**PRAKTIKUM PEMOGRAMAN BERORIENTASI OBYEK 1**

**KELAS C**

**“KONSTRUKTOR”**

****

**DISUSUN OLEH :**

Nama : Johanes Yogtan Wicaksono Raharja

NIM : 215314105

**TEKNIK INFORMATIKA**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA YOGYAKARTA**

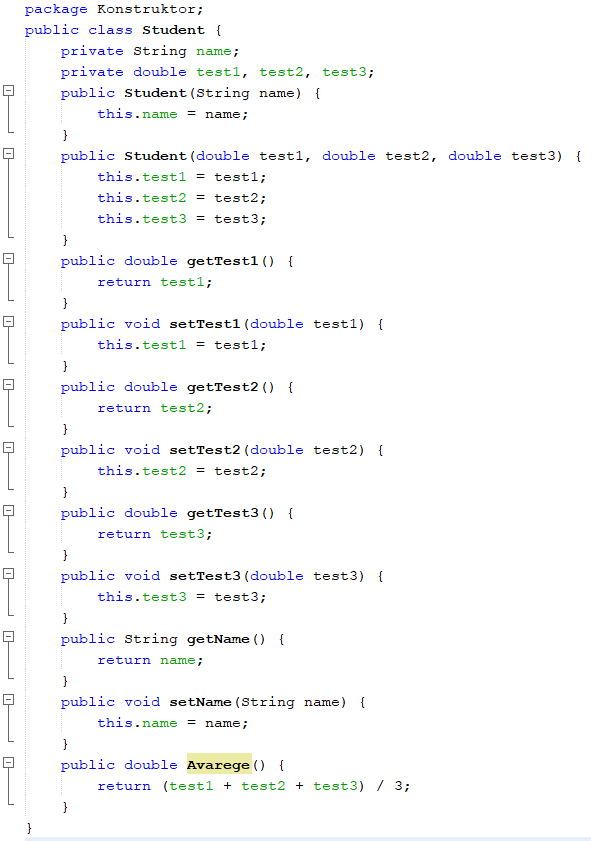
**2022**

1. Class Student dan StudentMain
2. Diagram Class Student

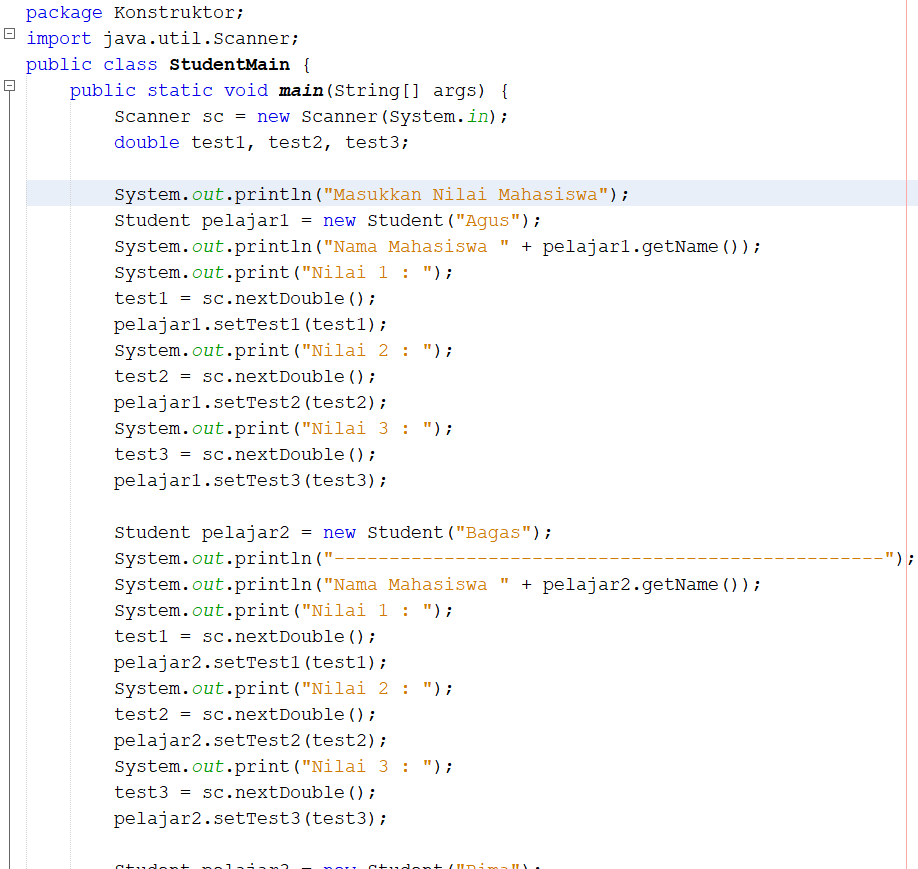
|  |
| --- |
| **Student** |
| -name : String  -test1 : double  -test2 : double  -test3 : double |
| <<constructor>>Student(String  +getTest1() : double  +setTest1 (double) : void  +getTest2() : double  +setTest2 (double) : void  +getTest3() : double  +setTest3 (double) : void  +getName() : String  +setName (String) : void  +Average() : double |

1. Capture Program

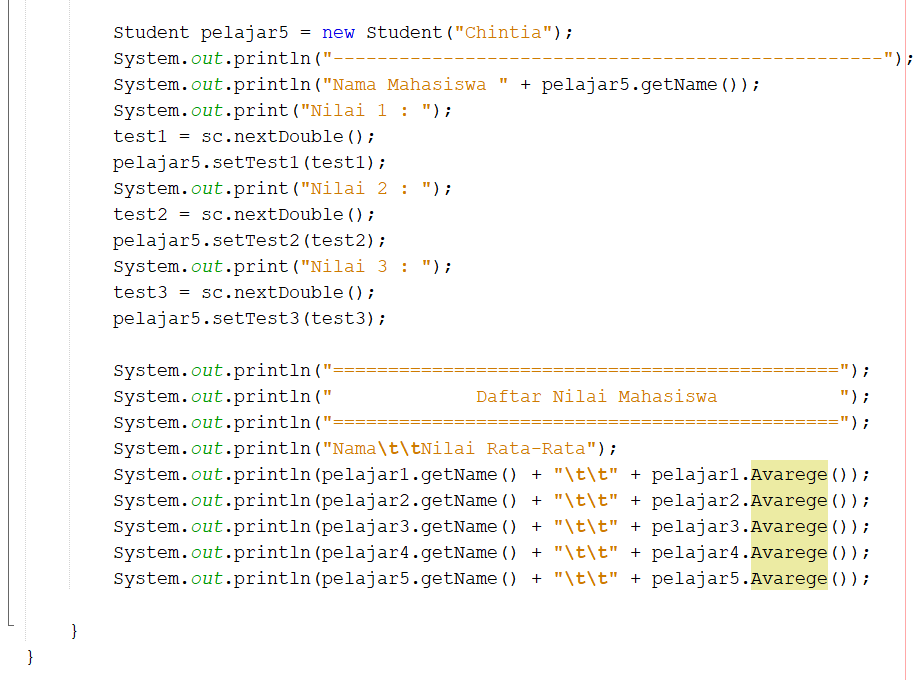
Class Student



Class StudentMain





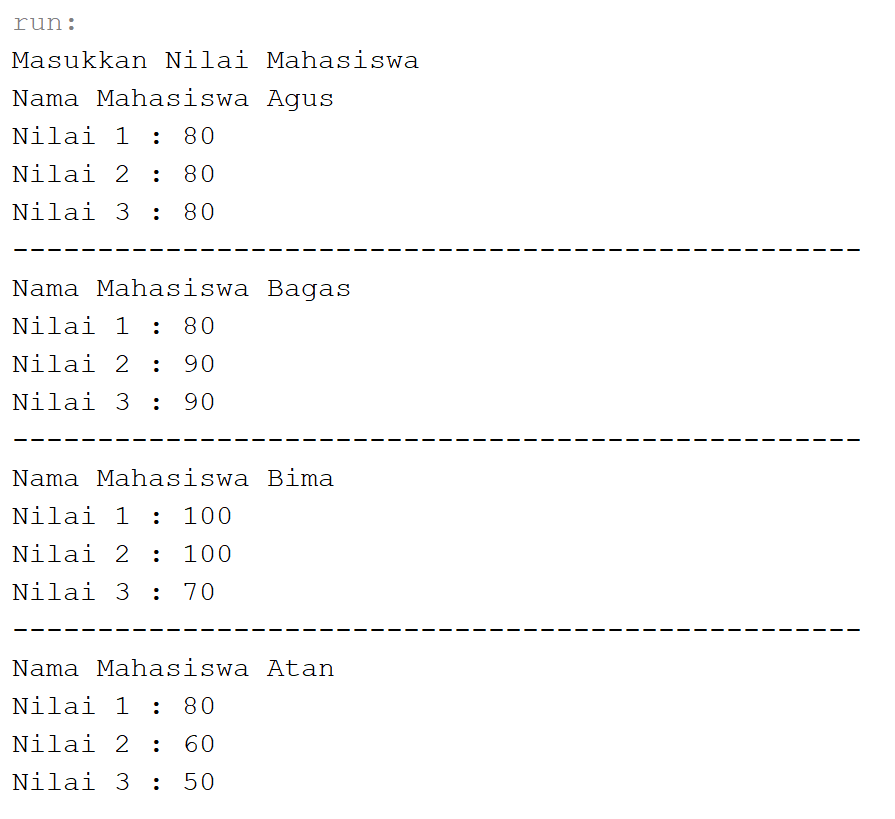


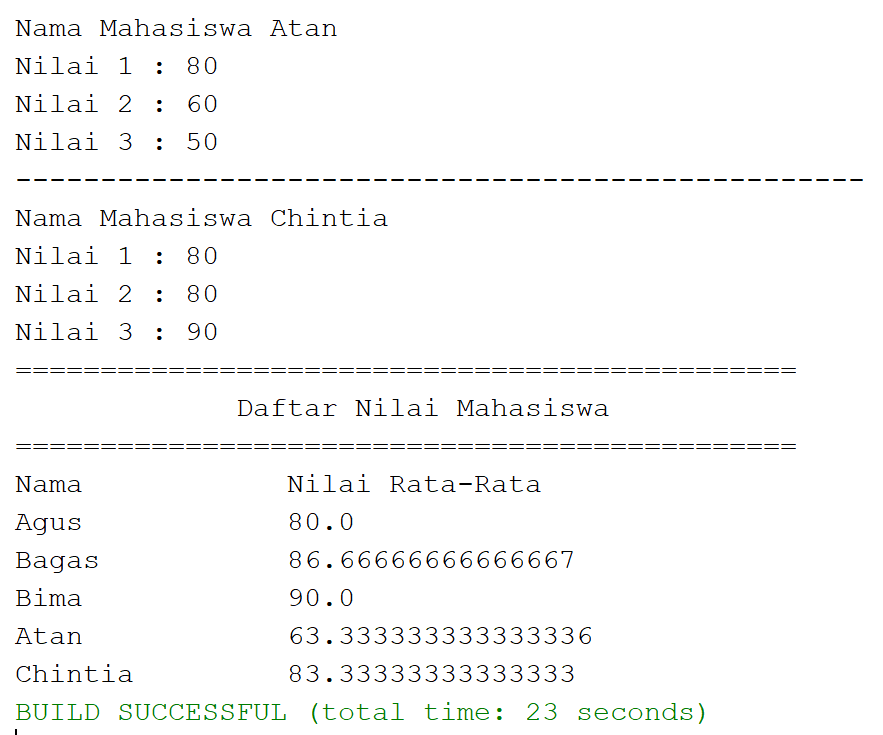
1. Penjelasan Program Berbaris

|  |  |  |
| --- | --- | --- |
| **No.** | **Teks Program (Student)** | **Arti/penjelasan** |
| 1 | package Konstruktor; | Deklarasi package |
| 2 | public class Student { | Deklarasi class untuk membentuk obyek baru |
| 3 | private String name; | Memesan variabel bertipe String dan didefinisikan secara privat |
| 4 | private double test1, test2, test3; | Memesan variabel bertipe double dan didefinisikan secara privat |
| 5 | public Student(String name) { | Constructor untuk mengawali variabel name |
| 6 | this.name = name; | Deklarasi menyimpan data ke kelas dari parameter |
| 7 | } |  |
| 8 | public Student(double test1, double test2, double test3) { | Constructor untuk mengawali variabel test1, test2, test3 |
| 9 | this.test1 = test1; | Perintah metode get(menampilkan) data obyek |
| 10 | this.test2 = test2; | Deklarasi menyimpan data ke kelas dari parameter |
| 11 | this.test3 = test3; | Deklarasi menyimpan data ke kelas dari parameter |
| 12 | } |  |
| 13 | public double getTest1() { | Perintah metode get(menampilkan) data obyek |
| 14 | return test1; | Deklarasi mengirim isi test1 |
| 15 | } |  |
