LAPORAN RESMI

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

POLYMORPHISM



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. Class Fans

|  |
| --- |
| public class Fans{  private String name;  public Fans(){  name = "noname";  }  public Fans(String name){  this.name = name;  }  public void showName(){  System.out.print("\n" + name + " : ");  }  public void watchingPerformance(){  System.out.println(this.name + " : " + "melihat idolanya dari youtube");  }  public void watchingPerformance(Musician musician){  showName();  System.out.print("melihat idolanya");  musician.perform();  }  } |

1. Class KpopFans

|  |
| --- |
| public class KpopFans extends Fans{    public KpopFans(){  super();  }  public KpopFans(String name){  super(name);  }  public void watchingPerformance(Musician musician, String expression){  super.showName();  System.out.print(expression + " melihat idolanya");  musician.perform();  }  } |

1. Class Musician

|  |
| --- |
| public class Musician{  public void perform(){  System.out.print("Beraksi di atas panggung");  }  } |

1. Class Singer

|  |
| --- |
| public class Musician{  public void perform(){  System.out.print("Beraksi di atas panggung");  }  } |

1. Class Kpop

|  |
| --- |
| public class Kpop extends Singer{  public void perform(){  super.perform();  System.out.print(", dan ngedance");  }  } |

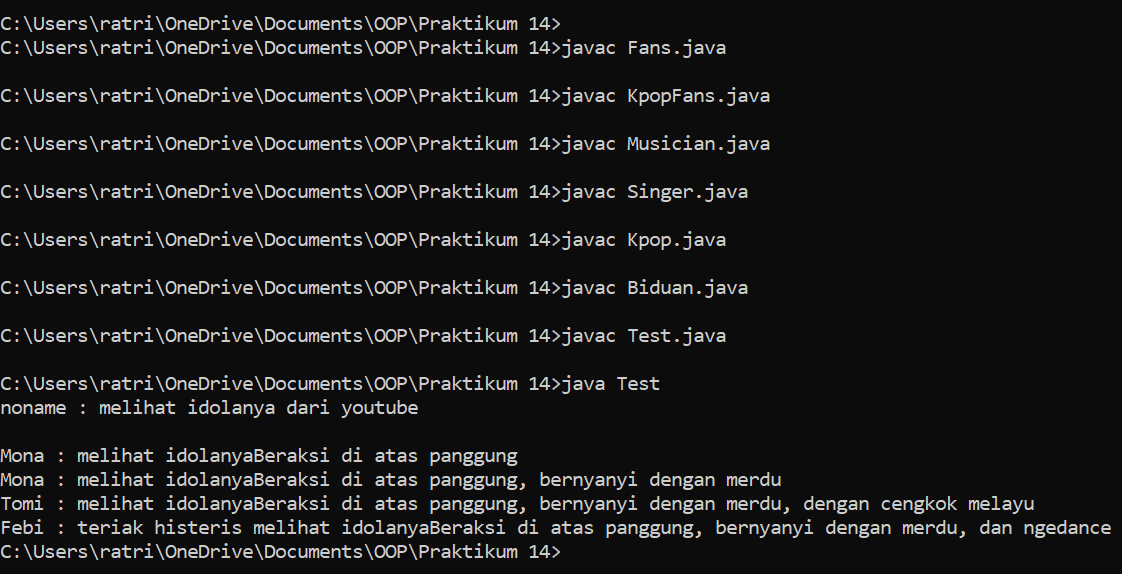
1. Class Biduan

|  |
| --- |
| public class Biduan extends Singer{  public void perform(){  super.perform();  System.out.print(", dengan cengkok melayu");  }  } |

1. Class Test

|  |
| --- |
| public class Test {  public static void main(String args[]){  Fans fans1 = new Fans();  Fans fans2 = new Fans("Mona");  Fans fans3 = new KpopFans("Tomi");  KpopFans fans4 = new KpopFans("Febi");  fans1.watchingPerformance();  fans2.watchingPerformance(new Musician());  fans2.watchingPerformance(new Singer());  fans3.watchingPerformance(new Biduan());  fans4.watchingPerformance(new Kpop(), "teriak histeris");  }  } |

1. Hasil Kompile



**LATIHAN**

1. Tunjukkan contoh overloading dalam percobaan diatas!
2. Overloading Konstruktor

|  |
| --- |
| public class Fans{  private String name;  public Fans(){  name = "noname";  }  public Fans(String name){  this.name = name;  }  } |

|  |
| --- |
| public class KpopFans extends Fans{    public KpopFans(){  super();  }  public KpopFans(String name){  super(name);  }  } |

1. Overloading method

|  |
| --- |
| public void watchingPerformance(){  System.out.println(this.name + " : " + "melihat idolanya dari youtube");  }  public void watchingPerformance(Musician musician){  showName();  System.out.print("melihat idolanya");  musician.perform();  } |

1. Tunjukkan contoh overriding method dan overriden method pada percobaan di atas
2. Overriding method

|  |
| --- |
| public class Singer extends Musician{  public void perform(){  super.perform();  System.out.print(", bernyanyi dengan merdu");  }  } |

|  |
| --- |
| public class Kpop extends Singer{  public void perform(){  super.perform();  System.out.print(", dan ngedance");  }  } |

|  |
| --- |
| public class Biduan extends Singer{  public void perform(){  super.perform();  System.out.print(", dengan cengkok melayu");  }  } |

1. Overriden method

|  |
| --- |
| public class Musician{  public void perform(){  System.out.print("Beraksi di atas panggung");  }  } |

1. Tunjukkan contoh overloading yang terjadi dalam satu class, pada percobaan di atas

|  |
| --- |
| public void watchingPerformance(){  System.out.println(this.name + " : " + "melihat idolanya dari youtube");  }  public void watchingPerformance(Musician musician){  showName();  System.out.print("melihat idolanya");  musician.perform();  } |

1. Tunjukkan contoh overloading yang terjadi antara superclass dan subclass pada percobaan diatas (Fans dan KpopFans)

|  |
| --- |
| public void watchingPerformance(){  System.out.println(this.name + " : " + "melihat idolanya dari youtube");  }  public void watchingPerformance(Musician musician, String expression){  super.showName();  System.out.print(expression + " melihat idolanya");  musician.perform();  } |

1. Tunjukkan contoh virtual method invocation yang terjadi pada percobaan di atas

|  |
| --- |
| fans2.watchingPerformance(new Musician());  fans2.watchingPerformance(new Singer());  fans3.watchingPerformance(new Biduan());  fans4.watchingPerformance(new Kpop(), "teriak histeris"); |

1. