

**LAPORAN PRAKTIKUM  
PEMROGRAMAN II  
MODUL 3**



**ENKAPSULASI & COLLECTION**

**Oleh:**

**Indra Suryadilaga      NIM. 2410817310014**

**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 : Indra Suryadilaga  
NIM : 2410817310014

Menyetujui,  
Asisten Praktikum

Mengetahui,  
Dosen Penanggung Jawab Praktikum

Galih Aji Sabdaraya  
NIM. 2310817210005

Andreyan Rizky Baskara, S.Kom.,  
M.Kom.  
NIP. 19930703 201903 1 011

## DAFTAR ISI

LEMBAR PENGESAHAN .....	2
DAFTAR ISI.....	3
DAFTAR GAMBAR .....	4
DAFTAR TABEL .....	5
SOAL 1 .....	6
A. Source Code .....	7
B. Output Program.....	9
C. Pembahasan.....	9
SOAL 2 .....	12
A. Source Code .....	13
A. Output Program.....	19
B. Pembahasan.....	19
SOAL 3 .....	23
A. Source Code .....	25
B. Output Program.....	30
C. Pembahasan.....	31
GITHUB.....	34

## **DAFTAR GAMBAR**

Gambar 1. Screenshot Hasil Jawaban Soal 1 .....	9
Gambar 2. Screenshot Hasil Jawaban Soal 2 .....	19
Gambar 3. Screenshot Hasil Jawaban Soal 3 .....	30

## DAFTAR TABEL

Tabel 1.1 Soal No. 1.....	6
Tabel 1.2 MainSoal1.java Source Code.....	7
Tabel 1.3 Dadu.java Source Code.....	7
Tabel 1.4 LemeparDadu.java Source Code.....	7
Tabel 2.1 Soal No. 2.....	12
Tabel 2.2 Soal2Main.java Source Code.....	13
Tabel 2.3 Negara.java Source Code.....	17
Tabel 3.1 Soal No. 3.....	23
Tabel 3.2 Mahasiswa.java Source Code .....	25
Tabel 3.3 DaftarMahasiswa.java Source Code .....	26
Tabel 3.3 Soal3Main.java Source Code.....	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 1.1 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 1.2 MainSoal1.java Source Code*

File: MainSoal1.java	
1	package
2	modulTiga.PRAK201_2410817310014_IndraSuryadilaga;
3	
4	public class Main {
5	public static void main(String[] args) {
6	LemparDadu lemparDadu = new LemparDadu();
7	lemparDadu.inputJumlahDadu();
8	lemparDadu.lemparDadu();
9	lemparDadu.tampilkanHasil();
10	}
11	}

*Tabel 1.3 Dadu.java Source Code*

File: Dadu.java	
1	package
2	modulTiga.PRAK201_2410817310014_IndraSuryadilaga;
3	
4	import java.util.Random;
5	
6	public class Dadu {
7	private int mataDadu;
8	
9	public void acakNilai() {
10	Random random = new Random();
11	this.mataDadu = random.nextInt(6) + 1;
12	}
13	
14	public int getMataDadu() {
15	return mataDadu;
16	}
17	}

*Tabel 1.4 LemeparDadu.java Source Code*

File: LemparDadu.java	
1	package
2	modulTiga.PRAK201_2410817310014_IndraSuryadilaga;
3	
4	import java.util.LinkedList;

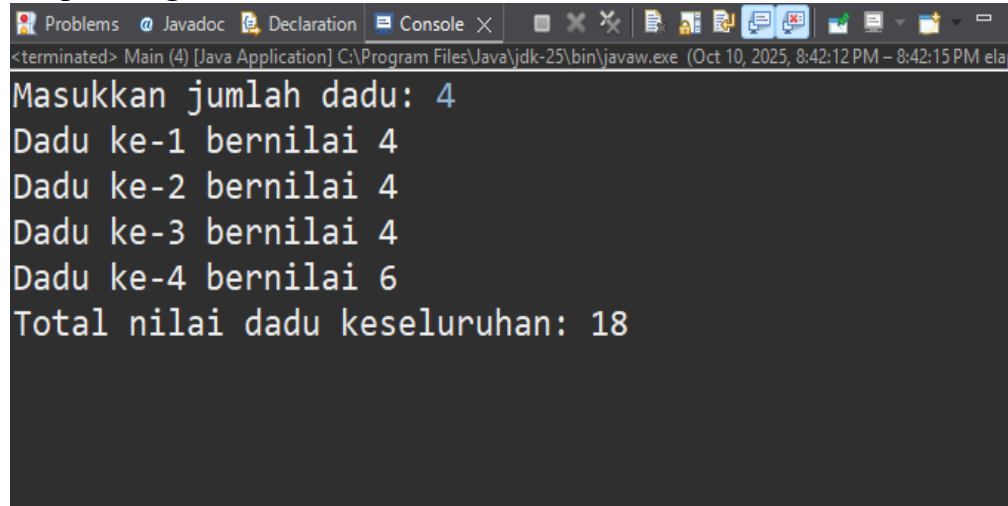
```

5 import java.util.Scanner;
6
7 public class Lempardadu {
8     private LinkedList<Dadu> daftarDadu = new
9     LinkedList<>();
10    private int jumlahDadu;
11    private int totalNilai;
12
13    public void inputJumlahDadu() {
14        Scanner input = new Scanner(System.in);
15        while (true) {
16            System.out.print("Masukkan jumlah dadu:
17 ");
18            jumlahDadu = input.nextInt();
19            if (jumlahDadu > 0) break;
20            System.out.println("Jumlah dadu harus
21 lebih dari 0!");
22        }
23        input.close();
24    }
25
26    public void lemparDadu() {
27        totalNilai = 0;
28        daftarDadu.clear();
29        for (int i = 0; i < jumlahDadu; i++) {
30            Dadu d = new Dadu();
31            d.acakNilai();
32            daftarDadu.add(d);
33            System.out.println("Dadu ke-" + (i + 1)
34 + " bernilai " + d.getMataDadu());
35            totalNilai += d.getMataDadu();
36        }
37    }
38
39    public void tampilkanHasil() {
40        System.out.println("Total nilai dadu
41 keseluruhan: " + totalNilai);
42    }
43 }