| 16 | public void setTest1(double test1) { | Perintah metode set(membaca) data obyek |
| 17 | this.test1 = test1; | Deklarasi menyimpan data ke kelas dari parameter |
| 18 | } |  |
| 19 | public double getTest2() { | Perintah metode get(menampilkan) data obyek |
| 20 | return test2; | Deklarasi mengirim isi test2 |
| 21 | } |  |
| 22 | public void setTest2(double test2) { | Perintah metode set(membaca) data obyek |
| 23 | this.test2 = test2; | Deklarasi menyimpan data ke kelas dari parameter |
| 24 | } |  |
| 25 | public double getTest3() { | Perintah metode get(menampilkan) data obyek |
| 26 | return test3; | Deklarasi mengirim isi test3 |
| 27 | } |  |
| 28 | public void setTest3(double test3) { | Perintah metode set(membaca) data obyek |
| 29 | this.test3 = test3; | Deklarasi menyimpan data ke kelas dari parameter |
| 30 | } |  |
| 31 | public String getName() { | Perintah metode get(menampilkan) data obyek |
| 32 | return name; | Deklarasi mengirim isi name |
| 33 | } |  |
| 34 | public void setName(String name) { | Perintah metode set(membaca) data obyek |
| 35 | this.name = name; | Deklarasi menyimpan data ke kelas dari parameter |
| 36 | } |  |
| 37 | public double Avarege() { | Membuat perintah ke obyek untuk menjalankan method Averege |
| 38 | return (test1 + test2 + test3) / 3; | Deklarasi mengirim isi hasil kali |
| 39 | } |  |
| 40 | } |  |

|  |  |  |
| --- | --- | --- |
| **No.** | **Teks Program (StudentMain)** | **Arti/penjelasan** |
|  | package Konstruktor; | Deklarasi package |
|  | import java.util.Scanner; | Deklarasi Scanner |
|  | public class StudentMain { | Deklarasi class untuk membentuk obyek baru |
|  | public static void main(String[] args) { | Deklarasi metode main |
|  | Scanner sc = new Scanner(System.in); | Mengimpor perlengkapan scanner |
|  | double test1, test2, test3; | Memesan variabel bertipe double |
|  |  |  |
|  | System.out.println("Masukkan Nilai Mahasiswa"); | Meminta user memasukkan data |
|  | Student pelajar1 = new Student("Agus"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar1 dan memasukkan datanya |
|  | System.out.println("Nama Mahasiswa " + pelajar1.getName()); | Memberi perintah obyek pelajar1 untuk menampilkan(get) metode data getName () |
|  | System.out.print("Nilai 1 : "); | Meminta user memasukkan data |
|  | test1 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test1 |
|  | pelajar1.setTest1(test1); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test1 |
|  | System.out.print("Nilai 2 : "); | Meminta user memasukkan data |
|  | test2 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test2 |
|  | pelajar1.setTest2(test2); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test2 |
|  | System.out.print("Nilai 3 : "); | Meminta user memasukkan data |
|  | test3 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test3 |
|  | pelajar1.setTest3(test3); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test3 |
|  |  |  |
|  | Student pelajar2 = new Student("Bagas"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar2 dan memasukkan datanya |
|  | System.out.println("--------------------------------------------------"); | Sebuah variasi |
|  | System.out.println("Nama Mahasiswa " + pelajar2.getName()); | Memberi perintah obyek pelajar2 untuk menampilkan(get) metode data getName () |
|  | System.out.print("Nilai 1 : "); | Meminta user memasukkan data |
|  | test1 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test1 |
|  | pelajar2.setTest1(test1); | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test1 |
|  | System.out.print("Nilai 2 : "); | Meminta user memasukkan data |
|  | test2 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test2 |
|  | pelajar2.setTest2(test2); | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test2 |
|  | System.out.print("Nilai 3 : "); | Meminta user memasukkan data |
|  | test3 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test3 |
|  | pelajar2.setTest3(test3); | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test3 |
|  |  |  |
|  | Student pelajar3 = new Student("Bima"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar3 dan memasukkan datanya |
|  | System.out.println("--------------------------------------------------"); | Sebuah variasi |
|  | System.out.println("Nama Mahasiswa " + pelajar3.getName()); | Memberi perintah obyek pelajar3 untuk menampilkan(get) metode data getName () |
|  | System.out.print("Nilai 1 : "); | Meminta user memasukkan data |
|  | test1 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test1 |
|  | pelajar3.setTest1(test1); | Deklarasi perintah obyek pelajar3 ke method setNama untuk membaca data test1 |
|  | System.out.print("Nilai 2 : "); | Meminta user memasukkan data |
|  | test2 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test2 |
|  | pelajar3.setTest2(test2); | Deklarasi perintah obyek pelajar3 ke method setNama untuk membaca data test2 |
|  | System.out.print("Nilai 3 : "); | Meminta user memasukkan data |
|  | test3 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test3 |
|  | pelajar3.setTest3(test3); | Deklarasi perintah obyek pelajar3 ke method setNama untuk membaca data test3 |
|  |  |  |
|  | Student pelajar4 = new Student("Atan"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelaja4 dan memasukkan datanya |
|  | System.out.println("--------------------------------------------------"); | Sebuah variasi |
|  | System.out.println("Nama Mahasiswa " + pelajar4.getName()); | Memberi perintah obyek pelajar4 untuk menampilkan(get) metode data getName () |
|  | System.out.print("Nilai 1 : "); | Meminta user memasukkan data |
|  | test1 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test1 |
|  | pelajar4.setTest1(test1); | Deklarasi perintah obyek pelajar4 ke method setNama untuk membaca data test1 |
|  | System.out.print("Nilai 2 : "); | Meminta user memasukkan data |
|  | test2 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test2 |
|  | pelajar4.setTest2(test2); | Deklarasi perintah obyek pelajar4 ke method setNama untuk membaca data test2 |
|  | System.out.print("Nilai 3 : "); | Meminta user memasukkan data |
|  | test3 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test3 |
|  | pelajar4.setTest3(test3); | Deklarasi perintah obyek pelajar4 ke method setNama untuk membaca data test3 |
|  |  |  |
|  | Student pelajar5 = new Student("Chintia"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar5 dan memasukkan datanya |
|  | System.out.println("--------------------------------------------------"); | Sebuah variasi |
|  | System.out.println("Nama Mahasiswa " + pelajar5.getName()); | Memberi perintah obyek pelajar5 untuk menampilkan(get) metode data getName () |
|  | System.out.print("Nilai 1 : "); | Meminta user memasukkan data |
|  | test1 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test1 |
|  | pelajar5.setTest1(test1); | Deklarasi perintah obyek pelajar5 ke method setNama untuk membaca data test1 |
|  | System.out.print("Nilai 2 : "); | Meminta user memasukkan data |
|  | test2 = sc.nextDouble(); | Menyimpan data dari keyboard ke variabel test2 |
|  | pelajar5.setTest2(test2); | Deklarasi perintah obyek pelajar5 ke method setNama untuk membaca data test2 |
|  | System.out.print("Nilai 3 : "); | Menyimpan data dari keyboard ke variabel test3 |
|  | test3 = sc.nextDouble(); | Deklarasi perintah obyek pelajar5 ke method setNama untuk membaca data test3 |
|  | pelajar5.setTest3(test3); | Deklarasi perintah obyek pelajar5 ke method setNama untuk membaca data test3 |
|  |  |  |
|  | System.out.println("=============================================="); | Swebuah variasi |
|  | System.out.println(" Daftar Nilai Mahasiswa "); | Deklarasi sebuah kalimat |
|  | System.out.println("=============================================="); | Sebuah variasi |
|  | System.out.println("Nama\t\tNilai Rata-Rata"); | Deklarasi sebuah kalimat |
|  | System.out.println(pelajar1.getName() + "\t\t" + pelajar1.Avarege()); | Memberi perintah obyek pelajar1 untuk menampilkan(get) metode data getName() dan metode Avarage() |
|  | System.out.println(pelajar2.getName() + "\t\t" + pelajar2.Avarege()); | Memberi perintah obyek pelajar2 untuk menampilkan(get) metode data getName() dan metode Avarage() |
|  | System.out.println(pelajar3.getName() + "\t\t" + pelajar3.Avarege()); | Memberi perintah obyek pelajar3 untuk menampilkan(get) metode data getName() dan metode Avarage() |
|  | System.out.println(pelajar4.getName() + "\t\t" + pelajar4.Avarege()); | Memberi perintah obyek pelajar4 untuk menampilkan(get) metode data getName() dan metode Avarage() |
|  | System.out.println(pelajar5.getName() + "\t\t" + pelajar5.Avarege()); | Memberi perintah obyek pelajar5 untuk menampilkan(get) metode data getName() dan metode Avarage() |
|  | } |  |
|  | } |  |

