

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 3**



ENKAPSULASI DAN COLLECTION

Oleh:

Rizki Adhitiya Maulana

NIM. 2410817110014

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
OKTOBER 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN II
MODUL 3

Laporan Praktikum Pemrograman II Modul 3: Enkapsulasi dan Collection ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rizki Adhitiya Maulana
NIM : 2410817110014

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Jovan Gilbert Natamasindah
NIM. 2310817310002

Irham Maulani Abdul Gani, S.Kom.,
M.Kom.
NIP. 199710312025061009

DAFTAR ISI

LEMBAR PENGESAHAN	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
SOAL 1	6
A. Source Code	7
B. Output Program.....	8
C. Pembahasan.....	9
SOAL 2	13
A. Source Code	15
B. Output Program.....	18
C. Pembahasan.....	18
SOAL 3	25
A. Source Code	28
B. Output Program.....	32
C. Pembahasan.....	32
TAUTAN GIT.....	39

DAFTAR GAMBAR

Gambar 1. Screenshot Output Program Soal No.1	8
Gambar 2. Screenshot Output Program Soal No.2	18
Gambar 3. Screenshot Output Program Soal No.3	32

DAFTAR TABEL

Tabel 1. Contoh Input dan Output Soal No.1	6
Tabel 2. Source Code Dice.java Soal No.1	7
Tabel 3. Source Code Main1soal.java Soal No.1.....	7
Tabel 4. Contoh Input dan Output Soal No.2	13
Tabel 5. Source Code Country.java Soal No.2	15
Tabel 6. Source Code Soal2main.java Soal No.2	16
Tabel 7. Contoh Input dan Output Soal No.3	25
Tabel 8. Source Code Student.java Soal No.3	28
Tabel 9. Source Code Soal3main.java Soal No.3	28

SOAL 1

Buatlah program yang mengimplementasikan enkapsulasi dan collection dengan ketentuan seperti berikut.

- Buatlah kelas dengan nama Dadu.
- Terdapat sebuah method `acakNilai()` yang akan memberikan nilai acak ketika objek dadu diinisiasi dengan nilai antara 1 – 6.
- Gunakan collection dengan tipe `LinkedList` untuk menyimpan objek dadu yang diinputkan.
- Program bersifat dinamis.
- Input baris pertama adalah banyaknya jumlah dadu.
- Output adalah nilai tiap objek dadu yang telah diinisiasi.
- Output paling akhir adalah total jumlah nilai semua objek dadu.
- Karena nilai dadu yang dihasilkan acak, maka nilai dadu output tidak harus sama dengan contoh table dibawah.

Tabel 1. Contoh Input dan Output Soal No.1

Input	Output
3	Dadu ke-1 bernilai 2 Dadu ke-2 bernilai 3 Dadu ke-3 bernilai 5 Total nilai dadu keseluruhan 10
4	Dadu ke-1 bernilai 1 Dadu ke-2 bernilai 3 Dadu ke-3 bernilai 6 Dadu ke-4 bernilai 1 Total nilai dadu keseluruhan 11
1	Dadu ke-1 bernilai 2 Total nilai dadu keseluruhan 2

Simpan coding anda dengan nama package: **soal1**

A. Source Code

Tabel 2. Source Code Dice.java Soal No.1

1	package soall;
2	
3	import java.util.LinkedList;
4	import java.util.Random;
5	
6	public class Dice
7	{
8	private int value;
9	
10	public Dice()
11	{
12	roll();
13	}
14	
15	public int getValue()
16	{
17	return value;
18	}
19	
20	public void roll()
21	{
22	Random rand = new Random();
23	this.value = rand.nextInt(6) + 1;
24	}
25	
26	public void displayDice(LinkedList<Dice> diceList)
27	{
28	int total = 0;
29	for (int i = 0; i < diceList.size(); i++)
30	{
31	int currentValue =
32	diceList.get(i).getValue();
33	System.out.println("Dadu ke-" + (i + 1) + "
34	bernilai " + currentValue);
35	total += currentValue;
36	}
37	System.out.println("Total nilai dadu keseluruhan
	" + total);
	}
	}

Tabel 3. Source Code MainIsaal.java Soal No.1

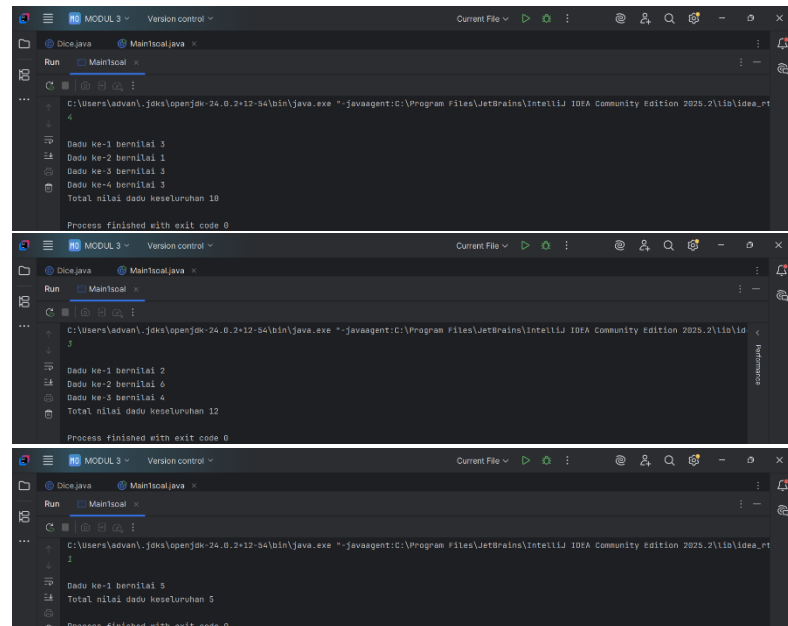
1	package soall;
2	
3	import java.util.LinkedList;
4	import java.util.Scanner;

```

5
6 public class Main1soal
7 {
8     public static void main(String[] args)
9     {
10         Scanner input = new Scanner(System.in);
11         LinkedList<Dice> diceList = new LinkedList<>();
12
13         int diceCount = input.nextInt();
14         System.out.println();
15
16         for (int i = 0; i < diceCount; i++)
17         {
18             diceList.add(new Dice());
19         }
20
21         Dice rollDice = new Dice();
22         rollDice.displayDice(diceList);
23
24         input.close();
25     }
26 }

```

B. Output Program



Gambar 1. Screenshot Output Program Soal No.1

C. Pembahasan

- Dice.java

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `import java.util.LinkedList;` dimana berfungsi untuk mengimpor kelas LinkedList dari pustaka Java. LinkedList yang ada digunakan untuk menyimpan sekumpulan objek secara berurutan.

Pada baris [4] terdapat `import java.util.Random;` dimana berfungsi untuk mengimpor kelas Random dari pustaka Java. Random yang ada digunakan untuk menghasilkan angka dadu dari 1-6 secara acak.

Pada baris [6] terdapat `public class Dice` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [8] terdapat `private int value;` dimana berfungsi untuk menyimpan value dalam tipe data integer, sehingga variabel yang ada akan menyimpan bilangan bulat di dalamnya.

Pada baris [10] sampai [13] terdapat `konstruktor public Dice()` dimana berfungsi untuk memberi nilai awal pada dadu ketika ingin membuat objek. `Roll()` akan dipanggil untuk setiap objek Dice yang dibuat dan secara otomatis akan mendapat nilai dari 1 sampai 6 secara acak.

Pada baris [15] sampai [18] terdapat `public int getValue()` dimana berfungsi untuk mengambil nilai dari variabel value. `return value;` akan mengembalikan isi dari variabel value agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [20] sampai [24] terdapat `public void roll()` dimana berfungsi untuk mengacak nilai dadu yang akan diberikan pada objek dadu. `Random rand = new Random();` untuk membuat objek yang berfungsi menghasilkan angka acak yang nantinya digunakan untuk menentukan nilai dadu. `this.value = rand.nextInt(6) + 1;` untuk memberikan nilai

acak dari sampai 6 untuk variabel value, sehingga objek dadu yang dibuat memiliki nilai masing-masing.

Pada baris [26] sampai [36] terdapat `public void displayDice(LinkedList<Dice> diceList)` dimana berfungsi untuk menampilkan seluruh nilai dadu yang tersimpan di dalam LinkedList. `Int total = 0;` untuk menyimpan jumlah dari semua nilai dadu dan diberi nilai awal 0 karena sebelum penjumlahan dimulai. `for (int i = 0; i < diceList.size(); i++)` untuk melakukan perulangan agar program bisa menelusuri setiap objek dadu di dalam daftar. `int currentValue = diceList.get(i).getValue();` untuk mengambil nilai dari dadu ke-*i* yang ada di dalam diceList, lalu menyimpannya ke dalam variabel currentValue. `System.out.println("Dadu ke-" + (i + 1) + " bernilai " + currentValue);` untuk menampilkan ke layar nomor urut dadu beserta nilai acaknya. `total += currentValue;` untuk menjumlah nilai dari setiap dadu dan hasil penjumlahannya akan dimasukkan ke variabel total. `System.out.println("Total nilai dadu keseluruhan " + total);` digunakan untuk menampilkan hasil penjumlahan seluruh nilai dadu.

Pada baris [7] sampai [37] terdapat `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

- Main1soal.java

Pada baris [1] terdapat `package soal1;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal1.

Pada baris [3] terdapat `import java.util.LinkedList;` dimana berfungsi untuk mengimpor kelas LinkedList dari pustaka Java. LinkedList yang ada digunakan untuk menyimpan sekumpulan objek secara berurutan.

Pada baris [4] terdapat `import java.util.Scanner;` dimana berfungsi untuk mengimpor class Scanner yang berada di package java.util

sehingga program dapat membaca input yang dimasukkan oleh pengguna melalui keyboard.

Pada baris [6] terdapat `public class Main1soal` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [8] terdapat `public static void main(String[] args)` dimana berfungsi sebagai method main atau titik awal program, artinya program akan dijalankan mulai dari kode ini.

Pada baris [10] terdapat `Scanner input = new Scanner(System.in);` dimana berfungsi untuk menghadirkan objek Scanner dengan nama input untuk membaca atau mengambil input dari pengguna melalui keyboard dan memasukkannya ke dalam sistem (`System.in`).

Pada baris [11] terdapat `LinkedList<Dice> diceList = new LinkedList<>();` dimana berfungsi untuk membuat objek LinkedList bernama diceList dengan tipe data Dice.

Pada baris [13] terdapat `int diceCount = input.nextInt();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel diceCount.

Pada baris [14] terdapat `System.out.println();` dimana berfungsi untuk untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [16] sampai [19] terdapat `for (int i = 0; i < diceCount; i++)` dimana berfungsi untuk membuat dan menambahkan objek dadu ke dalam daftar sebanyak jumlah yang dimasukkan oleh pengguna. `diceList.add(new Dice());` untuk objek dadu baru dari kelas Dice dan langsung memasukkannya ke dalam LinkedList.

Pada baris [21] `Dice rollDice = new Dice();` dimana berfungsi untuk membuat objek baru dari class Dice.

Pada baris [22] `rollDice.displayDice(diceList);` dimana berfungsi untuk menampilkan seluruh nilai dari objek-objek dadu yang telah dibuat dan tersimpan dalam `diceList`.

Pada baris [24] `input.close();` dimana berfungsi untuk menutup objek Scanner setelah selesai digunakan.

Pada baris [7] sampai [26] terdapat `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

SOAL 2

Buatlah program yang mengimplementasikan enkapsulasi dan collection dengan ketentuan seperti berikut.

- a. Buatlah kelas dengan nama Negara.
- b. Terdapat 5 attribute pada kelas Negara, yaitu nama, jenis kepemimpinan, nama pemimpin, tanggal kemerdekaan, bulan kemerdekaan, tahun kemerdekaan.
- c. Inisiasi nilai attribute dilakukan pada constructor.
- d. Gunakan collection dengan tipe LinkedList untuk menyimpan objek Negara.
- e. Gunakan collection dengan tipe HashMap untuk menyimpan daftar nama bulan. Nama bulan diambil dari hashmap berdasarkan angka bulan yang diinputkan.
- f. Program bersifat dinamis.
- g. Input baris pertama adalah banyaknya negara. Input baris berikutnya adalah data negara. Jika jenis kepemimpinan adalah monarki maka tidak perlu menginputkan tanggal kemerdekaan.
- h. Output adalah detail dari setiap objek negara yang telah diinputkan

Tabel 4. Contoh Input dan Output Soal No.2

Input
2
Indonesia
Presiden
Joko Widodo
17
8
1945
Palestina
Presiden
Mahmoud Abbas
15
11
1988
Output

<p>Negara Indonesia mempunyai Presiden bernama Joko Widodo Deklarasi Kemerdekaan pada Tanggal 17 Agustus 1945</p> <p>Negara Palestina mempunyai Presiden bernama Mahmoud Abbas Deklarasi Kemerdekaan pada Tanggal 15 November 1988</p>
Input
<p>3 Thailand Monarki Maha Vajiralongkorn Indonesia Presiden Joko Widodo 17 8 1945 Malaysia perdana menteri Ismail Sabri Yaakob 31 8 1957</p>
Output
<p>Negara Thailand mempunyai Raja bernama Maha Vajiralongkorn</p> <p>Negara Indonesia mempunyai Presiden bernama Joko Widodo Deklarasi Kemerdekaan pada Tanggal 17 Agustus 1945</p> <p>Negara Malaysia mempunyai Perdana Menteri bernama Ismail Sabri Yaakob Deklarasi Kemerdekaan pada Tanggal 31 Agustus 1957</p>

Simpan coding anda dengan nama package: **soal2**

A. Source Code

Tabel 5. Source Code Country.java Soal No.2

1	package soal2;
2	
3	public class Country
4	{
5	private String name;
6	private String leadershipType;
7	private String leaderName;
8	private int independenceDay;
9	private int independenceMonth;
10	private int independenceYear;
11	
12	public Country(String name, String leadershipType, String leaderName, int independenceDay, int independenceMonth, int independenceYear)
13	{
14	this.name = name;
15	this.leadershipType = leadershipType;
16	this.leaderName = leaderName;
17	this.independenceDay = independenceDay;
18	this.independenceMonth = independenceMonth;
19	this.independenceYear = independenceYear;
20	}
21	
22	public String getName()
23	{
24	return name;
25	}
26	
27	public String getLeadershipType()
28	{
29	return leadershipType;
30	}
31	
32	public String getLeaderName()
33	{
34	return leaderName;
35	}
36	
37	public int getIndependenceDay()
38	{
39	return independenceDay;
40	}
41	
42	public int getIndependenceMonth()
43	{
44	return independenceMonth;

45	}
46	
47	public int getIndependenceYear()
48	{
49	return independenceYear;
50	}
51	
52	public void displayInfo(String monthName)
53	{
54	if (getLeadershipType().equals("presiden"))
55	{
56	System.out.println("Negara " + getName() + " mempunyai Presiden bernama " + getLeaderName());
57	System.out.println("Deklarasi Kemerdekaan pada Tanggal " + getIndependenceDay() + " " + monthName + " " + getIndependenceYear());
58	}
59	
60	else if (getLeadershipType().equals("perdana menteri"))
61	{
62	System.out.println("Negara " + getName() + " mempunyai Perdana Menteri bernama " + getLeaderName());
63	System.out.println("Deklarasi Kemerdekaan pada Tanggal " + getIndependenceDay() + " " + monthName + " " + getIndependenceYear());
64	}
65	
66	else
67	{
68	System.out.println("Negara " + getName() + " mempunyai Raja bernama " + getLeaderName());
69	}
70	System.out.println();
71	}
72	}

Tabel 6. Source Code Soal2main.java Soal No.2

1	package soal2;
2	
3	import java.util.HashMap;
4	import java.util.LinkedList;
5	import java.util.Scanner;
6	
7	public class Main2soal
8	{
9	public static void main(String[] args)
10	{
11	Scanner input = new Scanner(System.in);


```

12
13     HashMap<Integer, String> monthName = new
HashMap<>();
14     monthName.put(1, "Januari");
15     monthName.put(2, "Februari");
16     monthName.put(3, "Maret");
17     monthName.put(4, "April");
18     monthName.put(5, "Mei");
19     monthName.put(6, "Juni");
20     monthName.put(7, "Juli");
21     monthName.put(8, "Agustus");
22     monthName.put(9, "September");
23     monthName.put(10, "Oktober");
24     monthName.put(11, "November");
25     monthName.put(12, "Desember");
26
27     LinkedList<Country> countryList = new
LinkedList<>();
28
29     int countryCount = input.nextInt();
30     input.nextLine();
31
32     for (int i = 0; i < countryCount; i++)
33     {
34         String name = input.nextLine();
35         String leadershipType = input.nextLine();
36         String leaderName = input.nextLine();
37
38         int independenceDay = 0;
39         int independenceMonth = 0;
40         int independenceYear = 0;
41
42         if
(!leadershipType.equalsIgnoreCase("monarki"))
43         {
44             independenceDay = input.nextInt();
45             independenceMonth = input.nextInt();
46             independenceYear = input.nextInt();
47             input.nextLine();
48         }
49
50         countryList.add(new Country(name,
leadershipType, leaderName, independenceDay,
independenceMonth, independenceYear));
51     }
52
53     System.out.println();
54
55     for (Country country : countryList)

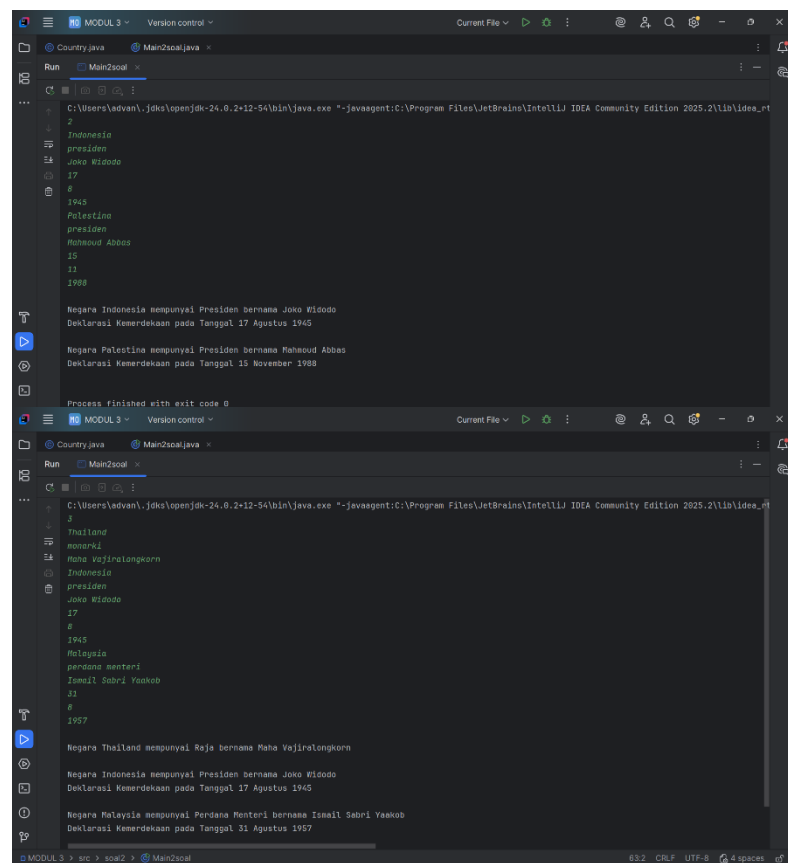
```

```

56         {
57             String monthText =
monthName.get(country.getIndependenceMonth());
58             country.displayInfo(monthText);
59         }
60
61         input.close();
62     }
63 }

```

B. Output Program



Gambar 2. Screenshot Output Program Soal No.2

C. Pembahasan

- Country.java

Pada baris [1] terdapat *package soal2;* dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal2.

Pada baris [3] terdapat *public class Country* dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [5] terdapat *private String name;* dimana berfungsi untuk menyimpan name dalam tipe data String, sehingga variabel yang ada akan menyimpan karakter di dalamnya.

Pada baris [6] terdapat *private String leadershipType;* dimana berfungsi untuk menyimpan leadershipType dalam tipe data String, sehingga variabel yang ada akan menyimpan karakter di dalamnya.

Pada baris [7] terdapat *private String leaderName;* dimana berfungsi untuk menyimpan leaderName dalam tipe data String, sehingga variabel yang ada akan menyimpan karakter di dalamnya.

Pada baris [8] terdapat *private int independenceDay;* dimana berfungsi untuk menyimpan independenceDay dalam tipe data integer, sehingga variabel yang ada akan menyimpan bilangan bulat di dalamnya.

Pada baris [9] terdapat *private int independenceMonth;* dimana berfungsi untuk menyimpan independenceMonth dalam tipe data integer, sehingga variabel yang ada akan menyimpan bilangan bulat di dalamnya.

Pada baris [10] terdapat *private int independenceYear;* dimana berfungsi untuk menyimpan independenceYear dalam tipe data integer, sehingga variabel yang ada akan menyimpan bilangan bulat di dalamnya.

Pada baris [12] sampai [20] terdapat konstruktor *public Country(String name, String leadershipType, String leaderName, int independenceDay, int independenceMonth, int independenceYear)* dimana berfungsi untuk memberikan nilai awal pada seluruh atribut ketika objek Country dibuat.

```
this.name = name, this.leadershipType = leadershipType, this.leaderName = leaderName, this.independenceDay = independenceDay, this.independenceMonth = independenceMonth dan
```

this.independenceYear = independenceYear akan menyimpan nilai yang dimasukkan ke dalam variabel-variabel yang sebelumnya sudah dibuat.

Pada baris [22] sampai [25] terdapat *public String getName()* dimana berfungsi untuk mengambil nilai dari variabel *name*. *return name;* akan mengembalikan isi dari variabel *name* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [27] sampai [30] terdapat *public String getLeadershipType()* dimana berfungsi untuk mengambil nilai dari variabel *leadershipType*. *return leadershipType;* akan mengembalikan isi dari variabel *leadershipType* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [32] sampai [35] terdapat *public String getLeaderName()* dimana berfungsi untuk mengambil nilai dari variabel *leaderName*. *return leaderName;* akan mengembalikan isi dari variabel *leaderName* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [37] sampai [40] terdapat *public int getIndependenceDay()* dimana berfungsi untuk mengambil nilai dari variabel *IndependenceDay*. *return IndependenceDay;* akan mengembalikan isi dari variabel *IndependenceDay* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [42] sampai [45] terdapat *public int getIndependenceMonth()* dimana berfungsi untuk mengambil nilai dari variabel *IndependenceMonth*. *return IndependenceMonth;* akan mengembalikan isi dari variabel *IndependenceMonth* agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [47] sampai [50] terdapat *public int getIndependenceYear()* dimana berfungsi untuk mengambil nilai dari

variabel `IndependenceYear`. `return IndependenceYear;` akan mengembalikan isi dari variabel `IndependenceYear` agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [52] sampai [71] terdapat `public void displayInfo(String monthName)` dimana berfungsi untuk menampilkan seluruh informasi lengkap negara ke layar.