```



## B. Output Program

A screenshot of a Java IDE's console window. The window title bar shows 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console text reads: '<terminated> Main (4) [Java Application] C:\Program Files\Java\jdk-25\bin\javaw.exe (Oct 10, 2025, 8:42:12 PM - 8:42:15 PM elapsed)'. Below this, the program output is displayed in a monospaced font: 'Masukkan jumlah dadu: 4', 'Dadu ke-1 bernilai 4', 'Dadu ke-2 bernilai 4', 'Dadu ke-3 bernilai 4', 'Dadu ke-4 bernilai 6', and 'Total nilai dadu keseluruhan: 18'.

Gambar 1. Screenshot Hasil Jawaban Soal 1

## C. Pembahasan

Pada program ini, digunakan tiga kelas utama yaitu Main, LemparDadu, dan Dadu. Program ini mengimplementasikan enkapsulasi dan collection dengan tipe LinkedList untuk menyimpan beberapa objek dadu yang nilainya diacak secara dinamis oleh pengguna.

### 1. Kelas Main

Kelas Main berfungsi sebagai *entry point* program, yaitu tempat eksekusi dimulai. Baris `package modulTiga.PRAK201_2410817310014_IndraSuryadilaga;` digunakan untuk mendefinisikan lokasi file di dalam folder praktikum.

Objek `lemparDadu` dibuat dari kelas `LemparDadu`. Kemudian, tiga method utama dipanggil secara berurutan:

- `inputJumlahDadu()` : untuk menerima input jumlah dadu dari pengguna.
- `lemparDadu()` : untuk mengacak nilai setiap dadu dan menyimpannya ke dalam *collection*.
- `tampilkanHasil()` : untuk menampilkan total nilai dari semua dadu yang dilempar.

## 2. Kelas Dadu

Kelas Dadu merupakan representasi dari sebuah objek dadu. Atribut mataDadu disembunyikan dengan akses `private` agar tidak bisa diakses langsung dari luar kelas. Sebagai gantinya, digunakan *getter method* `getMataDadu()` untuk mengambil nilainya, dan *method* `acakNilai()` untuk menentukan nilainya.

Method `acakNilai()` menggunakan kelas `Random` dari pustaka `java.util` untuk menghasilkan angka acak antara 1 hingga 6. Setiap kali objek Dadu dibuat, method ini akan menentukan nilai acaknya. Dengan cara ini, setiap objek Dadu menyimpan nilai yang berbeda-beda tergantung hasil acakannya.

## 3. Kelas LemparDadu

```
private    LinkedList<Dadu>    daftarDadu    =    new
LinkedList<>();
```

 menunjukkan penggunaan **collection** bertipe `LinkedList` untuk menyimpan kumpulan objek Dadu. `LinkedList` dipilih karena bersifat dinamis, ukurannya bisa berubah sesuai jumlah input dari pengguna.

Method `inputJumlahDadu()` digunakan untuk menerima input dari pengguna menggunakan kelas `Scanner`. Perulangan `while (true)` memastikan input yang dimasukkan harus bernilai positif. Jika pengguna memasukkan angka nol atau negatif, maka program akan menampilkan pesan kesalahan dan meminta input ulang.

Method `lemparDadu()` program terlebih dahulu mereset nilai total dadu menjadi 0 dan mengosongkan koleksi `daftarDadu` agar tidak menyimpan data dari percobaan sebelumnya. Selanjutnya, program membuat sejumlah objek Dadu sesuai dengan jumlah yang dimasukkan pengguna. Setiap objek Dadu kemudian memanggil method `acakNilai()` untuk menghasilkan nilai acak antara 1 hingga 6. Setelah nilai setiap dadu diperoleh, program menampilkannya ke layar dan menjumlahkannya ke dalam variabel `totalNilai` untuk menghitung total keseluruhan nilai dari seluruh dadu yang dilempar.

Method `tampilkanHasil()` berfungsi untuk menampilkan jumlah total seluruh nilai dadu yang telah dilempar.

## 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 2.1 Soal No. 2*

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

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

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

#### A. Source Code

*Tabel 2.2 Soal2Main.java Source Code*

<b>File: MainSoal2.java</b>	
1	package
2	modulTiga.PRAK202_2410817310014_IndraSuryadilaga;
3	import java.util.*;
4	
5	public class Main {

```

6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9
10        int jumlahNegara;
11        boolean jumlahValid = false;
12        LinkedList<Negara> daftarNegara = new
13        LinkedList<>();
14        HashMap<Integer, String> namaBulan = new
15        HashMap<>();
16
17        namaBulan.put(1, "Januari");
18        namaBulan.put(2, "Februari");
19        namaBulan.put(3, "Maret");
20        namaBulan.put(4, "April");
21        namaBulan.put(5, "Mei");
22        namaBulan.put(6, "Juni");
23        namaBulan.put(7, "Juli");
24        namaBulan.put(8, "Agustus");
25        namaBulan.put(9, "September");
26        namaBulan.put(10, "Oktober");
27        namaBulan.put(11, "November");
28        namaBulan.put(12, "December");
29
30        //input jumlah negara dengan validasi
31        jumlah negara harus lebih dari 0
32        do {
33            jumlahNegara = input.nextInt();
34            if (jumlahNegara < 1) {
35                System.out.println("jumlah
36        negara yang anda masukan tidak valid, jumlah negara
37        harus lebih dari 0");
38                continue;
39            } else {
40                jumlahValid = true;
41            }
42            input.nextLine();
43        }while (!jumlahValid);
44
45        //input data lengkap negara
46        for (int i = 0; i < jumlahNegara; i++) {
47            //input nama, jenis kepemimpinan,
48            dan nama pemeimpin
49            String nama = input.nextLine();
50

```

```

51         String jenisKepemimpinan =
52         input.nextLine();
53         String namaPemimpin = input.nextLine();
54
55         //input tanggal, bulan, dan tahun
56         kemerdekaan
57         if
58         (jenisKepemimpinan.equalsIgnoreCase("monarki")) {
59             //Negara monarki tidak memiliki
60             tanggal kemerdekaan
61             daftarNegara.add(new Negara(nama,
62             jenisKepemimpinan, namaPemimpin));
63         } else {
64             //Perulangan untuk Validasi Data
65             Kemerdekaan
66             int tglKemerdekaan;
67             int blnKemerdekaan;
68             int thnKemerdekaan;
69             boolean tanggalValid = false;
70             do {
71                 tglKemerdekaan =
72                 input.nextInt();
73                 blnKemerdekaan =
74                 input.nextInt();
75                 thnKemerdekaan =
76                 input.nextInt();
77                 input.nextLine();
78
79                 if (blnKemerdekaan < 1 ||
80                 blnKemerdekaan > 12 || thnKemerdekaan <= 0) {
81                     System.out.println("Bulan
82                     atau Tahun tidak valid. Silakan masukkan kembali
83                     tahun kemerdekaan.");
84                     continue;
85                 }
86
87                 // Menentukan jumlah hari
88                 maksimum dalam sebulan (termasuk tahun kabisat)
89                 int maxTanggal;
90                 if (blnKemerdekaan == 2) {
91                     // Cek tahun kabisat (leap
92                     year)
93                     boolean isKabisat =
94                     (thnKemerdekaan % 4 == 0 && thnKemerdekaan % 100 !=
95                     0) || (thnKemerdekaan % 400 == 0);

```

```

96         maxTanggal = isKabisat ? 29
97 : 28;
98     } else if (blnKemerdekaan == 4
99 || blnKemerdekaan == 6 || blnKemerdekaan == 9 ||
100 blnKemerdekaan == 11) {
101         maxTanggal = 30;
102     } else {
103         maxTanggal = 31;
104     }
105
106         // Memeriksa apakah tanggal
107 yang dimasukkan sesuai dengan rentang bulan dan
108 tahun
109         if (tglKemerdekaan >= 1 &&
110 tglKemerdekaan <= maxTanggal) {
111             tanggalValid = true;
112         } else {
113             System.out.println("=>
114 ERROR: Tanggal " + tglKemerdekaan + " tidak ada
115 pada bulan ke-" + blnKemerdekaan + " tahun " +
116 thnKemerdekaan + ". Silakan masukkan kembali data
117 kemerdekaan.");
118         }
119
120     } while (!tanggalValid);
121
122     daftarNegara.add(new Negara(nama,
123 jenisKepemimpinan, namaPemimpin, tglKemerdekaan,
124 blnKemerdekaan, thnKemerdekaan));
125     }
126 }
127
128     //output data negara dengan perulangan.
129     System.out.println();
130     for (Negara negara : daftarNegara) {
131         if
132 (negara.getJenisKepemimpinan().equalsIgnoreCase("mo
133 narki")) {
134             negara.TampilkanInfo("");
135         } else {
136             String bulanNama =
137 namaBulan.get(negara.getBlnKemerdekaan());
138             negara.TampilkanInfo(bulanNama);
139         }
140     }

