LAPORAN RESMI

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

OVERLOADING DAN OVERRIDING



Fadilah Fahrul Hardiansyah S.ST., M. Kom

Ratri Maria Manik

3121600039

D4 TEKNIK INFORMATIKA – B

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

TA 2022/2023

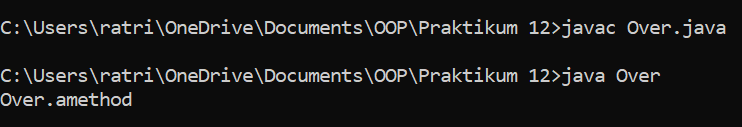
1. **LATIHAN**
2. Overriding
3. Class Base

|  |
| --- |
| class Base{  private void amethod(int iBase){  System.out.println("Base.amethod");  }  } |

1. Class Over

|  |
| --- |
| class Over extends Base{  public static void main(String argv[]){  Over o = new Over();  int iBase=0;  o.amethod(iBase);  }  public void amethod(int iOver){  System.out.println("Over.amethod");  }  } |

1. Hasil kompile



Analisa :

Pada program diatas, overriding method ditunjukkan pada baris 14, terdapat method dari subclass yang memiliki nama dan parameter yang sama dengan method superclass yaitu “amethod(int)”, sedangkan pada baris 2 merupakan overridden method.

1. Overloading
2. Class MyParent

|  |
| --- |
| class MyParent {  int x, y;  MyParent(int x, int y){  this.x = x;  this.y = y;  }  public int addMe(int x, int y){  return this.x + x + y + this.y;  }  public int addMe(MyParent myPar){  return addMe(myPar.x, myPar.y);  }  } |

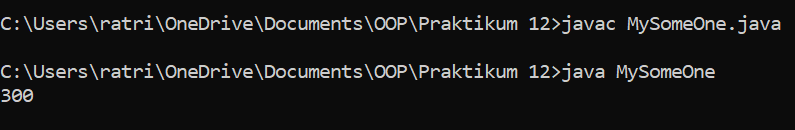
1. Class MyChild.java

|  |
| --- |
| class MyChild extends MyParent{  int z;  MyChild (int x, int y, int z) {  super(x,y);  this.z = z;  }  public int addMe(int x, int y, int z){  return this.x + x + this.y + y + this.z + z;  }  public int addMe(MyChild myChi){  return addMe(myChi.x, myChi.y, myChi.z);  }  public int addMe(int x, int y){  return this.x + x + this.y + y;  }  } |

1. Class MySomeOne.java

|  |
| --- |
| public class MySomeOne{  public static void main(String args[]) {  MyChild myChi = new MyChild(10, 20, 30);  MyParent myPar = new MyParent(10, 20);  int x = myChi.addMe(10, 20, 30);  int y = myChi.addMe(myChi);  int z = myPar.addMe(myPar);  System.out.println(x + y + z);  }  } |

1. Hasil kompile



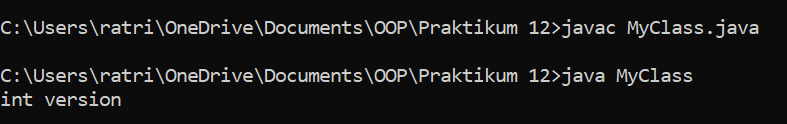
Analisa :

Pada class MyParent overloading method terletak pada baris ke 8 dan 12, sedangkan pada subclassnya, overloading method terletak pada baris ke 9 dan 13. Pada class MySomeOne baris ke 5 dan ke 6 merupakan pemanggilan method addme()

1. Overloading
2. Class MyClass

|  |
| --- |
| class MyClass{  void myMethod(int i) {  System.out.println("int version");  }  void myMethod(String s) {  System.out.println("String version");  }  public static void main(String args[]){  MyClass obj = new MyClass();  char ch = 'c';  obj.myMethod(ch);  }  } |

1. Hasil kompile



Analisa :

Pada program di atas, terdapat 2 overloading method. Pada baris ke 13 dilakukan pemanggilan method.

1. Mengimplementasikan UML class diagram dalam program
2. Class Orang

|  |
| --- |
| public class Orang{  protected String nama;  protected int umur;  public Orang(String nama){  this.nama = nama;  }  public Orang(String nama, int umur){  this.nama = nama;  this.umur = umur;  }  } |

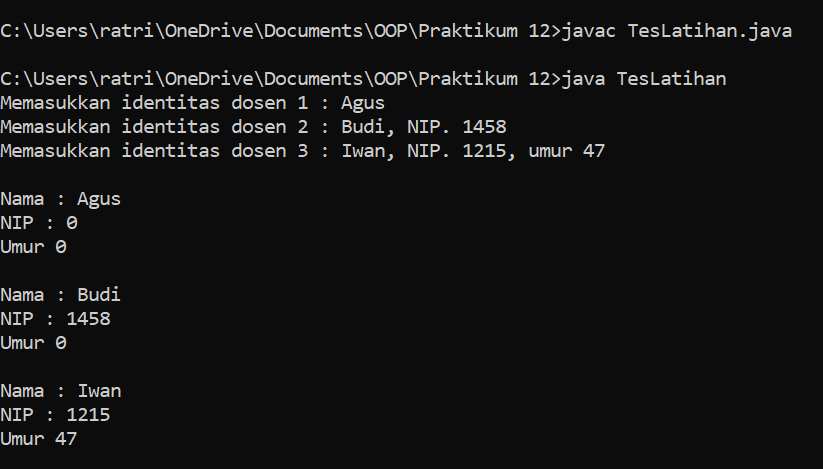
1. Class Dosen

|  |
| --- |
| public class Dosen extends Orang{  private int nip;  public Dosen(String nama){  super(nama);  }  public Dosen(String nama, int nip){  super(nama);  this.nip = nip;  }  public Dosen(String nama, int nip, int umur){  super(nama, umur);  this.nip = nip;  }  public void Info(){  System.out.println("Nama : " + nama);  System.out.println("NIP : " + nip);  System.out.println("Umur " + umur);  }  } |

1. Class TesLatihan

|  |
| --- |
| public class TesLatihan{  public static void main(String args[]){  System.out.println("Memasukkan identitas dosen 1 : Agus");  Dosen dosen1 = new Dosen("Agus");  System.out.println("Memasukkan identitas dosen 2 : Budi, NIP. 1458");  Dosen dosen2 = new Dosen("Budi", 1458);  System.out.println("Memasukkan identitas dosen 3 : Iwan, NIP. 1215, umur 47");  Dosen dosen3 = new Dosen("Iwan", 1215, 47);  System.out.println();  dosen1.Info();  System.out.println();  dosen2.Info();  System.out.println();  dosen3.Info();  }  } |

1. Hasil kompile



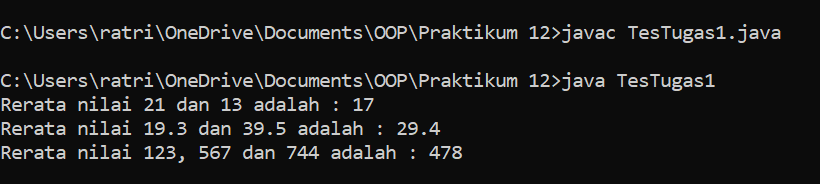
1. **TUGAS**
2. Mengimplementasikan UML class diagram dalam program
3. Class RerataNilai

|  |
| --- |
| public class RerataNilai{  public int average(int x, int y){  return (x+y) / 2;  }  public double average(double x, double y){  return (x+y) / 2;  }    public int average(int x, int y, int z){  return (x+y+z) / 3;  }  } |

1. Class TesTugas1

|  |
| --- |
| public class TesTugas1{  public static void main(String args[]){  RerataNilai rn = new RerataNilai();  System.out.println("Rerata nilai 21 dan 13 adalah : " + rn.average(21, 13));  System.out.println("Rerata nilai 19.3 dan 39.5 adalah : " + rn.average(19.3, 39.5));  System.out.println("Rerata nilai 123, 567 dan 744 adalah : " + rn.average(123, 567, 744));  }  } |

1. Hasil kompile



1. Mengimplementasikan UML class diagram dalam program
2. Class Katak

|  |
| --- |
| public class Katak{  private int umur;  private String nama;  public Katak(int umur, String nama){  this.umur = umur;  this.nama = nama;  }  public String caraBergerak(){  return "melompat";  }  public int getUmur(){  return umur;  }  public String getNama(){  return nama;  }  } |

1. Class Kecebong

|  |
| --- |
| public class Kecebong extends Katak{  private double panjangEkor;  public Kecebong(int umur, String nama, double panjang){  super(umur, nama);  this.panjangEkor = panjang;  }  public String caraBergerak(){  return "berenang";  }  public double getPanjangEkor(){  return panjangEkor;  }  } |

1. Class TesTugas2

|  |
| --- |
| public class TesTugas2{  public static void main(String[] args){  System.out.println("Objek \tumur \tnama \t\tpanjangEkor \tcaraBergerak");  System.out.println("============================================================");  Katak O1 = new Katak(5, "Froggy");  System.out.printf("O1 \t%-7d %-15s %-15s %s\n", O1.getUmur(), O1.getNama(), "", O1.caraBergerak());  Kecebong O2 = new Kecebong(2, "Junior Frog", 10);  System.out.printf("O2 \t%-7d %-15s %-15.0f %s\n", O2.getUmur(), O2.getNama(), O2.getPanjangEkor(), O2.caraBergerak());  System.out.println("============================================================");  }  } |

1. Hasil kompile

