finished with exit code 0

```

##### 6. Source code

```

36 // MODIFIKASI NOMOR 6
37 fruits.push( item: "Mango");
38 fruits.push( item: "Guava");
39 fruits.push( item: "Avocado");
40 fruits.sort(String::compareTo);

```

##### Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
Banana Orange Watermelon Leci Salak
[Banana, Orange, Watermelon, Leci, Salak]
Salak Leci Watermelon Orange Banana
Avocado Durian Guava Mango Melon Strawberry
Avocado Durian Guava Mango Melon Strawberry
Avocado Durian Guava Mango Melon Strawberry
Process finished with exit code 0

```

Data sudah diurutkan secara ascending.

## Percobaan 3

Pada percobaan 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.

### Langkah 1

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

```
1 public class Mahasiswa {
2     String nim;
3     String nama;
4     String notelp;
5
6     public Mahasiswa() {
7
8     }
9
10    public Mahasiswa(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 }
```

### Langkah 2

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

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 public class ListMahasiswa {
5     List<Mahasiswa> mahasiswas = new ArrayList<>();
6 }
```

### Langkah 3

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

```
8      public void tambah(Mahasiswa... mahasiswa) {
9          mahasiswa.addAll(Arrays.asList(mahasiswa));
10     }
11
12     public void hapus(int index) {
13         mahasiswa.remove(index);
14     }
15
16     public void update(int index, Mahasiswa mhs) {
17         mahasiswa.set(index, mhs);
18     }
19
20     public void tampil() {
21         mahasiswa.stream().forEach(mhs -> {
22             System.out.println(" " + mhs.toString());
23         });
24     }
```

#### Langkah 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 < mahasiswa.size(); i++) {
28             if (nim.equals(mahasiswa.get(i).nim)) {
29                 return i;
30             }
31         }
32         return -1;
33     }
```

#### Langkah 5

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

```
35 public static void main(String[] args) {  
36     ListMahasiswa lm = new ListMahasiswa();  
37     Mahasiswa m = new Mahasiswa( nim: "201234", nama: "Noureen", notelp: "021xx1");  
38     Mahasiswa m1 = new Mahasiswa( nim: "201235", nama: "Akhleema", notelp: "021xx2");  
39     Mahasiswa m2 = new Mahasiswa( nim: "201236", nama: "Shannum", notelp: "021xx3");  
40     // menambahkan objek mahasiswa  
41     lm.tambah(m, m1, m2);  
42     // menampilkan list mahasiswa  
43     lm.tampil();  
44     // update mahasiswa  
45     lm.update(lm.linearSearch( nim: "201235"),  
46         new Mahasiswa( nim: "201235", nama: "Akhleema Lela", notelp: "021xx2"));  
47     System.out.println();  
48     lm.tampil();  
49 }
```

### Verifikasi Hasil Percobaan

```
Run: ListMahasiswa x  
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"  
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}
```

### Pertanyaan Percobaan

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

### Jawaban Percobaan

1. Menggunakan konsep varargs, kelebihan dari fitur ini merupakan dapat menerima banyak argumen maka menjadi mudah untuk membuat metode yang perlu mengambil sejumlah variabel argumen.
2. Source code

```
44 // MODIFIKASI NOMOR 2
45 // lm.update(lm.linearSearch("201235"),
46 //     new Mahasiswa("201235", "Akhleema Lela", "021xx2"));
47 lm.update(lm.binarySearch( nim: "201235"),
48     new Mahasiswa( nim: "201235", nama: "Akhleema Lela", notelp: "021xx2"));
49 System.out.println();
50 lm.tampil();
51 }
52
53 Comparator<Mahasiswa> c = new Comparator<Mahasiswa>() {
54     @Override
55     public int compare(Mahasiswa o1, Mahasiswa o2) {
56         return o1.nim.compareTo(o2.nim);
57     }
58 };
59
60 public int binarySearch(String nim) {
61     return Collections.binarySearch(mahasiswas, new Mahasiswa(nim, nama: null, notelp: null), c);
62 }
```

Disini saya membuat comparator untuk membandingkan nim pada objek mahasiswa.

Output

```
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "
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}

Process finished with exit code 0
```

### 3. Source code

```

52      // MODIFIKASI NOMOR 3
53      System.out.println();
54      lm.sortAsc();
55      System.out.println("Sort Ascending berdasarkan nim");
56      lm.tampil();
57      System.out.println();
58      lm.sortDesc();
59      System.out.println("Sort Descencing berdasarkan nim");
60      lm.tampil();
61  }
62
63  void sortAsc() {
64      mahasiswa.sort(c);
65  }
66  void sortDesc() {
67      mahasiswa.sort(c.reversed());
68  }
69
70  Comparator<Mahasiswa> c = new Comparator<Mahasiswa>() {
71      @Override
72      public int compare(Mahasiswa o1, Mahasiswa o2) {
73          return o1.nim.compareTo(o2.nim);
74      }
75  };

```

### Output

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe" "-j
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}

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

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

Process finished with exit code 0

```

## Tugas Praktikum

### Nomor 1

Implementasikan stack menggunakan collection dengan contoh kasus tumpukan daftar film sesuai dengan fitur-fitur yang ditunjukkan pada gambar di bawah ini!

#### Ilustrasi Program

##### Menu Awal dan Tambah Data (Push)

```
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info semua Judul Film
5. Keluar
=====
Pilih:
1
ID Film      :
2333
Judul Film   :
Dongkrak Antik
Tahun Tayang :
1982
Director    :
Warkop DKI
```

##### Cetak Semua Data Film

```
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info semua Judul Film
5. Keluar
=====
Pilih:
4
Film{ID Film=1, Judul Film=Spider-Man: No Way Home, Tahun Tayang=2021, Director:Jon Watts}
Film{ID Film=2, Judul Film=Dr. Strange: The Madness of Multiverse, Tahun Tayang=2022, Director:Sam Raimi}
Film{ID Film=3, Judul Film=Skyfall, Tahun Tayang=2012, Director:Sam Mendes}
Film{ID Film=4, Judul Film=Aquaman, Tahun Tayang=2018, Director:James Wan}
Film{ID Film=5, Judul Film=Dark Knight Rises, Tahun Tayang=2012, Director:Christopher Nolan}
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI}
```

##### Cek Judul Film di tumpukan teratas

```
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info semua Judul Film
5. Keluar
=====
Pilih:
3
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI}
```

##### Hapus Film Teratas



```
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info semua Judul Film
5. Keluar
=====

Pilih:
2
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI}
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info semua Judul Film
5. Keluar
=====

Pilih:
4
Film{ID Film=1, Judul Film=Spider-Man: No Way Home, Tahun Tayang=2021, Director:Jon Watts}
Film{ID Film=2, Judul Film=Dr. Strange: The Madness of Multiverse, Tahun Tayang=2022, Director:Sam Raimi}
Film{ID Film=3, Judul Film=Skyfall, Tahun Tayang=2012, Director:Sam Mendes}
Film{ID Film=4, Judul Film=Aquaman, Tahun Tayang=2018, Director:James Wan}
Film{ID Film=5, Judul Film=Dark Knight Rises, Tahun Tayang=2012, Director:Christopher Nolan}
```

## Jawaban Nomor 1

### Source code Class Film\_26

```
Film_26.java x MainTugas1_26.java x
1 public class Film_26 {
2     int idFilm, tahun;
3     String judulFilm, director;
4
5     Film_26(int idFilm, String judulFilm, int tahun, String director) {
6         this.idFilm = idFilm;
7         this.judulFilm = judulFilm;
8         this.tahun = tahun;
9         this.director = director;
10    }
11 }
```

## Source code Class MainTugas1\_26

```

1  import java.util.Scanner;
2  import java.util.Stack;
3
4  public class MainTugas1_26 {
5      public static void main(String[] args) {
6          Scanner input = new Scanner(System.in);
7          int menu;
8          Stack<Film_26> film26Stack = new Stack<>();
9          // tambahan data untuk keperluan test
10         film26Stack.push(new Film_26(1, "Spider-Man: No Way Home", 2021, "Jon Watts"));
11         film26Stack.push(new Film_26(2, "Dr. Strange: The Madness of Multiverse", 2022, "Sam Raimi"));
12         film26Stack.push(new Film_26(3, "Skyfall", 2012, "Sam Mendes"));
13         film26Stack.push(new Film_26(4, "Aquaan", 2018, "James Wan"));
14         film26Stack.push(new Film_26(5, "Dark Knight Rises", 2012, "Christopher Nolan"));
15         do {
16             System.out.println("=====");
17             System.out.println("Data Film Layar Lebar Sepanjang Masa");
18             System.out.println("=====");
19             System.out.println();
20             System.out.println("1. Input Judul Film");
21             System.out.println("2. Hapus Data Film Teratas");
22             System.out.println("3. Cek Judul Film Teratas");
23             System.out.println("4. Info Semua Judul Film");
24             System.out.println("5. Keluar");
25             System.out.println("=====");
26             System.out.print("Pilih: ");
27             menu = input.nextInt();
28
29             switch (menu) {
30                 case 1 -> {
31                     int id_film, tahun;
32                     String judul_film, director;
33                     System.out.print("ID Film\t\t : ");
34                     id_film = input.nextInt();
35                     input.nextLine();
36                     System.out.print("Judul Film\t : ");
37                     judul_film = input.nextLine();
38                     System.out.print("Tahun Tayang : ");
39                     tahun = input.nextInt();
40                     input.nextLine();
41                     System.out.print("Director\t : ");
42                     director = input.nextLine();
43                     Film_26 newData = new Film_26(id_film, judul_film, tahun, director);
44                     film26Stack.push(newData);
45                 }
46                 case 2 -> {
47                     Film_26 pop = film26Stack.pop();
48                     System.out.printf("Film{ID Film=%d, Judul Film=%s, Tahun Tayang=%d, Director:%s\n",
49                                     pop.idFilm, pop.judulFilm, pop.tahun, pop.director);
50                 }
51                 case 3 -> {
52                     Film_26 peek = film26Stack.peek();
53                     System.out.printf("Film{ID Film=%d, Judul Film=%s, Tahun Tayang=%d, Director:%s\n",
54                                     peek.idFilm, peek.judulFilm, peek.tahun, peek.director);
55                 }
56                 case 4 -> film26Stack.forEach(film_26 -> {
57                     System.out.printf("Film{ID Film=%d, Judul Film=%s, Tahun Tayang=%d, Director:%s\n",
58                                     film_26.idFilm, film_26.judulFilm, film_26.tahun, film_26.director);
59                 });
60                 case 5 -> System.exit(status: 0);
61             }
62         } while (menu <= 5 && menu >= 1);
63     }
64 }

```

## Output Menu Awal dan Tambah Data

```

C:\Users\Asus TUF DT\.jdk\openjdk-17
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info Semua Judul Film
5. Keluar

Pilih: 1
ID Film      : 2333
Judul Film   : Dongkrak Antik
Tahun Tayang : 1982
Director     : Warkop DKI

```

## Output Cetak Semua Data Film

```

=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info Semua Judul Film
5. Keluar

Pilih: 4
Film{ID Film=1, Judul Film=Spider-Man: No Way Home, Tahun Tayang=2021, Director:Jon Watts
Film{ID Film=2, Judul Film=Dr. Strange: The Madness of Multiverse, Tahun Tayang=2022, Director:Sam Raimi
Film{ID Film=3, Judul Film=Skyfall, Tahun Tayang=2012, Director:Sam Mendes
Film{ID Film=4, Judul Film=Aquaan, Tahun Tayang=2018, Director:James Wan
Film{ID Film=5, Judul Film=Dark Knight Rises, Tahun Tayang=2012, Director:Christopher Nolan
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI

```

## Output Judul Film di Tumpukan Teratas