```



141	
142	input.close();
143	}
144	}

*Tabel 2.3 Negara.java Source Code*

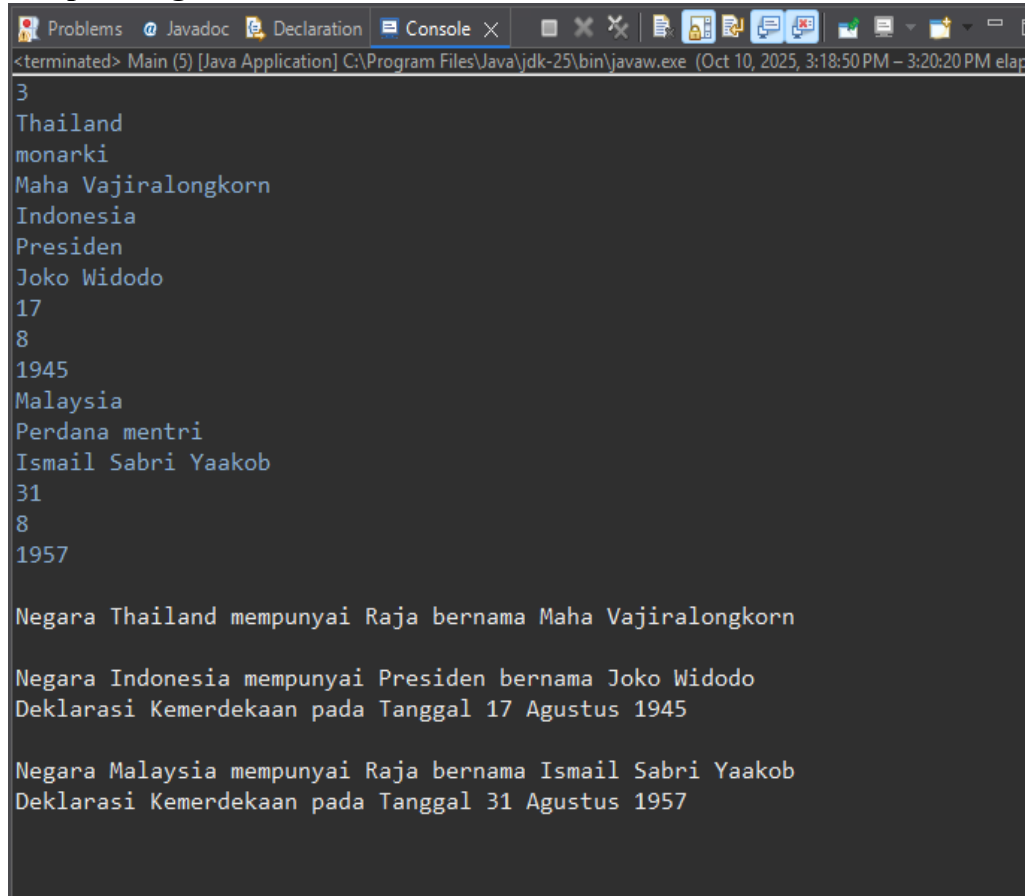
File: Negara.java	
1	package
2	modulTiga.PRAK202_2410817310014_IndraSuryadilaga;
3	
4	public class Negara {
5	//atribut
6	private String nama;
7	private String jenisKepemimpinan;
8	private String namaPemimpin;
9	private int tglKemerdekaan;
10	private int blnKemerdekaan;
11	private int thnKemerdekaan;
12	
13	//constructor <u>negara dengan kemerdekaan</u>
14	public Negara(String nama, String
15	jenisKepemimpinan, String namaPemimpin, int
16	tglKemerdekaan, int blnKemerdekaan, int
17	thnKemerdekaan) {
18	this.nama = nama;
19	this.jenisKepemimpinan =
20	jenisKepemimpinan;
21	this.namaPemimpin = namaPemimpin;
22	this.tglKemerdekaan = tglKemerdekaan;
23	this.blnKemerdekaan = blnKemerdekaan;
24	this.thnKemerdekaan = thnKemerdekaan;
25	}
26	
27	//constructor <u>negara tanpa kemerdekaan</u>
28	public Negara(String nama, String
29	jenisKepemimpinan, String namaPemimpin) {
30	this.nama = nama;
31	this.jenisKepemimpinan =
32	jenisKepemimpinan;
33	this.namaPemimpin = namaPemimpin;
34	}
35	
36	//Getter
37	public String getNama() {

```

38         return nama;
39     }
40
41     public String getJenisKepemimpinan() {
42         return jenisKepemimpinan;
43     }
44
45     public String getNamaPemimpin() {
46         return namaPemimpin;
47     }
48
49     public int getTglKemerdekaan() {
50         return tglKemerdekaan;
51     }
52
53     public int getBlnKemerdekaan() {
54         return blnKemerdekaan;
55     }
56
57     public int getThnKemerdekaan() {
58         return thnKemerdekaan;
59     }
60
61     //method menampilkan informasi negara
62     public void TampilkanInfo(String namaBulan) {
63         System.out.println("Negara " + nama + "
64 mempunyai "
65             +
66 (jenisKepemimpinan.equalsIgnoreCase("presiden") ?
67 "Presiden" : "Raja")
68             + " bernama " + namaPemimpin);
69
70         if
71 (!jenisKepemimpinan.equalsIgnoreCase("monarki")) {
72             System.out.println("Deklarasi
73 Kemerdekaan pada Tanggal "
74                 + tglKemerdekaan + " " +
75 namaBulan + " " + thnKemerdekaan);
76         }
77         System.out.println();
78     }
79
80 }

