**LAPORAN AKHIR PRAKTIKUM**

**PEMROGRAMAN II**

****

**Oleh:**

**Muhammad Ryan Rizky Rahmadi NIM. 2210817310001**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS LAMBUNG MANGKURAT**

**2023**

# LEMBAR PENGESAHAN

**LAPORAN AKHIR PRAKTIKUM PEMROGRAMAN II**

Laporan Praktikum Pemrograman II

Modul 0 : Input, Output, dan Logika Sederhana.

Modul 1 : Struktur Dasar Kode Java

Modul 2 : Class dan Objek

Modul 3 : Enkapulasi dan Collection

Modul 4 : Pewarisan/Inheritance & Diagram Class

Modul 5 : Polimofisme

Modul 6 : Graphical User Interface

ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman II. Laporan Akhir Prakitkum ini dikerjakan oleh:

Nama Praktikan : Muhammad Ryan Rizky Rahmadi

NIM : 2210817310001

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Bachrul Uluum  NIM. 2010817210025 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Andreyan Rizky Baskara, S.Kom., M.Kom.  NIP. 19930703 201903 1 011 |

# DAFTAR ISI

[LEMBAR PENGESAHAN 2](#_Toc153975465)

[DAFTAR ISI 3](#_Toc153975466)

[DAFTAR TABEL 6](#_Toc153975467)

[DAFTAR GAMBAR 8](#_Toc153975468)

[MODUL 0: INPUT, OUTPUT, DAN LOGIKA SEDERHANA 9](#_Toc153975469)

[SOAL 1 9](#_Toc153975470)

[A. Source Code 9](#_Toc153975471)

[B. Output Program 9](#_Toc153975472)

[C. Pembahasan 9](#_Toc153975473)

[SOAL 2 9](#_Toc153975474)

[A. Source Code 10](#_Toc153975475)

[B. Output Program 10](#_Toc153975476)

[C. Pembahasan 10](#_Toc153975477)

[SOAL 3 11](#_Toc153975478)

[A. Source Code 11](#_Toc153975479)

[B. Output Program 11](#_Toc153975480)

[C. Pembahasan 12](#_Toc153975481)

[SOAL 4 13](#_Toc153975482)

[A. Source Code 13](#_Toc153975483)

[B. Output Program 13](#_Toc153975484)

[C. Pembahasan 14](#_Toc153975485)

[SOAL 5 15](#_Toc153975486)

[A. Source Code 15](#_Toc153975487)

[B. Output Program 15](#_Toc153975488)

[C. Pembahasan 15](#_Toc153975489)

[MODUL 1: STRUKTUR DASAR KODE JAVA 17](#_Toc153975490)

[SOAL 1 17](#_Toc153975491)

[A. Source Code 17](#_Toc153975492)

[B. Output Program 19](#_Toc153975493)

[C. Pembahasan 19](#_Toc153975494)

[SOAL 2 20](#_Toc153975495)

[A. Source Code 20](#_Toc153975496)

[B. Output Program 20](#_Toc153975497)

[C. Pembahasan 21](#_Toc153975498)

[SOAL 3 22](#_Toc153975499)

[A. Source Code 22](#_Toc153975500)

[B. Output Program 23](#_Toc153975501)

[C. Pembahasan 23](#_Toc153975502)

[SOAL 4 24](#_Toc153975503)

[A. Source Code 24](#_Toc153975504)

[B. Output Program 26](#_Toc153975505)

[C. Pembahasan 26](#_Toc153975506)

[SOAL 5 27](#_Toc153975507)

[A. Source Code 27](#_Toc153975508)

[B. Output Program 27](#_Toc153975509)

[C. Pembahasan 27](#_Toc153975510)

[MODUL 2: CLASS DAN OBJEK 29](#_Toc153975511)

[SOAL 1 29](#_Toc153975512)

[A. Source Code 29](#_Toc153975513)

[B. Output Program 31](#_Toc153975514)

[C. Pembahasan 31](#_Toc153975515)

[SOAL 2 32](#_Toc153975516)

[A. Source Code 32](#_Toc153975517)

[B. Output Program 33](#_Toc153975518)

[C. Pembahasan 33](#_Toc153975519)

[SOAL 3 35](#_Toc153975520)

[A. Source Code 35](#_Toc153975521)

[B. Output Program 36](#_Toc153975522)

[C. Pembahasan 36](#_Toc153975523)

[MODUL 3: ENKAPULASI DAN COLLECTION 38](#_Toc153975524)

[SOAL 1 38](#_Toc153975525)

[A. Source Code 38](#_Toc153975526)

[B. Output Program 40](#_Toc153975527)

[C. Pembahasan 40](#_Toc153975528)

[SOAL 2 41](#_Toc153975529)

[A. Source Code 43](#_Toc153975530)

[B. Output Program 45](#_Toc153975531)

[C. Pembahasan 45](#_Toc153975532)

[SOAL 3 47](#_Toc153975533)

[A. Source Code 48](#_Toc153975534)

[B. Output Program 52](#_Toc153975535)

[C. Pembahasan 52](#_Toc153975536)

[MODUL 4: PEWARISAN/INHERITANCE & DIAGRAM KELAS 54](#_Toc153975537)

[SOAL 1 54](#_Toc153975538)

[A. Source Code 54](#_Toc153975539)

[B. Output Program 55](#_Toc153975540)

[C. Pembahasan 55](#_Toc153975541)

[SOAL 2 57](#_Toc153975542)

[A. Source Code 59](#_Toc153975543)

[B. Output Program 61](#_Toc153975544)

[C. Pembahasan 61](#_Toc153975545)

[MODUL 5: POLIMORFISME 63](#_Toc153975546)

[SOAL 63](#_Toc153975547)

[A. Source Code 67](#_Toc153975548)

[B. Output Program 70](#_Toc153975549)

[C. Pembahasan 70](#_Toc153975550)

[MODUL 6: GRAPHICAL USER INTEFACE 73](#_Toc153975551)

[SOAL 73](#_Toc153975552)

[A. Source Code 74](#_Toc153975553)

[B. Output Program 77](#_Toc153975554)

[C. Pembahasan 77](#_Toc153975555)

# DAFTAR TABEL

**MODUL 0: HOW TO PROGRAM**

[Tabel 1.1 Soal 1 9](#_Toc153974882)

[Tabel 2.1 Source Code Soal 1 9](#_Toc153974883)

[Tabel 3.2 Soal 2 10](#_Toc153974884)

[Tabel 4.2 Source Code Soal 2 10](#_Toc153974885)

[Tabel 5.3 Soal 3 11](#_Toc153974886)

[Tabel 6.3 Source Code Soal 3 11](#_Toc153974887)

[Tabel 7.4 Soal 3 13](#_Toc153974888)

[Tabel 8.4 Source Code Soal 4 13](#_Toc153974889)

[Tabel 9.5 Soal 5 15](#_Toc153974890)

[Tabel 10.5 Source Code Soal 5 15](#_Toc153974891)

**Modul 1 : Struktur Dasar Kode Java**

[Tabel 11.1 Soal 1 17](#_Toc153974892)

[Tabel 12.1 Source Code Soal 1 19](#_Toc153974893)

[Tabel 13.2 Soal 2 20](#_Toc153974894)

[Tabel 14.2 Source Code Soal 2 20](#_Toc153974895)

[Tabel 15.3 Soal 3 22](#_Toc153974896)

[Tabel 16.3 Source Code Soal 3 22](#_Toc153974897)

[Tabel 17.4 Soal 3 24](#_Toc153974898)

[Tabel 18.4 Source Code Soal 4 26](#_Toc153974899)

[Tabel 19.5 Soal 5 27](#_Toc153974900)

[Tabel 20.5 Source Code Soal 5 27](#_Toc153974901)

**Modul 2 : Class dan Objek**

[Tabel 21.1 Soal 1 29](#_Toc153974902)

[Tabel 22.1 Source Code Soal 1 Buah 30](#_Toc153974903)

[Tabel 23.1 Source Code Soal 1 Main 30](#_Toc153974904)

[Tabel 24.2 Soal 2 32](#_Toc153974905)

[Tabel 25.2 Source Code Soal 2 Kopi 33](#_Toc153974906)

[Tabel 26.2 Source Code Soal 2 Main 33](#_Toc153974907)

[Tabel 27.3 Soal 3 35](#_Toc153974908)

[Tabel 28.3 Source Code Soal 3 Pegawai 35](#_Toc153974909)

[Tabel 29.3 Source Code Soal 3 Main 36](#_Toc153974910)

**Modul 3 : Enkapulasi dan Collection**

[Tabel 30.1 Soal 1 38](#_Toc153974911)

[Tabel 31.1 Source Code Soal 1 Dadu 39](#_Toc153974912)

[Tabel 32.1 Source Code Soal 1 Main 39](#_Toc153974913)

[Tabel 33.2 Soal 2 42](#_Toc153974914)

[Tabel 34.2 Source Code Soal 2 Negara 44](#_Toc153974915)

[Tabel 35.2 Source Code Soal 2 Main 45](#_Toc153974916)

[Tabel 36.3 Soal 3 48](#_Toc153974917)

[Tabel 37.3 Source Code Soal 3 Kopi 49](#_Toc153974918)

[Tabel 38.3 Source Code Soal 3 Main 51](#_Toc153974919)

**Modul 4 : Pewarisan/Inheritance & Diagram Class**

[Tabel 39.1 Soal 1 54](#_Toc153974920)

[Tabel 40.1 Source Code Soal 1 Peliharaan 55](#_Toc153974921)

[Tabel 41.1 Source Code Soal 1 Man 55](#_Toc153974922)

[Tabel 42.2 Soal 2 58](#_Toc153974923)

[Tabel 43.2 Source Code Soal 2 HewanPeliharaan 59](#_Toc153974924)

[Tabel 44.2 Source Code Soal 2 Anjing 60](#_Toc153974925)

[Tabel 45.2 Source Code Soal 2 Kucing 60](#_Toc153974926)

[Tabel 46.2 Source Code Soal 2 Main 61](#_Toc153974927)

**Modul 5 : Polimofisme**

[Tabel 47.1 Soal 1 67](#_Toc153974928)

[Tabel 48.1 Source Code Soal 1 Spehere 67](#_Toc153974929)

[Tabel 49.1 Source Code Soal 1 Shape 68](#_Toc153974930)

[Tabel 50.1 Source Code Soal 1 Rectangle 68](#_Toc153974931)

[Tabel 51.1 Source Code Soal 1 Cylinder 69](#_Toc153974932)

[Tabel 52.1 Source Code Soal 1 Paint 69](#_Toc153974933)

[Tabel 53.1 Source Code Soal 1 PaintThings 70](#_Toc153974934)

**Modul 6 : Graphical User Interface**

[Tabel 54.1 Source Code Soal 1 Mahasiswa 75](#_Toc153974935)

[Tabel 55.1 Source Code Soal 1 Main 77](#_Toc153974936)

# DAFTAR GAMBAR

**MODUL 0: HOW TO PROGRAM**

[Gambar 1.1 Screenshoot Output Soal 1 9](#_Toc153974937)

[Gambar 2.2 Output Screenshoot Soal 2 10](#_Toc153974938)

[Gambar 3.3 Output Screenshoot Soal 3 11](#_Toc153974939)

[Gambar 4.4 Output Screenshoot Soal 4 13](#_Toc153974940)

[Gambar 5.5 Output Screenshoot Soal 5 15](#_Toc153974941)

**Modul 1 : Struktur Dasar Kode Java**

[Gambar 6.1 Screenshoot Output Soal 1 19](#_Toc153974942)

[Gambar 7.2 Output Screenshoot Soal 2 20](#_Toc153974943)

[Gambar 8.3 Output Screenshoot Soal 3 23](#_Toc153974944)

[Gambar 9.4 Output Screenshoot Soal 4 26](#_Toc153974945)

[Gambar 10.5 Output Screenshoot Soal 5 27](#_Toc153974946)

**Modul 2 : Class dan Objek**

[Gambar 11.1 Screenshoot Output Soal 1 31](#_Toc153974947)

[Gambar 12.2 Output Screenshoot Soal 2 33](#_Toc153974948)

[Gambar 13.3 Output Screenshoot Soal 3 36](#_Toc153974949)

**Modul 3 : Enkapulasi dan Collection**

[Gambar 14.1 Screenshoot Output Soal 1 40](#_Toc153974950)

[Gambar 15.2 Output Screenshoot Soal 2 45](#_Toc153974951)

[Gambar 16.3 Output Screenshoot Soal 3 52](#_Toc153974952)

**Modul 4 : Pewarisan/Inheritance & Diagram Class**

[Gambar 17.1 Screenshoot Output Soal 1 55](#_Toc153974953)

[Gambar 18.2 Output Screenshoot Soal 2 61](#_Toc153974954)

**Modul 5 : Polimofisme**

[Gambar 1.1 Screenshoot Output Soal 1 70](#_Toc153974955)

**Modul 6 : Graphical User Interface**

[Gambar 1.1 Screenshoot Output Soal 1 77](#_Toc153974956)