```

=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info Semua Judul Film
5. Keluar

Pilih: 3
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI

```

## Output Hapus Film Teratas

```
=====
Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info Semua Judul Film
5. Keluar

=====
Pilih: 2
Film{ID Film=2333, Judul Film=Dongkrak Antik, Tahun Tayang=1982, Director:Warkop DKI
=====

Data Film Layar Lebar Sepanjang Masa
=====

1. Input Judul Film
2. Hapus Data Film Teratas
3. Cek Judul Film Teratas
4. Info Semua Judul Film
5. Keluar

=====
Pilih: 4
Film{ID Film=1, Judul Film=Spider-Man: No Way Home, Tahun Tayang=2021, Director:Jon Watts
Film{ID Film=2, Judul Film=Dr. Strange: The Madness of Multiverse, Tahun Tayang=2022, Director:Sam Raimi
Film{ID Film=3, Judul Film=Skyfall, Tahun Tayang=2012, Director:Sam Mendes
Film{ID Film=4, Judul Film=Aquaan, Tahun Tayang=2018, Director:James Wan
Film{ID Film=5, Judul Film=Dark Knight Rises, Tahun Tayang=2012, Director:Christopher Nolan
```

## Nomor 2

Buatlah implementasi program daftar nilai mahasiswa semester, minimal memiliki 3 class yaitu Mahasiswa, Nilai, dan Mata Kuliah. Khusus untuk data Mahasiswa dan Mata Kuliah harus sudah Page 7 of 18 diinisialisasi, sehingga ketika memasukkan data nilai data mahasiswa dan mata kuliah cukup menginputkan nim ataupun kode mata kuliah. Yang paling penting adalah antara objek mahasiswa, matakuliah, dan nilai harus saling terhubung.

### 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

## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

```
*****
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
```

### Jawaban Nomor 2

#### Source code Class Mahasiswa\_26

```
MainTugas2_26.java x Mahasiswa_26.java x Nilai_26.java x MataKul
1 public class Mahasiswa_26 {
2     String nim, nama, telf;
3     Mahasiswa_26(String nim, String nama, String telf) {
4         this.nim = nim;
5         this.nama = nama;
6         this.telf = telf;
7     }
8 }
```

## Source code Class Nilai\_26

```
1 public class Nilai_26 {
2     int sks;
3     String nim, kode, nama, mataKuliah, nilai;
4
5     public Nilai_26(String kode, String nim, String nama, String mataKuliah, int sks, String nilai) {
6         this.kode = kode;
7         this.nim = nim;
8         this.sks = sks;
9         this.nilai = nilai;
10        this.nama = nama;
11        this.mataKuliah = mataKuliah;
12    }
13 }
```

## Source code Class MataKuliah\_26

```
1 public class MataKuliah_26 {
2     int sks;
3     String kode, nama;
4     MataKuliah_26(String kode, String nama, int sks) {
5         this.kode = kode;
6         this.nama = nama;
7         this.sks = sks;
8     }
9 }
```

## Source code Class MainTugas2\_26