```

## A. Output Program

A screenshot of a Java IDE's console window. The window title bar shows 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console text is as follows:

```
<terminated> Main (5) [Java Application] C:\Program Files\Java\jdk-25\bin\javaw.exe (Oct 10, 2025, 3:18:50 PM – 3:20:20 PM elapsed)
3
Thailand
monarki
Maha Vajiralongkorn
Indonesia
Presiden
Joko Widodo
17
8
1945
Malaysia
Perdana menteri
Ismail Sabri Yaakob
31
8
1957

Negara Thailand mempunyai Raja bernama Maha Vajiralongkorn

Negara Indonesia mempunyai Presiden bernama Joko Widodo
Deklarasi Kemerdekaan pada Tanggal 17 Agustus 1945

Negara Malaysia mempunyai Raja bernama Ismail Sabri Yaakob
Deklarasi Kemerdekaan pada Tanggal 31 Agustus 1957
```

Gambar 2. Screenshot Hasil Jawaban Soal 2

## B. Pembahasan

Program ini terdiri dari dua kelas utama, yaitu Main dan Negara. Kelas Main berfungsi sebagai driver class atau kelas utama yang berisi method main() sebagai titik masuk (entry point) eksekusi program. Sedangkan kelas Negara berperan sebagai blueprint (cetakan objek) untuk menyimpan informasi lengkap tentang sebuah negara, seperti nama, jenis kepemimpinan, pemimpin, serta data kemerdekaan.

### 1. Kelas Negara

Kelas **Negara** memiliki enam atribut:

- a) nama: menyimpan nama negara.

- b) `jenisKepemimpinan`: menyimpan jenis sistem pemerintahan (misalnya *presiden* atau *monarki*).
- c) `namaPemimpin`: menyimpan nama pemimpin negara.
- d) `tglKemerdekaan`: menyimpan tanggal kemerdekaan.
- e) `blnKemerdekaan`: menyimpan bulan kemerdekaan dalam bentuk angka.
- f) `thnKemerdekaan`: menyimpan tahun kemerdekaan.

Atribut-atribut tersebut dibuat `private` untuk menerapkan prinsip enkapsulasi, yaitu pembatasan akses langsung dari luar kelas agar data lebih aman.

Selanjutnya terdapat dua constructor. Constructor pertama menerima enam parameter, digunakan untuk negara yang memiliki tanggal kemerdekaan (seperti *Indonesia*). Constructor kedua hanya memiliki tiga parameter, digunakan untuk negara dengan sistem *monarki* yang tidak memiliki tanggal kemerdekaan. setiap nilai parameter disimpan ke atribut melalui kata kunci `this`.

Selain itu, kelas ini juga menyediakan beberapa getter seperti `getNama()`, `getJenisKepemimpinan()`, dan `getTglKemerdekaan()` agar atribut tetap bisa diakses secara aman dari luar kelas.

Kemudian terdapat method penting bernama **`TampilkanInfo(String namaBulan)`**, yang berfungsi untuk menampilkan informasi lengkap tentang negara. Dengan pengkondisian Jika jenis kepemimpinan adalah **`monarki`**, maka hanya menampilkan nama negara, jenis kepemimpinan dan nama pemimpin. Namun jika bukan `monarki`, maka juga ditambahkan keterangan tanggal kemerdekaan

## 2. Kelas Main

Langkah pertama dalam method `main()` adalah membuat objek `Scanner` untuk membaca input dari pengguna, serta mendeklarasikan dua struktur data: `LinkedList` dan `HashMap`. Kode berikut digunakan:

- a) `Scanner input = new Scanner(System.in);`
- b) `LinkedList<Negara> daftarNegara = new LinkedList<>();`
- c) `HashMap<Integer, String> namaBulan = new HashMap<>();`

Setelah struktur data dibuat, program mengisi objek HashMap dengan pasangan *key-value* yang merepresentasikan Integer untuk urutan bulan dan String untuk nama bulan. Dengan cara ini, ketika pengguna memasukkan angka bulan seperti 8, program dapat secara otomatis menampilkan teks "Agustus" pada bagian output.

Selanjutnya, program meminta pengguna untuk memasukkan jumlah negara yang akan diinput `jumlahNegara = input.nextInt()`; Program kemudian melakukan validasi agar jumlah negara yang dimasukkan harus lebih dari 0. Jika pengguna memasukkan angka yang tidak valid, maka program akan menampilkan pesan kesalahan dan meminta input ulang. Proses validasi ini dilakukan menggunakan perulangan *do-while*, memastikan program tidak akan lanjut sebelum pengguna memberikan jumlah negara yang benar.

Setelah jumlah negara valid, program mulai meminta input data setiap negara melalui perulangan `for (int i = 0; i < jumlahNegara; i++) { ... }`. Dalam perulangan ini, pengguna diminta memasukkan tiga data pertama yaitu **nama negara**, **jenis kepemimpinan**, dan **nama pemimpin**. Jika jenis kepemimpinan yang dimasukkan adalah *monarki*, maka negara tersebut tidak memerlukan data tanggal kemerdekaan. Program langsung membuat objek negara menggunakan constructor yang hanya memiliki tiga parameter `new Negara(nama, jenisKepemimpinan, namaPemimpin)`.