1. Capture Input dan Output



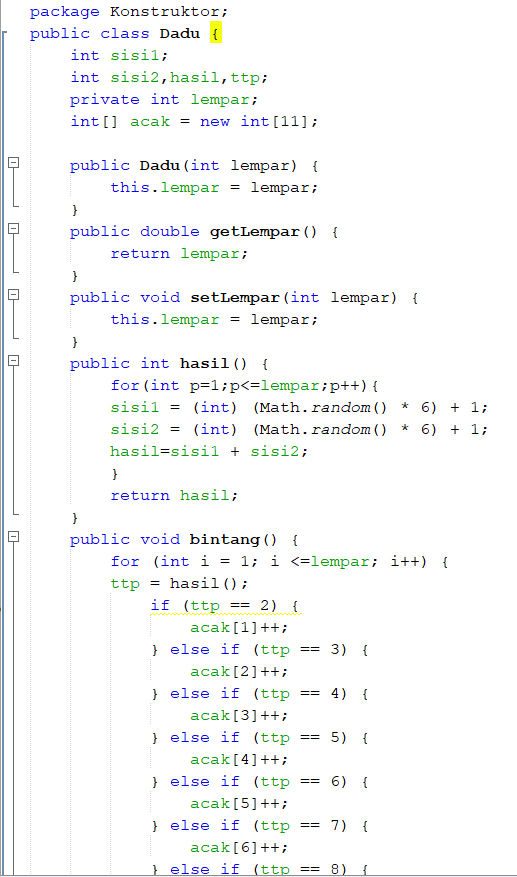


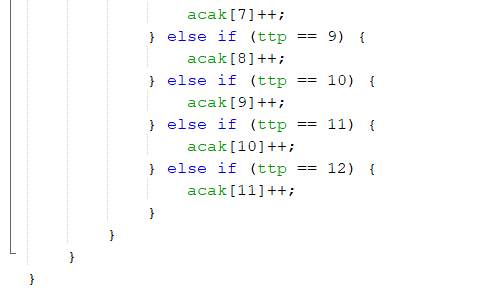
1. Class Dadu dan DuaDadu
2. Diagram Class Dadu

|  |
| --- |
| **Student** |
| -sisi1 : int  -sisi2 : int  -hasil : int  -ttp : int  -lempar : int |
| <<constructor>>Dadau(Int)  +getLempar() : double  +setLempar (double) : void  +hasil() : int  +bintang() : void |

