

**LAPORAN PRAKTIKUM
PEMROGRAMAN II
MODUL 3**



ENKAPSULASI DAN COLLECTION

Oleh:

Ryan Muhammad Irfan NIM. 2210817310013

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
OKTOBER 2023**

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 : Ryan Muhammad Irfan
NIM : 2210817310013

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Bachrul Uluum
NIM. 2010817210025

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.....	8
C. Pembahasan.....	9
D. Tautan GIT.....	13
SOAL 2.....	14
A. Source Code	16
B. Output Program.....	19
C. Pembahasan.....	20
D. Tautan GIT.....	32
SOAL 3.....	33
A. Source Code	35
B. Output Program.....	38
C. Pembahasan.....	39
D. Tautan GIT.....	49

DAFTAR GAMBAR

Gambar 1. Screenshot Hasil Jawaban Soal 1 (1).....	8
Gambar 2. Screenshot Hasil Jawaban Soal 1 (2).....	8
Gambar 3. Screenshot Hasil Jawaban Soal 1 (3).....	9
Gambar 4. Screenshot Hasil Jawaban Soal 2 (1).....	19
Gambar 5. Screenshot Hasil Jawaban Soal 2 (2).....	20
Gambar 6. Screenshot Hasil Jawaban Soal 3	38

DAFTAR TABEL

Tabel 1. Soal 1 (Output)	6
Tabel 2. Source Code Soal 1(Dadu.java)	7
Tabel 3. Source Code Soal 1 (DaduMain.java)	8
Tabel 4. Soal 2 (Output)	15
Tabel 5. Source Code Soal 2 (Negara.java).....	17
Tabel 6. Source Code Soal 2 (NegaraMain.java)	18
Tabel 7. Soal 3 (Output)	35
Tabel 8. Source Code Soal 3 (Mahasiswa.java)	35
Tabel 9. Source Code Soal 3 (MahasiswaMain.java).....	37

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.

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

Tabel 1. Soal 1 (Output)

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

A. Source Code

Dadu.java	
1	package soal1;
2	import java.util.Random;
3	
4	public class Dadu {
5	private int nilai;
6	
7	public int getNilai() {
8	return nilai;
9	}
10	
11	public void acakNilai() {
12	Random nilaiRandom = new Random();
13	nilai = nilaiRandom.nextInt(6) + 1;
14	}
15	}

Tabel 2. Source Code Soal 1(Dadu.java)

DaduMain.java	
1	package soal1;
2	
3	import java.util.LinkedList;
4	import java.util.Scanner;
5	
6	public class DaduMain {
7	public static void main(String[] args) {
8	Scanner userinput = new Scanner(System.in);
9	LinkedList<Dadu> daduList = new LinkedList<>();
10	
11	int jumlahDadu = userinput.nextInt();
12	
13	for (int i = 1; i <= jumlahDadu; i++) {
14	Dadu dadu = new Dadu();
15	dadu.acakNilai();
16	daduList.add(dadu);
17	System.out.println("Dadu ke-" + i + "
18	bernilai " + dadu.getNilai());
19	}
20	
21	int totalNilai = 0;
22	for (Dadu dadu : daduList) {
23	totalNilai += dadu.getNilai();
24	}
25	
26	

27	System.out.println("Total nilai dadu keseluruhan
28	" + totalNilai);
	}
	}

Tabel 3. Source Code Soal 1 (DaduMain.java)

B. Output Program

```

Run    DaduMain x
C:\Users\Ryan muhammad irfan\.jdk\openjdk-11
3
Dadu ke-1 bernilai 3
Dadu ke-2 bernilai 6
Dadu ke-3 bernilai 1
Total nilai dadu keseluruhan 10
Process finished with exit code 0

```

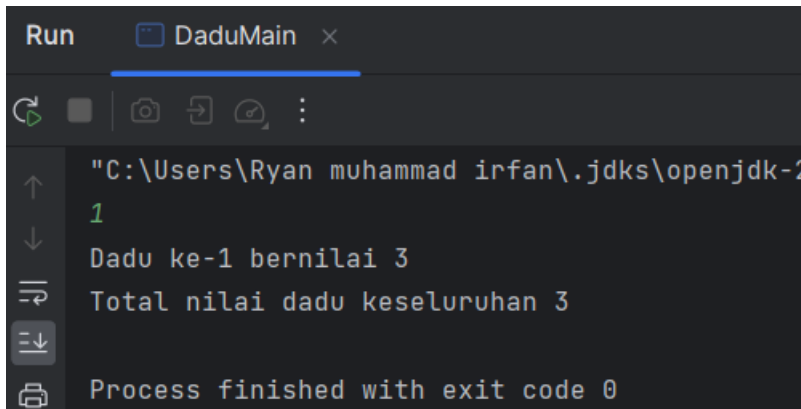
Gambar 1. Screenshot Hasil Jawaban Soal 1 (1)

```

Run    DaduMain x
C:\Users\Ryan muhammad irfan\.jdk\openjdk-11
4
Dadu ke-1 bernilai 6
Dadu ke-2 bernilai 3
Dadu ke-3 bernilai 3
Dadu ke-4 bernilai 4
Total nilai dadu keseluruhan 16
Process finished with exit code 0

```

Gambar 2. Screenshot Hasil Jawaban Soal 1 (2)



```
Run DaduMain x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-2
1
Dadu ke-1 bernilai 3
Total nilai dadu keseluruhan 3
Process finished with exit code 0
```

Gambar 3. Screenshot Hasil Jawaban Soal 1 (3)

C. Pembahasan

Dadu.java

Pada baris [1], `package soal1;`

“package soal1” berfungsi untuk mendeklarasikan package dengan nama soal1. Class Dadu berada dalam package soal1.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [2], `import java.util.Random;`

“import java.util.Random” berfungsi untuk mengimpor class ‘Random’ dari package ‘java.util’. Kelas ‘Random’ digunakan untuk menghasilkan nilai acak.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4-15], `public class Dadu`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama Dadu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [5], `private int nilai;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“int” merupakan tipe data dari atribut nilai.

“nilai” merupakan atribut dari sebuah class Dadu.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7-9], `public int getNilai() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“int” merupakan tipe data yang dikembalikan oleh method `getNilai()`.

“`getNilai()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘nilai’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [8], `return nilai;`

“`return nilai`” berfungsi untuk mengembalikan nilai atribut ‘nilai’ ketika method `getNilai ()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11-14], `public void acakNilai () {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“void” berfungsi menyatakan method tersebut tidak dapat mengembalikan nilai.

“`acakNilai ()`” merupakan sebuah method dengan nama `acakNilai`. Dapat dipanggil pada pada class lain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [12], `Random nilaiRandom = new Random();`

“`Random nilaiRandom = new Random()`” berfungsi untuk membuat objek baru dengan nama `nilaiRandom` dari kelas `Random`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13], `nilai = nilaiRandom.nextInt(6) + 1;`

“nilai” merupakan atribut untuk menyimpan objek `nilaiRandom` dengan method `.nextInt(6) + 1`.

“`nextInt(6) + 1`” berfungsi untuk menghasilkan nilai acak antara 0 hingga 5. Dengan menambahkan ‘+ 1’ maka akan menggeser nilai ke dalam rentang 1 hingga 6.

“;” berfungsi untuk mengakhiri sebuah perintah.

DaduMain.java

Pada baris [1], `package soal1;`

“package soal1” berfungsi untuk mendeklarasikan package dengan nama soal1. Class `DaduMain` berada dalam package soal1.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `import java.util.LinkedList;`

Mengimpor class `LinkedList` yang terdapat di dalam package `java.util`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `import java.util.Scanner;`

Mengimpor class scanner yang terdapat di dalam package `java.util`. Berfungsi untuk membaca input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6-28], `public class DaduMain`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama `DaduMain`.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [7-27], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [8], `Scanner userInput = new Scanner(System.in);`

Berfungsi untuk membuat sebuah objek dari class `Scanner` atau dalam hal ini variabel `userinput` akan berisi object dari `Scanner` class dengan cara memanggil konstruktor `Scanner(System.in)`. `System.in` adalah input stream yang digunakan untuk menerima input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9], `LinkedList<Dadu> daduList = new LinkedList<>();`

“`LinkedList<Dadu>`” berfungsi untuk mendeklarasikan linked list. `<Dadu>` merupakan tipe objeknya. Dalam hal ini linkedlist ‘`daduList`’ akan berisi objek-objek dari Class `Dadu`.

