**LAPORAN PRAKTIKUM**

**PEMROGRAMAN II**

**MODUL 3**

****

**ENKAPSULASI & COLLECTION**

**Oleh:**

**Indra Suryadilaga NIM. 2410817310014**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**OKTOBER 2025**

# LEMBAR PENGESAHAN

**LAPORAN PRAKTIKUM PEMROGRAMAN II**

**MODUL 3**

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

Nama Praktikan : Indra Suryadilaga

NIM : 2410817310014

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Galih Aji Sabdaraya  NIM. 2310817210005 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Andreyan Rizky Baskara, S.Kom., M.Kom.  NIP. 19930703 201903 1 011 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc210844338)

[DAFTAR ISI 3](#_Toc210844339)

[DAFTAR GAMBAR 4](#_Toc210844340)

[DAFTAR TABEL 5](#_Toc210844341)

[SOAL 1 6](#_Toc210844342)

[A. Source Code 7](#_Toc210844343)

[B. Output Program 9](#_Toc210844344)

[C. Pembahasan 9](#_Toc210844345)

[SOAL 2 12](#_Toc210844346)

[A. Source Code 13](#_Toc210844347)

[A. Output Program 19](#_Toc210844348)

[B. Pembahasan 19](#_Toc210844349)

[SOAL 3 23](#_Toc210844350)

[A. Source Code 25](#_Toc210844351)

[B. Output Program 30](#_Toc210844352)

[C. Pembahasan 31](#_Toc210844353)

[GITHUB 34](#_Toc210844354)

# DAFTAR GAMBAR

[Gambar 1. Screenshot Hasil Jawaban Soal 1 9](#_Toc210687247)

[Gambar 2. Screenshot Hasil Jawaban Soal 2 19](#_Toc210687248)

[Gambar 3. Screenshot Hasil Jawaban Soal 3 30](#_Toc210687249)

# DAFTAR TABEL

[Tabel 1.1 Soal No. 1 6](#_Toc211028698)

[Tabel 1.2 MainSoal1.java Source Code 7](#_Toc211028699)

[Tabel 1.3 Dadu.java Source Code 7](#_Toc211028700)

[Tabel 1.4 LemeparDadu.java Source Code 7](#_Toc211028701)

[Tabel 2.1 Soal No. 2 12](#_Toc211028702)

[Tabel 2.2 Soal2Main.java Source Code 13](#_Toc211028703)

[Tabel 2.3 Negara.java Source Code 17](#_Toc211028704)

[Tabel 3.1 Soal No. 3 23](#_Toc211028705)

[Tabel 3.2 Mahasiswa.java Source Code 25](#_Toc211028706)

[Tabel 3.3 DaftarMahasiswa.java Source Code 26](#_Toc211028707)

[Tabel 3.3 Soal3Main.java Source Code 28](#_Toc211028708)

# SOAL 1

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

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

contoh table dibawah.

Tabel 1.1 Soal No. 1

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

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

## Source Code

Tabel 1.2 MainSoal1.java Source Code

|  |  |
| --- | --- |
| **File: MainSoal1.java** | |
| 1  2  3  4  5  6  7  8  9  10  11 | package modulTiga.PRAK201\_2410817310014\_IndraSuryadilaga;  public class Main {  public static void main(String[] args) {  LemparDadu lemparDadu = new LemparDadu();  lemparDadu.inputJumlahDadu();  lemparDadu.lemparDadu();  lemparDadu.tampilkanHasil();  }  } |

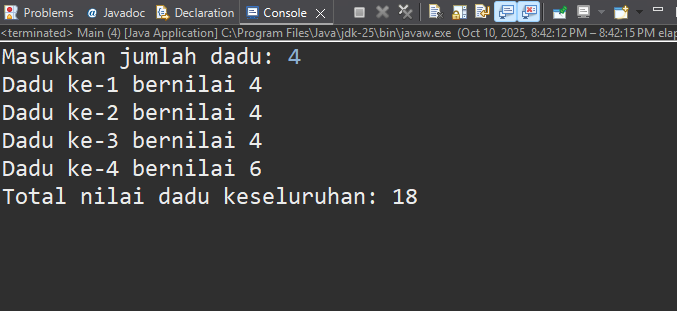
Tabel 1.3 Dadu.java Source Code

|  |  |
| --- | --- |
| **File: Dadu.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | package modulTiga.PRAK201\_2410817310014\_IndraSuryadilaga;  import java.util.Random;  public class Dadu {  private int mataDadu;  public void acakNilai() {  Random random = new Random();  this.mataDadu = random.nextInt(6) + 1;  }  public int getMataDadu() {  return mataDadu;  }  } |

Tabel 1.4 LemeparDadu.java Source Code

|  |  |
| --- | --- |
| **File: LemparDadu.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | package modulTiga.PRAK201\_2410817310014\_IndraSuryadilaga;  import java.util.LinkedList;  import java.util.Scanner;  public class LemparDadu {  private LinkedList<Dadu> daftarDadu = new LinkedList<>();  private int jumlahDadu;  private int totalNilai;  public void inputJumlahDadu() {  Scanner input = new Scanner(System.***in***);  while (true) {  System.***out***.print("Masukkan jumlah dadu: ");  jumlahDadu = input.nextInt();  if (jumlahDadu > 0) break;  System.***out***.println("Jumlah dadu harus lebih dari 0!");  }  input.close();  }  public void lemparDadu() {  totalNilai = 0;  daftarDadu.clear();  for (int i = 0; i < jumlahDadu; i++) {  Dadu d = new Dadu();  d.acakNilai();  daftarDadu.add(d);  System.***out***.println("Dadu ke-" + (i + 1) + " bernilai " + d.getMataDadu());  totalNilai += d.getMataDadu();  }  }  public void tampilkanHasil() {  System.***out***.println("Total nilai dadu keseluruhan: " + totalNilai);  }  } |

## Output Program



Gambar 1. Screenshot Hasil Jawaban Soal 1

## Pembahasan

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

1. **Kelas Main**

Kelas Main berfungsi sebagai *entry point* program, yaitu tempat eksekusi dimulai. Baris package modulTiga.PRAK201\_2410817310014\_IndraSuryadilaga; digunakan untuk mendefinisikan lokasi file di dalam folder praktikum.

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

1. inputJumlahDadu(): untuk menerima input jumlah dadu dari pengguna.
2. lemparDadu(): untuk mengacak nilai setiap dadu dan menyimpannya ke dalam *collection*.
3. tampilkanHasil() : untuk menampilkan total nilai dari semua dadu yang dilempar.
4. **Kelas Dadu**

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

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

1. **Kelas LemparDadu**

private LinkedList<Dadu> daftarDadu = new LinkedList<>(); menunjukkan penggunaan **collection** bertipe LinkedList untuk menyimpan kumpulan objek Dadu. LinkedList dipilih karena bersifat dinamis, ukurannya bisa berubah sesuai jumlah input dari pengguna.