# MODUL 0: INPUT, OUTPUT, DAN LOGIKA SEDERHANA

## SOAL 1

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

Selamat Pagi, Nama Anda Selamat Siang, Nama Anda Selamat Malam, Nama Anda

**Output**

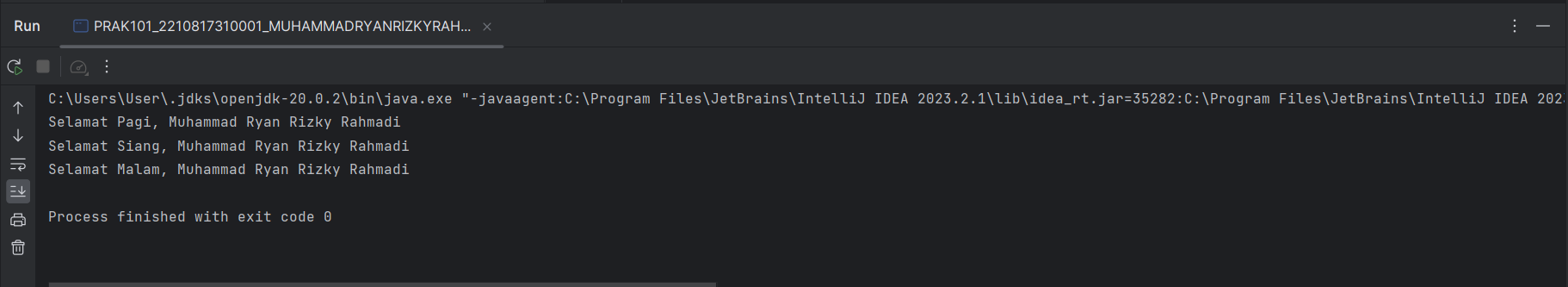
Tabel .1 Soal 1

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | Public class PRAK101\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args){  System.*out*.println("Selamat Pagi, Muhammad Ryan Rizky Rahmadi");  System.*out*.println("Selamat Siang, Muhammad Ryan Rizky Rahmadi");  System.*out*.println("Selamat Malam, Muhammad Ryan Rizky Rahmadi");  } } |

Tabel .1 Source Code Soal 1

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

Line 1 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 11 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 4 – 9 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

## SOAL 2

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

##############################

##############################

**Output**

|  |  |  |
| --- | --- | --- |
| # |  | # |
| # | Nama Anda | # |

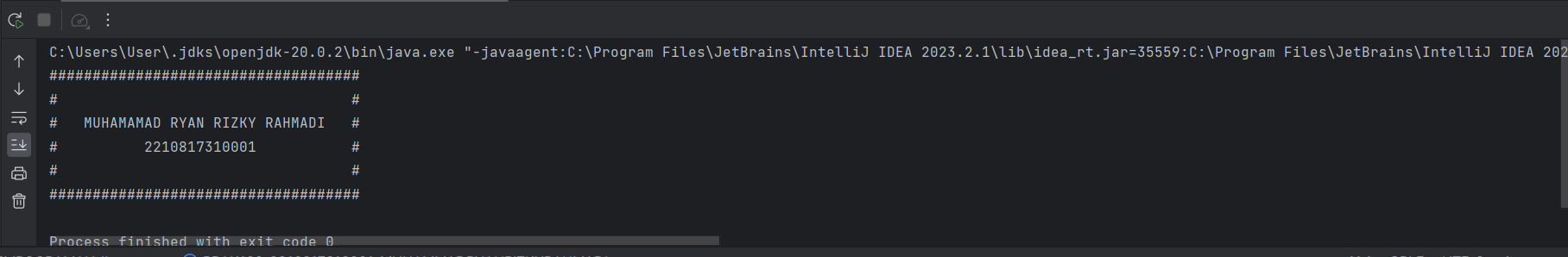
Tabel .2 Soal 2

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public class PRAK102\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args){  System.*out*.println ("####################################");  System.*out*.println ("# #");  System.*out*.println ("# MUHAMAMAD RYAN RIZKY RAHMADI #");  System.*out*.println ("# 2210817310001 #");  System.*out*.println ("# #");  System.*out*.println ("####################################");  } } |

Tabel .2 Source Code Soal 2

### Output Program



Gambar .2 Output Screenshoot Soal 2

### Pembahasan

Line 1 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

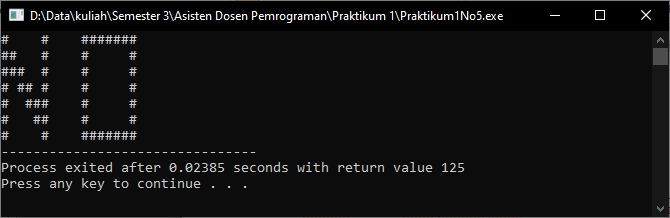
Line 2 dan 17 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 4 – 15 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

## SOAL 3

Buatlah program yang dapat menghasilkan output sebagaimana berikut:



**Output**

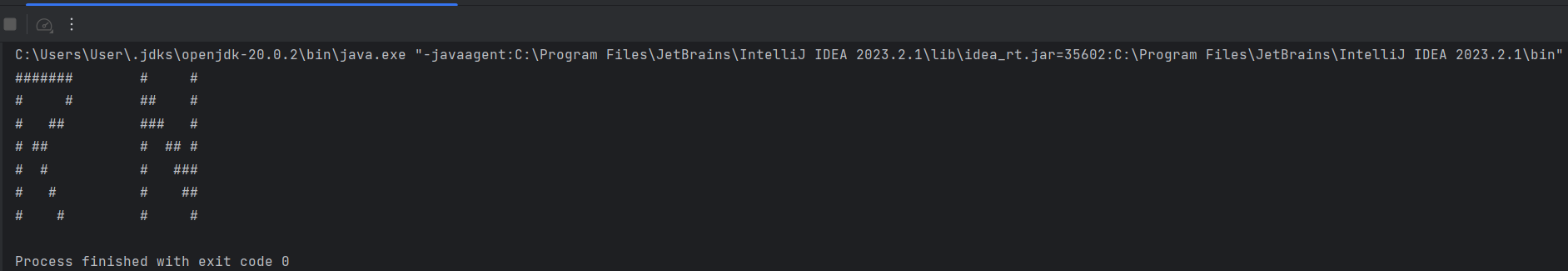
Tabel .3 Soal 3

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | public class PRAK103\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args){  System.*out*.println ("####### # #");  System.*out*.println ("# # ## #");  System.*out*.println ("# ## ### #");  System.*out*.println ("# ## # ## #");  System.*out*.println ("# # # ###");  System.*out*.println ("# # # ##");  System.*out*.println ("# # # #");  }  } |

Tabel .3 Source Code Soal 3

### Output Program



Gambar .3 Output Screenshoot Soal 3

### Pembahasan

Line 1 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 13 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 4 – 10 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

## SOAL 4

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

|  |
| --- |
| **Output** |
| Perkenalkan, Nama Saya <Nama>, Umur Saya <Umur> |

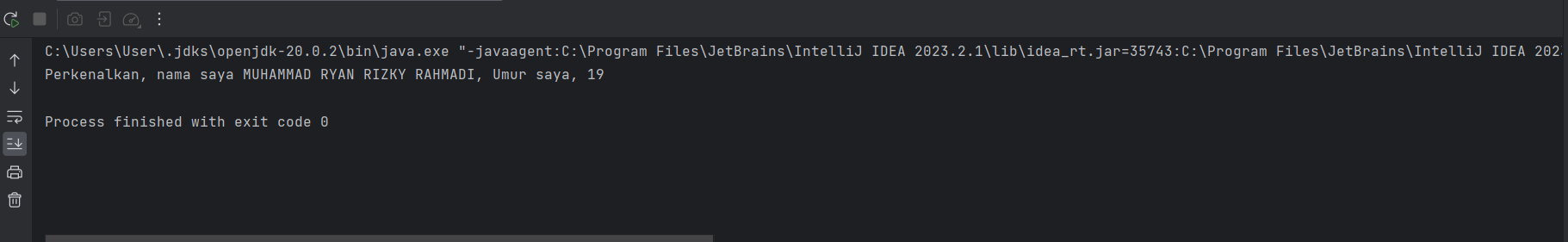
Tabel .4 Soal 3

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public class PRAK104\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args) {  String namy = ("MUHAMMAD RYAN RIZKY RAHMADI");  int umur = (19);  System.*out*.println ("Perkenalkan, nama saya " + (namy)+", Umur saya, " + (umur));  } } |

Tabel .4 Source Code Soal 4

### Output Program



Gambar .4 Output Screenshoot Soal 4

### Pembahasan

Line 1 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 10 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 4 : *String*, tipe data teks yang berisikan karakter lebih.

Line 5 : *int,* untuk penulisan angka bulat.

Line 7: *System.out.println*  mencetak sesuatu dari program sebagai *output.*

## SOAL 5

Buatlah program yang dapat menghasilkan output sebagaimana berikut:

|  |
| --- |
| **Input** |
| Masukan Makanan Favorit: Bakso  Masukan Hobi: Renang |
| **Output** |
| Aku Suka Makan Bakso, dan Hobiku Renang |

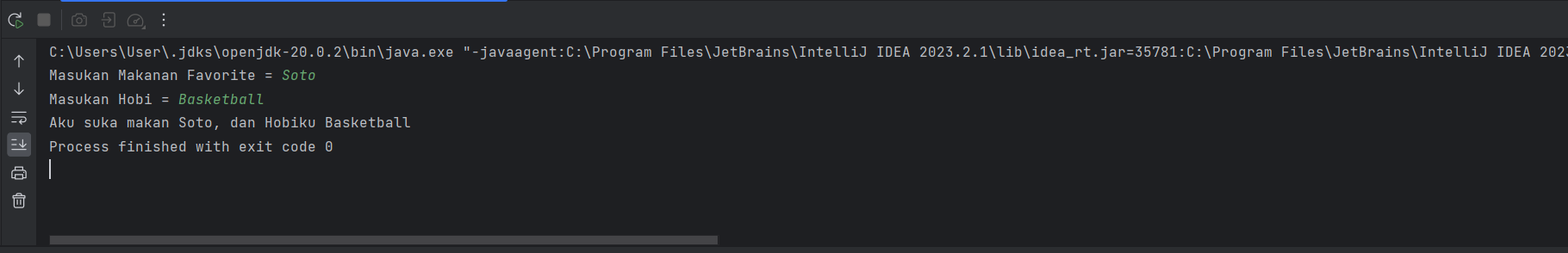
Tabel .5 Soal 5

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | import java.util.Scanner;  public class PRAK105\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args) {  String makan;  String hoby;  Scanner namy = new Scanner(System.*in*);   System.*out*.print("Masukan Makanan Favorite = ");  makan = namy.next();   System.*out*.print("Masukan Hobi = ");  hoby = namy.next();   System.*out*.print("Aku suka makan "+ (makan)+", dan Hobiku "+(hoby));  } } |

Tabel .5 Source Code Soal 5

### Output Program



Gambar .5 Output Screenshoot Soal 5

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 3 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 20 : {}, penanda awal dan akhir *block code*.

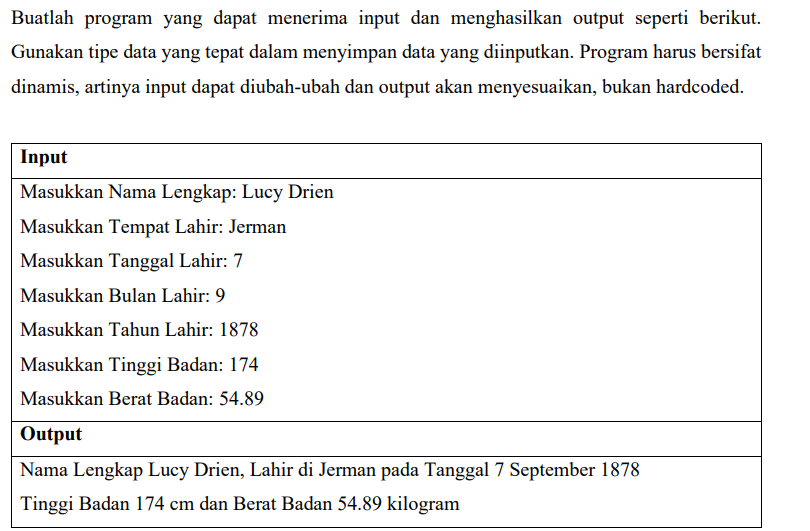
Line 5 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 6-7 : *String*, tipe data teks yang berisikan karakter lebih.

Line 10 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

# MODUL 1: STRUKTUR DASAR KODE JAVA

## SOAL 1



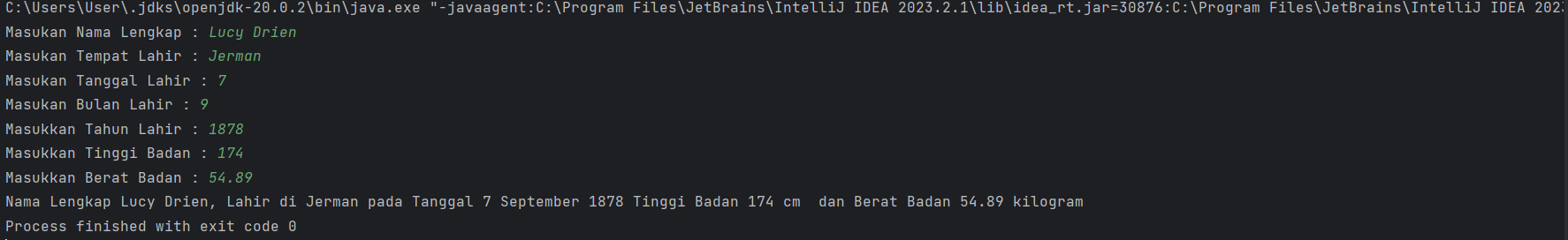
Tabel .1 Soal 1

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  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  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83 | import java.util.Scanner;  public class PRAK101\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args){  String namapanjang;  String tempatlahir, namabulan = "";  int tanggallahir,bulanlahir, tahunlahir, tinggibadan;  float beratbadan;  Scanner namy = new Scanner (System.*in*);  System.*out*.print("Masukan Nama Lengkap : ");  namapanjang = namy.nextLine();  System.*out*.print("Masukan Tempat Lahir : ");  tempatlahir = namy.next();  System.*out*.print("Masukan Tanggal Lahir : ");  tanggallahir = namy.nextInt();  System.*out*.print("Masukan Bulan Lahir : ");  bulanlahir = namy.nextInt();  switch (bulanlahir){  case 1:  namabulan = "Januari";  break;  case 2:   namabulan = "Febuari";  break;  case 3:  namabulan = "Maret" ;   break;  case 4:   namabulan = "April";  break;  case 5:   namabulan = "Mei";  break;  case 6:   namabulan = "Juni";  break;  case 7:   namabulan = "Juli";  break;  case 8:   namabulan = "Agustus";  break;  case 9:  namabulan = "September";  break;  case 10:  namabulan = "Oktobor";  break;  case 11:  namabulan = "November";  break;  case 12:  namabulan = "Desember";  break;   }  System.*out*.print("Masukkan Tahun Lahir : ");  tahunlahir = namy.nextInt();  System.*out*.print("Masukkan Tinggi Badan : ");  tinggibadan = namy.nextInt();  System.*out*.print("Masukkan Berat Badan : ");  beratbadan = namy.nextFloat();   System.*out*.print("Nama Lengkap "+ (namapanjang)+", Lahir di "+ (tempatlahir)  + " pada Tanggal "+ (tanggallahir)+" "+ (namabulan)+" "+(tahunlahir)+ " Tinggi Badan "+(tinggibadan)+" cm "+ " dan Berat Badan "+(beratbadan)+" kilogram");    } } |

Tabel .1 Source Code Soal 1

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 2 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 83 : {}, penanda awal dan akhir *block code*.