“`daduList = new LinkedList<>()`” berfungsi untuk inisialisasi dari linkedlist. `daduList` akan menjadi objek linked list yang digunakan untuk menyimpan objek-objek dari class `Dadu`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11], `int jumlahDadu = userInput.nextInt();`

“Int” merupakan tipe datanya, “jumlahDadu” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextInt()`. `<tipe data> <nama variabel> = <nilai variabel>`.

“userinput” merupakan object dari class `scanner` yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai Int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13-18], `for (int i = 1; i <= jumlahDadu; i++) {...}` berfungsi untuk melakukan perulangan dengan operasi tertentu sebanyak jumlahDadu. For (inisialisasi; kondisi; kenaikan).

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [14], `Dadu dadu = new Dadu();`

“Dadu” merupakan sebuah class.

“dadu” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Dadu.

“new Dadu” membuat objek baru dari kelas Dadu

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `dadu.acakNilai();`

“dadu.acakNilai()” berfungsi untuk memanggil method acakNilai pada objek “dadu” dari class Dadu. Digunakan untuk menghasilkan nilai acak pada objek dadu.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16], `daduList.add(dadu);`

“daduList.add(dadu)” berfungsi untuk menambahkan objek dad uke linkedlist ‘daduList’. Nilai acak pada objek dadu tersebut dimasukan ke dalam linkedlist sehingga dapat disimpan dan diakses.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17], `System.out.println("Dadu ke-" + i + " bernilai " + dadu.getNilai());`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Dadu ke-" + value i + “ bernilai “ + nilai dengan objek dadu diambil melalui method getter “getNilai()”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21], `int totalNilai = 0;`

“int” merupakan tipe datanya, “totalNilai” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya 0. <tipe data> <nama variabel> = <nilai variabel>.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22-24], `for (Dadu dadu : daduList){...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap elemen di dalam linkedlist. Digunakan untuk mengulangi elemen – elemen di dalam daduList hingga semua

elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek dadu di dalam daduList telah diakses satu persatu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [23], `totalNilai += dadu.getNilai();`

“`totalNilai += dadu.getNilai()`” berfungsi untuk menambahkan nilai dari setiap dadu di dalam linked list ‘daduList’ ke dalam variabel `totalNilai`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26], `System.out.println("Total nilai dadu keseluruhan " + totalNilai);`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Total nilai dadu keseluruhan " + value `totalNilai`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%203/src/soal1>

SOAL 2

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

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

Input
2 Indonesia presiden Joko Widodo 17 8 1945 Palestina presiden Mahmoud Abbas 15 11 1988
Output
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
Input
3 Thailand monarki Maha Vajiralongkorn Indonesia presiden Joko Widodo 17 8 1945 Malaysia perdana menteri Ismail Sabri Yaakob 31 8 1957
Output
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

Tabel 4. Soal 2 (Output)

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

A. Source Code

Negara.java	
1	package soal2;
2	
3	import java.util.HashMap;
4	
5	public class Negara {
6	private String namaNegara, jenisKepemimpinan,
	namaPemimpin;
7	private int tanggalKemerdekaan, bulanKemerdekaan,
	tahunKemerdekaan;
8	
9	public Negara(String namaNegara, String
	jenisKepemimpinan, String namaPemimpin, int
	tanggalKemerdekaan, int bulanKemerdekaan, int
	tahunKemerdekaan) {
10	this.namaNegara = namaNegara;
11	this.jenisKepemimpinan = jenisKepemimpinan;
12	this.namaPemimpin = namaPemimpin;
13	this.tanggalKemerdekaan = tanggalKemerdekaan;
14	this.bulanKemerdekaan = bulanKemerdekaan;
15	this.tahunKemerdekaan = tahunKemerdekaan;
16	}
17	
18	public String getNamaNegara() {
19	return namaNegara;
20	}
21	
22	public String getJenisKepemimpinan() {
23	return jenisKepemimpinan;
24	}
25	
26	public String getNamaPemimpin() {
27	return namaPemimpin;
28	}
29	
30	public int getTanggalKemerdekaan() {
31	return tanggalKemerdekaan;
32	}
33	
34	public int getBulanKemerdekaan() {
35	return bulanKemerdekaan;
36	}
37	
38	public int getTahunKemerdekaan() {
39	return tahunKemerdekaan;
40	}

41	
42	public String namaBulan(int bulanKemerdekaan) {
43	HashMap<Integer, String> namaBulan = new
44	HashMap<>();
	namaBulan.put(1, "Januari");
45	namaBulan.put(2, "Februari");
46	namaBulan.put(3, "Maret");
47	namaBulan.put(4, "April");
48	namaBulan.put(5, "Mei");
49	namaBulan.put(6, "Juni");
50	namaBulan.put(7, "Juli");
51	namaBulan.put(8, "Agustus");
52	namaBulan.put(9, "September");
53	namaBulan.put(10, "Oktober");
54	namaBulan.put(11, "November");
55	namaBulan.put(12, "Desember");
56	
57	return namaBulan.get(bulanKemerdekaan);
58	}
59	}

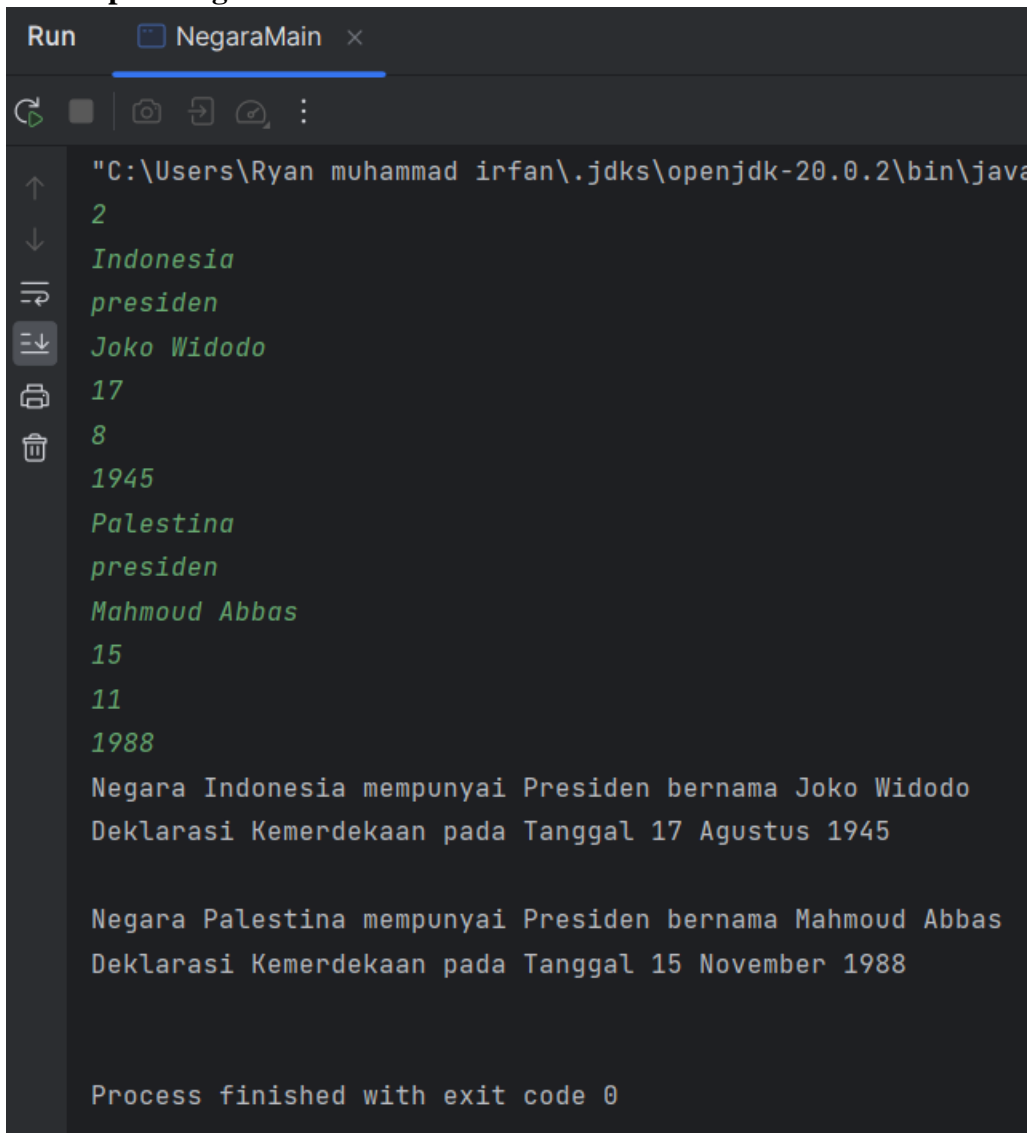
Tabel 5. Source Code Soal 2 (Negara.java)