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

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

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

# SOAL 2

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

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

Tabel 2.1 Soal No. 2

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

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

## Source Code

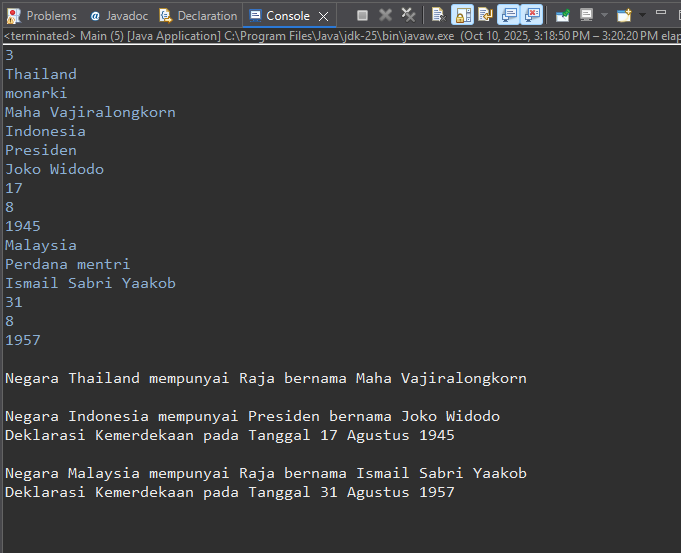
Tabel 2.2 Soal2Main.java Source Code

|  |  |
| --- | --- |
| **File: MainSoal2.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144 | package modulTiga.PRAK202\_2410817310014\_IndraSuryadilaga;  import java.util.\*;  public class Main {  public static void main(String[] args) {  Scanner input = new Scanner(System.***in***);    int jumlahNegara;  boolean jumlahValid = false;  LinkedList<Negara> daftarNegara = new LinkedList<>();  HashMap<Integer, String> namaBulan = new HashMap<>();  namaBulan.put(1, "Januari");  namaBulan.put(2, "Febuari");  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, "December");    //input jumlah negara dengan validasi jumlah negara harus lebih dari 0  do {  jumlahNegara = input.nextInt();  if (jumlahNegara < 1) {  System.***out***.println("jumlah negara yang anda masukan tidak valid,jumlah negara harus lebih dari 0");  continue;  } else {  jumlahValid = true;  }  input.nextLine();  }while (!jumlahValid);    //input data lengkap negara  for (int i = 0; i < jumlahNegara; i++) {  //input nama, jenis kepemimpinan, dan nama pemeimpin  String nama = input.nextLine();  String jenisKepemimpinan = input.nextLine();  String namaPemimpin = input.nextLine();    //input tanggal, bulan, dan tahun kemerdekaan  if (jenisKepemimpinan.equalsIgnoreCase("monarki")) {  //Negara monarki tidak memiliki tanggal kemerdekaan  daftarNegara.add(new Negara(nama, jenisKepemimpinan, namaPemimpin));  } else {  //Perulangan untuk Validasi Data Kemerdekaan  int tglKemerdekaan;  int blnKemerdekaan;  int thnKemerdekaan;  boolean tanggalValid = false;  do {  tglKemerdekaan = input.nextInt();  blnKemerdekaan = input.nextInt();  thnKemerdekaan = input.nextInt();  input.nextLine();  if (blnKemerdekaan < 1 || blnKemerdekaan > 12 || thnKemerdekaan <= 0) {  System.***out***.println("Bulan atau Tahun tidak valid. Silakan masukkan kembali tahun kemerdekaan.");  continue;  }  // Menentukan jumlah hari maksimum dalam sebulan (termasuk tahun kabisat)  int maxTanggal;  if (blnKemerdekaan == 2) {  // Cek tahun kabisat (leap year)  boolean isKabisat = (thnKemerdekaan % 4 == 0 && thnKemerdekaan % 100 != 0) || (thnKemerdekaan % 400 == 0);  maxTanggal = isKabisat ? 29 : 28;  } else if (blnKemerdekaan == 4 || blnKemerdekaan == 6 || blnKemerdekaan == 9 || blnKemerdekaan == 11) {  maxTanggal = 30;  } else {  maxTanggal = 31;  }  // Memeriksa apakah tanggal yang dimasukkan sesuai dengan rentang bulan dan tahun  if (tglKemerdekaan >= 1 && tglKemerdekaan <= maxTanggal) {  tanggalValid = true;  } else {  System.***out***.println("=> ERROR: Tanggal " + tglKemerdekaan + " tidak ada pada bulan ke-" + blnKemerdekaan + " tahun " + thnKemerdekaan + ". Silakan masukkan kembali data kemerdekaan.");  }  } while (!tanggalValid);  daftarNegara.add(new Negara(nama, jenisKepemimpinan, namaPemimpin, tglKemerdekaan, blnKemerdekaan, thnKemerdekaan));  }  }    //output data negara dengan perulangan.  System.***out***.println();  for (Negara negara : daftarNegara) {  if (negara.getJenisKepemimpinan().equalsIgnoreCase("monarki")) {  negara.TampilkanInfo("");  } else {  String bulanNama = namaBulan.get(negara.getBlnKemerdekaan());  negara.TampilkanInfo(bulanNama);  }  }    input.close();  }  } |

Tabel 2.3 Negara.java Source Code

|  |  |
| --- | --- |
| **File: Negara.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80 | package modulTiga.PRAK202\_2410817310014\_IndraSuryadilaga;  public class Negara {  //atribut  private String nama;  private String jenisKepemimpinan;  private String namaPemimpin;  private int tglKemerdekaan;  private int blnKemerdekaan;  private int thnKemerdekaan;  //constructor negara dengan kemerdekaan  public Negara(String nama, String jenisKepemimpinan, String namaPemimpin, int tglKemerdekaan, int blnKemerdekaan, int thnKemerdekaan) {  this.nama = nama;  this.jenisKepemimpinan = jenisKepemimpinan;  this.namaPemimpin = namaPemimpin;  this.tglKemerdekaan = tglKemerdekaan;  this.blnKemerdekaan = blnKemerdekaan;  this.thnKemerdekaan = thnKemerdekaan;  }    //constructor negara tanpa kemerdekaan  public Negara(String nama, String jenisKepemimpinan, String namaPemimpin) {  this.nama = nama;  this.jenisKepemimpinan = jenisKepemimpinan;  this.namaPemimpin = namaPemimpin;  }    //Getter  public String getNama() {  return nama;  }  public String getJenisKepemimpinan() {  return jenisKepemimpinan;  }    public String getNamaPemimpin() {  return namaPemimpin;  }    public int getTglKemerdekaan() {  return tglKemerdekaan;  }    public int getBlnKemerdekaan() {  return blnKemerdekaan;  }  public int getThnKemerdekaan() {  return thnKemerdekaan;  }    //method menampilkan informasi negara  public void TampilkanInfo(String namaBulan) {  System.***out***.println("Negara " + nama + " mempunyai "  + (jenisKepemimpinan.equalsIgnoreCase("presiden") ? "Presiden" : "Raja")  + " bernama " + namaPemimpin);  if (!jenisKepemimpinan.equalsIgnoreCase("monarki")) {  System.***out***.println("Deklarasi Kemerdekaan pada Tanggal "  + tglKemerdekaan + " " + namaBulan + " " + thnKemerdekaan);  }  System.***out***.println();  }    } |

## Output Program



Gambar 2. Screenshot Hasil Jawaban Soal 2

## Pembahasan

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

1. **Kelas Negara**

Kelas **Negara** memiliki enam atribut:

1. nama: menyimpan nama negara.
2. jenisKepemimpinan: menyimpan jenis sistem pemerintahan (misalnya *presiden* atau *monarki*).
3. namaPemimpin: menyimpan nama pemimpin negara.
4. tglKemerdekaan: menyimpan tanggal kemerdekaan.
5. blnKemerdekaan: menyimpan bulan kemerdekaan dalam bentuk angka.
6. thnKemerdekaan: menyimpan tahun kemerdekaan.

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

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

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

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

1. **Kelas Main**

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

1. Scanner input = new Scanner(System.in);
2. LinkedList<Negara> daftarNegara = new LinkedList<>();
3. HashMap<Integer, String> namaBulan = new HashMap<>();.

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

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

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

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

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

# SOAL 3

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

1. Buatlah kelas dengan nama Mahasiswa
2. Terdapat 2 atribut pada kelas, yaitu nama dan nim
3. Inisiasi nilai atribut dilakukan pada konstruktor
4. Terdapat getter untuk 2 atribut tadi
5. Gunakan collection dengan tipe ArrayList untuk menyimpan objek Mahasiswa
6. Program bersifat dinamis dan interaktif
7. Programa dapat melakukan operasi seperti berikut:
   * Tambah Mahasiswa, menambahkan objek baru ke ArrayList
   * Hapus Mahasiswa, menhapus data mahasiswa dari ArrayList berdasarkan NIM
   * Cari Mahasiswa, menampilkan data mahasiswa berdasarkan NIM yang di input oleh pengguna
   * Tampilkan seluruh data Mahasiswa, menampilkan seluruh data mahasiswa dari ArrayList. Tampilkan nama dan NIM mahasiswa
   * Keluar, program berhenti dan seluruh data pada ArrayList dihapus, ketika program dijalankan ulang, ArrayList masih kosong

Tabel 3.1 Soal No. 3

|  |
| --- |
| **Output** |
| Menu:  1. Tambah Mahasiswa  2. Hapus Mahasiswa berdasarkan NIM  3. Cari Mahasiswa berdasarkan NIM  4. Tampilkan Daftar Mahasiswa  0. Keluar  Pilihan: 1  Masukkan Nama Mahasiswa: Bachrul Uluum  Masukkan NIM Mahasiswa (harus unik): 2010817210025  Mahasiswa Bachrul Uluum ditambahkan.  Menu:  1. Tambah Mahasiswa  2. Hapus Mahasiswa berdasarkan NIM  3. Cari Mahasiswa berdasarkan NIM  4. Tampilkan Daftar Mahasiswa  0. Keluar  Pilihan: 1  Masukkan Nama Mahasiswa: Muhammad Aulia Akbar  Masukkan NIM Mahasiswa (harus unik): 2010817210023  Mahasiswa Muhammad Aulia Akbar ditambahkan.  Menu:  1. Tambah Mahasiswa  2. Hapus Mahasiswa berdasarkan NIM  3. Cari Mahasiswa berdasarkan NIM  4. Tampilkan Daftar Mahasiswa  0. Keluar  Pilihan: 4  Daftar Mahasiswa:  NIM: 2010817210025, Nama: Bachrul Uluum  NIM: 2010817210023, Nama: Muhammad Aulia Akbar  1. Tambah Mahasiswa  2. Hapus Mahasiswa berdasarkan NIM  3. Cari Mahasiswa berdasarkan NIM  4. Tampilkan Daftar Mahasiswa  0. Keluar  Pilihan: 2  Masukkan NIM Mahasiswa yang akan dihapus: 2010817210025  Mahasiswa dengan NIM 2010817210025 dihapus.  Menu:  1. Tambah Mahasiswa  2. Hapus Mahasiswa berdasarkan NIM  3. Cari Mahasiswa berdasarkan NIM  4. Tampilkan Daftar Mahasiswa  0. Keluar  Pilihan: 0  Terima kasih! |

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

## Source Code

Tabel 3.2 Mahasiswa.java Source Code

|  |  |
| --- | --- |
| **File: Mahasiswa.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | package modulTiga.PRAK203\_2410817310014\_IndraSuryadilaga;  public class Mahasiswa {  //atribut  private String nama;  private String nim;    //constructor  Mahasiswa(String nama, String nim) {  this.nama = nama;  this.nim = nim;  }      //getter  public String getNama() {  return this.nama;  }    public String getNim() {  return this.nim;  }  } |

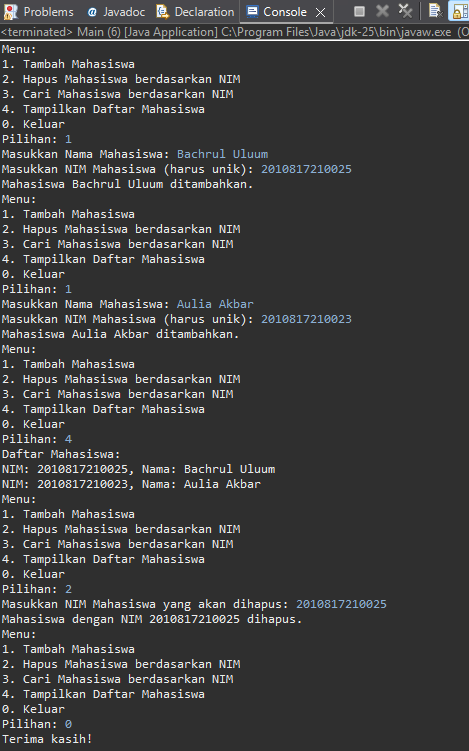
Tabel 3.3 DaftarMahasiswa.java Source Code