Line 5 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 6-7 : *String*, tipe data teks yang berisikan karakter lebih.

Line 8-9: *int,* tipe data *integer* atau menerima bilangan bulat.

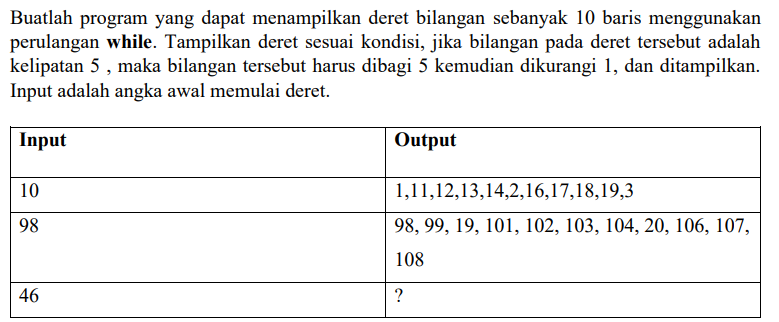
Line 10 : *float* tipe data *float* atau menerima bilangan desimal.

Line 11, : *Scanner* untuk menginput data.

Line 13,16,20,23,64,67,71,dan 75 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

Line 26-62 : *Switch,* membuat suatu menu yang di sebut *case* untuk membadandingkan sebuah nilai variable.

## SOAL 2



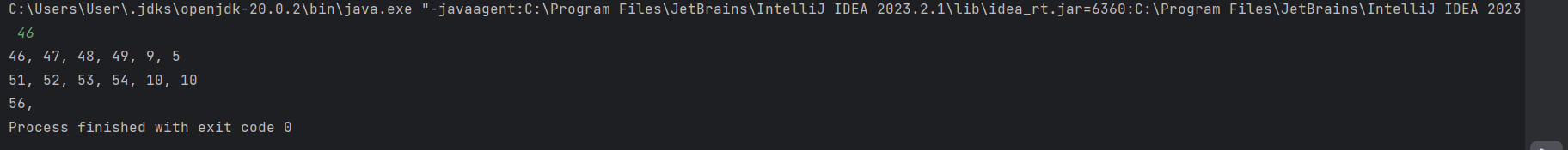
Tabel .2 Soal 2

### Source Code

|  |  |
| --- | --- |
| 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 | import java.util.Scanner;  public class PRAK102\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args) {  Scanner namy = new Scanner(System.*in*);  System.*out*.print(" ");  int angka = namy.nextInt();  int derets = 1;  while (derets <= 11) {  int bilangan;  if (angka % 5 == 0) {  bilangan = (angka / 5) - 1;  } else {  bilangan = angka;  }  System.*out*.print(bilangan + ", ");  if (derets % 5 == 0) {  System.*out*.println(derets);  }  angka++;  derets++;  }  } } |

Tabel .2 Source Code Soal 2

### Output Program



Gambar .2 Output Screenshoot Soal 2

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 3 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 25 : {}, penanda awal dan akhir *block code*.

Line 5 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 6 : *Scanner* untuk menginput data.

Line 7 dan 19 : *System.out.print*  mencetak sesuatu dari program sebagai *output.*

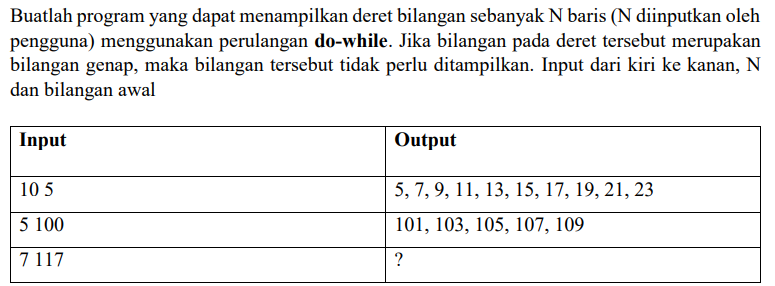
Line 8-9, dan 11: *int,* tipe data *integer* atau menerima bilangan bulat.

Line 10 : *While* untuk perulangan pada kondisi hingga terpenuhi.

Line 12,14 dan 18 : *If Else* untuk memberikan kondisi *true* atau *false.*

Line 21-22 : ++ atau *increment*  untuk menambah variable satu angka.

## SOAL 3



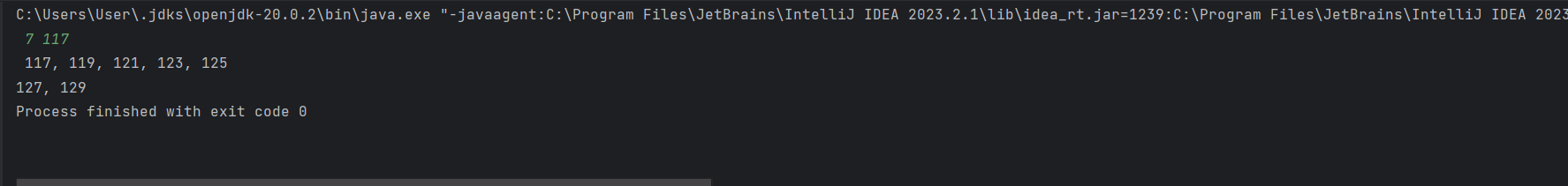
Tabel .3 Soal 3

### Source Code

|  |  |
| --- | --- |
| 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 | import java.util.Scanner;  public class PRAK103\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args) {  Scanner namy = new Scanner(System.*in*);  System.*out*.print(" ");  int N = namy.nextInt();  System.*out*.print(" ");  int bilanganpertama = namy.nextInt();  int baris = 1;  int bilangan = bilanganpertama;  do {  if (bilangan % 2 != 0) {  System.*out*.print(bilangan);  if (baris % 5 != 0 && baris != N) {  System.*out*.print(", ");  }  bilangan += 2;  }  if (baris % 5 == 0) {  System.*out*.println();  }  baris++;  } while (baris <= N);  } } |

Tabel .3 Source Code Soal 3

### Output Program



Gambar .3 Output Screenshoot Soal 3

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 3 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 27 : {}, penanda awal dan akhir *block code*.

Line 5 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 6 : *Scanner* untuk menginput data.

Line 7, 9, 15, 17, dan 22 : *System.out.print*  mencetak sesuatu dari program sebagai *output.*

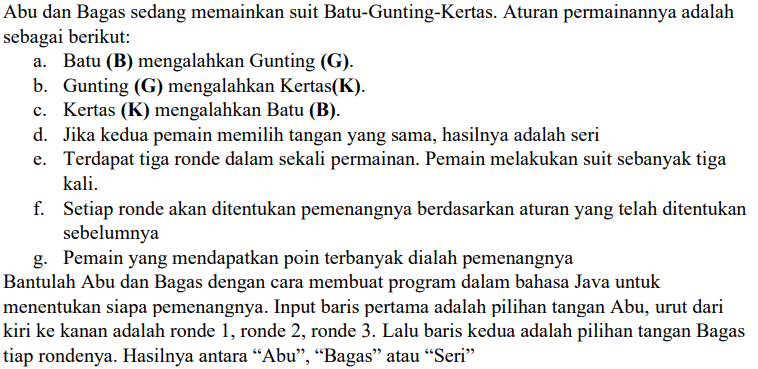
Line 8, 10 - 12: *int,* tipe data *integer* atau menerima bilangan bulat.

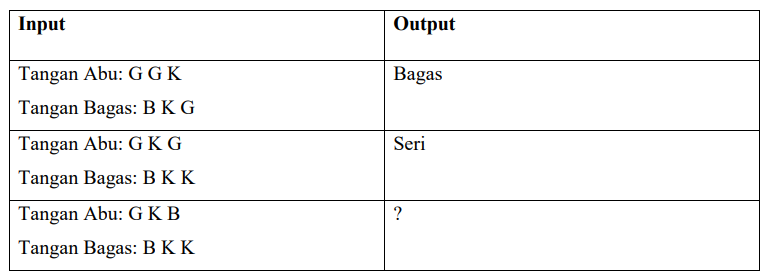
Line 13 dan 25 : *Do While* untuk perulangan terleb.ih dahulu, lalu memerika kondisi atau syaratnya

Line 14 dan 21 : *If Else* untuk memberikan kondisi *true* atau *false.*

Line 24 : ++ atau *increment*  untuk menambah variable satu angka.

## SOAL 4





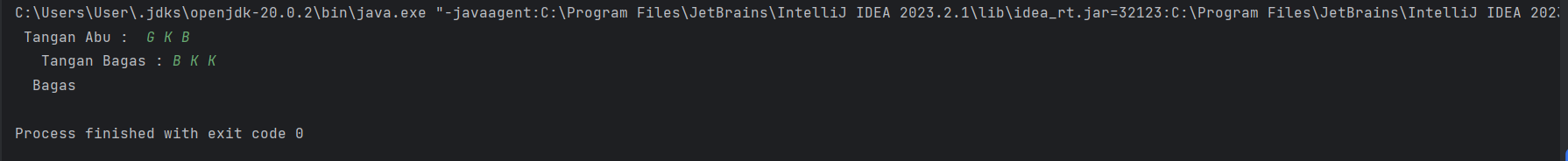
Tabel .4 Soal 3

### Source Code

|  |  |
| --- | --- |
| 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 | import java.util.Scanner;  public class PRAK104\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String[] args) {  Scanner namy = new Scanner(System.*in*);   System.*out*.print(" Tangan Abu : ");  char abuRonde1 = namy.next().charAt(0);  System.*out*.print(" ");  char abuRonde2 = namy.next().charAt(0);  System.*out*.print(" ");  char abuRonde3 = namy.next().charAt(0);  System.*out*.print(" Tangan Bagas : ");  char bagasRonde1 = namy.next().charAt(0);  System.*out*.print(" ");  char bagasRonde2 = namy.next().charAt(0);  System.*out*.print(" ");  char bagasRonde3 = namy.next().charAt(0);   int abuScore = 0;  int bagasScore = 0;   if ((abuRonde1 == 'B' && bagasRonde1 == 'G') ||  (abuRonde1 == 'G' && bagasRonde1 == 'K') ||  (abuRonde1 == 'K' && bagasRonde1 == 'B')) {  abuScore++;  } else if ((abuRonde1 == 'G' && bagasRonde1 == 'B') ||  (abuRonde1 == 'K' && bagasRonde1 == 'G') ||  (abuRonde1 == 'B' && bagasRonde1 == 'K')) {  bagasScore++;  }  if ((abuRonde2 == 'B' && bagasRonde2 == 'G') ||  (abuRonde2 == 'G' && bagasRonde2 == 'K') ||  (abuRonde2 == 'K' && bagasRonde2 == 'B')) {  abuScore++;  } else if ((abuRonde2 == 'G' && bagasRonde2 == 'B') ||  (abuRonde2 == 'K' && bagasRonde2 == 'G') ||  (abuRonde2 == 'B' && bagasRonde2 == 'K')) {  bagasScore++;  }  if ((abuRonde3 == 'B' && bagasRonde3 == 'G') ||  (abuRonde3 == 'G' && bagasRonde3 == 'K') ||  (abuRonde3 == 'K' && bagasRonde3 == 'B')) {  abuScore++;  } else if ((abuRonde3 == 'G' && bagasRonde3 == 'B') ||  (abuRonde3 == 'K' && bagasRonde3 == 'G') ||  (abuRonde3 == 'B' && bagasRonde3 == 'K')) {  bagasScore++;  }  if (abuScore > bagasScore) {  System.*out*.println("Abu");  } else if (abuScore < bagasScore) {  System.*out*.println("Bagas");  } else {  System.*out*.println("SERI");  }  } } |

Tabel .4 Source Code Soal 4

### Output Program



Gambar .4 Output Screenshoot Soal 4

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 3 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 77 : {}, penanda awal dan akhir *block code*.

Line 5 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 6 : *Scanner* untuk menginput data.

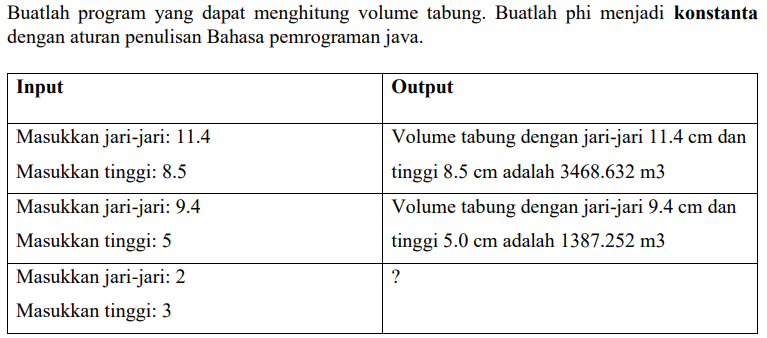
Line 8, 10, 12, 14, 16, 18, 70,72, dan 74: *System.out.print*  mencetak sesuatu dari program sebagai *output.*

Line 10 : *char* tipe data *char* atau menerima bilangan dan symbol.