Pada baris [54] sampai [58] terdapat `if (getLeadershipType().equals("presiden"))` dimana berfungsi untuk memeriksa apakah jenis kepemimpinan negara adalah presiden. Jika benar, maka program akan menampilkan informasi negara yang berada di `System.out.println` ke layar.

Pada baris [60] sampai [64] terdapat `else if getLeadershipType().equals("perdana menteri")` dimana berfungsi untuk memeriksa apakah jenis kepemimpinan negara adalah perdana menteri. Jika benar, maka program akan menampilkan informasi negara yang berada di `System.out.println` ke layar.

Pada baris [66] sampai [69] terdapat `else` dimana berfungsi untuk menangani kondisi ketika jenis kepemimpinan negara bukan presiden maupun perdana menteri. Jika kondisi tersebut terjadi, maka program akan menampilkan informasi negara yang berada di `System.out.println` ke layar.

Pada baris [70] terdapat `System.out.println();` dimana berfungsi untuk untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [4] sampai [72] terdapat `{ }` dimana digunakan untuk membatasi blok kode yang ada pada program.

- Soal2main.java

Pada baris [1] terdapat `package soal2;` dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal2.

Pada baris [3] terdapat `import java.util.HashMap;` dimana berfungsi untuk mengimpor kelas HashMap dari pustaka Java. HashMap digunakan untuk menyimpan data dalam bentuk pasangan key value.

Pada baris [4] terdapat `import java.util.LinkedList;` dimana berfungsi untuk mengimpor kelas LinkedList dari pustaka Java. LinkedList yang ada digunakan untuk menyimpan sekumpulan objek secara berurutan.

Pada baris [5] terdapat `import java.util.Scanner;` dimana berfungsi untuk mengimpor class Scanner yang berada di package java.util sehingga program dapat membaca input yang dimasukkan oleh pengguna melalui keyboard.

Pada baris [7] terdapat `public class Main2soal` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [9] terdapat `public static void main(String[] args)` dimana berfungsi sebagai method main atau titik awal program, artinya program akan dijalankan mulai dari kode ini.

Pada baris [11] terdapat `Scanner input = new Scanner(System.in);` dimana berfungsi untuk menghadirkan objek Scanner dengan nama input untuk membaca atau mengambil input dari pengguna melalui keyboard dan memasukkannya ke dalam sistem (`System.in`).

Pada baris [13] sampai [25] terdapat `HashMap<Integer, String> monthName = new HashMap<>()` dimana berfungsi untuk menyimpan pasangan data antara angka bulan dan nama bulan. Setiap angka dari 1 hingga 12 akan dipasangkan dengan nama bulan yang sesuai, dengan begitu program dapat menampilkan nama bulan berdasarkan angka yang dimasukkan pengguna.

Pada baris [27] terdapat `LinkedList<Country> countryList = new LinkedList<>()`; dimana berfungsi untuk membuat daftar kosong

untuk menampung objek-objek Country yang akan dimasukkan berdasarkan input pengguna.

Pada baris [29] terdapat `int countryCount = input.nextInt();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `countryCount`.

Pada baris [30] terdapat `input.nextLine();` dimana berfungsi untuk menghapus karakter newline setelah input.

Pada baris [32] sampai [51] terdapat `for (int i = 0; i < countryCount; i++)` dimana berfungsi mengulangi proses input data negara sebanyak jumlah negara yang dimasukkan sebelumnya.

Pada baris [34] sampai [36] terdapat `String name, String leadershipType` dan `String leaderName = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel-variabel yang dituju.

Pada baris [38] sampai [40] terdapat `int independenceDay, independenceMonth` dan `independenceYear = 0;` dimana berfungsi untuk memberikan nilai awal 0 pada ketiga variabel tersebut.

Pada baris [42] sampai [48] terdapat `if (!leadershipType.equalsIgnoreCase("monarki"))` dimana berfungsi untuk menangani kondisi ketika jenis kepemimpinan negara bukan monarki. Jika kondisi tersebut terjadi, maka program akan meminta pengguna untuk memasukkan `independenceDay`, `independenceMonth` dan `independenceYear`. Kemudian, apabila jenis kepemimpinan negara adalah monarki, bagian dari program akan meminta pengguna untuk memasukkan `independenceDay`, `independenceMonth` dan `independenceYear` akan di skip.

Pada baris [50] terdapat `countryList.add(new Country(...))` dimana berfungsi untuk membuat objek `Country` baru berdasarkan input pengguna, lalu menambahkannya ke dalam daftar `countryList`.

Pada baris [53] terdapat `System.out.println()`; dimana berfungsi untuk untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [55] sampai [59] terdapat `for (Country country : countryList` dimana berfungsi untuk menampilkan informasi dari setiap objek Country yang telah disimpan di dalam daftar. `String monthText = monthName.get(country.getIndependenceMonth());` untuk mengambil nama bulan berdasarkan angka bulan kemerdekaan yang di masukkan pengguna. `country.displayInfo(monthText);` untuk menampilkan seluruh informasi negara yang dimasukkan, tergantung dengan jenis kepemimpinannya.

Pada baris [61] terdapat `input.close()`; dimana berfungsi untuk menutup objek Scanner setelah selesai digunakan.

Pada baris [8] sampai [63] terdapat `{}` dimana digunakan untuk membatasi blok kode yang ada pada program.

SOAL 3

Buatlah program yang mengimplementasikan enkapsulasi dan collection dengan ketentuan seperti berikut.