Namun, jika jenis kepemimpinan bukan *monarki*, maka program meminta tambahan input berupa tanggal, bulan, dan tahun kemerdekaan. Input tersebut kemudian divalidasi agar sesuai dengan kalender, termasuk pengecekan tahun kabisat untuk bulan Februari. Setelah valid, objek negara dibuat menggunakan constructor dengan enam parameter `new Negara(nama, jenisKepemimpinan, namaPemimpin, tgl, bln, thn)`.

Setelah seluruh data negara berhasil dimasukkan, program menampilkan informasi dari masing-masing objek menggunakan perulangan `for-each` `for` `(Negara negara : daftarNegara) {...}` Dalam proses ini, program membedakan dua jenis output. Jika negara memiliki sistem *monarki*, maka hanya ditampilkan nama negara dan nama rajanya. Sebaliknya, jika negara memiliki sistem pemerintahan lain seperti *presiden* atau *perdana menteri*, maka akan ditampilkan juga tanggal, bulan dan tahun kemerdekaan.

### 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 3.1 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

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 3.2 Mahasiswa.java Source Code*

##### **File: Mahasiswa.java**

```
1 package
2 modulTiga.PRAK203_2410817310014_IndraSuryadilaga;
3
4 public class Mahasiswa {
5     //atribut
6     private String nama;
7     private String nim;
8
9     //constructor
10    Mahasiswa(String nama, String nim) {
11        this.nama = nama;
12        this.nim = nim;
13    }
14
15
16    //getter
17    public String getNama() {
18        return this.nama;
19    }
20
21    public String getNim() {
22        return this.nim;
23    }
24 }
```

Tabel 3.3 DaftarMahasiswa.java Source Code

File: DaftarMahasiswa.java	
1	package
2	modulTiga.PRAK203_2410817310014_IndraSuryadilaga;
3	
4	import java.util.ArrayList;
5	import java.util.Iterator;
6	import java.util.Scanner;
7	
8	public class DaftarMahasiswa {
9	Scanner input = new Scanner(System.in);
10	
11	ArrayList<Mahasiswa> daftarMahasiswa = new
12	ArrayList<>();
13	
14	//tambah Mahasiswa
15	public void tambahMahasiswa(String nama, String
16	nim) {
17	//nim harus unik
18	for (Mahasiswa m : daftarMahasiswa) {
19	if (m.getNim().equals(nim)) {
20	System.out.println("NIM sudah
21	digunakan! Gagal menambahkan mahasiswa.\n");
22	return;
23	}
24	}
25	daftarMahasiswa.add(new Mahasiswa(nama,
26	nim));
27	System.out.println("Mahasiswa " + nama + "
28	ditambahkan.\n");
29	}
30	
31	//hapus mahasiswa berdasar kan nim
32	public void hapusMahasiswa(String nim) {
33	Iterator<Mahasiswa> iterator =
34	daftarMahasiswa.iterator();
35	while (iterator.hasNext()) {
36	Mahasiswa m = iterator.next();
37	if (m.getNim().equals(nim)) {
38	iterator.remove();
39	System.out.println("Mahasiswa dengan
40	NIM " + nim + " dihapus.\n");
41	return;
42	}

```

43     }
44     System.out.println("Mahasiswa dengan NIM " +
45 nim + " tidak ditemukan.\n");
46     }
47
48     //mencari mahasiswa menggunakan nim
49     public void cariMahasiswa(String nim) {
50         for (Mahasiswa m : daftarMahasiswa) {
51             if (m.getNim().equals(nim)) {
52                 System.out.println("Data
53 Mahasiswa Ditemukan:");
54                 System.out.println("NIM: " +
55 m.getNim() + ", Nama: " + m.getNama() + "\n");
56                 return;
57             }
58         }
59         System.out.println("Mahasiswa dengan NIM "
60 + nim + " tidak ditemukan.\n");
61     }
62
63     // Menampilkan semua data mahasiswa
64     public void tampilkanSemua() {
65         if (daftarMahasiswa.isEmpty()) {
66             System.out.println("Daftar Mahasiswa
67 kosong.\n");
68             return;
69         }
70         System.out.println("Daftar Mahasiswa:");
71         for (Mahasiswa m : daftarMahasiswa) {
72             System.out.println("NIM: " + m.getNim()
73 + ", Nama: " + m.getNama());
74         }
75         System.out.println();
76     }
77
78     // Menghapus semua data (saat keluar)
79     public void hapusSemua() {
80         daftarMahasiswa.clear();
81     }
82 }