NegaraMain.java	
1	package soal2;
2	
3	import java.util.LinkedList;
4	import java.util.Scanner;
5	
6	public class NegaraMain {
7	public static void main(String[] args) {
8	Scanner userInput = new Scanner(System.in);
9	int jumlahNegara = userInput.nextInt();
10	LinkedList<Negara> daftarNegara = new LinkedList<>();
11	userInput.nextLine();
12	
13	for (int i = 0; i < jumlahNegara; i++){
14	String namaNegara = userInput.nextLine();
15	String jenisKepemimpinan = userInput.nextLine();
16	String namaPemimpin = userInput.nextLine();
17	
18	if (jenisKepemimpinan.equalsIgnoreCase("monarki")) {
19	Negara N = new Negara(namaNegara, jenisKepemimpinan,
20	namaPemimpin, 0, 0, 0);
21	daftarNegara.add(N);
22	}
23	else {
24	int tanggalKemerdekaan = userInput.nextInt();
25	int bulanKemerdekaan = userInput.nextInt();
26	int tahunKemerdekaan = userInput.nextInt();
27	userInput.nextLine();
28	

29	Negara N = new Negara(namaNegara, jenisKepemimpinan,
30	namaPemimpin, tanggalKemerdekaan, bulanKemerdekaan,
31	tahunKemerdekaan);
32	daftarNegara.add(N);
33	}
34	}
35	for (Negara N : daftarNegara){
36	if
37	(N.getJenisKepemimpinan().equalsIgnoreCase("monarki")){
38	System.out.println("Negara " + N.getNamaNegara() + "
39	mempunyai Raja bernama " + N.getNamaPemimpin() + "\n");
40	}
41	else
42	(N.getJenisKepemimpinan().equalsIgnoreCase("presiden")) {
43	System.out.println("Negara " + N.getNamaNegara() + "
44	mempunyai Presiden bernama " + N.getNamaPemimpin());
45	System.out.println("Deklarasi Kemerdekaan pada
46	Tanggal " + N.getTanggalKemerdekaan() + " " +
47	N.namaBulan(N.getBulanKemerdekaan()) + " " + N.getTahunKemerdekaan()
48	+ "\n");
49	}
50	else
51	if(N.getJenisKepemimpinan().equalsIgnoreCase("perdana menteri")) {
52	System.out.println("Negara " + N.getNamaNegara() + "
53	mempunyai Perdana Menteri bernama " + N.getNamaPemimpin());
54	System.out.println("Deklarasi Kemerdekaan pada
55	Tanggal " + N.getTanggalKemerdekaan() + " " +
56	N.namaBulan(N.getBulanKemerdekaan()) + " " + N.getTahunKemerdekaan()
57	+ "\n");
58	}
59	}
60	}

Tabel 6. Source Code Soal 2 (NegaraMain.java)

B. Output Program

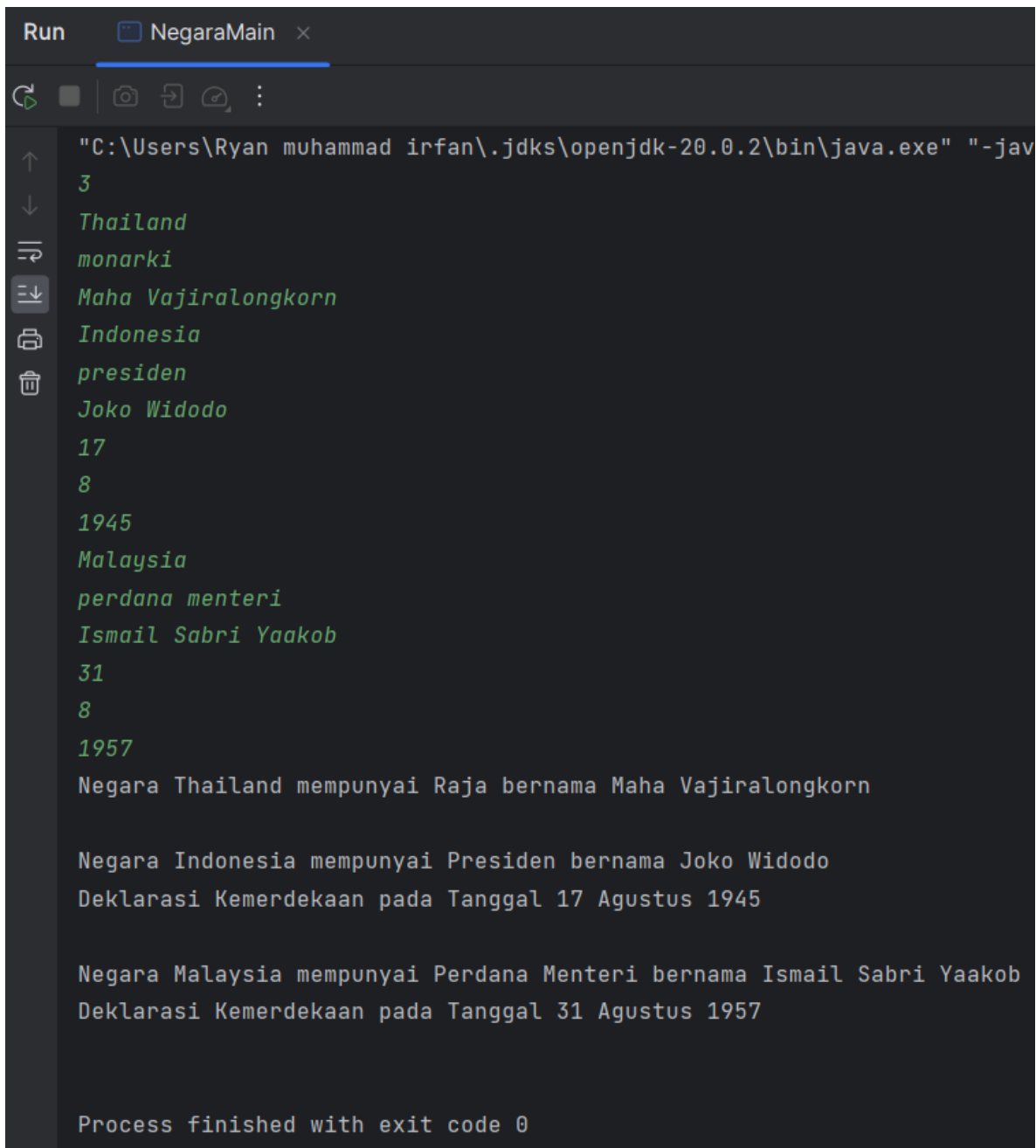


```
Run NegaraMain x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-20.0.2\bin\java
2
Indonesia
presiden
Joko Widodo
17
8
1945
Palestina
presiden
Mahmoud Abbas
15
11
1988
Negara Indonesia mempunyai Presiden bernama Joko Widodo
Deklarasi Kemerdekaan pada Tanggal 17 Agustus 1945

Negara Palestina mempunyai Presiden bernama Mahmoud Abbas
Deklarasi Kemerdekaan pada Tanggal 15 November 1988

Process finished with exit code 0
```

Gambar 4. Screenshot Hasil Jawaban Soal 2 (1)



```
Run NegaraMain x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-20.0.2\bin\java.exe" "-jav
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

Process finished with exit code 0
```

Gambar 5. Screenshot Hasil Jawaban Soal 2 (2)

C. Pembahasan

Negara.java

Pada baris [1], `package soal2;`

“`package soal2`” berfungsi untuk mendeklarasikan package dengan nama `soal2`. Class `Negara` berada dalam package `soal2`.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `import java.util.HashMap;`

“import java.util.HashMap” berfungsi untuk mengimpor class ‘HashMap’ dari package ‘java.util’.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [5-60], `public class Negara`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“Negara” berfungsi untuk membuat class yang dalam baris ini diberi nama Negara.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [6], `private String namaNegara, jenisKepemimpinan, namaPemimpin;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut namaNegara, jenisKepemimpinan, namaPemimpin.

“namaNegara, jenisKepemimpinan, namaPemimpin” merupakan atribut dari sebuah class Negara.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [7], `private int tanggalKemerdekaan, bulanKemerdekaan, tahunKemerdekaan;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“Int” merupakan tipe data dari atribut tanggalKemerdekaan, bulanKemerdekaan, tahunKemerdekaan.

“tanggalKemerdekaan, bulanKemerdekaan, tahunKemerdekaan” merupakan atribut dari sebuah class Negara.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9-16], `public Negara(String namaNegara, String jenisKepemimpinan, String namaPemimpin, int tanggalKemerdekaan, int bulanKemerdekaan, int tahunKemerdekaan) {...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“Negara(String namaNegara, String jenisKepemimpinan, String namaPemimpin, int tanggalKemerdekaan, int bulanKemerdekaan, int tahunKemerdekaan)” merupakan sebuah constructor dari class Negara. Constructor ini menerima parameter String namaNegara,

String jenisKepemimpinan, String namaPemimpin, int tanggalKemerdekaan, int bulanKemerdekaan, int tahunKemerdekaan.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [10], `this.namaNegara = namaNegara;`

“`this.namaNegara = namaNegara`” berfungsi untuk menginisialisasi atribut “namaNegara” dengan nilai yang diterima oleh parameter “namaNegara”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11], `this.jenisKepemimpinan = jenisKepemimpinan;`

“`this.jenisKepemimpinan = jenisKepemimpinan`” berfungsi untuk menginisialisasi atribut “jenisKepemimpinan” dengan nilai yang diterima oleh parameter “jenisKepemimpinan”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [12], `this.namaPemimpin = namaPemimpin;`

“`this.namaPemimpin = namaPemimpin`” berfungsi untuk menginisialisasi atribut “namaPemimpin” dengan nilai yang diterima oleh parameter “namaPemimpin”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13], `this.tanggalKemerdekaan = tanggalKemerdekaan;`

“`this.tanggalKemerdekaan = tanggalKemerdekaan`” berfungsi untuk menginisialisasi atribut “tanggalKemerdekaan” dengan nilai yang diterima oleh parameter “tanggalKemerdekaan”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `this.bulanKemerdekaan = bulanKemerdekaan;`

“`this.bulanKemerdekaan = bulanKemerdekaan`” berfungsi untuk menginisialisasi atribut “namaNegara” dengan nilai yang diterima oleh parameter “bulanKemerdekaan”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `this.tahunKemerdekaan = tahunKemerdekaan;`

“`this.tahunKemerdekaan = tahunKemerdekaan`” berfungsi untuk menginisialisasi atribut “tahunKemerdekaan” dengan nilai yang diterima oleh parameter “tahunKemerdekaan”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [18-20], `public String getNamaNegara() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getNamaNegara()`.

“`getNamaNegara()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘namaNegara’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [19], `return namaNegara;`

“return namaNegara” berfungsi untuk mengembalikan nilai atribut ‘namaNegara’ ketika method `getNamaNegara()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22-24], `public String getJenisKepemimpinan() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getJenisKepemimpinan()`.

“`getJenisKepemimpinan()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘jenisKepemimpinan’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [23], `return jenisKepemimpinan;`

“return jenisKepemimpinan” berfungsi untuk mengembalikan nilai atribut ‘jenisKepemimpinan’ ketika method `getJenisKepemimpinan()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26-28], `public String getNamaPemimpin() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getNamaPemimpin()`.

“`getNamaPemimpin()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘namaPemimpin’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [27], `return namaPemimpin;`

“return namaPemimpin” berfungsi untuk mengembalikan nilai atribut ‘namaPemimpin’ ketika method `getNamaPemimpin()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [30-32], `public int getTanggalKemerdekaan() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“int” merupakan tipe data yang dikembalikan oleh method `getTanggalKemerdekaan()`.

“`getTanggalKemerdekaan()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘tanggalKemerdekaan’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [31], `return tanggalKemerdekaan;`

“return tanggalKemerdekaan” berfungsi untuk mengembalikan nilai atribut ‘tanggalKemerdekaan’ ketika method `getTanggalKemerdekaan()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [34-36], `public int getBulanKemerdekaan() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“int” merupakan tipe data yang dikembalikan oleh method `getBulanKemerdekaan()`.

“`getBulanKemerdekaan()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘bulanKemerdekaan’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [35], `return bulanKemerdekaan;`

“`return bulanKemerdekaan`” berfungsi untuk mengembalikan nilai atribut ‘bulanKemerdekaan’ ketika method `getBulanKemerdekaan()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [38-40], `public int getTahunKemerdekaan() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“int” merupakan tipe data yang dikembalikan oleh method `getTahunKemerdekaan()`.

“`getTahunKemerdekaan()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘tahunKemerdekaan’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [39], `return tahunKemerdekaan;`

“`return tahunKemerdekaan`” berfungsi untuk mengembalikan nilai atribut ‘tahunKemerdekaan’ ketika method `getTahunKemerdekaan()` dipanggil.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [42], `public String namaBulan(int bulanKemerdekaan) {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari class lain.

“String” berfungsi menyatakan method tersebut bertipe data String.

“`namaBulan()`” merupakan sebuah method dengan nama `namaBulan`. Dapat dipanggil pada class lain.

“(int bulanKemerdekaan)” berfungsi untuk menerima satu parameter bertipe data integer dengan nama ‘bulanKemerdekaan’.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [43], `HashMap<Integer, String> namaBulan = new HashMap<>();`

“`HashMap<Integer, String>`” berfungsi untuk mendeklarasikan `HashMap` yang dapat menyimpan pasangan kunci dan nilai. Integer merupakan kunci dan String merupakan

nilainya. Dalam hal ini bilangan bulat sebagai kunci untuk mengakses namaBulan dalam bentuk String.

“namaBulan = new HashMap<>()” berfungsi untuk inisialisasi dari HashMap. namaBulan akan menjadi objek HashMap yang digunakan untuk menyimpan objek-objek..

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [44-55],

```
namaBulan.put(1, "Januari");  
namaBulan.put(2, "Februari");  
namaBulan.put(3, "Maret");  
namaBulan.put(4, "April");  
namaBulan.put(5, "Mei");  
namaBulan.put(6, "Juni");  
namaBulan.put(7, "Juli");  
namaBulan.put(8, "Agustus");  
namaBulan.put(9, "September");  
namaBulan.put(10, "Oktober");  
namaBulan.put(11, "November");  
namaBulan.put(12, "Desember");
```

“ namaBulan.put” berfungsi untuk menambahkan pasangan kunci-nilai ke dalam HashMap ‘namaBulan’. Dalam hal ini sebagai contoh (1, "Januari") = (kunci, nilai).

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [57], return namaBulan.get(bulanKemerdekaan);

“return namaBulan.get(bulanKemerdekaan)” berfungsi untuk mengembalikan nilai ‘namaBulan’ yang sesuai dengan bulanKemerdekaan yang diinputkan menggunakan metode get.

“;” berfungsi untuk mengakhiri sebuah perintah.

NegaraMain.java

Pada baris [1], package soal2;

“package soal2” berfungsi untuk mendeklarasikan package dengan nama soal2. Class NegaraMain berada dalam package soal2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], import java.util.LinkedList;

Mengimpor class LinkedList yang terdapat di dalam package java.util.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], import java.util.Scanner;

Mengimpor class scanner yang terdapat di dalam package java.util. Berfungsi untuk membaca input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6-47], `public class NegaraMain`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama NegaraMain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [7-46], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [8], `Scanner userInput = new Scanner(System.in);`

Berfungsi untuk membuat sebuah objek dari class Scanner atau dalam hal ini variabel userInput akan berisi object dari Scanner class dengan cara memanggil konstruktor Scanner(System.in). System in adalah input stream yang digunakan untuk menerima input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [9], `int jumlahNegara = userInput.nextInt();`

“Int” merupakan tipe datanya, “jumlahNegara” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai Int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [10], `LinkedList<Negara> daftarNegara = new LinkedList<>();`

“LinkedList<Negara>” berfungsi untuk mendeklarasikan linked list. <Negara> merupakan tipe objeknya. Dalam hal ini linkedlist ‘daftarNegara’ akan berisi objek-objek dari Class Negara.

“daftarNegara = new LinkedList<>()” berfungsi untuk inisialisasi dari linkedlist. daftarNegara akan menjadi objek linked list yang digunakan untuk menyimpan objek-objek dari class Negara.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11], `userinput.nextLine()` ;

Baris ini berfungsi untuk membersihkan baris baru yang masih tersisa dalam buffer masukan. Baris ini diperlukan karena sebelumnya `nextInt()` hanya bisa membaca angka, tidak bisa membaca karakter baris baru yang mungkin tersisa dalam buffer setelahnya.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13-31], `for (int i = 1; i <= jumlahNegara; i++){...}`

berfungsi untuk melakukan perulangan dengan operasi tertentu sebanyak jumlahNegara. `for` (inisialisasi; kondisi; kenaikan).

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [14], `String namaNegara = userinput.nextLine()` ;

“String” merupakan tipe datanya, “namaNegara” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `String jenisKepemimpinan = userinput.nextLine()` ;

“String” merupakan tipe datanya, “jenisKepemimpinan” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16], `String namaPemimpin = userinput.nextLine()` ;

“String” merupakan tipe datanya, “namaPemimpin” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextLine()” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [18-21],

```
if (jenisKepemimpinan.equalsIgnoreCase("monarki")) {...}
```

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“equalsIgnoreCase” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan, tanpa memperhatikan huruf besar dan huruf kecil

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [19], `Negara N = new Negara(namaNegara, jenisKepemimpinan, namaPemimpin, 0, 0, 0);`

“Negara” merupakan sebuah class.

“N ” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Negara.

“new Negara” membuat objek baru dari kelas Negara

“(namaNegara, jenisKepemimpinan, namaPemimpin, 0, 0, 0)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter namaNegara: value namaNegara, jenisKepemimpinan: value jenisKepemimpinan, namaPemimpin: value namaPemimpin, tanggalKemerdekaan: 0, bulanKemerdekaan: 0, tahunKemerdekaan: 0.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [20], `daftarNegara.add(N);`

“daftarNegara.add(N)” berfungsi untuk menambahkan objek N ke linkedlist ‘daftarNegara’.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [22-30], `else{...}`

“else” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya dari if sebelumnya salah/tidak terpenuhi.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [23], `int tanggalKemerdekaan = userInput.nextInt();`

“int” merupakan tipe datanya, “tanggalKemerdekaan” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [24], `int bulanKemerdekaan = userinput.nextInt();`

“int” merupakan tipe datanya, “bulanKemerdekaan” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [25], `int tahunKemerdekaan = userinput.nextInt();`

“int” merupakan tipe datanya, “tahunKemerdekaan” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26], `userinput.nextLine();`

Baris ini berfungsi untuk membersihkan baris baru yang masih tersisa dalam buffer masukan. Baris ini diperlukan karena sebelumnya `nextInt()` hanya bisa membaca angka, tidak bisa membaca karakter baris baru yang mungkin tersisa dalam buffer setelahnya.

Pada baris [28], `Negara N = new Negara(namaNegara, jenisKepemimpinan, namaPemimpin, tanggalKemerdekaan, bulanKemerdekaan, tahunKemerdekaan);`

“Negara” merupakan sebuah class.

“N” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Negara.

“new Negara” membuat objek baru dari kelas Negara

“(namaNegara, jenisKepemimpinan, namaPemimpin, tanggalKemerdekaan, bulanKemerdekaan, tahunKemerdekaan)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter
namaNegara: value namaNegara, jenisKepemimpinan: value jenisKepemimpinan,
namaPemimpin: value namaPemimpin, tanggalKemerdekaan: value tanggalKemerdekaan,
bulanKemerdekaan: value bulanKemerdekaan, tahunKemerdekaan: value tahunKemerdekaan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [29], `daftarNegara.add(N);`

“`daftarNegara.add(N)`” berfungsi untuk menambahkan objek N ke linkedlist ‘daftarNegara’.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [33-35], `for (Negara N : daftarNegara) {...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap elemen di dalam linkedlist. Digunakan untuk mengulangi elemen – elemen di dalam daftarNegara hingga semua elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek N di dalam daftarNegara telah diakses satu persatu.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [34-36],

`if (N.getJenisKepemimpinan().equalsIgnoreCase("monarki")) {...}`

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“`N.getJenisKepemimpinan()`” berfungsi untuk mengambil value jenisKepemimpinan dengan objek N diambil melalui method getter “`getJenisKepemimpinan()`”.

“`.equalsIgnoreCase`” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan, tanpa memperhatikan huruf besar dan huruf kecil

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [35], `System.out.println("Negara " + N.getNamaNegara() + " mempunyai Raja bernama " + N.getNamaPemimpin() + "\n");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Negara " + value namaNegara dengan objek N diambil melalui method getter “`getNamaNegara()`” + " mempunyai Raja bernama " + value namaPemimpin dengan objek N diambil melalui method getter “`getNamaPemimpin()`” + “`\n`”.

“`\n`” berfungsi untuk pindah ke baris berikutnya (newline).

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [37-40], `else if`

`(N.getJenisKepemimpinan().equalsIgnoreCase("presiden")) {`

“else if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisi dari if tidak terpenuhi.

“`N.getJenisKepemimpinan()`” berfungsi untuk mengambil value jenisKepemimpinan dengan objek N diambil melalui method getter “`getJenisKepemimpinan()`”.

“`.equalsIgnoreCase`” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan, tanpa memperhatikan huruf besar dan huruf kecil

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [38], `System.out.println("Negara " + N.getNamaNegara() + " mempunyai Presiden bernama " + N.getNamaPemimpin());`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Negara " + value namaNegara dengan objek N melalui method getter “`getNamaNegara()`” + " mempunyai Presiden bernama " + value namaPemimpin dengan objek N melalui method getter “`getNamaPemimpin()`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [39], `System.out.println("Deklarasi Kemerdekaan pada Tanggal " + N.getTanggalKemerdekaan() + " " + N.namaBulan(N.getBulanKemerdekaan()) + " " + N.getTahunKemerdekaan() + "\n");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Deklarasi Kemerdekaan pada Tanggal " + value tanggalKemerdekaan dengan objek N melalui method getter “`getTanggalKemerdekaan()`” + " " + value namaBulan dengan objek N melalui method “`getBulanKemerdekaan()`” + " " + value tahunKemerdekaan dengan objek N diambil melalui method “`getTahunKemerdekaan()`” + “`\n`”

“`\n`” berfungsi untuk pindah ke baris berikutnya (newline).