- a. Buatlah kelas dengan nama Mahasiswa.
- b. Terdapat 2 atribut pada kelas, yaitu nama dan nim.
- c. Inisiasi nilai atribut dilakukan pada konstruktor.
- d. Terdapat getter untuk 2 atribut tadi.
- e. Gunakan collection dengan tipe ArrayList untuk menyimpan objek Mahasiswa.
- f. Program bersifat dinamis dan interaktif.
- g. Program dapat melakukan operasi seperti berikut:
 - Tambah Mahasiswa, menambahkan objek baru ke ArrayList
 - Hapus Mahasiswa, menghapus data mahasiswa dari ArrayList berdasarkan NIM
 - Cari Mahasiswa, menampilkan data mahasiswa berdasarkan NIM yang di input oleh pengguna
 - Tampilkan seluruh data Mahasiswa, menampilkan seluruh data mahasiswa dari ArrayList. Tampilkan nama dan NIM mahasiswa
 - Keluar, program berhenti dan seluruh data pada ArrayList dihapus, ketika program dijalankan ulang, ArrayList masih kosong

Tabel 7. Contoh Input dan Output Soal No.3

Output
Menu: 1. Tambah Mahasiswa 2. Hapus Mahasiswa berdasarkan NIM 3. Cari Mahasiswa berdasarkan NIM 4. Tampilkan Daftar Mahasiswa 0. Keluar Pilihan: 1 Masukkan Nama Mahasiswa: Bachrul Uluum Masukkan NIM Mahasiswa (harus unik): 2010817210025

Mahasiswa Bachrul Uluum ditambahkan.

Menu:

1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
0. Keluar

Pilihan: 1

Masukkan Nama Mahasiswa: Muhammad Aulia Akbar

Masukkan NIM Mahasiswa (harus unik): 2010817210023

Mahasiswa Muhammad Aulia Akbar ditambahkan.

Menu:

1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
0. Keluar

Pilihan: 4

Daftar Mahasiswa:

NIM: 2010817210025, Nama: Bachrul Uluum

NIM: 2010817210023, Nama: Muhammad Aulia Akbar

Menu:

1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
0. Keluar

Pilihan: 2

Masukkan NIM Mahasiswa yang akan dihapus: 2010817210025

Mahasiswa dengan NIM 2010817210025 dihapus.

Menu:

1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM

3. Cari Mahasiswa berdasarkan NIM

4. Tampilkan Daftar Mahasiswa

0. Keluar

Pilihan: 0

Terima kasih!

Simpan coding anda dengan nama package: **soal3**

A. Source Code

Tabel 8. Source Code Student.java Soal No.3

1	package soal3;
2	
3	public class Student
4	{
5	private String name;
6	private String nim;
7	
8	public Student(String name, String nim)
9	{
10	this.name = name;
11	this.nim = nim;
12	}
13	
14	public String getName()
15	{
16	return name;
17	}
18	
19	public String getNim()
20	{
21	return nim;
22	}
23	}

Tabel 9. Source Code Soal3main.java Soal No.3

1	package soal3;
2	
3	import java.util.ArrayList;
4	import java.util.Scanner;
5	
6	public class Main3soal
7	{
8	public static void main(String[] args)
9	{
10	ArrayList<Student> studentList = new
11	ArrayList<>();
12	Scanner input = new Scanner(System.in);
13	while (true)
14	{
15	System.out.println("\nMenu:");
16	System.out.println("1. Tambah Mahasiswa");
17	System.out.println("2. Hapus Mahasiswa
	berdasarkan NIM");
18	System.out.println("3. Cari Mahasiswa
	berdasarkan NIM");

19	System.out.println("4. Tampilkan Daftar Mahasiswa");
20	System.out.println("0. Keluar");
21	System.out.print("Pilihan: ");
22	String choice = input.nextLine();
23	
24	switch (choice)
25	{
26	case "1":
27	System.out.print("Masukkan Nama Mahasiswa: ");
28	String name = input.nextLine();
29	System.out.print("Masukkan NIM Mahasiswa (harus unik): ");
30	String nim = input.nextLine();
31	
32	boolean nimExists = false;
33	for (int i = 0; i <
34	studentList.size(); i++)
35	{
36	if
37	(studentList.get(i).getNim().equals(nim))
38	{
39	nimExists = true;
40	break;
41	}
42	if (nimExists)
43	{
44	System.out.println("NIM sudah digunakan. Mahasiswa tidak ditambahkan.");
45	}
46	
47	else
48	{
49	studentList.add(new
50	Student(name, nim));
51	System.out.println("Mahasiswa "
52	+ name + " ditambahkan.");
53	}
54	break;
55	
56	case "2":
57	System.out.print("Masukkan NIM Mahasiswa yang akan dihapus: ");
58	String nimDelete = input.nextLine();
	boolean foundDelete = false;

```

59         for (int i = 0; i <
studentList.size(); i++)
60         {
61             if
(studentList.get(i).getNim().equals(nimDelete))
62             {
63                 studentList.remove(i);
64
System.out.println("Mahasiswa dengan NIM " + nimDelete +
" dihapus.");
65
                 foundDelete = true;
66                 break;
67             }
68         }
69
70         if (!foundDelete)
71         {
72             System.out.println("Mahasiswa
dengan NIM " + nimDelete + " tidak ditemukan.");
73         }
74         break;
75
76         case "3":
77             System.out.print("Masukkan NIM
Mahasiswa yang dicari: ");
78             String nimSearch = input.nextLine();
79             boolean found = false;
80
81             for (int i = 0; i <
studentList.size(); i++)
82             {
83                 if
(studentList.get(i).getNim().equals(nimSearch))
84                 {
85
System.out.println("Mahasiswa ditemukan:");
86                 System.out.println("NIM: " +
studentList.get(i).getNim() + ", Nama: " +
studentList.get(i).getName());
87                 found = true;
88                 break;
89             }
90         }
91
92         if (!found)
93         {
94             System.out.println("Mahasiswa
dengan NIM " + nimSearch + " tidak ditemukan.");
95         }