```

Tabel 3.4 Soal3Main.java Source Code

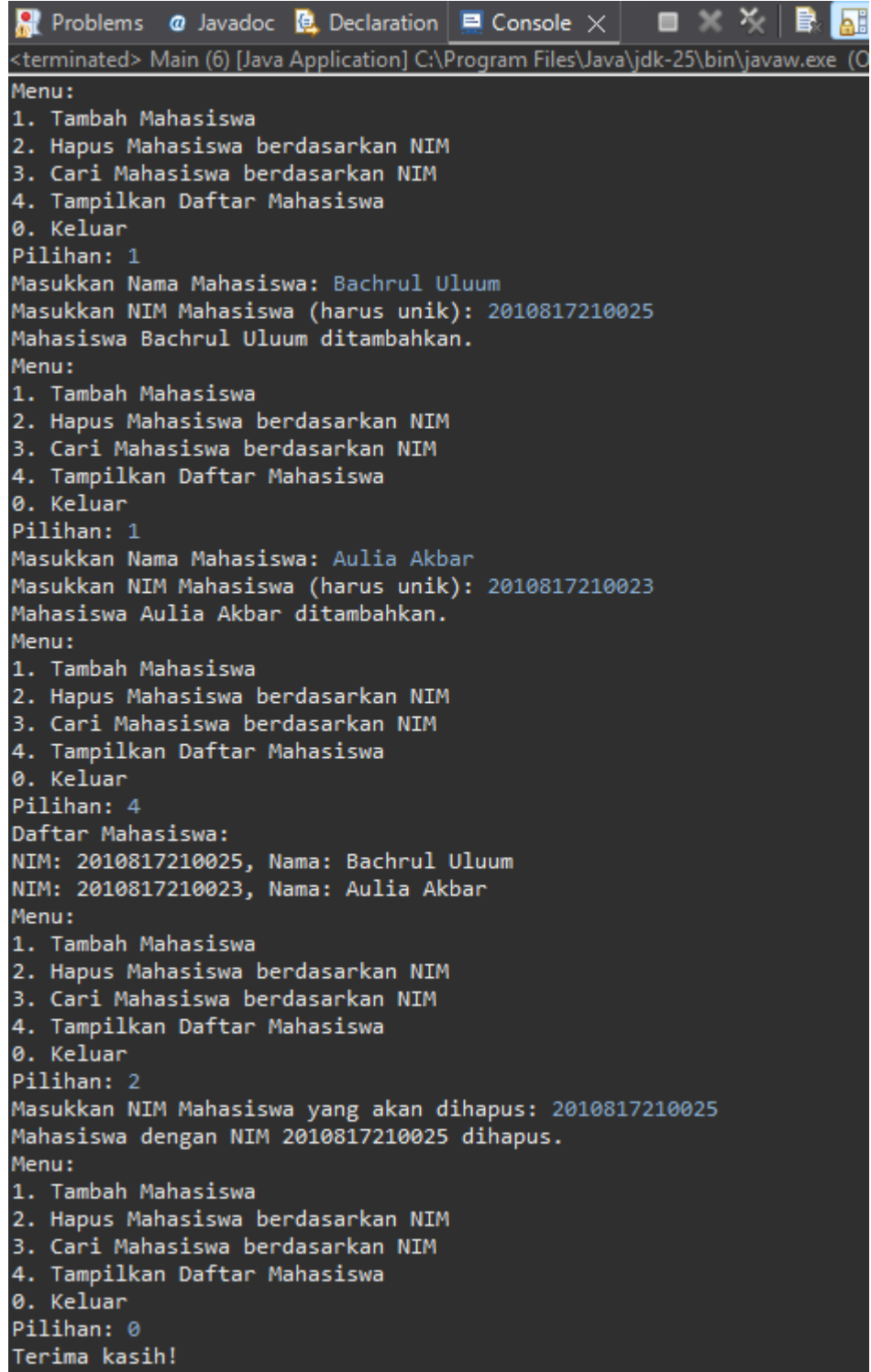
File: Soal3Main.java	
1	package
2	modulTiga.PRAK203_2410817310014_IndraSuryadilaga;
3	import java.util.Scanner;
4	
5	public class Main {
6	
7	public static void main(String[] args) {
8	Scanner input = new Scanner(System.in);
9	DaftarMahasiswa daftar = new
10	DaftarMahasiswa();
11	int pilihan;
12	
13	do {
14	System.out.println("Menu:");
15	System.out.println("1. Tambah
16	Mahasiswa");
17	System.out.println("2. Hapus Mahasiswa
18	berdasarkan NIM");
19	System.out.println("3. Cari Mahasiswa
20	berdasarkan NIM");
21	System.out.println("4. Tampilkan Daftar
22	Mahasiswa");
23	System.out.println("0. Keluar");
24	System.out.print("Pilihan: ");
25	pilihan = input.nextInt();
26	input.nextLine(); // <u>buang</u> newline
27	
28	switch (pilihan) {
29	case 1:
30	System.out.print("Masukkan Nama
31	Mahasiswa: ");
32	String nama = input.nextLine();
33	System.out.print("Masukkan NIM
34	Mahasiswa (harus unik): ");
35	String nim = input.nextLine();
36	daftar.tambahMahasiswa(nama,
37	nim);
38	break;
39	
40	case 2:
41	System.out.print("Masukkan NIM
42	Mahasiswa yang akan dihapus: ");

```

43         nim = input.nextLine();
44         daftar.hapusMahasiswa(nim);
45         break;
46
47         case 3:
48             System.out.print("Masukkan NIM
49 Mahasiswa yang dicari: ");
50             nim = input.nextLine();
51             daftar.cariMahasiswa(nim);
52             break;
53
54         case 4:
55             daftar.tampilkanSemua();
56             break;
57
58         case 0:
59             daftar.hapusSemua();
60             System.out.println("Terima
61 kasih!");
62             break;
63
64         default:
65             System.out.println("Pilihan
66 tidak valid.\n");
67             break;
68     }
69 } while (pilihan != 0);
70
71 input.close();
72 }
73 }
74 }

```

## B. Output Program



```
<terminated> Main (6) [Java Application] C:\Program Files\Java\jdk-25\bin\javaw.exe (C
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: Aulia Akbar
Masukkan NIM Mahasiswa (harus unik): 2010817210023
Mahasiswa 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: 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!
```

Gambar 3. Screenshot Hasil Jawaban Soal 3

### **C. Pembahasan**

Program ini terdiri dari dua kelas utama, yaitu Main dan Mahasiswa. Kelas Main berfungsi sebagai driver class atau kelas utama yang berisi method main() sebagai titik masuk eksekusi program. Sedangkan kelas Mahasiswa berperan sebagai blueprint untuk menyimpan informasi lengkap tentang seorang mahasiswa, seperti nama dan nim.

