

**LAPORAN PRAKTIKUM  
PEMROGRAMAN II  
MODUL 3**



**ENKAPSULASI & COLLECTION**

**Oleh:**

**Daniel Noprianto**

**NIM. 2410817110010**

**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 : Daniel Noprianto  
NIM : 2410817110010

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 .....	14
B. Output Program.....	20
C. Pembahasan.....	21
SOAL 3 .....	24
A. Source Code .....	26
B. Output Program.....	31
C. Pembahasan.....	32
TAUTAN GIT HUB .....	34

## **DAFTAR GAMBAR**

Gambar 1. Screenshot Hasil Jawaban Soal 1 .....	9
Gambar 2. Screenshot Hasil Jawaban Soal 2 .....	20
Gambar 3. Screenshot Hasil Jawaban Soal 3 .....	31

## DAFTAR TABEL

Tabel 1. Soal No. 1.....	6
Tabel 2. Source Code Soal 1 .....	7
Tabel 3. Source Code Soal 1 .....	7
Tabel 4. Soal No. 2.....	12
Tabel 5. Source Code Soal 2 .....	14
Tabel 6. Source Code Soal 2 .....	17
Tabel 7. Soal No. 3.....	24
Tabel 8. Source Code Soal 3 .....	26
Tabel 12. Source Code Soal 3 .....	27

## 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. 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 Soal 1*

File: Dadu.java	
1	package
	modulTiga.PRAK301_2410817110010_DanielNoprianto;
2	import java.util.Random;
3	
4	public class Dadu {
5	private int nilai;
6	
7	public Dadu() {
8	acakNilai();
9	}
10	
11	private void acakNilai() {
12	Random acak = new Random();
13	this.nilai = acak.nextInt(6) + 1;
14	}
15	
16	public int getNilai() {
17	return this.nilai;
18	}
19	}

*Tabel 3. Source Code Soal 1*

File: MainSoal1.java	
1	package
	modulTiga.PRAK301_2410817110010_DanielNoprianto;
2	
3	import java.util.LinkedList;
4	import java.util.Scanner;
5	
6	public class MainSoal1 {
7	
8	public static void main(String[] args) {
9	Scanner scan = new Scanner(System.in);
10	int jumlahDadu = 0;
11	
12	while (true) {

```

13         System.out.print("");
14
15         if (scan.hasNextInt()) {
16             jumlahDadu = scan.nextInt();
17             if (jumlahDadu > 0) {
18                 break;
19             } else {
20                 System.out.println("ERROR:
Jumlah dadu harus > 0");
21             }
22             } else {
23                 System.out.println("ERROR:      Input
HARUS sebuah angka!");
24                 scan.next();
25             }
26         }
27
28         System.out.println();
29
30         LinkedList<Dadu>    kumpulanDadu    =    new
LinkedList<>();
31         for (int i = 0; i < jumlahDadu; i++) {
32             kumpulanDadu.add(new Dadu());
33         }
34
35         int totalNilai = 0;
36         for (int i = 0; i < kumpulanDadu.size(); i++)
37         {
38             Dadu dadu = kumpulanDadu.get(i);
39             int nilai = dadu.getNilai();
40             System.out.println("Dadu ke-" + (i + 1)
+ " bernilai " + nilai);
41             totalNilai += nilai;
42         }
43
44         System.out.println("Total      nilai      dadu
keseluruhan " + totalNilai);
45         scan.close();
46     }
47

```



## B. Output Program

```
3
|
Dadu ke-1 bernilai 4
Dadu ke-2 bernilai 6
Dadu ke-3 bernilai 2
Total nilai dadu keseluruhan 12
```

*Gambar 1. Screenshot Hasil Jawaban Soal 1*

## C. Pembahasan

File `Dadu.java` berfungsi sebagai blueprint atau cetak biru dari objek dadu. Di dalamnya terdapat atribut nilai yang bertipe `int` dan dideklarasikan sebagai `private` agar tidak dapat diakses langsung dari luar class, sehingga sudah menerapkan konsep enkapsulasi. Ketika sebuah objek Dadu dibuat melalui konstruktor `public Dadu()`, program secara otomatis akan memanggil method `acakNilai()` yang bertugas mengisi atribut nilai dengan angka acak dari 1 sampai 6 menggunakan class `Random`. Metode `getNilai()` disediakan sebagai getter untuk mengakses nilai lemparan dadu dari luar class. Dengan struktur ini, setiap objek Dadu memiliki nilai acak yang unik saat dibuat.