Line 24-69 : *If Else* untuk memberikan kondisi *true* atau *false.*

Line 30, 37, 45, 52, 60, dan 67 : ++ atau *increment*  untuk menambah variable satu angka.

## SOAL 5



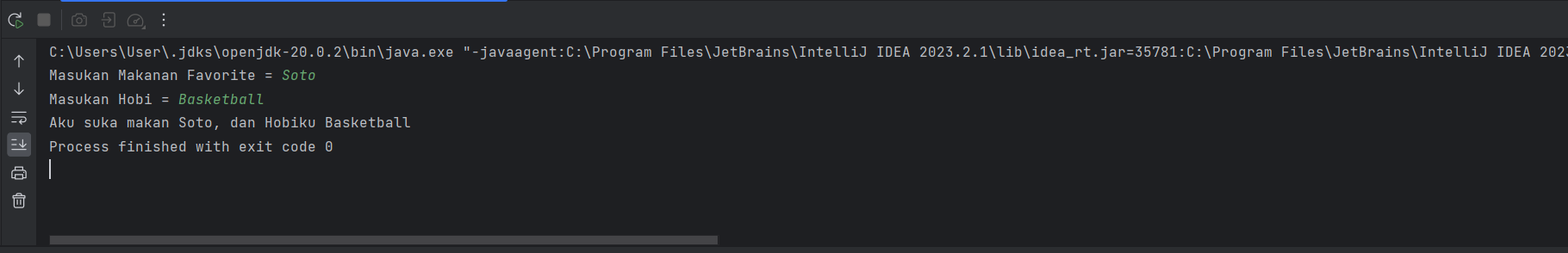
Tabel .5 Soal 5

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | import java.util.Scanner; public class PRAK105\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI {  public static void main(String [] args){  float jarijari, tinggi, rumus;  double pi = 3.14;  Scanner namy =new Scanner(System.*in*);  System.*out*.print("Masukkan Jari-Jari : ");  jarijari = namy.nextFloat();  System.*out*.print("Masukkan Tinggi : ");  tinggi = namy.nextFloat();  rumus = (float) (pi \* (jarijari \*jarijari\*tinggi));   System.*out*.print("Volume tabung dengan jari-jari "+jarijari+" dan tinggi "+tinggi+" cm adalah "+rumus+" m3");   } } |

Tabel .5 Source Code Soal 5

### Output Program



Gambar .5 Output Screenshoot Soal 5

### Pembahasan

Line 1 : *Import,*  perintah untuk program memasukkan bahasa java sehingga aktif.

Line 2 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 22 : {}, penanda awal dan akhir *block code*.

Line 4 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 5 : *float* tipe data *float* atau menerima bilangan desimal.

Line 6 : *double* tipe data *double* atau menerima bilangan desimal.

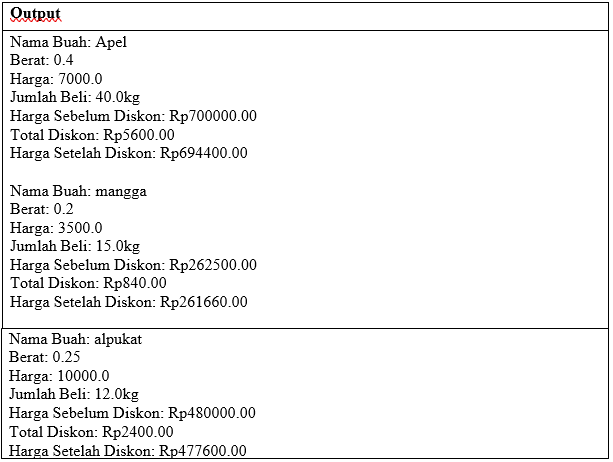
Line 7, : *Scanner* untuk menginput data.

Line 8, 10, dan 15 : *System.out.print*  mencetak sesuatu dari program sebagai *output.*

# MODUL 2: CLASS DAN OBJEK

## SOAL 1

Reno membeli buah-buahan di toko buah. Setiap buah memiliki diskon atau potongan harga 2% per 4kg. Apel memiliki harga Rp7.000 per 0.4kg. Mangga memiliki harga Rp3500 per 0.2kg. Alpukat memiliki harga Rp10.000 per 0.25kg. Dari informasi tersebut, buatlah sebuah program dalam bahasa java yang mengimplementasikan konsep **class** dan **object.** Ketentuan : Di **class main** terdapat 3 buah instansiasi objek dan **tidak boleh ada operasi apapun selain** instansiasi objek dan memanggil method

****

Tabel .1 Soal 1

### Source Code

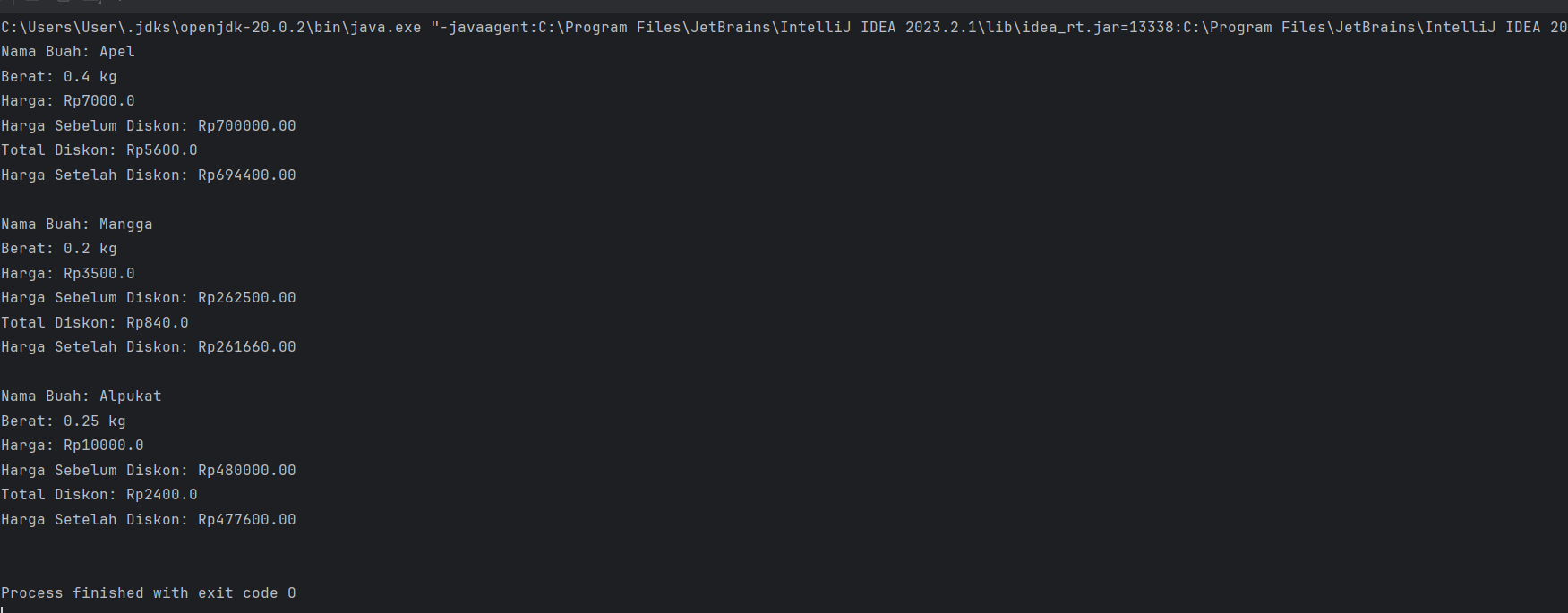
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26  27  28  29 30  31  32  33  34 | package PRAK201\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI;  public class Buah {  String nama;  double berat,harga, jumlahBeli,hargaSebelumDiskon,hargaSesudahDiskon,totalDiskon ;  public Buah(String namaBuah, double beratBuah, int hargaBuah, double jumlahBeli){  this.nama = namaBuah;  this.berat = beratBuah;  this.harga = hargaBuah;  this.jumlahBeli = jumlahBeli;  this.hargaSebelumDiskon = harga\*(jumlahBeli/beratBuah);  this.totalDiskon = (int)(jumlahBeli/4)\*(4\*harga)\*0.02;  this.hargaSesudahDiskon = hargaSebelumDiskon-totalDiskon;  }  public void display(){  System.out.println("Nama Buah: "+nama);  System.out.println("Berat: "+berat+" kg");  System.out.println("Harga: Rp"+harga);  System.out.println("Harga Sebelum Diskon: Rp"+ String.format ("%.2f",hargaSebelumDiskon));  System.out.println("Total Diskon: Rp"+totalDiskon);  System.out.println("Harga Setelah Diskon: Rp"+String.format("%.2f",hargaSesudahDiskon)+"\n");  }  } |

Tabel .1 Source Code Soal 1 Buah

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package PRAK201\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI; public class Main {  public static void main(String[] args) {  Buah Apel1= new Buah("Apel", 0.4, 7000, 40);  Buah Mangga1 = new Buah("Mangga", 0.2, 3500, 15);  Buah Alpukat1 = new Buah("Alpukat", 0.25, 10000, 12);   Apel1.display();  Mangga1.display();  Alpukat1.display();  } } |

Tabel .1 Source Code Soal 1 Main

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

#### Buah

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2, 9, dan 24 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 13 : {}, penanda awal dan akhir *block code*.

Line 4 : *String*, tipe data teks yang berisikan karakter lebih.

Line 5 : *double* tipe data *double* atau menerima bilangan desimal.

Line 25-31 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

Line 28 dan 31 : *String.format(),* sebuah format untuk meng-edit sebuah *string.*

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

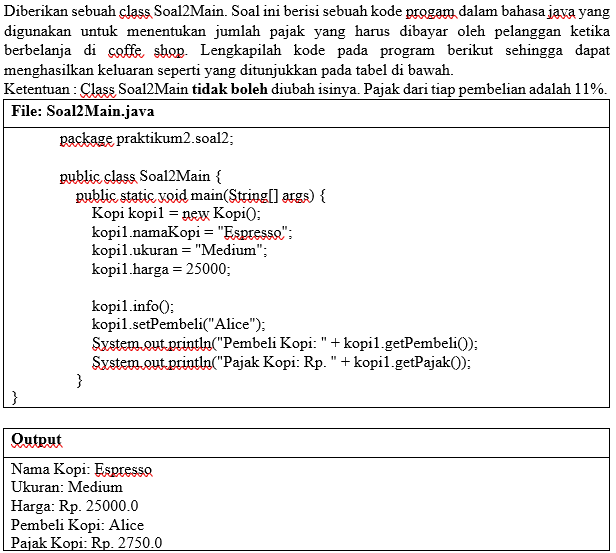
Line 2 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 13 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 9-11 : *Display,* merupakan fungsi menampilkan data yang diprogram.

## SOAL 2



Tabel .2 Soal 2

### Source Code

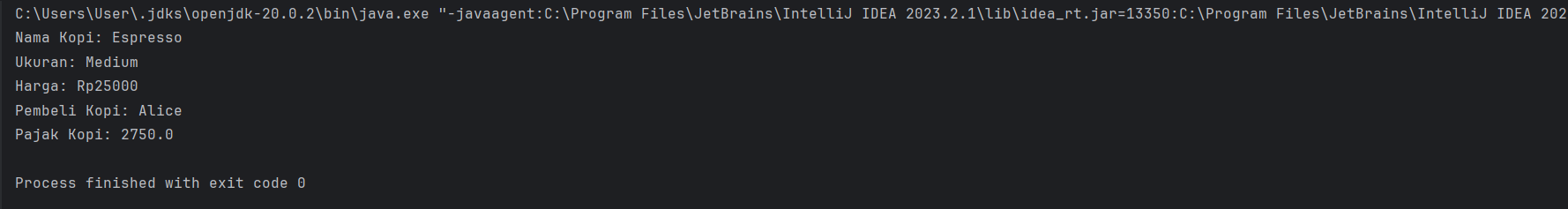
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23 | package PRAK202\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI; public class Kopi {  String namaKopi,ukuran,namaPembeli;  int harga;  double pajak;  public void info(){  System.*out*.println("Nama Kopi: "+namaKopi);  System.*out*.println("Ukuran: "+ukuran);  System.*out*.println("Harga: Rp"+harga);  }  public void setPembeli(String namaPembeli){  this.namaPembeli=namaPembeli;  }  public String getPembeli(){  return namaPembeli;  }  public double pajak(){  pajak=harga\*0.11;  return pajak;  }  } |

Tabel .2 Source Code Soal 2 Kopi

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | package PRAK202\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI; public class Soal2Main {  public static void main(String[] args) {  Kopi kopi1 = new Kopi();  kopi1.namaKopi = "Espresso";  kopi1.ukuran = "Medium";  kopi1.harga= 25000;   kopi1.info();  kopi1.setPembeli("Alice");  System.*out*.println("Pembeli Kopi: "+kopi1.getPembeli());  System.*out*.println("Pajak Kopi: "+ kopi1.pajak());  } } |

Tabel .2 Source Code Soal 2 Main

### Output Program



Gambar .2 Output Screenshoot Soal 2

### Pembahasan

#### Kopi

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2, 11, 14, dan 17 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 23 : {}, penanda awal dan akhir *block code*.

Line 3 : *String*, tipe data teks yang berisikan karakter lebih.

Line 4 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 5 : *double* tipe data *double* atau menerima bilangan desimal.

Line 7-9 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

Line 15 dan 20 : *return* untuk mengembalikan nilai.

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

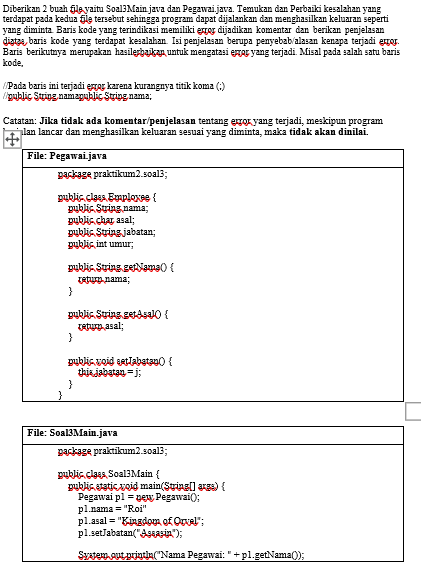
Line 2 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 15 : {}, penanda awal dan akhir *block code*.

Line 3 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

Line 12 dan 13 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

## SOAL 3



Tabel .3 Soal 3

### Source Code

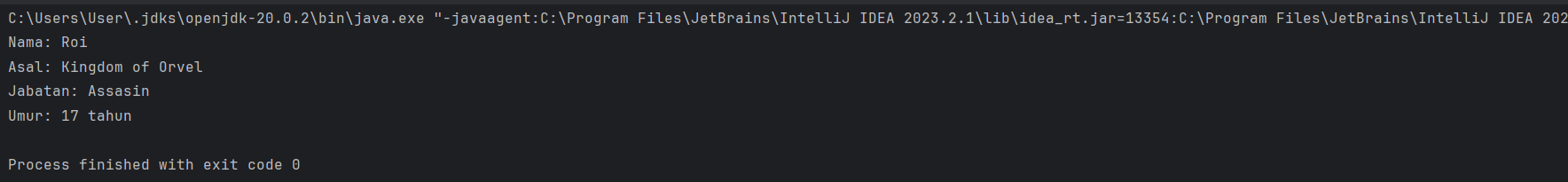
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | package PRAK203\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI; public class Pegawai {  public String nama;  public String asal;//tipe data berbeda yang harusnya string tetapi char.  public String jabatan;  public int umur;  public String getNama() {  return nama;  }  //tipe data yang harusnya string tetapi char.  public String getAsal() {  return asal;  }  public void setJabatan(String j) {  this.jabatan = j;  } } |

Tabel .3 Source Code Soal 3 Pegawai

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26 | package PRAK203\_2210817310001\_MUHAMMADRYANRIZKYRAHMADI; public class Soal3Main {  public static void main(String[] args){  Pegawai p1 = new Pegawai();  //p1.nama = "Roi" erornya karena tidak ada ;  p1.nama = "Roi";  p1.asal = "Kingdom of Orvel";  p1.setJabatan ("Assasin");  //p1.jabatan = "Assasin"; terjadi error karena deklarasi tidak sesuai dengan yang ada pada class Pegawai, jadi ditambahkan method setJabatan  p1.umur = 17; //Belum terdapat pendeklarasian pada property umur   //Output yang diperlukan adalah "Nama: Roi", jadi kata Pegawai dihapus aja.  System.*out*.println("Nama: " + p1.getNama());  System.*out*.println("Asal: " + p1.getAsal());  System.*out*.println("Jabatan: " + p1.jabatan);  //Untuk mencetak kata tahun diakhir kalimat, jadi tambahkan (+ " tahun")  System.*out*.println("Umur: " + p1.umur + " tahun");  } } |

Tabel .3 Source Code Soal 3 Main

### Output Program



Gambar .3 Output Screenshoot Soal 3

### Pembahasan

#### Pegawai

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2-8, 12, dan 15 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 2 dan 18 : {}, penanda awal dan akhir *block code*.

Line 3-8 : *String*, tipe data teks yang berisikan karakter lebih.

Line 7 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 4 dan 11: // atau *comments* membuat note atau komentar pada *workspace*.

Line9 dan 13: *return* untuk mengembalikan nilai.

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 26 : {}, penanda awal dan akhir *block code*.

Line 4 : *Static*, fungsi yang dapat dieksekusi langsung tanpa harus melakukan instansiasi objek. *Void*, tidak ada nilai balik atau *return.* *String[] args*, adalah parameters.

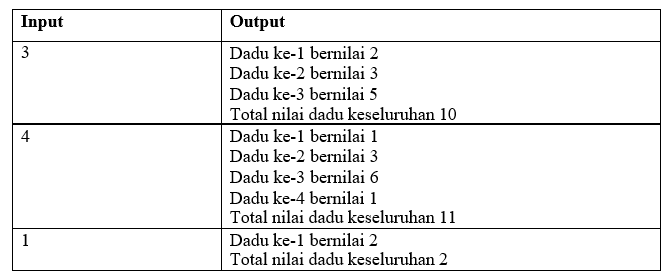
Line 6, 10, 16, dan 22 : // atau *comments* membuat note atau komentar pada *workspace*.

Line 18-20, dan 24 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

# MODUL 3: ENKAPULASI DAN COLLECTION

## 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 Soal 1

### Source Code

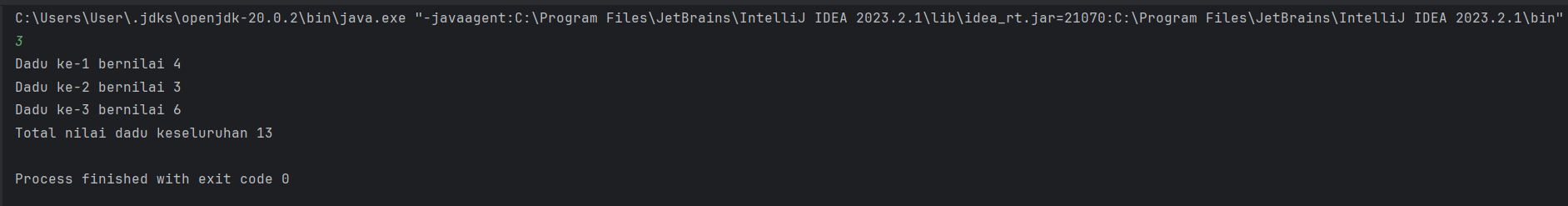
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26  27  28 | package PRAK301\_soal1;  import java.util.LinkedList;  public class Dadu {  public int input;  public int min = 1;  public int max = 6;  int total;   public void setInput(int input){this.input = input;}   void acakNilai(){  LinkedList<Integer> ll = new LinkedList<Integer>();  ll.add(1);ll.add(2);ll.add(3);ll.add(4);ll.add(5);ll.add(6);  for (int i = 0; i < input; i++){  int random\_int = (int)Math.*floor*(Math.*random*()\*(max-min+1)+min);  System.*out*.println("Dadu ke-" + (i + 1) + " bernilai "+ random\_int);  total += random\_int;  }  System.*out*.println("Total nilai dadu keseluruhan " +total);  } } |

Tabel .1 Source Code Soal 1 Dadu

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | package PRAK301\_soal1;  import java.util.\*;  public class Main {  public static void main(String[] args) {  Scanner input = new Scanner(System.*in*);  Dadu dd = new Dadu();  int masukkan = input.nextInt();   dd.setInput(masukkan);  dd.acakNilai();  } } |

Tabel .1 Source Code Soal 1 Main

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

#### Dadu

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 5-9, dan 11: *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 5 dan 28 : {}, penanda awal dan akhir *block code*.

Line 6-9 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 14 : *LinkedList,*  suatu cara untuk menyimpan data dengan struktur.

Line 20 : *Math.floor,* sebuah metode mengembalikan nilai *double*  sama dengan bilangan bulat terbesar.

Line 25 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 5-6 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 5 dan 14 : {}, penanda awal dan akhir *block code*.

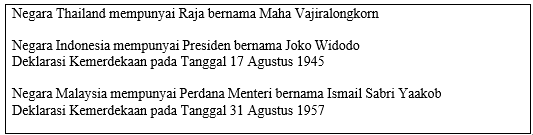
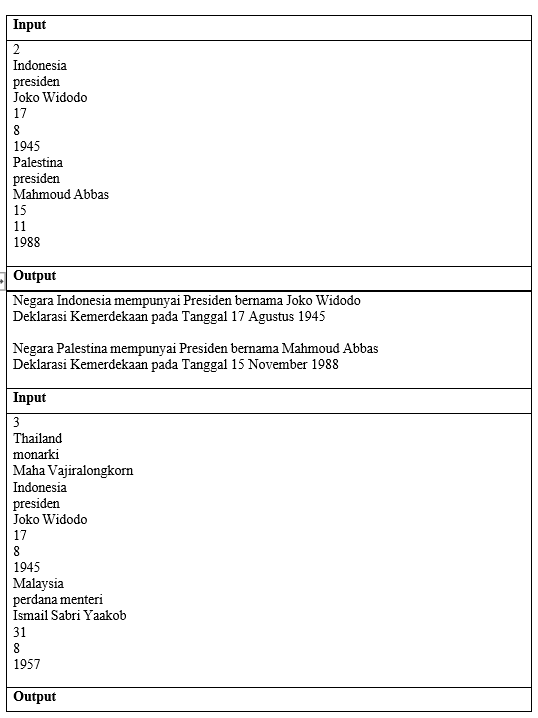
Line 7, : *Scanner* untuk menginput data.

Line 9 : *int,* tipe data *integer* atau menerima bilangan bulat.

## 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 Soal 2

### Source Code

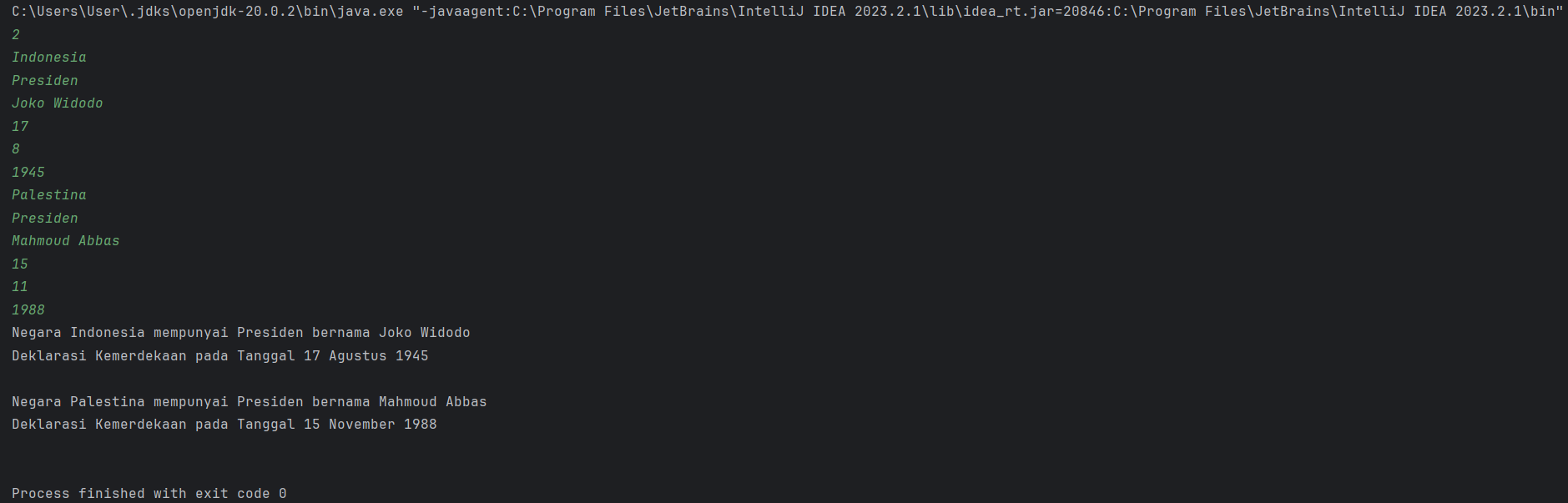
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70 | package PRAK302\_soal2;  import java.util.\*;  public class Negara {  private String nama, jenis, nama\_p;  private int tanggal, bulan, tahun;   // Konstruktor  public Negara(String nama, String jenis, String nama\_p, int tanggal, int bulan, int tahun){  this.nama = nama;  this.jenis = jenis;  this.nama\_p = nama\_p;  this.tanggal = tanggal;  this.bulan = bulan;  this.tahun = tahun;  }   // Encapsulation  public int getBulan(){  return bulan;  }   void getData(){  Scanner scn = new Scanner(System.*in*);  nama=scn.nextLine();  jenis=scn.nextLine();  nama\_p=scn.nextLine();  if (jenis.equalsIgnoreCase("presiden")||jenis.equalsIgnoreCase("perdana menteri")) {  tanggal = scn.nextInt();  bulan = scn.nextInt();  tahun = scn.nextInt();  }  }   void tampilkan(){  HashMap<Integer, String> month = new HashMap<Integer,String>();  month.put(1, "Januari");  month.put(2, "Februari");  month.put(3, "Maret");  month.put(4, "April");  month.put(5, "Mei");  month.put(6, "Juni");  month.put(7, "Juli");  month.put(8, "Agustus");  month.put(9, "September");  month.put(10, "Oktober");  month.put(11, "November");  month.put(12, "Desember");  if (jenis.equalsIgnoreCase("presiden")||jenis.equalsIgnoreCase("perdana menteri")) {  System.*out*.println("Negara "+nama+" mempunyai "+jenis+" bernama "+nama\_p);}  else if (jenis.equalsIgnoreCase("monarki")){  System.*out*.println("Negara "+nama+" mempunyai Raja"+" bernama "+nama\_p+"\n");  }  if (jenis.equalsIgnoreCase("presiden")||jenis.equalsIgnoreCase("perdana menteri")) {  System.*out*.println("Deklarasi Kemerdekaan pada Tanggal "+tanggal+" "+month.get(getBulan())+" "+tahun+"\n");}  } } |

Tabel .2 Source Code Soal 2 Negara

|  |  |
| --- | --- |
| 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 | package PRAK302\_soal2;  import java.util.\*; import java.util.Scanner;  public class Main {  public static void main(String[] args) {  String nama = "", jenis="", nama\_p="";  int tanggal=0, bulan=0, tahun=0;  Scanner input = new Scanner(System.*in*);  Negara negara = new Negara(nama, jenis, nama\_p, tanggal, bulan, tahun);  LinkedList<Negara> llNegara = new LinkedList<Negara>();   int ulang;  ulang = input.nextInt();   for (int i=0; i<ulang; i++){  Negara c = new Negara(nama, jenis, nama\_p, tanggal, bulan, tahun);  c.getData();  llNegara.add(c);  }   for (int i = 0; i < llNegara.size(); i++){  Negara nIndex = llNegara.get(i);  nIndex.tampilkan();  }  } } |

Tabel .2 Source Code Soal 2 Main

### Output Program



Gambar .2 Output Screenshoot Soal 2

### Pembahasan

#### Negara

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 5, 10, dan 22 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6-7 : *Private,* kunci pada java yang dapat diakses oleh kelas itu sendiri.

Line 5 dan 70 : {}, penanda awal dan akhir *block code*.

Line 10 : *String*, tipe data teks yang berisikan karakter lebih.

Line 11 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 41 : *HashMap*, adalah *class* yang berisi sekumpulan pasangan nilai atau  *value*  dan *key.*

Line 43-54 : *month.put,* metode menambahkan *key* bulan.

Line 55 dan 60 : *If Else* untuk memberikan kondisi *true* atau *false.*

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3-4: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 6-7 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6 dan 31 : {}, penanda awal dan akhir *block code*.

Line 8 : *String*, tipe data teks yang berisikan karakter lebih.

Line 9 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 10, : *Scanner* untuk menginput data.

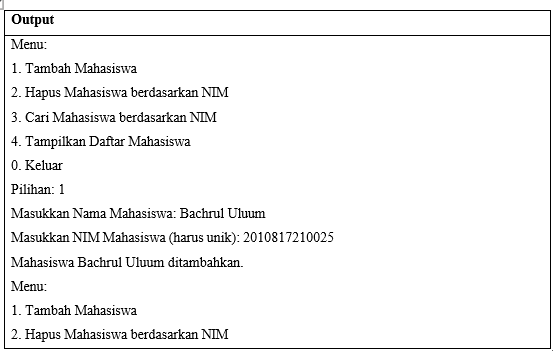
Line 13-14 : *LinkedList,*  suatu cara untuk menyimpan data dengan struktur.

Line 19 dan 26 : *for,*  untuk perulangan pada *statement* secara berulang – ulang selama terpenuhi.

# 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 Soal 3

### Source Code

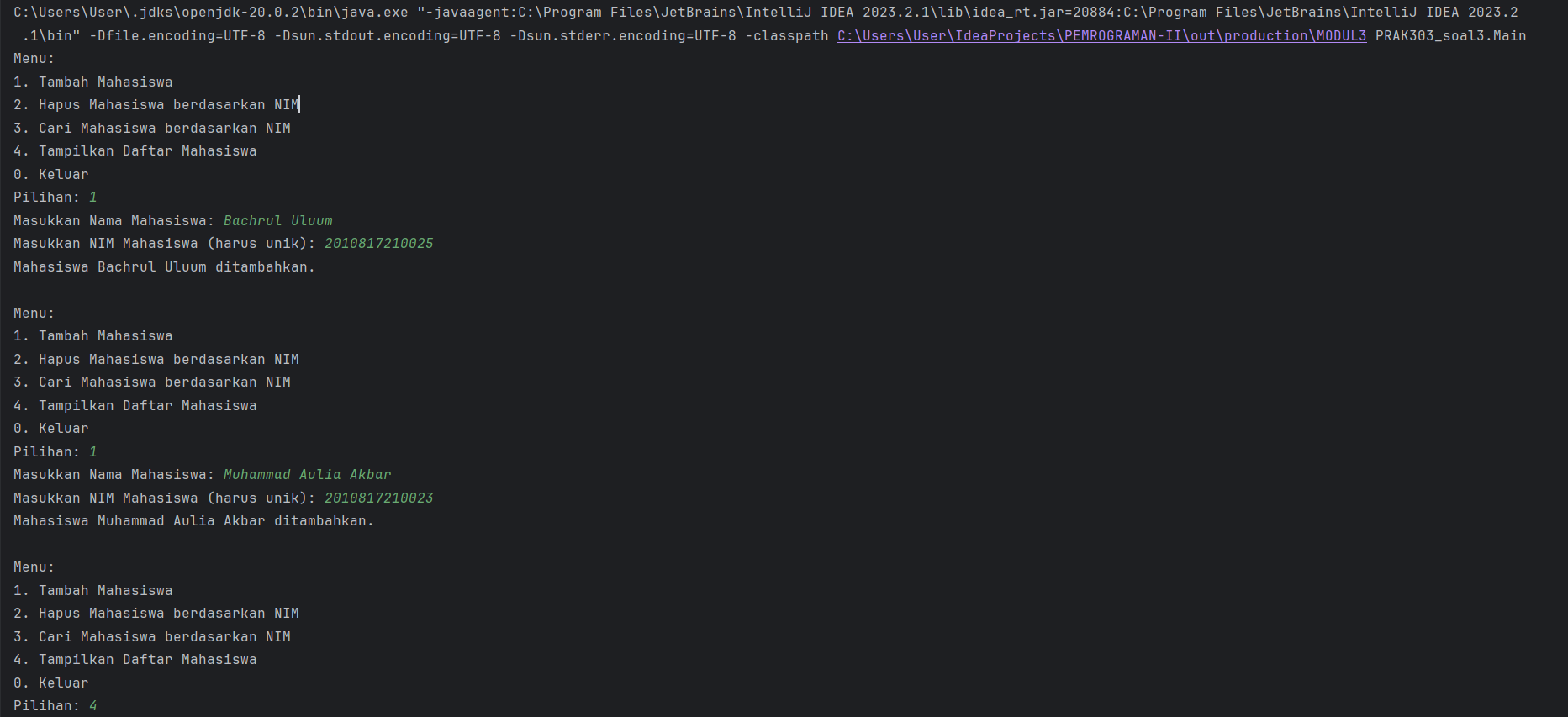
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | package PRAK303\_soal3;  class Mahasiswa {  private String nama;  private String nim;   public Mahasiswa(String nama, String nim) {  this.nama = nama;  this.nim = nim;  }   public String getNama() {  return nama;  }   public String getNim() {  return nim;  } } |

Tabel .3 Source Code Soal 3 Kopi

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  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 | package PRAK303\_soal3;  import java.util.ArrayList; import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner sc = new Scanner(System.*in*);  ArrayList<Mahasiswa> mahasiswaList = new ArrayList<>();   while (true) {  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: ");  int pilihan = sc.nextInt();   if (pilihan == 0) {  mahasiswaList.clear();  System.*out*.println("Terima kasih!");  break;  } else if (pilihan == 1) {  System.*out*.print("Masukkan Nama Mahasiswa: ");  sc.nextLine();  String nama = sc.nextLine();  System.*out*.print("Masukkan NIM Mahasiswa (harus unik): ");  String nim = sc.nextLine();   boolean isNimExist = false;  for (Mahasiswa mahasiswa : mahasiswaList) {  if (mahasiswa.getNim().equals(nim)) {  isNimExist = true;  break;  }  }   if (isNimExist) {  System.*out*.println("NIM sudah digunakan.");  } else {  Mahasiswa mahasiswa = new Mahasiswa(nama, nim);  mahasiswaList.add(mahasiswa);  System.*out*.println("Mahasiswa " + nama + " ditambahkan.");  }  } else if (pilihan == 2) {  System.*out*.print("Masukkan NIM Mahasiswa yang akan dihapus: ");  String nim = sc.next();   boolean isNimExist = false;  for (Mahasiswa mahasiswa : mahasiswaList) {  if (mahasiswa.getNim().equals(nim)) {  isNimExist = true;  mahasiswaList.remove(mahasiswa);  System.*out*.println("Mahasiswa dengan NIM " + nim + " dihapus.");  break;  }  }   if (!isNimExist) {  System.*out*.println("NIM tidak ditemukan.");  }  } else if (pilihan == 3) {  System.*out*.print("Masukkan NIM Mahasiswa yang akan dicari: ");  String nim = sc.next();   boolean isNimExist = false;  for (Mahasiswa mahasiswa : mahasiswaList) {  if (mahasiswa.getNim().equals(nim)) {  isNimExist = true;  System.*out*.println("NIM: " + mahasiswa.getNim() + ", Nama: " + mahasiswa.getNama());  break;  }  }  if (!isNimExist) {  System.*out*.println("NIM tidak ditemukan.");  }  } else if (pilihan == 4) {  System.*out*.println("Daftar Mahasiswa:");  for (Mahasiswa mahasiswa : mahasiswaList) {  System.*out*.println("NIM: " + mahasiswa.getNim() + ", Nama: " + mahasiswa.getNama());  }  } else {  System.*out*.println("Pilihan tidak valid.");  }  System.*out*.println();  }  } } |

Tabel 38.3 Source Code Soal 3 Main

### Output Program



Gambar 16.3 Output Screenshoot Soal 3

### Pembahasan

#### Kopi

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 4-5 : *Private,* kunci pada java yang dapat diakses oleh kelas itu sendiri.

Line 7,12, dan 16 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 5 dan 70 : {}, penanda awal dan akhir *block code*.

Line 4-5, 7, 12, dan 16 : *String*, tipe data teks yang berisikan karakter lebih.

Line 13 dan 17 : *return* untuk mengembalikan nilai.

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3-4: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 6-7 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6 dan 107 : {}, penanda awal dan akhir *block code*.

Line 8, : *Scanner* untuk menginput data.

Line 9: *ArrayList,*  untuk menyimpan beberapa nilai dalam satu variable.

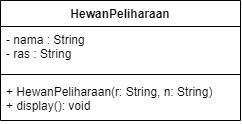
Line 12 : *While,* perulangan yang selalu dilakukan, jika perulangan tidak diketahui.

Line 13-23, 28, 31, 35, 48, 54, 58, 67, 74, 78, 86, 98, 102, dan 104 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

Line 26 dan 30 : *If Else* untuk memberikan kondisi *true* atau *false.*

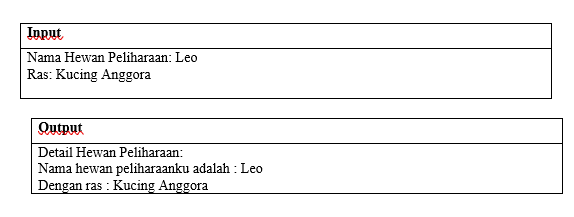
# MODUL 4: PEWARISAN/INHERITANCE & DIAGRAM KELAS

## SOAL 1

Diberikan class diagram seperti berikut: (isi program harus sesuai dengan class diagram)Buatlah kelas dengan nama Dadu.

Implementasikan class diagram diatas menjadi program menggunakan Bahasa pemrograman java dengan ketentuan seperti berikut:

1. Inisiasi objek **harus** menggunakan konstruktor.
2. Output ditampilkan **dengan hanya** memanggil method display.
3. Tidak boleh ada fungsi mencetak di class main.



Tabel .1 Soal 1

### Source Code

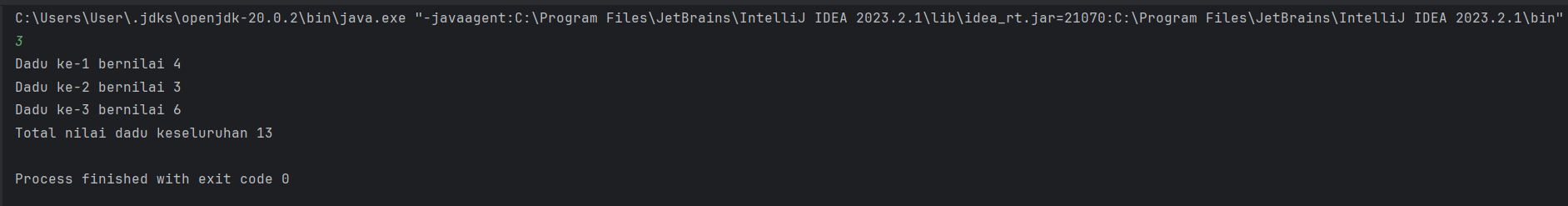
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21 | package soal1;  public class peliharaan {  private String nama;  private String ras;   public peliharaan(String r, String n) {  ras = r;  nama = n;  }  public void display() {  System.*out*.println("Detail Hewan Peliharaan:");  System.*out*.println("Nama hewan peliharaanku adalah : " + nama);  System.*out*.println("Dengan ras : " + ras);  }  //ryann } |

Tabel .1 Source Code Soal 1 Peliharaan

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20 | package soal1;  import java.util.Scanner; public class Main {  public static void main(String[] args) {  Scanner isian = new Scanner(System.*in*);   System.*out*.print("Nama Hewan Peliharaan: ");  String namaHewan = isian.nextLine();   System.*out*.print("Ras: ");  String rasHewan = isian.nextLine();   peliharaan hewanPeliharaan = new peliharaan(rasHewan, namaHewan);  isian.close();  hewanPeliharaan.display();  } } |

Tabel .1 Source Code Soal 1 Man

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

#### Peliharaan

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 3 dan 11: *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 21 : {}, penanda awal dan akhir *block code*.

Line 4 dan 5 : *String*, tipe data teks yang berisikan karakter lebih.

Line 12, 14, dan 16 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 5 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

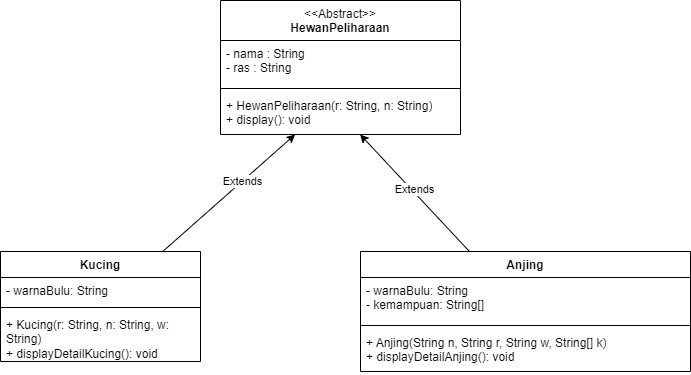
Line 5 dan 20 : {}, penanda awal dan akhir *block code*.