```
1 import java.util.*;
2
3 public class MainTugas2_26 {
4     static List<Mahasiswa_26> mahasiswa26s = new ArrayList<>();
5     static List<MataKuliah_26> mataKuliah26s = new ArrayList<>();
6     static List<Nilai_26> nilai26s = new ArrayList<>();
7     public static void main(String[] args) {
8         // INPUT DATA MAHASISWA DAN MATA KULIAH
9         mahasiswa26s.add(new Mahasiswa_26( nim: "20001", nama: "Thalhah", telf: "021xxx"));
10        mahasiswa26s.add(new Mahasiswa_26( nim: "20002", nama: "Zubair", telf: "021xxx"));
11        mahasiswa26s.add(new Mahasiswa_26( nim: "20003", nama: "Abdur-Rahman", telf: "021xxx"));
12        mahasiswa26s.add(new Mahasiswa_26( nim: "20004", nama: "Sa'ad", telf: "021xxx"));
13        mahasiswa26s.add(new Mahasiswa_26( nim: "20005", nama: "Sa'id", telf: "021xxx"));
14        mahasiswa26s.add(new Mahasiswa_26( nim: "20006", nama: "Ubaidah", telf: "021xxx"));
15        mataKuliah26s.add(new MataKuliah_26( kode: "00001", nama: "Internet of Things", sks: 3));
16        mataKuliah26s.add(new MataKuliah_26( kode: "00002", nama: "Algoritma dan Struktur Data", sks: 2));
17        mataKuliah26s.add(new MataKuliah_26( kode: "00003", nama: "Algoritma dan Pemrograman", sks: 2));
18        mataKuliah26s.add(new MataKuliah_26( kode: "00004", nama: "Praktikum Algoritma dan Struktur Data", sks: 3));
19        mataKuliah26s.add(new MataKuliah_26( kode: "00005", nama: "Praktikum Algoritma dan Pemrograman", sks: 3));
20
21        Scanner input = new Scanner(System.in);
22        int menu;
23
24        do {
25            System.out.println("*****");
26            System.out.println("SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER");
27            System.out.println("*****");
28            System.out.println();
29            System.out.println("1. Input Nilai");
30            System.out.println("2. Tampil Nilai");
31            System.out.println("3. Mencari Nilai Mahasiswa");
32            System.out.println("4. Urut Data Nilai");
33            System.out.println("5. Keluar");
34            System.out.print("Pilih : ");
35            menu = input.nextInt();
36        } while (menu < 5);
37    }
38 }
```



```

35 switch (menu) {
36     case 1 -> {
37         String nim, kodeNilai, kodeMK;
38         int indexMahasiswa, indexMatakuliah;
39         String nilai;
40         System.out.println("Masukkan data");
41         System.out.print("Kode\t\t : ");
42         kodeNilai = input.next();
43         System.out.print("Nilai\t\t : ");
44         nilai = input.next();
45         tampilMahasiswa();
46         System.out.print("Pilih mahasiswa by nim: ");
47         nim = input.next();
48         indexMahasiswa = Collections.binarySearch(mahasiswa26s,
49             new Mahasiswa_26(nim, nama: null, telf: null, mahasiswa26Comparator);
50         String newName = mahasiswa26s.get(indexMahasiswa).nama;
51         tampilMatakuliah();
52         System.out.print("Pilih MK by kode: ");
53         kodeMK = input.next();
54         indexMatakuliah = Collections.binarySearch(mataKuliah26s,
55             new MataKuliah_26(kodeMK, nama: null, sks: 0), mataKuliah26Comparator);
56         String newNameMK = mataKuliah26s.get(indexMatakuliah).nama;
57         int newSKS = mataKuliah26s.get(indexMatakuliah).sks;
58         Nilai_26 newNilai = new Nilai_26(kodeNilai, nim, newName, newNameMK, newSKS, nilai);
59         nilai26s.add(newNilai);
60     }
61     case 2 -> tampilNilai();

```

```

62     case 3 -> {
63         String searchNIM;
64         int indexSearch;
65         tampilNilai();
66         System.out.print("Masukkan data mahasiswa [nim] : ");
67         searchNIM = input.next();
68         indexSearch = Collections.binarySearch(nilai26s,
69             new Nilai_26( kode: null, searchNIM, nama: null, mataKuliah: null, sks: 0 , nilai: null),
70             nilai26Comparator);
71         System.out.println("NIM\t\tNama\t\tMata Kuliah\t\tSKS\t\tNilai");
72         System.out.println(nilai26s.get(indexSearch).nim + "\t\t" + nilai26s.get(indexSearch).nama +
73             "\t\t" + nilai26s.get(indexSearch).mataKuliah + " " + nilai26s.get(indexSearch).sks +
74             "\t\t" + nilai26s.get(indexSearch).nilai);
75     }
76     case 4 -> {
77         nilai26s.sort(sortNilaiComparator);
78         tampilNilai();
79     }
80     case 5 -> System.exit(status: 0);
81 }
82 } while (menu <= 5 && menu >= 0);
83 }
84

```