File `MainSoal1.java` merupakan pusat logika utama dari program dan berfungsi untuk mengatur seluruh alur eksekusi mulai dari menerima input pengguna, membuat objek, menyimpan data, hingga menampilkan hasil akhir. Program dimulai dengan mengimpor library `java.util.Scanner` untuk membaca input dari pengguna dan `java.util.LinkedList` sebagai struktur data dinamis tempat menyimpan objek dadu. Kemudian di dalam method `main`, dibuat sebuah objek `Scanner` bernama `scan` dan variabel `jumlahDadu` bertipe `int` yang akan digunakan untuk menyimpan banyaknya dadu yang ingin dilempar oleh pengguna.

Langkah pertama program adalah meminta pengguna untuk memasukkan jumlah dadu. Proses ini dilengkapi dengan validasi input menggunakan perulangan `while (true)`. Program akan memeriksa apakah input yang diberikan adalah angka

dengan `scan.hasNextInt()`. Jika pengguna memasukkan selain angka, program akan menampilkan pesan “ERROR: Input HARUS sebuah angka!” dan mengulang proses input. Jika pengguna memasukkan angka tetapi nilainya kurang dari atau sama dengan nol, program juga akan menolak input tersebut dengan pesan “ERROR: Jumlah dadu harus > 0”. Penggunaan validasi ini bertujuan untuk memastikan program hanya menerima data yang benar sebelum melanjutkan proses berikutnya. Jika input valid, maka nilai tersebut disimpan ke dalam variabel `jumlahDadu` dan perulangan berhenti.

Setelah input jumlah dadu valid diperoleh, program membuat sebuah `LinkedList<Dadu>` bernama `kumpulanDadu`. Struktur data ini dipilih karena dapat menyimpan banyak objek Dadu secara dinamis sesuai jumlah yang dimasukkan pengguna. Program kemudian melakukan perulangan `for` sebanyak nilai `jumlahDadu`. Pada setiap iterasi, dibuat sebuah objek baru dari class Dadu dengan `new Dadu()`. Saat objek dibuat, konstruktor pada class Dadu secara otomatis mengacak nilai dadu dari 1 sampai 6. Objek yang sudah dibuat kemudian dimasukkan ke dalam `LinkedList` menggunakan method `add()`. Proses ini menghasilkan sekumpulan objek dadu dengan nilai acak yang siap ditampilkan.

Setelah semua objek dadu berhasil dibuat, program melanjutkan ke proses perhitungan dan penampilan hasil. Program membuat variabel `totalNilai` untuk menyimpan jumlah dari semua nilai dadu. Kemudian dijalankan perulangan `for` kedua, kali ini untuk mengakses setiap objek dadu yang sudah disimpan dalam `LinkedList`. Pada setiap iterasi, objek Dadu diambil menggunakan `get(i)` dan nilainya diperoleh melalui pemanggilan method `getNilai()`. Nilai setiap dadu kemudian dicetak ke layar dalam format “Dadu ke-i bernilai X” untuk memberikan informasi hasil lemparan satu per satu. Nilai tersebut juga ditambahkan ke dalam `totalNilai` untuk keperluan perhitungan total.

Setelah perulangan selesai, program mencetak hasil akhir berupa total nilai semua dadu yang telah dilempar dengan perintah `System.out.println("Total nilai dadu keseluruhan " + totalNilai);`. Ini menjadi ringkasan dari

seluruh lemparan dadu. Sebagai langkah terakhir, objek Scanner ditutup menggunakan `scan.close()` untuk mencegah kebocoran resource.

Secara keseluruhan, file `MainSoal1.java` menyusun alur program secara sistematis mulai dari input, validasi, pembuatan objek dadu, penyimpanan dalam struktur data dinamis, pengolahan nilai, hingga penampilan hasil akhir.