#### **1. Kelas Mahasiswa**

dibuat kelas bernama Mahasiswa yang berfungsi sebagai representasi dari satu entitas mahasiswa. Kelas ini menerapkan prinsip enkapsulasi, di mana atribut nama dan nim diset sebagai private, sehingga tidak dapat diakses langsung dari luar kelas. Kedua atribut ini diinisialisasi melalui konstruktor Mahasiswa(String nama, String nim) yang otomatis menyimpan data ketika objek mahasiswa dibuat.

Untuk mengakses nilai atribut tersebut, disediakan dua getter, yaitu getName() dan getNim(). Dengan cara ini, data mahasiswa lebih aman dan hanya dapat diakses melalui metode yang telah ditentukan.

#### **2. Kelas DaftarMahasiswa**

Kelas DaftarMahasiswa yang berfungsi untuk mengelola kumpulan objek mahasiswa. Di dalam kelas ini digunakan collection bertipe ArrayList<Mahasiswa> untuk menyimpan daftar mahasiswa secara dinamis. ArrayList dipilih karena dapat menambah, menghapus, dan mencari data dengan mudah tanpa batasan ukuran tertentu.

Kelas DaftarMahasiswa memiliki metode tambahMahasiswa(String nama, String nim) untuk menambahkan mahasiswa baru ke dalam ArrayList. Namun, sebelum data ditambahkan, terlebih dahulu memeriksa apakah NIM ada yang sama agar bersifat unik. Pemeriksaan dilakukan dengan melakukan iterasi terhadap seluruh data dalam daftar menggunakan perulangan for-each. Jika ditemukan NIM yang sama, program menampilkan pesan kesalahan dan proses penambahan dibatalkan. Jika tidak ada duplikasi, objek Mahasiswa baru dibuat dan dimasukkan ke dalam ArrayList.

Metode `hapusMahasiswa(String nim)` digunakan untuk menghapus data mahasiswa berdasarkan NIM. Proses ini memanfaatkan Iterator, yang memungkinkan penghapusan elemen dari `ArrayList` saat sedang diiterasi tanpa menyebabkan error. Jika NIM yang dicari ditemukan, objek tersebut dihapus dan pesan konfirmasi ditampilkan. Jika tidak ditemukan, program memberikan notifikasi bahwa mahasiswa dengan NIM tersebut tidak ada dalam daftar.

Metode `cariMahasiswa(String nim)` berfungsi menampilkan data mahasiswa berdasarkan NIM yang dimasukkan pengguna. Program melakukan pencarian linear di seluruh elemen `ArrayList` dan membandingkan NIM setiap objek dengan input pengguna. Jika cocok, maka program menampilkan informasi lengkap mahasiswa tersebut NIM dan Nama. Bila tidak ditemukan, sistem menampilkan pesan bahwa data mahasiswa tidak ada.

Metode `tampilkanSemua()` digunakan untuk menampilkan seluruh isi daftar mahasiswa. Jika daftar kosong, maka program menampilkan pesan bahwa tidak ada data yang tersimpan. Jika ada, maka setiap elemen dalam `ArrayList` dicetak ke layar dengan format: NIM: ..., Nama: ....

Sebelum program berakhir, terdapat metode `hapusSemua()` yang digunakan untuk menghapus seluruh isi `ArrayList` dengan memanggil fungsi `clear()`. Hal ini memastikan bahwa ketika program dijalankan kembali, data lama tidak tersisa dan `ArrayList` kembali dalam keadaan kosong.

### **3. Kelas Main**

Kelas Main menjadi titik masuk program. Di dalamnya dibuat objek `DaftarMahasiswa` dan menu interaktif berbasis do-while loop yang menampilkan lima opsi utama:

- a) Tambah Mahasiswa
- b) Hapus Mahasiswa berdasarkan NIM
- c) Cari Mahasiswa berdasarkan NIM
- d) Tampilkan Daftar Mahasiswa



e) Keluar dari program.

Pengguna memasukkan pilihan melalui Scanner. Berdasarkan nilai pilihan, program memanggil metode yang sesuai dari objek DaftarMahasiswa. Struktur switch-case digunakan untuk mengatur alur logika program dengan efisien. Setelah pengguna memilih opsi 0, program menutup scanner, menghapus seluruh data, dan menampilkan pesan “Terima kasih!” sebelum berhenti.

## **GITHUB**

[https://github.com/IndraSuryadilaga/Pemrograman\\_II](https://github.com/IndraSuryadilaga/Pemrograman_II)