Line 7, : *Scanner* untuk menginput data.

Line 9 dan 12 : *String*, tipe data teks yang berisikan karakter lebih.

Line 9 : *int,* tipe data *integer* atau menerima bilangan bulat.

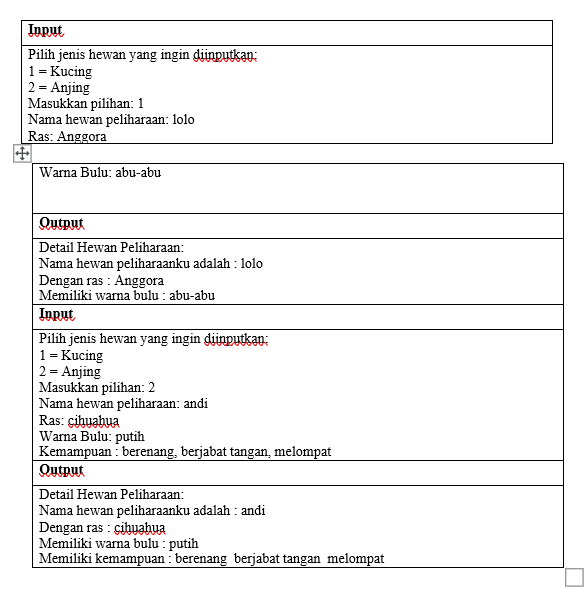
# SOAL 2



Implementasikan class diagram diatas menjadi program menggunakan Bahasa pemrograman java dengan ketentuan seperti berikut:

* 1. Inisiasi objek harus menggunakan konstruktor.
  2. Isi baris program harus sesuai dengan apa yang digambarkan di dalam kelas diagram.
  3. Menampilkan detail dari class Kucing maupun Anjing harus melalui method yang ada, tidak boleh menggunakan fungsi cetak dari java

Isi detail dari class Kucing dan Anjing harus memanggil method display pada parent class terlebih dahulu.



Tabel .2 Soal 2

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20 | package soal2;  public abstract class HewanPeliharaan {  private String nama;  private String ras;   public HewanPeliharaan(String r, String n) {  ras = r;  nama = n;  }   public void display() {  System.*out*.println("Detail Hewan Peliharaan:");  System.*out*.println("Nama hewan peliharaanku adalah : " + nama);  System.*out*.println("Dengan ras : " + ras);  } } |

Tabel .2 Source Code Soal 2 HewanPeliharaan

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24 | package soal2;  public class Anjing extends HewanPeliharaan {  private String warnaBulu;  private String[] kemampuan;   public Anjing(String n, String r, String w, String[] k) {  super(r, n);  warnaBulu = w;  kemampuan = k;  }   public void displayDetailAnjing() {  super.display();  System.*out*.println("Memiliki warna bulu : " + warnaBulu);  System.*out*.print("Kemampuan : ");  for (String k : kemampuan) {  System.*out*.print(" "+k);  }  } } |

Tabel .2 Source Code Soal 2 Anjing

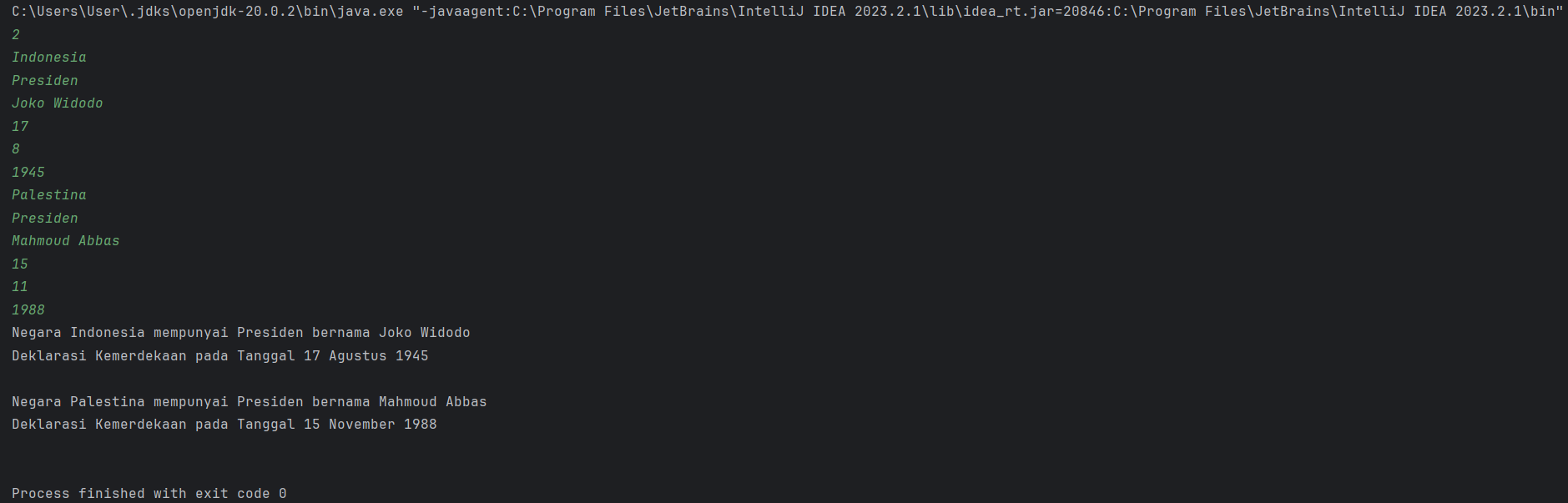
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package soal2;  public class Kucing extends HewanPeliharaan {  private String warnaBulu;   public Kucing(String r, String n, String w) {  super(r, n);  warnaBulu = w;  }   public void displayDetailKucing() {  super.display();  System.*out*.println("Memiliki warna bulu : " + warnaBulu);  } } |

Tabel .2 Source Code Soal 2 Kucing

|  |  |
| --- | --- |
| 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 | package soal2;  import java.util.Scanner;  public class Main {  public static void main(String[] args) {  Scanner masukan = new Scanner(System.*in*);  System.*out*.println("Pilih jenis hewan yang ingin diinputkan:");  System.*out*.println("1 = Kucing");  System.*out*.println("2 = Anjing");  System.*out*.print("Masukkan pilihan: ");  int pilihan = masukan.nextInt();   if (pilihan == 1) {  masukan.nextLine();  System.*out*.print("Nama hewan peliharaan: ");  String namaKucing = masukan.nextLine();  System.*out*.print("Ras: ");  String rasKucing = masukan.nextLine();  System.*out*.print("Warna Bulu: ");  String warnaBuluKucing = masukan.nextLine();   Kucing kucing = new Kucing(rasKucing, namaKucing, warnaBuluKucing);  kucing.displayDetailKucing();  } else if (pilihan == 2) {  masukan.nextLine();  System.*out*.print("Nama hewan peliharaan: ");  String namaAnjing = masukan.nextLine();  System.*out*.print("Ras: ");  String rasAnjing = masukan.nextLine();  System.*out*.print("Warna Bulu: ");  String warnaBuluAnjig = masukan.nextLine();  System.*out*.print("Kemampuan: ");  String kemampuanStr = masukan.nextLine();  String[] kemampuan = kemampuanStr.split(",");   Anjing anjing = new Anjing(namaAnjing, rasAnjing, warnaBuluAnjig, kemampuan);  anjing.displayDetailAnjing();  } else {  System.*out*.println("Pilihan tidak valid");  }   masukan.close();  } } |

Tabel .2 Source Code Soal 2 Main

### Output Program



Gambar .2 Output Screenshoot Soal 2

### Pembahasan

#### HewanPeliharaan

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 2: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 3, 7, dan 11: *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6-7 : *Private,* kunci pada java yang dapat diakses oleh kelas itu sendiri.

Line 5 dan 20 : {}, penanda awal dan akhir *block code*.

Line 10 : *String*, tipe data teks yang berisikan karakter lebih.

Line 20 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

#### Anjing

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3-4: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 3,7, dan 14 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6 dan 31 : {}, penanda awal dan akhir *block code*.

Line 8 : *String*, tipe data teks yang berisikan karakter lebih.

Line 7, 14, dan 21 : *System.out.println*  mencetak sesuatu dari program sebagai *output.*

#### Kucing

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3, 6, dan 11 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6 dan 16 : {}, penanda awal dan akhir *block code*.

Line 8 : *String*, tipe data teks yang berisikan karakter lebih.

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3-4: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 5 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6, : *Scanner* untuk menginput data.

Line 5 dan 51 : {}, penanda awal dan akhir *block code*.

Line 7 - 49 : *String*, tipe data teks yang berisikan karakter lebih.

Line 7 - 49 : *int,* tipe data *integer* atau menerima bilangan bulat.

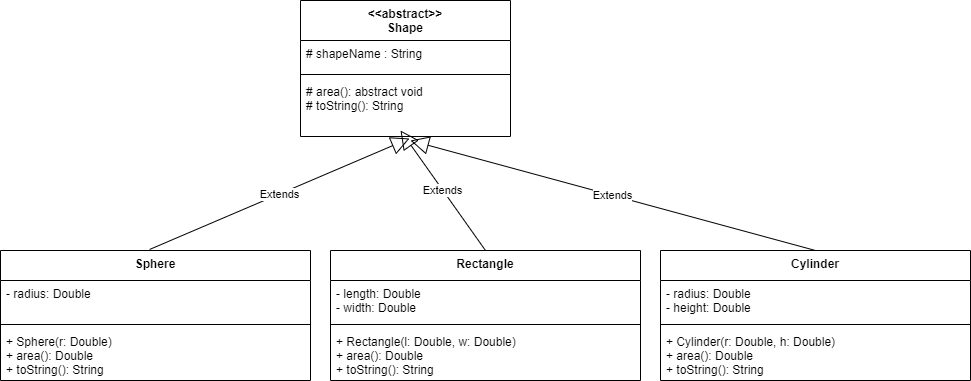
Line 15 dan 29 : *If Else* untuk memberikan kondisi *true* atau *false.*

# MODUL 5: POLIMORFISME

## SOAL

Pada praktikum kali ini anda akan diminta untuk membuat sebuah program yang dapat menghitung banyaknya liter cat yang digunakan untuk mewarnai bentuk ruang yang beragam.

Buatlah sebuah hierarki kelas abstrak Shape dimana memiliki 3 kelas anak yaitu Sphere, Rectangle, dan Cylinder seperti ditunjukkan oleh diagram kelas berikut.



Method area() digunakan untuk menghitung luas masing-masing objek. Berikut adalah formula yang digunakan untuk menghitung luas masing-masing bangun yang harus diimplementasikan.

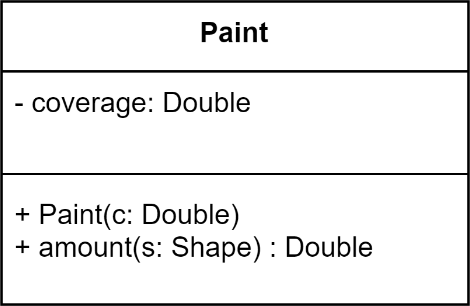
Sphere: 4 × 𝜋 × 𝑟𝑎𝑑𝑖𝑢𝑠2 Rectangle: 𝑙𝑒𝑛𝑔𝑡ℎ × 𝑤𝑖𝑑𝑡ℎ Cylinder: 𝜋 × 𝑟𝑎𝑑𝑖𝑢𝑠2 × ℎ𝑒𝑖𝑔ℎ𝑡

Method toString() digunakan untuk mengembalikan nilai String dari nama bangun

Berikut adalah ilustrasi dari kelas Sphere.java. Implementasikan kelas lainnya untuk Shape, Rectangle dan Cyclinder

|  |
| --- |
| **Contoh Ilustrasi Sphere.java** |
| public class Sphere extends Shape  {  private double radius; //radius in feet  //  // Constructor: Sets up the sphere.  //  public Sphere(double r)  {  super("Sphere"); radius = r;  }  //  // Returns the surface area of the sphere.  //  public double area()  {  return 4\*Math.PI\*(radius\*radius);  }  //  // Returns the sphere as a String.  //  public String toString()  {  return super.toString() + " of radius " + radius;  }  } |

Selanjutnya, Buatlah kelas Paint.java seperti ditunjukkan diagram kelas berikut.



Method amount digunakan untuk menghitung banyaknya liter cat yang digunakan dengan persamaan berikut:

𝑎𝑚𝑜𝑢𝑛𝑡 𝑜𝑓 𝑝𝑎𝑖𝑛𝑡 =

𝑎𝑟𝑒𝑎 𝑜𝑓 𝑠ℎ𝑎𝑝𝑒

𝑐𝑜𝑣𝑒𝑟𝑎𝑔𝑒

Lengkapi kode dibawah supaya menghasilkan keluaran yang diinginkan

|  |
| --- |
| **Paint.java** |
| public class Paint  {  private double coverage; //number of square feet per gallon  //  // Constructor: Sets up the paint object.  //  public Paint(double c)  {  coverage = c;  }  //  // Returns the amount of paint (number of gallons)  // needed to paint the shape given as the parameter.  //  public double amount(Shape s)  {  System.out.println ("Computing amount for " + s); return 0;  }  } |

Terakhir, Buatlah kelas main bernama PaintThings.java. Tambahkan beberapa hal berikut agar program berjalan sesuai yang diinginkan.

1. Instansiasi 3 bentuk objek:
   1. objek bernama deck berbentuk persegi panjang dengan ukuran Panjang 20cm dan lebar 30cm.
   2. objek bernama bigBall berbentuk bola dengan ukuran radius 15cm.
   3. objek bernama tank berbentuk silinder dengan ukuran radius 10cm dan tinggi 30cm.
2. Panggil fungsi yang tepat agar dapat menghitung jumlah cat yang diperlukan.