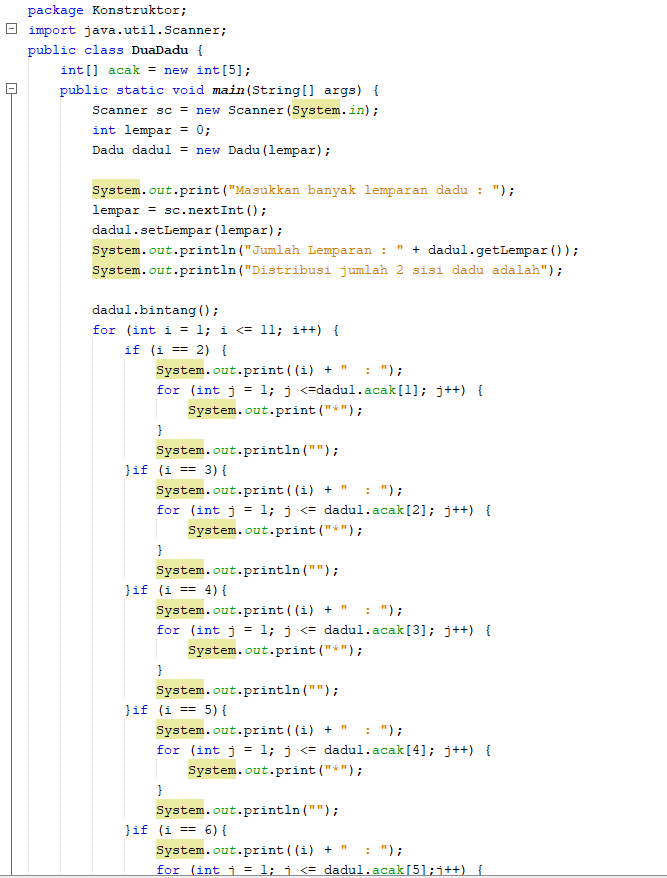
1. Capture Program

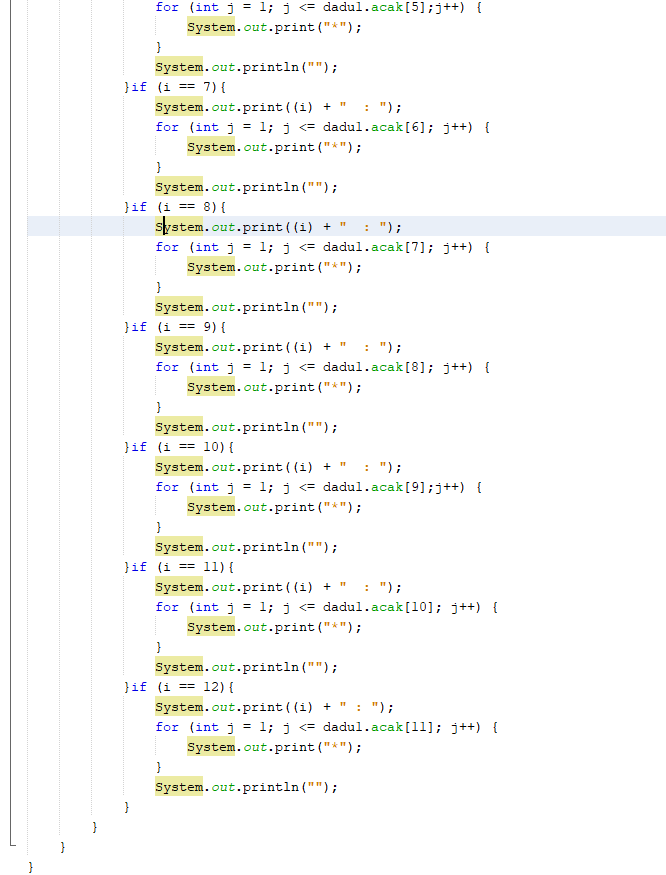
Class Dadu





Class DuaDadu





1. Penjelasan Program Berbaris

|  |  |  |
| --- | --- | --- |
| **No.** | **Teks Program (Student)** | **Arti/penjelasan** |
| 1 | package Konstruktor; | Deklarasi package |
| 2 | public class Dadu { | Deklarasi class untuk membentuk obyek baru |
| 3 | int sisi1; | Memesan variabel bertipe String dan didefinisikan secara privat |
| 4 | int sisi2,hasil,ttp; | Memesan variabel bertipe double dan didefinisikan secara privat |
| 5 | private int lempar; | Constructor untuk mengawali variabel name |
| 6 | int[] acak = new int[11]; | Deklarasi menyimpan data ke kelas dari parameter |
| 7 |  |  |
| 8 | public Dadu(int lempar) { | Constructor untuk mengawali variabel test1, test2, test3 |
| 9 | this.lempar = lempar; | Perintah metode get(menampilkan) data obyek |
| 10 | } | Deklarasi menyimpan data ke kelas dari parameter |
| 11 | public double getLempar() { | Deklarasi menyimpan data ke kelas dari parameter |
| 12 | return lempar; |  |
| 13 | } | Perintah metode get(menampilkan) data obyek |
| 14 | public void setLempar(int lempar) { | Deklarasi mengirim isi test1 |
| 15 | this.lempar = lempar; |  |
| 16 | } | Perintah metode set(membaca) data obyek |
| 17 | public int hasil() { | Deklarasi menyimpan data ke kelas dari parameter |
| 18 | for(int p=1;p<=lempar;p++){ |  |
| 19 | sisi1 = (int) (Math.random() \* 6) + 1; | Perintah metode get(menampilkan) data obyek |
| 20 | sisi2 = (int) (Math.random() \* 6) + 1; | Deklarasi mengirim isi test2 |
| 21 | hasil=sisi1 + sisi2; |  |
| 22 | } | Perintah metode set(membaca) data obyek |
| 23 | return hasil; | Deklarasi menyimpan data ke kelas dari parameter |
| 24 | } |  |
| 25 | public void bintang() { | Perintah metode get(menampilkan) data obyek |
| 26 | for (int i = 1; i <=lempar; i++) { | Deklarasi mengirim isi test3 |
| 27 | ttp = hasil(); |  |
| 28 | if (ttp == 2) { | Perintah metode set(membaca) data obyek |
| 29 | acak[1]++; | Deklarasi menyimpan data ke kelas dari parameter |
| 30 | } else if (ttp == 3) { |  |
| 31 | acak[2]++; | Perintah metode get(menampilkan) data obyek |
| 32 | } else if (ttp == 4) { | Deklarasi mengirim isi name |
| 33 | acak[3]++; |  |
| 34 | } else if (ttp == 5) { | Perintah metode set(membaca) data obyek |
| 35 | acak[4]++; | Deklarasi menyimpan data ke kelas dari parameter |
| 36 | } else if (ttp == 6) { |  |
| 37 | acak[5]++; | Membuat perintah ke obyek untuk menjalankan method Averege |
| 38 | } else if (ttp == 7) { | Deklarasi mengirim isi hasil kali |
| 39 | acak[6]++; |  |
| 40 | } else if (ttp == 8) { |  |
| 41 | acak[7]++; |  |
| 42 | } else if (ttp == 9) { |  |
| 43 | acak[8]++; |  |
| 44 | } else if (ttp == 10) { |  |
| 45 | acak[9]++; |  |
| 46 | } else if (ttp == 11) { |  |