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [41-44], `else if (N.getJenisKepemimpinan().equalsIgnoreCase("perdana menteri")) {`

“`else if`” berfungsi sebagai percabangan yang akan dieksekusi jika kondisi dari `if` tidak terpenuhi.

“`N.getJenisKepemimpinan()`” berfungsi untuk mengambil value jenisKepemimpinan dengan objek N diambil melalui method getter “`getJenisKepemimpinan()`”.

“`.equalsIgnoreCase`” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan, tanpa memperhatikan huruf besar dan huruf kecil

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [42], `System.out.println("Negara " + N.getNamaNegara() + " mempunyai Perdana Menteri bernama " + N.getNamaPemimpin());`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Negara " + value namaNegara dengan objek N melalui method getter “`getNamaNegara()`” + " mempunyai Perdana Menteri bernama " + value namaPemimpin dengan objek N melalui method getter “`getNamaPemimpin()`”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [43], `System.out.println("Deklarasi Kemerdekaan pada Tanggal " + N.getTanggalKemerdekaan() + " " +`

```
N.namaBulan(N.getBulanKemerdekaan()) + " " +  
N.getTahunKemerdekaan() + "\n");
```

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Deklarasi Kemerdekaan pada Tanggal " + value tanggalKemerdekaan dengan objek N melalui method getter “getTanggalKemerdekaan()” + " " + value namaBulan dengan objek N melalui method “getBulanKemerdekaan()” + " " + value tahunKemerdekaan dengan objek N diambil melalui method “getTahunKemerdekaan()” + "\n"

“\n” berfungsi untuk pindah ke baris berikutnya (newline).

“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%203/src/soal2>

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

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!

Tabel 7. Soal 3 (Output)

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

A. Source Code

Mahasiswa.java	
1	package soal3;
2	
3	public class Mahasiswa {
4	private String nama, nim;
5	
6	public Mahasiswa(String nama, String nim) {
7	this.nama = nama;
8	this.nim = nim;
9	}
10	
11	public String getNama() {
12	return nama;
13	}
14	
15	public String getNim() {
16	return nim;
17	}
18	}

Tabel 8. Source Code Soal 3 (Mahasiswa.java)

MahasiswaMain.java

```
1 package soal3;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class MahasiswaMain {
7     public static void main(String[] args) {
8         Scanner userInput = new Scanner(System.in);
9         ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();
10
11         while(true) {
12             System.out.println("Menu:");
13             System.out.println("1. Tambah Mahasiswa");
14             System.out.println("2. Hapus Mahasiswa berdasarkan NIM");
15             System.out.println("3. Cari Mahasiswa berdasarkan NIM");
16             System.out.println("4. Tampilkan Daftar Mahasiswa");
17             System.out.println("0. Keluar");
18             System.out.print("Pilihan: ");
19             int opsi = userInput.nextInt();
20
21             switch (opsi) {
22                 case 1:
23                     System.out.print("Masukkan Nama Mahasiswa: ");
24                     userInput.nextLine();
25                     String nama = userInput.nextLine();
26                     System.out.print("Masukkan NIM Mahasiswa (harus
unik): ");
27                     String nim = userInput.nextLine();
28
29                     boolean nimUnik = true;
30                     for (Mahasiswa mahasiswa : daftarMahasiswa) {
31                         if(mahasiswa.getNim().equals(nim)){
32                             nimUnik = false;
33                             break;
34                         }
35                     }
36                     if(nimUnik){
37                         Mahasiswa mhs = new Mahasiswa(nama, nim);
38                         daftarMahasiswa.add((mhs));
39                         System.out.println("Mahasiswa " + nama + "
ditambahkan.");
40                     }
41                     else {
42                         System.out.println("Mahasiswa dengan NIM " + nim
+ " sudah ada" );
43                     }
44                     break;
45
46                 case 2:
47                     System.out.print("Masukkan NIM Mahasiswa yang akan
dihapus: ");
48                     String hapusNim = userInput.next();
49                     for (Mahasiswa mahasiswa : daftarMahasiswa){
```

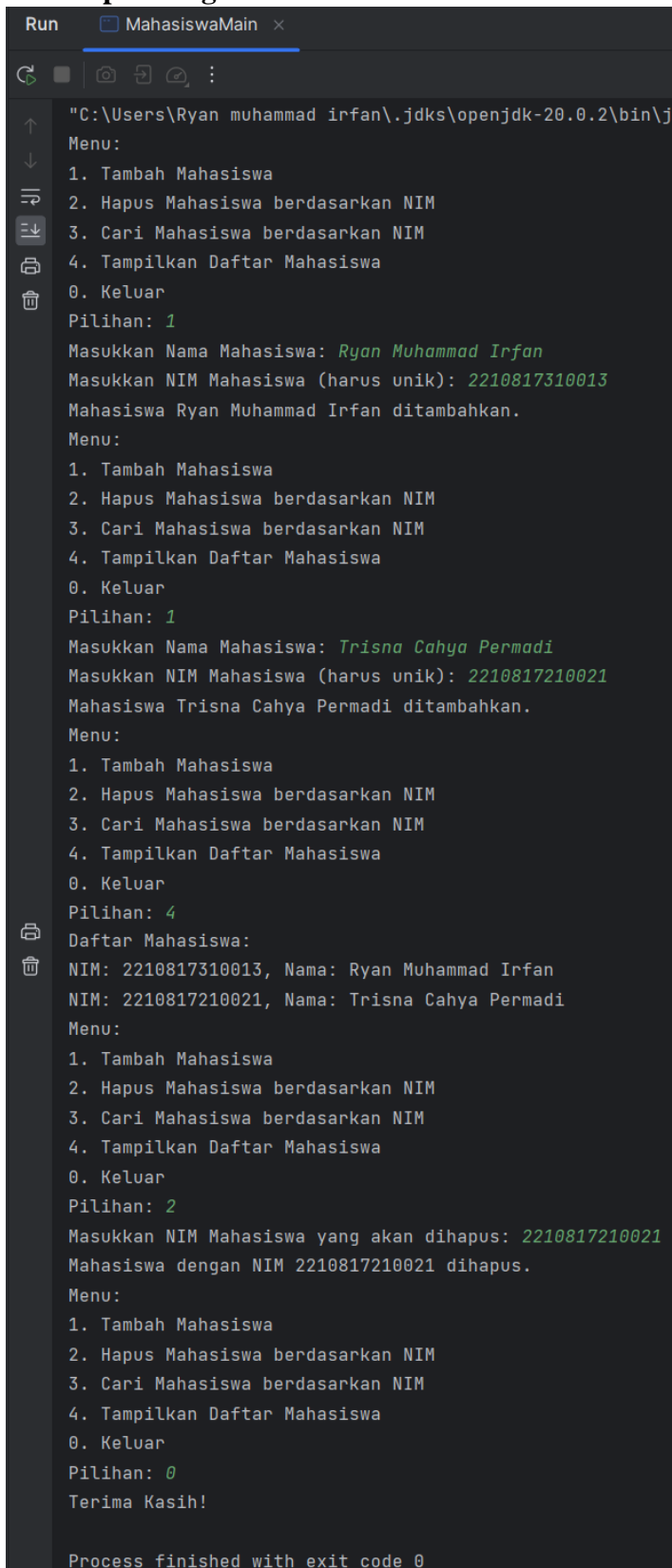
```

50         if (mahasiswa.getNim().equals(hapusNim)) {
51             daftarMahasiswa.remove(mahasiswa);
52             System.out.println("Mahasiswa dengan NIM " +
hapusNim + " dihapus.");
53             break;
54         }
55     }
56     break;
57
58     case 3:
59         System.out.print("Masukan NIM Mahasiswa yang akan
dicari: ");
60         String cariNim = userInput.next();
61         for (Mahasiswa mahasiswa : daftarMahasiswa) {
62             if (mahasiswa.getNim().equals(cariNim)) {
63                 daftarMahasiswa.remove(mahasiswa);
64                 System.out.println("Mahasiswa dengan NIM " +
cariNim + " atas nama " + mahasiswa.getNama() + " ditemukan.");
65                 break;
66             }
67             else {
68                 System.out.println("Mahasiswa dengan NIM " +
cariNim + " tidak ditemukan.");
69             }
70         }
71         break;
72
73     case 4:
74         for (Mahasiswa mahasiswa : daftarMahasiswa) {
75             System.out.println("NIM: " + mahasiswa.getNama()
+ ", Nama: " + mahasiswa.getNim());
76         }
77         break;
78
79     case 0:
80         System.out.println("Terima Kasih!");
81         System.exit(0);
82         break;
83
84     default:
85         System.out.println("Pilihan tidak valid, silahkan
pilih lagi.");
86     }
87 }
88 }
89 }