Petunjuk untuk kelas main PaintThings.java

import java.text.DecimalFormat; public class PaintThings

{

//

// Creates some shapes and a Paint object

// and prints the amount of paint needed

// to paint each shape.

//

public static void main (String[] args)

{

final double COVERAGE = 350;

Paint paint = new Paint(COVERAGE); Rectangle deck;

Sphere bigBall; Cylinder tank;

double deckAmt, ballAmt, tankAmt;

// Instantiate the three shapes to paint

// Compute the amount of paint needed for each shape

// Print the amount of paint for each. DecimalFormat fmt = new DecimalFormat("0.#");

System.out.println ("\nNumber of gallons of paint needed..."); System.out.println ("Deck " + fmt.format(deckAmt));

System.out.println ("Big Ball " + fmt.format(ballAmt)); System.out.println ("Tank " + fmt.format(tankAmt));

}

}

1. Jalankan program dan perhatikan hasil untuk ketiga bentuk yang berbeda, screenshot hasil yang didapatkan dan lampirkan di dalam source code.
2. Simpan coding anda dengan nama package: **soal1**
3. Pastikan terdapat screenshoot pada repositori github

Tabel .1 Soal 1

### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | package soal1;  import java.lang.Math; public class sphere extends shape {  private double radius;   public sphere(Double r){  super("Sphere");  radius = r;  }   public Double area(){return 4\*Math.*PI*\*(radius\*radius);}   public String toString(){return super.toString() + " of radius "+radius;} } |

Tabel .1 Source Code Soal 1 Spehere

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package soal1;  public abstract class shape {  protected String shapeName;   public shape(String name) {  shapeName = name;  }   protected abstract Double area();   public String toString() {  return shapeName;  } } |

Tabel .1 Source Code Soal 1 Shape

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | package soal1;  public class rectangle extends shape{  private Double length, width;   public rectangle(Double l,Double w){  super("Rectangle");  length = l;  width = w;  }   public Double area(){return length\*width;}   public String toString(){return super.toString() + " of length " + length + " and width "+width;} } |

Tabel .1 Source Code Soal 1 Rectangle

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | package soal1;  public class cylinder extends shape{  private Double radius, height;   public cylinder(Double r, Double h){  super("Cylinder");  radius = r;  height = h;  }   public Double area(){return Math.*PI*\*(radius\*radius)\*height;}   public String toString(){return super.toString() + " of radius " + radius + " and height "+height;} } |

Tabel .1 Source Code Soal 1 Cylinder

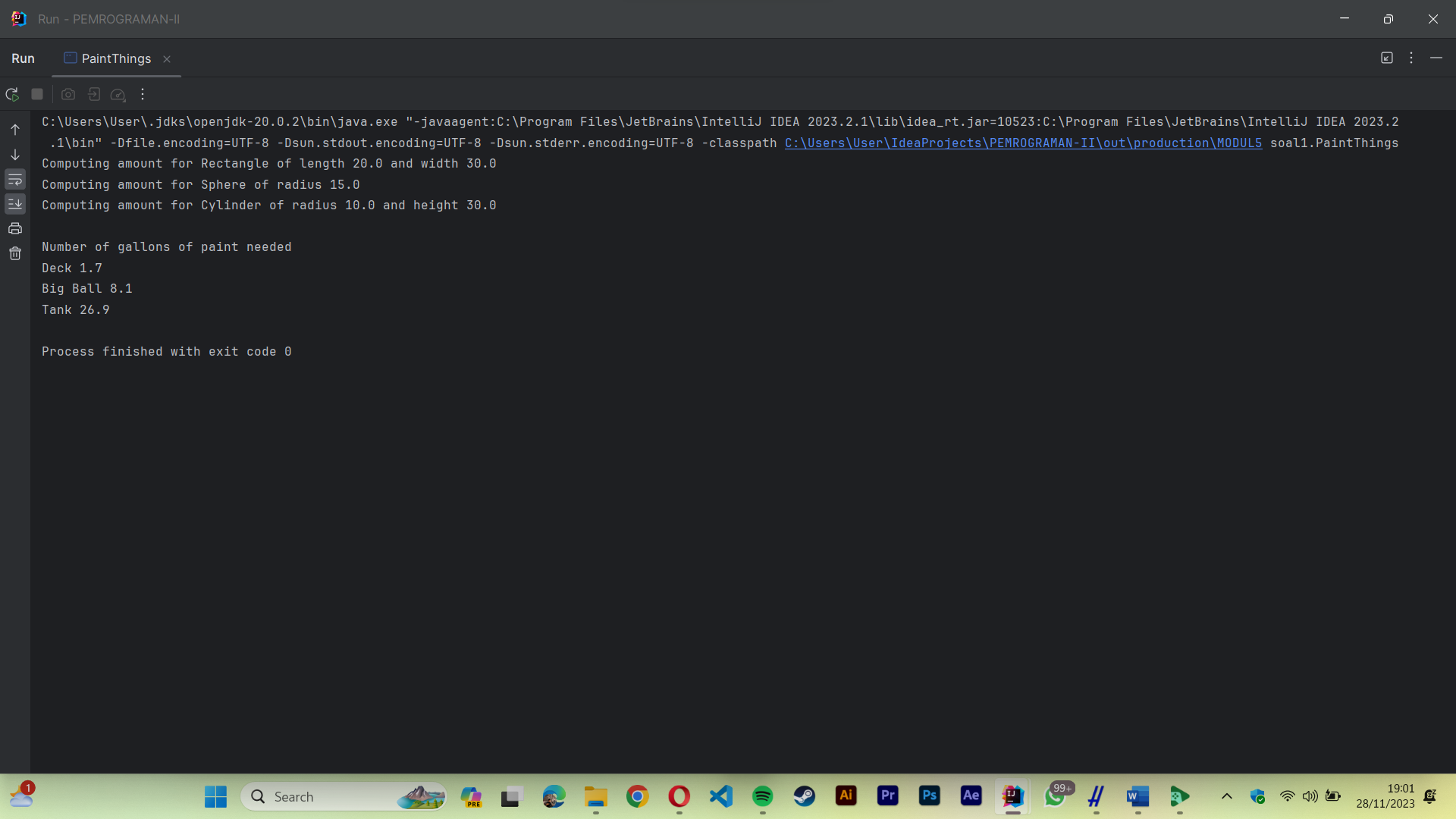
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | package soal1;  public class paint {  private double coverage;   public paint(double c){coverage = c;}   public double amount(shape s){  System.*out*.println("Computing amount for "+ s);  return s.area()/coverage;  } } |

Tabel .1 Source Code Soal 1 Paint

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | package soal1;  import java.text.DecimalFormat;  public class PaintThings {  public static void main(String[] args) {  final double COVERAGE = 350;  paint paint = new paint(COVERAGE);  rectangle deck;  sphere bigBall;  cylinder tank;  double deckAmt, ballAmt, tankAmt;   deck = new rectangle(20.0,30.0);  bigBall = new sphere(15.0);  tank = new cylinder(10.0,30.0);   deckAmt = paint.amount(deck);  ballAmt = paint.amount(bigBall);  tankAmt = paint.amount(tank);   DecimalFormat fmt = new DecimalFormat("0.#");  System.*out*.println("\nNumber of gallons of paint needed");  System.*out*.println("Deck " + fmt.format(deckAmt));  System.*out*.println("Big Ball " + fmt.format(ballAmt));  System.*out*.println("Tank " + fmt.format(tankAmt));  } } |

Tabel .1 Source Code Soal 1 PaintThings

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

#### Sphere

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 4, 7, 12, dan 15: *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 4 dan 17 : {}, penanda awal dan akhir *block code*.*.*

#### Shape

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3, 6, dan 12 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 3 dan 15 : {}, penanda awal dan akhir *block code*.

Line 10 : *Abstract,* sebagai ekspresi konsep umum tempat kelas yang lebih spesifik dapat diturunkan.

Line 13 : *Return,* mengembalikan nilai dari sebuah function.

#### Rectangle

Line 1 : Package, untuk pengelompokan class atau interface menjadi satu unit.

Line 3, 6, 12, dan 14: Public, kunci pada java yang dapat diakses kelas class lain.

Line 3 : *Extends ,*  kata kunci yang ditulis dengan kelas anak selama deklarasi kelas diikuti dengan nama kelas induk.

Line 3 dan 16: {}, penanda awal dan akhir block code.

#### Cylinder

Line 1 : Package, untuk pengelompokan class atau interface menjadi satu unit.

Line 3 : *Extends ,*  kata kunci yang ditulis dengan kelas anak selama deklarasi kelas diikuti dengan nama kelas induk.

Line 3, 6, 12, dan 15: Public, kunci pada java yang dapat diakses kelas class lain.

Line 3 dan 17 : {}, penanda awal dan akhir block code.

Line 15 : *Return,* mengembalikan nilai dari sebuah function.

#### Paint

Line 1 : Package, untuk pengelompokan class atau interface menjadi satu unit.

Line 3, 6, dan 8: Public, kunci pada java yang dapat diakses kelas class lain.

Line 3 dan 13 : {}, penanda awal dan akhir block code.

Line 9 : System.out.println mencetak sesuatu dari program sebagai output.

Line 11 : *Return,* mengembalikan nilai dari sebuah function.

#### PaintThings

Line 1 : Package, untuk pengelompokan class atau interface menjadi satu unit.

Line 2: Import, suatu perintah untuk memasukkan suatu method atau perintah dalam Bahasa Java.

Line 4 dan 11: Public, kunci pada java yang dapat diakses kelas class lain.

Line 4 dan 34 : {}, penanda awal dan akhir block code.

Line 24-31 : System.out.println mencetak sesuatu dari program sebagai output.

# MODUL 6: GRAPHICAL USER INTEFACE

## SOAL

Diberikan class diagram seperti berikut: (isi program harus sesuai dengan class diagram)



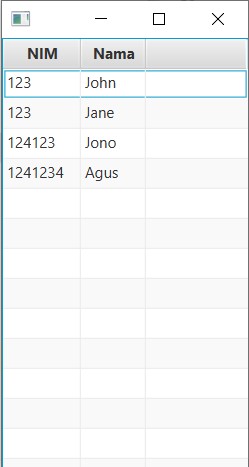
Implementasikan class diagram diatas menjadi class pada Bahasa pemrograman java. Class mahasiswa harus menerapkan setter dan getter.

Program harus menampilkan list data dalam bentuk tabel.

Kolom dari tabel adalah :

* NIM
* Nama

Kemudian buatlah 10 data secara *hardcode* untuk ditampilkan pada tabel Contoh program dapat dilihat sebagai berikut:



### Source Code

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  25  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | package com.example.modul\_6;  import javafx.fxml.FXML;  import javafx.scene.control.Label;  public class Mahasiswa {  private int id;  private String name;  private String nim;  public Mahasiswa(int id, String name, String nim) {  this.id = id;  this.name = name;  this.nim = nim;  }  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public String getNim() {  return nim;  }  public void setNim(String nim) {  this.nim=nim;  }  } |

Tabel .1 Source Code Soal 1 Mahasiswa

|  |  |
| --- | --- |
| 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 com.example.modul\_6;  import javafx.application.Application;  import javafx.scene.Scene;  import javafx.scene.control.TableColumn;  import javafx.scene.control.TableView;  import javafx.scene.control.cell.PropertyValueFactory;  import javafx.scene.layout.VBox;  import javafx.stage.Stage;  public class Main extends Application {  public static void main(String[] args) {  launch(args);  }  @Override  public void start(Stage primaryStage) {  TableView tableView = new TableView();  TableColumn<Mahasiswa, String> column1 =  new TableColumn<>("NIM");  column1.setCellValueFactory(  new PropertyValueFactory<>("nim"));  TableColumn<Mahasiswa, String> column2 =  new TableColumn<>("Nama");  column2.setCellValueFactory(  new PropertyValueFactory<>("name"));  tableView.getColumns().add(column1);  tableView.getColumns().add(column2);  tableView.getItems().add(  new Mahasiswa(1, "Audrey", "7000"));  tableView.getItems().add(  new Mahasiswa(2, "Dandellion", "7001"));  tableView.getItems().add(  new Mahasiswa(3, "Deyya", "7002"));  tableView.getItems().add(  new Mahasiswa(4, "Aina", "7003"));  tableView.getItems().add(  new Mahasiswa(5, "Dawan", "7004"));  tableView.getItems().add(  new Mahasiswa(6, "Helenah", "7005"));  tableView.getItems().add(  new Mahasiswa(7, "Lidama", "7006"));  tableView.getItems().add(  new Mahasiswa(8, "Aninda", "7007"));  tableView.getItems().add(  new Mahasiswa(9, "Isanhabang", "7008"));  tableView.getItems().add(  new Mahasiswa(10, "IrfanStis", "7009"));  VBox vbox = new VBox(tableView);  Scene scene = new Scene(vbox);  primaryStage.setScene(scene);  primaryStage.show();  }  } |

Tabel .1 Source Code Soal 1 Main

### Output Program



Gambar .1 Screenshoot Output Soal 1

### Pembahasan

#### Mahasiswa

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3 dan 4: *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 7 : *int,* tipe data *integer* atau menerima bilangan bulat.

Line 8 : *String*, tipe data teks yang berisikan karakter lebih.

Line 9 : *String*, tipe data teks yang berisikan karakter lebih.

Line 18, 22, 26,30, 35, dan 38 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 6 dan 42: {}, penanda awal dan akhir *block code*.*.*

#### Main

Line 1 : *Package,* untuk pengelompokan *class* atau *interface* menjadi satu unit.

Line 3 – 10 : *Import,*  suatu perintah untuk memasukkan suatu *method* atau perintah dalam Bahasa Java.

Line 12 : *Public*, kunci pada java yang dapat diakses kelas *class* lain.

Line 12 : *Extends ,*  kata kunci yang ditulis dengan kelas anak selama deklarasi kelas diikuti dengan nama kelas induk.

Line 12 dan 72 : {}, penanda awal dan akhir *block code*.

# LINK GITHUB

<https://github.com/RylenRawr/PEMROGRAMAN-II>