|  |  |
| --- | --- |
| **File: DaftarMahasiswa.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82 | package modulTiga.PRAK203\_2410817310014\_IndraSuryadilaga;  import java.util.ArrayList;  import java.util.Iterator;  import java.util.Scanner;  public class DaftarMahasiswa {  Scanner input = new Scanner(System.in);    ArrayList<Mahasiswa> daftarMahasiswa = new ArrayList<>();    //tambah Mahasiswa  public void tambahMahasiswa(String nama, String nim) {  //nim harus unik  for (Mahasiswa m : daftarMahasiswa) {  if (m.getNim().equals(nim)) {  System.out.println("NIM sudah digunakan! Gagal menambahkan mahasiswa.\n");  return;  }  }  daftarMahasiswa.add(new Mahasiswa(nama, nim));  System.out.println("Mahasiswa " + nama + " ditambahkan.\n");  }    //hapus mahasiswa berdasrkan nim  public void hapusMahasiswa(String nim) {  Iterator<Mahasiswa> iterator = daftarMahasiswa.iterator();  while (iterator.hasNext()) {  Mahasiswa m = iterator.next();  if (m.getNim().equals(nim)) {  iterator.remove();  System.out.println("Mahasiswa dengan NIM " + nim + " dihapus.\n");  return;  }  }  System.out.println("Mahasiswa dengan NIM " + nim + " tidak ditemukan.\n");  }    //mencari mahasiswa menggunakan nim  public void cariMahasiswa(String nim) {  for (Mahasiswa m : daftarMahasiswa) {  if (m.getNim().equals(nim)) {  System.out.println("Data Mahasiswa Ditemukan:");  System.out.println("NIM: " + m.getNim() + ", Nama: " + m.getNama() + "\n");  return;  }  }  System.out.println("Mahasiswa dengan NIM " + nim + " tidak ditemukan.\n");  }    // Menampilkan semua data mahasiswa  public void tampilkanSemua() {  if (daftarMahasiswa.isEmpty()) {  System.out.println("Daftar Mahasiswa kosong.\n");  return;  }  System.out.println("Daftar Mahasiswa:");  for (Mahasiswa m : daftarMahasiswa) {  System.out.println("NIM: " + m.getNim() + ", Nama: " + m.getNama());  }  System.out.println();  }    // Menghapus semua data (saat keluar)  public void hapusSemua() {  daftarMahasiswa.clear();  }  } |

Tabel 3.3 Soal3Main.java Source Code

|  |  |
| --- | --- |
| **File: Soal3Main.java** | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74 | package modulTiga.PRAK203\_2410817310014\_IndraSuryadilaga;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  DaftarMahasiswa daftar = new DaftarMahasiswa();  int pilihan;    do {  System.*out*.println("Menu:");  System.*out*.println("1. Tambah Mahasiswa");  System.*out*.println("2. Hapus Mahasiswa berdasarkan NIM");  System.*out*.println("3. Cari Mahasiswa berdasarkan NIM");  System.*out*.println("4. Tampilkan Daftar Mahasiswa");  System.*out*.println("0. Keluar");  System.*out*.print("Pilihan: ");  pilihan = input.nextInt();  input.nextLine(); // buang newline  switch (pilihan) {  case 1:  System.*out*.print("Masukkan Nama Mahasiswa: ");  String nama = input.nextLine();  System.*out*.print("Masukkan NIM Mahasiswa (harus unik): ");  String nim = input.nextLine();  daftar.tambahMahasiswa(nama, nim);  break;  case 2:  System.*out*.print("Masukkan NIM Mahasiswa yang akan dihapus: ");  nim = input.nextLine();  daftar.hapusMahasiswa(nim);  break;  case 3:  System.*out*.print("Masukkan NIM Mahasiswa yang dicari: ");  nim = input.nextLine();  daftar.cariMahasiswa(nim);  break;  case 4:  daftar.tampilkanSemua();  break;  case 0:  daftar.hapusSemua();  System.*out*.println("Terima kasih!");  break;  default:  System.*out*.println("Pilihan tidak valid.\n");  break;  }    } while (pilihan != 0);  input.close();  }  } |

## Output Program



Gambar 3. Screenshot Hasil Jawaban Soal 3

## Pembahasan

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

1. **Kelas Mahasiswa**

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

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

1. **Kelas DaftarMahasiswa**

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

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

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

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

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

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

1. **Kelas Main**

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

1. Tambah Mahasiswa
2. Hapus Mahasiswa berdasarkan NIM
3. Cari Mahasiswa berdasarkan NIM
4. Tampilkan Daftar Mahasiswa
5. Keluar dari program.

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

# GITHUB

<https://github.com/IndraSuryadilaga/Pemrograman_II>