```

Tabel 9. Source Code Soal 3 (MahasiswaMain.java)

B. Output Program



```
Run MahasiswaMain x
"C:\Users\Ryan muhammad irfan\.jdk\openjdk-20.0.2\bin\j
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: Ryan Muhammad Irfan
Masukkan NIM Mahasiswa (harus unik): 2210817310013
Mahasiswa Ryan Muhammad Irfan 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: Trisna Cahya Permadi
Masukkan NIM Mahasiswa (harus unik): 2210817210021
Mahasiswa Trisna Cahya Permadi 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: 2210817310013, Nama: Ryan Muhammad Irfan
NIM: 2210817210021, Nama: Trisna Cahya Permadi
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: 2210817210021
Mahasiswa dengan NIM 2210817210021 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!

Process finished with exit code 0
```

Gambar 6. Screenshot Hasil Jawaban Soal 3

C. Pembahasan

Mahasiswa.java

Pada baris [1], `package soal3;`

“package soal3” berfungsi untuk mendeklarasikan package dengan nama soal3. Class Mahasiswa berada dalam package soal3.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3-19], `public class Mahasiswa`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“Mahasiswa” berfungsi untuk membuat class yang dalam baris ini diberi nama Mahasiswa.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [4], `private String nama, nim;`

“private” berfungsi sebagai penanda bahwa atribut dapat memiliki akses terbatas. Hanya bisa diakses dari dalam class itu sendiri.

“String” merupakan tipe data dari atribut nama, nim.

“nama, nim” merupakan atribut dari sebuah class Mahasiswa.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6-9], `public Mahasiswa(String nama, String nim) {...}`

“public” berfungsi sebagai penanda bahwa constructor dapat diakses dari class lain.

“Mahasiswa(String nama, String nim)” merupakan sebuah constructor dari class Mahasiswa. Constructor ini menerima parameter String nama, String nim.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [7], `this.nama = nama;`

“this.nama = nama” berfungsi untuk menginisialisasi atribut “nama” dengan nilai yang diterima oleh parameter “nama”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [8], `this.nim = nim;`

“this.nim = nim” berfungsi untuk menginisialisasi atribut “nim” dengan nilai yang diterima oleh parameter “nim”.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11-13], `public String getNama() {...}`

“public” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getNama ()`.

“`getNama()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘nama’.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [12], `return nama;`

“`return nama`” berfungsi untuk mengembalikan nilai atribut ‘nama’ ketika method `getNama()` dipanggil.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15-17], `public String getNim() {...}`

“`public`” berfungsi sebagai penanda bahwa method ini dapat diakses dari luar class.

“String” merupakan tipe data yang dikembalikan oleh method `getNim ()`.

“`getNim()`” merupakan method getter. Digunakan untuk mengambil nilai dari atribut ‘nim’.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [16], `return nim;`

“`return nama`” berfungsi untuk mengembalikan nilai atribut ‘nim’ ketika method `getNim()` dipanggil.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

MahasiswaMain.java

Pada baris [1], `package soal3;`

“`package soal3`” berfungsi untuk mendeklarasikan package dengan nama soal3. Class MahasiswaMain berada dalam package soal2.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [3], `import java.util.ArrayList;`

Mengimpor class ArrayList yang terdapat di dalam package java.util.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [4], `import java.util.Scanner;`

Mengimpor class scanner yang terdapat di dalam package java.util. Berfungsi untuk membaca input dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [6-89], `public class MahasiswaMain`

“`public`” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“class” berfungsi untuk membuat class yang dalam baris ini diberi nama MahasiswaMain.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [7-88], `public static void main(String[] args)`

“public” berfungsi sebagai penanda bahwa class, objek, method, atau atribut dapat diakses dari class lain.

“static” berfungsi membuat suatu method tanpa perlu melakukan instansiasi terlebih dahulu.

“void” berfungsi untuk tidak mengembalikan nilai apapun.

“main” merupakan nama fungsi yang digunakan oleh java sebagai awal masuk ke program.

“String[] args” berfungsi sebagai parameter yang diperlukan oleh fungsi main. Parameter ini adalah array dari argument perintah yang bisa diteruskan ke program java.

Pada baris [8], `Scanner userInput = new Scanner(System.in);`

Berfungsi untuk membuat sebuah objek dari class Scanner atau dalam hal ini variabel userInput akan berisi object dari Scanner class dengan cara memanggil konstruktor Scanner(System.in). System in adalah input stream yang digunakan untuk menerima input dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [11-87], `while (true) {...}`

“while” merupakan sebuah perulangan yang mengulang sebuah blok kode selama kondisi yang ditentukan benar. “true” merupakan sebuah kondisi.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [12], `System.out.println("Menu:");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Menu:".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [13], `System.out.println("1. Tambah Mahasiswa");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "1. Tambah Mahasiswa".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [14], `System.out.println("2. Hapus Mahasiswa berdasarkan NIM");`

“System.out.println(...)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "2. Hapus Mahasiswa berdasarkan NIM".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [15], `System.out.println("3. Cari Mahasiswa berdasarkan NIM");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "3. Cari Mahasiswa berdasarkan NIM".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [16], `System.out.println("4. Tampilkan Daftar Mahasiswa");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "4. Tampilkan Daftar Mahasiswa".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [17], `System.out.println("0. Keluar");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "0. Keluar".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [18], `System.out.println("Pilihan: ");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "0. Keluar Pilihan: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [19], `int opsi = userInput.nextInt();`

“Int” merupakan tipe datanya, “opsi” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextInt()`. <tipe data> <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextInt()” berfungsi untuk membaca nilai Int dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [21-86], `switch (opsi){...}`

“switch” merupakan percabangan dengan membandingkan nilai sebuah variabel dengan beberapa nilai. Jika proses perbandingan tersebut menghasilkan true, maka blok code akan di eksekusi. (opsi) merupakan nama variabel.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [22], `case 1:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai opsi = 1 maka blok kode akan di eksekusi.

Pada baris [23], `System.out.print("Masukkan Nama Mahasiswa: ");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Masukkan Nama Mahasiswa: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [24], `userinput.nextLine()` ;

Baris ini berfungsi untuk membersihkan baris baru yang masih tersisa dalam buffer masukan. Baris ini diperlukan karena sebelumnya `nexInt()` hanya bisa membaca angka, tidak bisa membaca karakter baris baru yang mungkin tersisa dalam buffer setelahnya.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [25], `String nama = userinput.nextLine()` ;

“String” merupakan tipe datanya, “nama” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [26], `System.out.print("Masukkan NIM Mahasiswa (harus unik) : ")` ;

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Masukkan NIM Mahasiswa (harus unik): ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [27], `String nim= userinput.nextLine()` ;

“String” merupakan tipe datanya, “nim” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userinput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [29], `boolean nimUnik=true` ;

“boolean” merupakan tipe datanya, “nimUnik” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `true`. <tipe data> <nama variabel> = <nilai variabel>.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [30-35], `for (Mahasiswa mahasiswa : daftarMahasiswa){...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap elemen di dalam ArrayList. Digunakan untuk mengulangi elemen – elemen di dalam `daftarMahasiswa` hingga

semua elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek mahasiswa di dalam daftarMahasiswa telah diakses satu persatu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [31-34], `if (mahasiswa.getNim().equals(nim)) {...}`

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“mahasiswa.getNim()” berfungsi untuk mengambil value nim dengan objek mahasiswa diambil melalui method getter “getNim()”.