## 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. Soal No. 2*

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

1988
<b>Output</b>
<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>
<b>Input</b>
<p>3 Thailand monarki Maha Vajiralongkorn Indonesia presiden Joko Widodo 17 8 1945 Malaysia perdana menteri Ismail Sabri Yaakob 31 8 1957</p>
<b>Output</b>
<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 Soal 2

File: Negara.java	
1	package
	modulTiga.PRAK302_2410817110010_DanielNoprianto;
2	
3	public class Negara {
4	private String nama;
5	private String jenisKepemimpinan;
6	private String namaPemimpin;
7	private int tanggal;
8	private int bulan;
9	private int tahun;
10	private boolean monarki;
11	
12	// NonMonarki
13	public Negara(String nama, String
	jenisKepemimpinan, String namaPemimpin, int tanggal,
	int bulan, int tahun) {
14	this.nama = nama;
15	this.jenisKepemimpinan =
	jenisKepemimpinan;
16	this.namaPemimpin = namaPemimpin;
17	setTahun(tahun);
18	setBulan(bulan);
19	setTanggal(tanggal);
20	this.monarki = false;
21	}
22	
23	// Monarki
24	public Negara(String nama, String
	jenisKepemimpinan, String namaPemimpin) {
25	this.nama = nama;
26	this.jenisKepemimpinan =
	jenisKepemimpinan;
27	this.namaPemimpin = namaPemimpin;
28	this.monarki = true;
29	}
30	
31	// Tahun kabisat
32	private boolean TahunKabisat(int tahun) {
33	return (tahun % 400 == 0)    (tahun % 4 ==
	0 && tahun % 100 != 0);
34	}

```

34
35     // Maksimal hari dalam bulan
36     private int getMaximalHari(int bulan, int
37     tahun) {
38         switch (bulan) {
39             case 1: case 3: case 5: case 7: case
40             8: case 10: case 12:
41                 return 31;
42             case 4: case 6: case 9: case 11:
43                 return 30;
44             case 2:
45                 return TahunKabisat(tahun) ? 29
46                 : 28;
47             default:
48                 return 31;
49         }
50     }
51     // Set Tanggal
52     public boolean setTanggal(int tanggal) {
53         int maxHari = getMaximalHari(bulan,
54         tahun);
55         if (tanggal < 1 || tanggal > maxHari) {
56             System.out.println("ERROR: Tanggal
57             tidak valid untuk bulan " + bulan + " pada tahun " +
58             tahun + ".");
59             return false;
60         } else {
61             this.tanggal = tanggal;
62             return true;
63         }
64     }
65     // Set Bulan
66     public boolean setBulan(int bulan) {
67         if (bulan < 1 || bulan > 12) {
68             System.out.println("ERROR: Bulan
69             harus antara 1 - 12.");
70             return false;
71         } else {
72             this.bulan = bulan;
73             return true;
74         }
75     }

```

73	// Set Tahun	
74	public boolean setTahun(int tahun) {	
75	if (tahun <= 0    tahun > 2025) {	
76	System.out.println("ERROR: Tahun	
	tidak valid.");	
77	return false;	
78	} else {	
79	this.tahun = tahun;	
80	return true;	
81	}	
82	}	
83		
84	// Getter	
85	public String getNama() {	
86	return nama;	
87	}	
88		
89	public String getJenisKepemimpinan() {	
90	return jenisKepemimpinan;	
91	}	
92		
93	public String getNamaPemimpin() {	
94	return namaPemimpin;	
95	}	
96		
97	public int getTanggal() {	
98	return tanggal;	
99	}	
100		
101	public int getBulan() {	
102	return bulan;	
103	}	
104		
105	public int getTahun() {	
106	return tahun;	
107	}	
108		
109	public boolean isMonarki() {	
110	return monarki;	
111	}	
112	}	
113		