```

85 static Comparator<Mahasiswa_26> mahasiswa26Comparator = Comparator.comparing(o -> o.nim);
86
87 static Comparator<Nilai_26> sortNilaiComparator = Comparator.comparing(o -> o.nilai);
88
89 static Comparator<Nilai_26> nilai26Comparator = Comparator.comparing(o -> o.nim);
90
91 static Comparator<MataKuliah_26> mataKuliah26Comparator = Comparator.comparing(o -> o.kode);
92
93 static void tampilMahasiswa() {
94     System.out.println();
95     System.out.println("DAFTAR MAHASISWA");
96     System.out.println("*****");
97     System.out.println("NIM\tt\tNama\tt\tt\tt\tTelf");
98     mahasiswa26s.forEach(mahasiswa_26 ->
99         System.out.println(mahasiswa_26.nim + "\t\t" + mahasiswa_26.nama + "\t\t\t" + mahasiswa_26.telf));
100 }
101
102 static void tampilMatakuliah() {
103     System.out.println();
104     System.out.println("DAFTAR MATA KULIAH");
105     System.out.println("*****");
106     System.out.println("Kode\tt\tMata Kuliah\tt\tt\tt\tSKS");
107     mataKuliah26s.forEach(mataKuliah_26 ->
108         System.out.println(mataKuliah_26.kode + "\t\t" + mataKuliah_26.nama + "\t" + mataKuliah_26.sks));
109 }
110

```



```

111 static void tampilNilai() {
112     System.out.println();
113     System.out.println("DAFTAR NILAI MAHASISWA");
114     System.out.println("*****");
115     System.out.println("NIM\t\t\tNama\t\t\tMata Kuliah\t\t\t\t\tSKS\t\tNilai");
116     nilai26s.forEach(nilai_26 ->
117         System.out.println(nilai_26.nim + "\t\t" + nilai_26.nama + "\t\t" + nilai_26.mataKuliah
118             + "\t\t" + nilai_26.sks + "\t\t" + nilai_26.nilai));
119 }
120 }

```

## Output Menu Awal dan Penambahan Data

```

"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2\bin\java.exe"
*****
SISTEM PENGOLAHAN DATA NILAI MAHASISWA SEMESTER
*****
1. Input Nilai
2. Tampil Nilai
3. Mencari Nilai Mahasiswa
4. Urut Data Nilai
5. Keluar
Pilih : 1

```

```

Pilih : 1
Masukkan data
Kode      : 0001
Nilai     : 80.75

DAFTAR MAHASISWA
*****
NIM      Nama      Telf
20001    Thalbah    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

```

### Output 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    Thalbah    Internet of Things 3        80.75

```

### Output 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    Thalbah    Internet of Things 3        80.75
20002    Zubair     Praktikum Algoritma dan Pemrograman 3        90.00
Masukkan data mahasiswa [nim] : 20002
NIM      Nama      Mata Kuliah      SKS      Nilai
20002    Zubair     Praktikum Algoritma dan Pemrograman 3        90.00

```

### Output Pengurutan Data Nilai

Sebelumnya sudah saya tambah data agar pengurutan berdasarkan nilai terkecil makin terlihat.

```

*****
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
20006    Ubaidah    Algoritma dan Pemrograman 2        70.00
20001    Thalbah    Internet of Things 3        80.75
20002    Zubair     Praktikum Algoritma dan Pemrograman 3        90.00

```