```

```

96         break;
97
98         case "4":
99             System.out.println("Daftar
100 Mahasiswa:");
101             if (studentList.isEmpty())
102             {
103                 System.out.println("(Belum ada
104 data mahasiswa)");
105             }
106             else
107             {
108                 for (int i = 0; i <
109 studentList.size(); i++)
110                 {
111                     System.out.println("NIM: " +
112 studentList.get(i).getNim() + ", Nama: " +
113 studentList.get(i).getName());
114                 }
115             }
116             break;
117
118         case "0":
119             studentList.clear();
120             System.out.println("Terima kasih!");
121             input.close();
122             return;
123
124         default:
125             System.out.println("Pilihan tidak
126 valid. Silakan coba lagi.");
127     }
128 }
129 }
130 }

```

B. Output Program

```
Run Main3soal.java
Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
5. Keluar
Pilihan: 1
Masukkan Nama Mahasiswa: Bachrul Ulum
Masukkan NIM Mahasiswa (harus unik): 2010017210025
Mahasiswa Bachrul Ulum ditambahkan.

Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
5. Keluar
Pilihan: 2
Masukkan Nama Mahasiswa: Muhammad Aulia Akbar
Masukkan NIM Mahasiswa (harus unik): 2010017210023
Mahasiswa Muhammad Aulia Akbar ditambahkan.

Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
5. Keluar
Pilihan: 3
Masukkan NIM Mahasiswa yang dicari: 2010017210023
Mahasiswa ditemukan:
NIM: 2010017210023, Nama: Muhammad Aulia Akbar

Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
5. Keluar
Pilihan: 0
Terima kasih!

Process finished with exit code 0
```

Gambar 3. Screenshot Output Program Soal No.3

C. Pembahasan

- Student.java

Pada baris [1] terdapat *package soal3;* dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal3.

Pada baris [3] terdapat *public class Student* dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [5] terdapat *private String name;* dimana berfungsi untuk menyimpan name dalam tipe data String, sehingga variabel yang ada akan menyimpan karakter di dalamnya.

Pada baris [6] terdapat *private String nim;* dimana berfungsi untuk menyimpan nim dalam tipe data String, sehingga variabel yang ada akan menyimpan karakter di dalamnya.

Pada baris [8] sampai [12] terdapat *konstruktor public Student (String name, String nim)* dimana berfungsi untuk memberikan nilai awal pada seluruh atribut ketika objek Student dibuat. *this.name = name;* dan *this.nim = nim;* akan menyimpan nilai yang dimasukkan ke dalam variabel-variabel yang sebelumnya sudah dibuat.

Pada baris [14] sampai [17] *public String getName()* dimana berfungsi untuk mengambil nilai dari variabel name. *return name;* akan mengembalikan isi dari variabel name agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.

Pada baris [19] sampai [22] *public String getNim()* dimana berfungsi untuk mengambil nilai dari variabel nim. *return nim;* akan mengembalikan isi dari variabel nim agar dapat ditampilkan ketika diperlukan atau digunakan oleh program.]

Pada baris [4] sampai [23] terdapat *{ }* dimana digunakan untuk membatasi blok kode yang ada pada program.

- Main3soal.java

Pada baris [1] terdapat *package soal3;* dimana berfungsi untuk menandakan bahwa file program ini berada di dalam folder bernama soal3.

Pada baris [3] terdapat *import java.util.LinkedList;* dimana berfungsi untuk mengimpor kelas LinkedList dari pustaka Java. LinkedList yang ada digunakan untuk menyimpan sekumpulan objek secara berurutan.

Pada baris [4] terdapat `import java.util.Scanner;` dimana berfungsi untuk mengimpor class Scanner yang berada di package java.util sehingga program dapat membaca input yang dimasukkan oleh pengguna melalui keyboard.

Pada baris [6] terdapat `public class Soal3Main` dimana berfungsi untuk membuat class publik yang mana nantinya semua kode program yang dibuat akan berada di dalam class ini.

Pada baris [8] terdapat `public static void main(String[] args)` dimana berfungsi sebagai method main atau titik awal program, artinya program akan dijalankan mulai dari kode ini.

Pada baris [10] terdapat `ArrayList<Student> studentList = new ArrayList<>();` dimana berfungsi untuk membuat objek LinkedList bernama StudentList dengan tipe data Student.

Pada baris [11] terdapat `Scanner input = new Scanner(System.in);` dimana berfungsi untuk menghadirkan objek Scanner dengan nama input untuk membaca atau mengambil input dari pengguna melalui keyboard dan memasukkannya ke dalam sistem (`System.in`).

Pada baris [13] sampai [123] terdapat `while (true)` dimana berfungsi untuk membuat program berjalan terus-menerus sampai pengguna memanggil fungsi keluar yang ada untuk menghentikan program.

Pada baris [15] sampai [21] terdapat `System.out.println();` dan `System.out.print` dimana berfungsi untuk untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [22] terdapat `String choice = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel choice.

Pada baris [24] sampai [122] terdapat *switch (choice)* dimana berfungsi untuk menjalankan menu program berdasarkan pilihan yang dimasukkan oleh pengguna.

Pada baris [26] sampai [52] terdapat *case "1"* dimana berfungsi untuk mengurus pilihan menu nomor 1 apabila di masukkan oleh pengguna.

Pada baris [27] dan [29] terdapat *System.out.print* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [28] dan [30] terdapat *String name* dan *String nim = input.nextLine();* dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel *name* dan *nim*.

Pada baris [32] sampai [40] terdapat *boolean nimExists = false;* dimana berfungsi untuk mengecek apakah *nim* yang diinput sudah pernah dimasukkan atau belum. *for (int i = 0; i < studentList.size(); i++)* untuk menelusuri setiap data mahasiswa dalam daftar. *if (studentList.get(i).getNim().equals(nim))* untuk membandingkan *nim* yang diinput pengguna dengan *nim* yang ada di daftar, apabila ada *nim* yang sama maka *nimExists = true;* akan memberitahu kalau *nim* yang ada sudah digunakan dan sistem *case 1* akan langsung dihentikan.