“equals” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan,

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [32], `nimUnik = false;`
Menyatakan value dari nimUnik yaitu false.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [33], `break;`
Memberhentikan loop yang dijalankan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [36-40], `if (nimUnik) {...}`

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“nimUnik” berfungsi sebagai kondisi. Dalam hal ini kondisinya value dari nimUnik.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [37], `Mahasiswa mhs = new Mahasiswa(nama, nim);`

“mhs” merupakan nama variabel. Digunakan untuk merujuk ke objek yang akan dibuat dengan menggunakan class Mahasiswa.

“new Mahasiswa” membuat objek baru dari kelas Mahasiswa.

“(nama, nim)” berfungsi untuk mengisi value atribut dari parameter yang sudah ditentukan. Pada baris ini mengisi value atribut dengan parameter nama: value nama, nim: value nim.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [38], `daftarMahasiswa.add(mhs);`

“daftarMahasiswa.add(mhs)” berfungsi untuk menambahkan objek mhs ke ArrayList ‘daftarMahasiswa’.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [39], `System.out.println("Mahasiswa " + nama + " ditambahkan.");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini `"Mahasiswa " + value nama + "` ditambahkan.”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [41-43], `else{...}`

“`else`” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya dari `if` sebelumnya salah/tidak terpenuhi.

“`{...}`” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [42], `System.out.println("Mahasiswa dengan NIM " + nim + " sudah ada");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini `" Mahasiswa dengan NIM " + value nim + "` sudah ada”.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [44], `break;`

Memberhentikan case 1.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [46], `case 2:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai opsi = 2 maka blok kode akan dieksekusi.

Pada baris [47], `System.out.print("Masukkan NIM Mahasiswa yang akan dihapus: ");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak `"Masukkan NIM Mahasiswa yang akan dihapus: "`.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [48], `String hapusNim= userInput.nextLine();`

“`String`” merupakan tipe datanya, “`hapusNim`” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userinput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“`userinput`” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“`nextLine()`” berfungsi untuk membaca nilai `String` dari pengguna.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [49-55], `for (Mahasiswa mahasiswa : daftarMahasiswa){...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap elemen di dalam ArrayList. Digunakan untuk mengulangi elemen – elemen di dalam daftarMahasiswa hingga semua elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek mahasiswa di dalam daftarMahasiswa telah diakses satu persatu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [50-54], `if (mahasiswa.getNim().equals(hapusNim)) {...}`

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“mahasiswa.getNim()” berfungsi untuk mengambil value nim dengan objek mahasiswa diambil melalui method getter “getNim()”.

“equals” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan,

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [51], `daftarMahasiswa.remove((mahasiswa));`

“daftarMahasiswa.remove((mahasiswa))” berfungsi untuk menghapus objek mahasiswa dari ArrayList ‘daftarMahasiswa’.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [52], `System.out.println("Mahasiswa dengan NIM " + hapusNim + " dihapus.");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini "Mahasiswa dengan NIM " + value hapusNim + " dihapus".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [53], `break;`

Memberhentikan loop yang dijalankan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [56], `break;`

Memberhentikan case 2.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [58], `case 3:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai opsi = 3 maka blok kode akan di eksekusi.

Pada baris [59], `System.out.print("Masukan NIM Mahasiswa yang akan dicari: ");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Masukan NIM Mahasiswa yang akan dicari: ".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [60], `String cariNim= userInput.nextLine();`

“String” merupakan tipe datanya, “cariNim” merupakan variabel yang diberi sebuah penamaan, Lalu setelah itu adalah valuenya `userInput.nextLine()`. <tipe data> <nama variabel> = <nilai variabel>.

“userInput” merupakan object dari class scanner yang berfungsi mengambil input dari pengguna.

“nextLine()” berfungsi untuk membaca nilai String dari pengguna.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [61-69], `for (Mahasiswa mahasiswa : daftarMahasiswa){...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap element di dalam ArrayList. Digunakan untuk mengulangi elemen – elemen di dalam daftarMahasiswa hingga semua elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek mahasiswa di dalam daftarMahasiswa telah diakses satu persatu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [62-65], `if (mahasiswa.getNim().equals(cariNim)) {...}`

“if” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya benar. Jika salah maka tidak akan dieksekusi atau akan berlanjut ke percabangan else if atau else.

“mahasiswa.getNim()” berfungsi untuk mengambil value nim dengan objek mahasiswa diambil melalui method getter “getNim()”.

“equals” merupakan sebuah method yang membandingkan sebuah string apakah ada kesamaan,

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [63], `System.out.println("Mahasiswa dengan NIM "+ cariNim + " atas nama " + mahasiswa.getNama() + " ditemukan.");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini "Mahasiswa dengan NIM "+ value cariNim + " atas nama " + value nama dengan objek mahasiswa diambil melalui method “getNama()” + " ditemukan.”.

Pada baris [64], `break;`

Memberhentikan loop yang dijalankan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [66-68], `else{...}`

“else” berfungsi sebagai percabangan yang akan dieksekusi jika kondisinya dari if sebelumnya salah/tidak terpenuhi.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [67], `System.out.println("Mahasiswa dengan NIM "+ cariNim + " tidak ditemukan.");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini " Mahasiswa dengan NIM "+ value cariNim + " tidak ditemukan.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [70], `break;`

Memberhentikan case 3.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [72], `case 4:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai opsi = 4 maka blok kode akan di eksekusi.

Pada baris [73], `System.out.print("Daftar Mahasiswa:");`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Daftar Mahasiswa:".

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [74-76], `for (Mahasiswa mahasiswa : daftarMahasiswa){...}`

Ini merupakan sebuah for-each yang berfungsi untuk mengulangi setiap element di dalam ArrayList. Digunakan untuk mengulangi elemen – elemen di dalam daftarMahasiswa hingga semua elemen telah diakses. Pengulangan akan terus berlanjut sampai semua objek mahasiswa di dalam daftarMahasiswa telah diakses satu persatu.

“{...}” berfungsi untuk memulai dan mengakhiri blok kode.

Pada baris [75], `System.out.println("NIM: " + mahasiswa.getNim() + ", Nama: " + mahasiswa.getNama());`

“`System.out.println(...)`” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak “NIM: ” + value nim dengan objek mahasiswa diambil melalui method getter “`getNim()`” + “, Nama: ” + value nama dengan objek mahasiswa diambil melalui method getter “`getNama()`”.

Pada baris [77], `break;`

Memberhentikan case 4.

“`;`” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [79], `case 0:`

Merupakan nilai yang diperiksa. Dalam hal ini jika nilai opsi = 0 maka blok kode akan di eksekusi.

Pada baris [80], `System.out.print("Terima Kasih!");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak " Terima Kasih!:".

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [81], `System.exit(0);`

Memberhentikan eksekusi program java. Program akan berhasil dihentikan.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [82], `break;`

Memberhentikan case 0.

“;” berfungsi untuk mengakhiri sebuah perintah.

Pada baris [84], `default:`

Apabila tidak ada case yang sesuai maka default yang akan di eksekusi.

Pada baris [85], `System.out.println("Pilihan tidak valid, silahkan pilih lagi.");`

“System.out.println(…)” berfungsi untuk mencetak angka atau karakter dengan pindah ke baris berikutnya. Dalam baris ini mencetak "Pilihan tidak valid, silahkan pilih lagi.".

“;” berfungsi untuk mengakhiri sebuah perintah.

D. Tautan GIT

<https://github.com/ryanmi04/Praktikum-Pemrograman-2-Paralel-1/tree/main/PRAKTIKUM%203/src/soal3>