Tabel 6. Source Code Soal 2

File: MainSoal2.java	
1	package
	modulTiga.PRAK302_2410817110010_DanielNoprianto;
2	
3	import java.util.HashMap;
4	import java.util.LinkedList;
5	import java.util.Scanner;
6	
7	public class MainSoal2 {
8	
9	public static void main(String[] args) {
10	Scanner scan = new Scanner(System.in);
11	LinkedList<Negara> listNegara= new
	LinkedList<>();
12	HashMap<Integer, String> namaBulan = new
	HashMap<>();
13	
14	namaBulan.put(1, "Januari");
15	namaBulan.put(2, "Februari");
16	namaBulan.put(3, "Maret");
17	namaBulan.put(4, "April");
18	namaBulan.put(5, "Mei");
19	namaBulan.put(6, "Juni");
20	namaBulan.put(7, "Juli");
21	namaBulan.put(8, "Agustus");
22	namaBulan.put(9, "September");
23	namaBulan.put(10, "Oktober");
24	namaBulan.put(11, "November");
25	namaBulan.put(12, "Desember");
26	
27	int jumlahNegara = 0;
28	while(true) {
29	if(scan.hasNextInt()) {
30	jumlahNegara = scan.nextInt();
31	if (jumlahNegara > 0) {
32	break;
33	} else {
34	System.out.println("ERROR:
	Jumlah negara HARUS > 0");
35	}
36	} else {

```

37         System.out.println("ERROR:      Input
HARUS angka!");
38         scan.next();
39     }
40 }
41 scan.nextLine();
42
43     for(int i = 0; i < jumlahNegara; i++) {
44 //         Nama Negara
45         String nama = scan.nextLine();
46
47 //         Jenis Kepemimpinan
48         String jenis = scan.nextLine();
49
50 //         Nama Pemimpin
51         String pemimpin = scan.nextLine();
52
53         if (jenis.equalsIgnoreCase("monarki"))
54 {
55             listNegara.add(new      Negara(nama,
jenis, pemimpin));
56         } else {
57             int tgl = 0, bln = 0, thn = 0;
58
59             while (true) {
60 //                 Tanggal Kemerdekaan
61                 if (scan.hasNextInt()) {
62                     tgl = scan.nextInt();
63                     if (tgl >= 1 && tgl <= 31)
64 break;
65                 else
66 System.out.println("ERROR: Tanggal tidak valid.");
67             } else {
68                 System.out.println("ERROR:
Input tanggal harus angka.");
69                 scan.next();
70             }
71         }
72
73         while (true) {
74 //             Bulan Kemerdekaan
75             if (scan.hasNextInt()) {
76                 bln = scan.nextInt();
77                 if (bln >= 1 && bln <= 12)
78 break;

```

```

75         else
76         System.out.println("ERROR: Bulan tidak valid.");
77         } else {
78             System.out.println("ERROR:
79             Input bulan harus angka.");
80             scan.next();
81         }
82     }
83     while (true) {
84         // Tahun Kemerdekaan
85         if (scan.hasNextInt()) {
86             thn = scan.nextInt();
87             if (thn > 0) break;
88             else
89             System.out.println("ERROR: Tahun harus positif.");
90             } else {
91                 System.out.println("ERROR:
92                 Input tahun harus angka.");
93                 scan.next();
94             }
95             scan.nextLine();
96
97             listNegara.add(new Negara(nama,
98             jenis, pemimpin, tgl, bln, thn));
99         }
100     }
101     // Hasil
102     System.out.println("");
103     for (Negara negara : listNegara) {
104         String jabatan;
105         String jenisKepemimpinan =
106         negara.getJenisKepemimpinan();
107         if
108         (jenisKepemimpinan.equalsIgnoreCase("presiden")) {
109             jabatan = "Presiden";
110         } else if
111         (jenisKepemimpinan.equalsIgnoreCase("perdana
112         menteri")) {
113             jabatan = "Perdana Menteri";
114         } else {
115             jabatan = "Raja";

```

```

111         }
112
113         System.out.println("Negara " +
negara.getNama() + " mempunyai " + jabatan + "
bernama " + negara.getNamaPemimpin());
114
115         if (!negara.isMonarki()) {
116             System.out.println("Deklarasi
Kemerdekaan pada Tanggal " +
117                 negara.getTanggal() + " " +
namaBulan.get(negara.getBulan()) + " " +
negara.getTahun());
118         }
119
120         System.out.println();
121     }
122
123     scan.close();
124 }
125 }

```

## B. Output Program

```

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 Perdana Menteri bernama Ismail Sabri Yaakob
Deklarasi Kemerdekaan pada Tanggal 31 Agustus 1957

```

Gambar 2. Screenshot Hasil Jawaban Soal 2

### C. Pembahasan

File `Negara.java` ini berfungsi sebagai blueprint atau cetak biru objek yang akan digunakan dalam program utama. Dalam file ini, dibuat sebuah class bernama `Negara` yang merepresentasikan informasi dan karakteristik suatu negara, baik negara dengan sistem monarki maupun non-monarki. Class ini menerapkan atribut bersifat `private` sehingga hanya dapat diakses melalui method `getter` dan `setter`. Atribut yang disimpan mencakup `nama`, `jenisKepemimpinan`, `namaPemimpin`, `tanggal`, `bulan`, `tahun`, serta `monarki` yang bertipe `boolean` untuk membedakan apakah negara tersebut termasuk monarki atau tidak.

Class ini memiliki dua konstruktor. Konstruktor pertama digunakan untuk negara non-monarki, yang menerima parameter `nama negara`, `jenis kepemimpinan`, `nama pemimpin`, serta `tanggal`, `bulan`, dan `tahun kemerdekaan`. Pada konstruktor ini, nilai `tahun`, `bulan`, dan `tanggal` tidak langsung diset, melainkan melalui method `setTahun()`, `setBulan()`, dan `setTanggal()` untuk memastikan validasi data dilakukan. Konstruktor kedua digunakan untuk negara monarki, sehingga tidak membutuhkan input `tanggal kemerdekaan` dan langsung menandai atribut `monarki` sebagai `true`.

Selain itu, terdapat method bantu `TahunKabisat(int tahun)` yang digunakan untuk menentukan apakah suatu tahun merupakan tahun kabisat. Fungsi ini akan berpengaruh pada jumlah hari dalam bulan Februari. Kemudian ada method `getMaksimalHari(int bulan, int tahun)` yang mengembalikan jumlah maksimum hari dalam bulan tertentu, dengan memperhitungkan tahun kabisat.

File ini juga memiliki method `setter` dengan validasi: `setTanggal(int tanggal)` memastikan tanggal yang dimasukkan sesuai dengan jumlah maksimal hari pada bulan dan tahun yang dipilih. Jika tidak valid, maka pesan error akan ditampilkan. `setBulan(int bulan)` memastikan bulan berada dalam rentang 1–12. `setTahun(int tahun)` memastikan tahun lebih dari 0 dan tidak melebihi 2025. Selanjutnya, terdapat method `getter` untuk mengakses nilai atribut yang telah

disimpan secara aman. Penggunaan getter dan setter ini menjaga agar data tidak bisa dimodifikasi secara sembarangan dari luar class.

Selanjutnya file `MainSoal2.java` berfungsi sebagai program utama yang menjalankan seluruh logika proses input, validasi, penyimpanan data, dan output informasi negara. Program ini bekerja dengan cara menerima data negara dari pengguna, menyimpannya dalam sebuah struktur data, lalu menampilkannya kembali dengan format yang terstruktur.

Pada bagian awal program, dibuat objek `Scanner` untuk membaca input dari pengguna. Kemudian disiapkan sebuah `LinkedList<Negara>` untuk menampung objek negara yang diinputkan. Struktur data ini dipilih karena fleksibel dan mendukung penambahan elemen secara dinamis. Selain itu, terdapat sebuah `HashMap<Integer, String>` bernama `namaBulan` yang berfungsi untuk menghubungkan angka bulan (1–12) dengan nama bulan dalam bahasa Indonesia, seperti Januari, Februari, dan seterusnya. Ini memudahkan dalam menampilkan tanggal kemerdekaan negara dalam format yang lebih mudah dibaca.

Selanjutnya, program meminta pengguna untuk memasukkan jumlah negara yang akan diinputkan. Pada bagian ini terdapat validasi input, jika pengguna memasukkan angka kurang dari atau sama dengan nol, program akan menampilkan pesan error dan meminta input ulang. Jika pengguna memasukkan selain angka, program juga akan menolak input dan meminta ulang. Validasi ini memastikan bahwa jumlah negara yang diproses tidak akan bernilai salah atau menyebabkan error di tahap berikutnya.

Setelah jumlah negara valid diperoleh, program akan masuk ke dalam perulangan sebanyak jumlah negara tersebut. Dalam setiap iterasi, program akan meminta pengguna untuk memasukkan data negara: nama negara, jenis kepemimpinan (misalnya “presiden”, “perdana menteri”, atau “monarki”), dan nama pemimpin negara. Jika jenis kepemimpinan yang dimasukkan adalah “monarki”, maka program langsung membuat objek `Negara` menggunakan konstruktor `monarki` (tanpa tanggal kemerdekaan).

Namun, jika negara bukan monarki, maka pengguna diwajibkan memasukkan tanggal, bulan, dan tahun kemerdekaan. Pada setiap input tanggal, bulan, dan tahun,

program melakukan validasi ulang: tanggal harus antara 1–31, bulan harus antara 1–12, dan tahun harus lebih dari 0. Jika input tidak valid, pengguna diminta untuk mengulang hingga benar. Setelah semua valid, program membuat objek Negara dengan konstruktor non-monarki dan menambahkannya ke dalam `LinkedList`.

Setelah seluruh data negara dimasukkan, program masuk ke tahap output atau penampilan hasil. Program melakukan iterasi pada setiap objek Negara yang tersimpan dalam list. Untuk menampilkan jabatan pemimpin, program melakukan pengecekan jenis kepemimpinan: jika “presiden”, maka ditampilkan “Presiden”; jika “perdana menteri”, maka ditampilkan “Perdana Menteri”; jika tidak, maka ditampilkan “Raja”. Kemudian, ditampilkan nama negara dan nama pemimpin sesuai input.

Jika negara bukan monarki, maka ditambahkan pula informasi tanggal deklarasi kemerdekaan, dengan bulan yang ditampilkan dalam bentuk nama (bukan angka), menggunakan `HashMap` yang telah dibuat sebelumnya. Terakhir, setelah semua data ditampilkan, `Scanner` ditutup untuk mengakhiri proses input.

### 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  
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. 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 8. Source Code Soal 3*

File: Mahasiswa.java	
1	package
	modulTiga.PRAK303_2410817110010_DanielNoprianto;
2	
3	public class Mahasiswa {
4	private String nama;
5	private String nim;
6	
7	public Mahasiswa(String nama, String nim) {
8	this.nama = nama;
9	this.nim = <u>nim</u> ;
10	}
11	
12	public void setName(String nama) {
13	this.nama = nama;

14	}
15	
16	public String getNama() {
17	return nama;
18	}
19	
20	public void setNim(String nim) {
21	this.nim = nim;
22	}
23	
24	public String getNim() {
25	return nim;
26	}
27	
28	}
29	

*Tabel 9. Source Code Soal 3*

File: MainSoal3.java	
1	package
	modulTiga.PRAK303_2410817110010_DanielNoprianto;
2	
3	import java.util.ArrayList;
4	import java.util.Iterator;
5	import java.util.Scanner;
6	
7	public class MainSoal3 {
8	
9	public static void main(String[] args) {
10	Scanner input = new Scanner(System.in);
11	ArrayList<Mahasiswa> daftarMahasiswa = new
	ArrayList<>();
12	
13	int pilihan;
14	do {
15	System.out.println("Menu: ");
16	System.out.println("1.                   Tambah
	Mahasiswa");
17	System.out.println("2.                   Hapus
	Mahasiswa");
18	System.out.println("3.                   Cari
	Mahasiswa");
19	System.out.println("4.                   Tampilkan
	Daftar");

20	System.out.println("0. Keluar");
21	System.out.print("Pilihan: ");
22	
23	while (!input.hasNextInt()) {
24	System.out.println("ERROR: Input
25	harus angka!");
26	System.out.print("Pilihan: ");
27	input.next();
28	}
29	pilihan = input.nextInt();
30	input.nextLine();
31	switch (pilihan) {
32	case 1:
33	// Tambah Mahasiswa
34	System.out.print("Masukkan Nama
35	Mahasiswa: ");
36	String nama = input.nextLine();
37	System.out.print("Masukkan NIM
38	Mahasiswa (harus unik): ");
39	String nim = input.nextLine();
40	boolean sudahAda = false;
41	for (Mahasiswa m : daftarMahasiswa) {
42	if (m.getNim().equals(nim))
43	{
44	sudahAda = true;
45	break;
46	}
47	if (sudahAda) {
48	System.out.println("ERROR:
49	NIM sudah digunakan!");
50	} else {
51	daftarMahasiswa.add(new
52	Mahasiswa(nama, nim));
53	System.out.println("Mahasiswa " + nama + "
54	ditambahkan.");
55	} break;
	case 2:

56	// Hapus Mahasiswa	
57	System.out.print("Masukkan NIM	
	Mahasiswa yang akan dihapus: ");	
58	String hapusNim	=
	input.nextLine();	
59	boolean ditemukan = false;	
60		
61	Iterator<Mahasiswa> iterator =	
	daftarMahasiswa.iterator();	
62	while (iterator.hasNext()) {	
63	Mahasiswa m	=
	iterator.next();	
64	if	
	(m.getNim().equals(hapusNim)) {	
65	iterator.remove();	
66		
67	System.out.println("Mahasiswa dengan NIM " +	
	hapusNim + " dihapus.");	
	ditemukan = true;	
68	break;	
69	}	
70	}	
71		
72	if (!ditemukan) {	
73		
74	System.out.println("Mahasiswa dengan NIM " +	
	hapusNim + " tidak ditemukan.");	
	}	
75	break;	
76		
77	case 3:	
78	// Cari Mahasiswa	
79	System.out.print("Masukkan NIM	
	Mahasiswa yang dicari: ");	
80	String cariNim	=
	input.nextLine();	
81	boolean ketemu = false;	
82		
83	for (Mahasiswa m :	
	daftarMahasiswa) {	
84	if	
	(m.getNim().equals(cariNim)) {	
85	System.out.println("Data Mahasiswa Ditemukan:");	

86	System.out.println("NIM: " + m.getNim() + ", Nama: " + m.getNama());
87	ketemu = true;
88	break;
89	}
90	}
91	
92	if (!ketemu) {
93	System.out.println("Mahasiswa dengan NIM " + cariNim + " tidak ditemukan.");
94	}
95	break;
96	
97	case 4:
98	// Tampilkan Daftar
99	if (daftarMahasiswa.isEmpty())
100	{
101	System.out.println("Daftar Mahasiswa kosong.");
102	} else {
103	System.out.println("=== Daftar Mahasiswa ===");
104	for (Mahasiswa m : daftarMahasiswa) {
105	System.out.println("NIM: " + m.getNim() + ", Nama: " + m.getNama());
106	}
107	break;
108	
109	case 0:
110	daftarMahasiswa.clear();
111	System.out.println("Terima kasih!");
112	break;
113	
114	default:
115	System.out.println("ERROR: Pilihan tidak valid.");
116	}
117	
118	} while (pilihan != 0);

119	
120	input.close();
121	}
122	}

## B. Output Program

```

Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa
3. Cari Mahasiswa
4. Tampilkan Daftar
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
3. Cari Mahasiswa
4. Tampilkan Daftar
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
3. Cari Mahasiswa
4. Tampilkan Daftar
0. Keluar
Pilihan: 4
=== Daftar Mahasiswa ===
NIM: 2010817210025, Nama: Bachrul Uluum
NIM: 2010817210023, Nama: Muhammad Aulia Akbar
Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa
3. Cari Mahasiswa
4. Tampilkan Daftar
0. Keluar
Pilihan: 2
Masukkan NIM Mahasiswa yang akan dihapus: 2010817210025
Mahasiswa dengan NIM 2010817210025 dihapus.
Menu:
1. Tambah Mahasiswa
2. Hapus Mahasiswa
3. Cari Mahasiswa
4. Tampilkan Daftar
0. Keluar
Pilihan: 0
Terima kasih!

```

Gambar 3. Screenshot Hasil Jawaban Soal 3

### C. Pembahasan

File `Mahasiswa.java` berfungsi sebagai blueprint atau cetak biru dari objek yang digunakan pada program utama. Dalam file ini didefinisikan sebuah class bernama `Mahasiswa` yang merepresentasikan informasi dasar tentang satu mahasiswa. Class ini menyimpan dua atribut utama, yaitu `nama` dan `nim`, yang bertipe `String`. Kedua atribut tersebut dibuat dengan modifier akses `private` sehingga tidak dapat diakses langsung dari luar class.