Pada baris [42] sampai [45] terdapat *if (nimExists)* dimana berfungsi untuk menampilkan *System.out.println("NIM sudah digunakan. Mahasiswa tidak ditambahkan.");* ke layar apabila *nim* yang dimasukkan ternyata sudah digunakan.

Pada baris [47] sampai [52] terdapat *else* dimana berfungsi untuk menampilkan *System.out.println("Mahasiswa " + name + " ditambahkan.");* ke layar apabila informasi mahasiswa baru berhasil ditambahkan. Nama dan *nim* yang diinput pengguna akan dimasukkan ke dalam

arraylist `studentList.add(new Student(name, nim));` dan sistem case 1 akan dihentikan.

Pada baris [54] sampai [74] terdapat case “2” dimana berfungsi untuk mengurus pilihan menu nomor 2 apabila di masukkan oleh pengguna.

Pada baris [55] terdapat `System.out.print` dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [56] terdapat `String nimDelete = input.nextLine();` dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel `nimDelete`.

Pada baris [57] terdapat `boolean foundDelete = false;` dimana berfungsi untuk mengecek apakah nim yang diinput ditemukan atau tidak ada.

Pada baris [59] sampai [68] terdapat `for (int i = 0; i < studentList.size(); i++)` dimana berfungsi untuk menelusuri setiap data mahasiswa dalam daftar. `if (studentList.get(i).getNim().equals(nimDelete))` untuk membandingkan nim yang diinput pengguna dengan nim mahasiswa yang ada di daftar. `studentList.remove(i);` untuk menghapus data mahasiswa dari daftar berdasarkan input pengguna sebelumnya. `System.out.println("Mahasiswa dengan NIM " + nimDelete + " dihapus.");` untuk menampilkan pesan ke layar kalau data mahasiswa yang ada di hapus. `foundDelete = true;` untuk menandai bahwa data mahasiswa dengan nim yang dicari berhasil ditemukan untuk dihapus dan sistem case 2 akan dihentikan.

Pada baris [70] sampai [74] terdapat `if (!foundDelete)` dimana berfungsi untuk menampilkan pesan mahasiswa tidak ditemukan, apabila nim yang dicari tidak ada di daftar mahasiswa.

Pada baris [76] sampai [96] terdapat case “3” dimana berfungsi untuk mengurus pilihan menu nomor 3 apabila di masukkan oleh pengguna.

Pada baris [77] terdapat *System.out.print* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [78] terdapat *String nimSearch = input.nextLine();* dimana berfungsi untuk menyimpan input dari pengguna ke dalam variabel *nimSearch*.

Pada baris [79] terdapat *boolean found = false;* dimana berfungsi untuk mengecek apakah *nim* yang diinput ada atau tidak ada.

Pada baris [81] sampai [90] terdapat *for (int i = 0; i < studentList.size(); i++)* dimana berfungsi untuk dimana berfungsi untuk menelusuri setiap data mahasiswa dalam daftar. *if (studentList.get(i).getNim().equals(nimSearch))* untuk membandingkan *nim* yang diinput pengguna dengan *nim* mahasiswa yang ada di daftar. *System.out.println("Mahasiswa ditemukan:");* dan *System.out.println("NIM: " + studentList.get(i).getNim() + ", Nama: " + studentList.get(i).getName());* untuk menampilkan pesan ke layar apabila *nim* mahasiswa yang dicari ditemukan. *found = true;* untuk menandai bahwa data mahasiswa dengan *nim* yang dicari berhasil ditemukan dan sistem case 3 akan dihentikan.

Pada baris [92] sampai [96] terdapat *if (!found)* dimana berfungsi untuk menampilkan pesan mahasiswa tidak ditemukan, apabila *nim* yang dicari tidak ada di daftar mahasiswa.

Pada baris [98] sampai [112] terdapat terdapat case “4” dimana berfungsi untuk mengurus pilihan menu nomor 4 apabila di masukkan oleh pengguna.

Pada baris [99] terdapat *System.out.print* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [100] sampai [103] terdapat *if (studentList.isEmpty())* dimana berfungsi untuk mengecek apakah

daftar mahasiswa masih kosong, apabila kosong akan muncul tampilan belum ada data mahasiswa.

Pada baris [105] sampai [112] terdapat *else* dimana berfungsi untuk menampilkan seluruh data mahasiswa yang ada di dalam *studentList*. *for (int i = 0; i < studentList.size(); i++)* dimana berfungsi untuk menelusuri setiap data mahasiswa dalam daftar dan sistem case 4 akan dihentikan.

Pada baris [114] sampai [118] terdapat *case "0"* dimana berfungsi untuk mengurus pilihan menu nomor 0 apabila di masukkan oleh pengguna.

Pada baris [115] terdapat *studentList.clear();* dimana berfungsi untuk menghapus seluruh data mahasiswa dari daftar sebelum keluar dari program.

Pada baris [116] terdapat *System.out.println* dimana berfungsi untuk menampilkan teks atau nilai yang ada ke layar.

Pada baris [117] terdapat *input.close();* dimana berfungsi untuk menutup objek Scanner setelah selesai digunakan.

Pada baris [118] terdapat *return;* dimana berfungsi untuk menghentikan eksekusi program sepenuhnya.

Pada baris [120] sampai [121] terdapat *default:* dimana berfungsi untuk menangani input menu yang tidak valid. *System.out.println ("Pilihan tidak valid. Silakan coba lagi.");* dimana berfungsi untuk menampilkan teks pilihan tidak sesuai dengan daftar menu yang ada ke layar.

Pada baris [7] sampai [125] terdapat *{ }* dimana digunakan untuk membatasi blok kode yang ada pada program.

TAUTAN GIT

<https://github.com/Rizki-A-M/PRAKTIKUM-PEMROGRAMAN-II.git>