| 47 | acak[10]++; |  |
| 48 | } else if (ttp == 12) { |  |
| 49 | acak[11]++; |  |

|  |  |  |
| --- | --- | --- |
| 50 | } |  |
| 51 | } |  |
| 52 | } |  |
| 53 | } |  |

|  |  |  |
| --- | --- | --- |
| **No.** | **Teks Program (DuaDadu)** | **Arti/penjelasan** |
|  | package Konstruktor; | Deklarasi package |
|  | import java.util.Scanner; | Deklarasi Scanner |
|  | public class DuaDadu { | Deklarasi class untuk membentuk obyek baru |
|  | int[] acak = new int[5]; | Deklarasi metode main |
|  | public static void main(String[] args) { | Mengimpor perlengkapan scanner |
|  | Scanner sc = new Scanner(System.in); | Memesan variabel bertipe double |
|  | int lempar = 0; |  |
|  | Dadu dadu1 = new Dadu(lempar); | Meminta user memasukkan data |
|  |  | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar1 dan memasukkan datanya |
|  | System.out.print("Masukkan banyak lemparan dadu : "); | Memberi perintah obyek pelajar1 untuk menampilkan(get) metode data getName () |
|  | lempar = sc.nextInt(); | Meminta user memasukkan data |
|  | dadu1.setLempar(lempar); | Menyimpan data dari keyboard ke variabel test1 |
|  | System.out.println("Jumlah Lemparan : " + dadu1.getLempar()); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test1 |
|  | System.out.println("Distribusi jumlah 2 sisi dadu adalah"); | Meminta user memasukkan data |
|  |  | Menyimpan data dari keyboard ke variabel test2 |
|  | dadu1.bintang(); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test2 |
|  | for (int i = 1; i <= 11; i++) { | Meminta user memasukkan data |
|  | if (i == 2) { | Menyimpan data dari keyboard ke variabel test3 |
|  | System.out.print((i) + " : "); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test3 |
|  | for (int j = 1; j <=dadu1.acak[1]; j++) { |  |
|  | System.out.print("\*"); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar2 dan memasukkan datanya |
|  | } | Sebuah variasi |
|  | System.out.println(""); | Memberi perintah obyek pelajar2 untuk menampilkan(get) metode data getName () |
|  | }if (i == 3){ | Meminta user memasukkan data |
|  | System.out.print((i) + " : "); | Menyimpan data dari keyboard ke variabel test1 |
|  | for (int j = 1; j <= dadu1.acak[2]; j++) { | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test1 |
|  | System.out.print("\*"); | Meminta user memasukkan data |
|  | } | Menyimpan data dari keyboard ke variabel test2 |
|  | System.out.println(""); | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test2 |
|  | }if (i == 4){ | Meminta user memasukkan data |
|  | System.out.print((i) + " : "); | Menyimpan data dari keyboard ke variabel test3 |
|  | for (int j = 1; j <= dadu1.acak[3]; j++) { | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test3 |
|  | System.out.print("\*"); |  |
|  | } | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar3 dan memasukkan datanya |