Class ini memiliki satu buah konstruktor yang berfungsi untuk memberikan nilai awal pada atribut `nama` dan `nim` saat objek mahasiswa dibuat. Dengan demikian, setiap objek mahasiswa yang dibuat akan langsung memiliki data yang lengkap sejak awal. Selain itu, file ini juga memiliki method `getter` dan `setter`. `Getter` (`getNama()` dan `getNim()`) digunakan untuk mengambil nilai atribut, sedangkan `setter` (`setNama()` dan `setNim()`) digunakan untuk mengubah nilai atribut tersebut jika diperlukan. Dengan cara ini, pengelolaan data mahasiswa menjadi lebih terkontrol karena setiap perubahan nilai atribut harus melalui method `setter`, bukan dengan mengakses variabelnya secara langsung.

Sementara itu, file `MainSoal3.java` berfungsi sebagai program utama yang mengatur seluruh logika pengolahan data mahasiswa, mulai dari input pengguna, validasi data, penyimpanan ke dalam struktur data, hingga proses pencarian dan penghapusan data. Program ini menggunakan *Scanner* untuk menerima input dari pengguna dan `ArrayList<Mahasiswa>` untuk menyimpan kumpulan objek mahasiswa secara dinamis. Struktur data *ArrayList* dipilih karena fleksibel dan memungkinkan penambahan serta penghapusan data secara efisien tanpa harus menentukan ukuran tetap di awal program.

Di dalam program ini terdapat perulangan `do-while` yang berfungsi menampilkan menu utama secara berulang hingga pengguna memilih keluar (0). Menu tersebut terdiri atas empat fitur utama: menambah data, menghapus data, mencari data, dan menampilkan daftar mahasiswa. Sebelum mengeksekusi menu, program juga dilengkapi validasi input angka menggunakan



`hasNextInt()` , sehingga jika pengguna memasukkan karakter non-angka, program akan meminta input ulang dan tidak mengalami error.

Menu Tambah Mahasiswa meminta pengguna untuk menginput nama dan NIM mahasiswa. Program kemudian melakukan pengecekan apakah NIM yang dimasukkan sudah ada dalam daftar. Jika NIM tersebut sudah terdaftar, program akan menampilkan pesan error. Namun jika belum ada, program akan membuat objek baru Mahasiswa dan menambahkannya ke dalam *ArrayList*. Menu Hapus Mahasiswa meminta NIM, kemudian menggunakan *Iterator* untuk menelusuri daftar dan menghapus mahasiswa yang sesuai. Jika NIM tidak ditemukan, akan muncul pesan bahwa data tidak ditemukan. Menu Cari Mahasiswa memungkinkan pengguna mencari data berdasarkan NIM dan menampilkan informasi mahasiswa jika ditemukan. Menu Tampilkan Daftar Mahasiswa akan mencetak seluruh data mahasiswa yang tersimpan dalam *ArrayList*, atau memberi tahu jika daftar kosong. Ketika pengguna memilih Keluar, program akan menghapus seluruh data dan menutup proses input dengan *Scanner*.

Kedua file ini bekerja secara terintegrasi dan saling melengkapi. File `Mahasiswa.java` hanya berfungsi untuk mendefinisikan struktur data dan menyediakan mekanisme akses yang aman terhadap atribut melalui getter dan setter. Sedangkan `MainSoal3.java` bertugas sebagai pengendali logika dan interaksi dengan pengguna. Saat pengguna menambahkan data, program utama membuat objek Mahasiswa menggunakan konstruktor dari file blueprint. Ketika pengguna mencari atau menghapus data, program utama mengakses data mahasiswa tersebut menggunakan getter yang disediakan. Dengan cara ini, data mahasiswa terjaga keamanannya, sementara logika pengolahan data tetap fleksibel dan mudah dikembangkan.

## **TAUTAN GIT HUB**

[https://github.com/Daniel-N0/Pemrograman\\_II/tree/411e22d5fa2598aa30a4534b079a38cd4c7af8eb/modulTiga](https://github.com/Daniel-N0/Pemrograman_II/tree/411e22d5fa2598aa30a4534b079a38cd4c7af8eb/modulTiga)