|  |  |  |
| --- | --- | --- |
|  | System.out.println(""); | Deklarasi package |
|  | }if (i == 5){ | Deklarasi Scanner |
|  | System.out.print((i) + " : "); | Deklarasi class untuk membentuk obyek baru |
|  | for (int j = 1; j <= dadu1.acak[4]; j++) { | Deklarasi metode main |
|  | System.out.print("\*"); | Mengimpor perlengkapan scanner |
|  | } | Memesan variabel bertipe double |
|  | System.out.println(""); |  |
|  | }if (i == 6){ | Meminta user memasukkan data |
|  | System.out.print((i) + " : "); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar1 dan memasukkan datanya |
|  | for (int j = 1; j <= dadu1.acak[5];j++) { | Memberi perintah obyek pelajar1 untuk menampilkan(get) metode data getName () |
|  | System.out.print("\*"); | Meminta user memasukkan data |
|  | } | Menyimpan data dari keyboard ke variabel test1 |
|  | System.out.println(""); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test1 |
|  | }if (i == 7){ | Meminta user memasukkan data |
|  | System.out.print((i) + " : "); | Menyimpan data dari keyboard ke variabel test2 |
|  | for (int j = 1; j <= dadu1.acak[6]; j++) { | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test2 |
|  | System.out.print("\*"); | Meminta user memasukkan data |
|  | } | Menyimpan data dari keyboard ke variabel test3 |
|  | System.out.println(""); | Deklarasi perintah obyek pelajar1 ke method setNama untuk membaca data test3 |
|  | }if (i == 8){ |  |
|  | System.out.print((i) + " : "); | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar2 dan memasukkan datanya |
|  | for (int j = 1; j <= dadu1.acak[7]; j++) { | Sebuah variasi |
|  | System.out.print("\*"); | Memberi perintah obyek pelajar2 untuk menampilkan(get) metode data getName () |
|  | } | Meminta user memasukkan data |
|  | System.out.println(""); | Menyimpan data dari keyboard ke variabel test1 |
|  | }if (i == 9){ | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test1 |
|  | System.out.print((i) + " : "); | Meminta user memasukkan data |
|  | for (int j = 1; j <= dadu1.acak[8]; j++) { | Menyimpan data dari keyboard ke variabel test2 |
|  | System.out.print("\*"); | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test2 |
|  | } | Meminta user memasukkan data |
|  | System.out.println(""); | Menyimpan data dari keyboard ke variabel test3 |
|  | }if (i == 10){ | Deklarasi perintah obyek pelajar2 ke method setNama untuk membaca data test3 |
|  | System.out.print((i) + " : "); |  |
|  | for (int j = 1; j <= dadu1.acak[9];j++) { | Class Student () dipakai untuk membuat obyek baru yang bernama pelajar3 dan memasukkan datanya |
|  | System.out.print("\*"); |  |
|  | } |  |
|  | System.out.println(""); |  |
|  | }if (i == 11){ |  |
|  | System.out.print((i) + " : "); |  |
|  | for (int j = 1; j <= dadu1.acak[10]; j++) { |  |
|  | System.out.print("\*"); |  |
|  | } |  |
|  | System.out.println(""); |  |
|  | }if (i == 12){ |  |
|  | System.out.print((i) + " : "); |  |
|  | for (int j = 1; j <= dadu1.acak[11]; j++) { |  |
|  | System.out.print("\*"); |  |
|  | } |  |
|  | System.out.println(""); |  |
|  | } |  |
|  | } |  |
|  | } |  |

1. Capture Input dan Output

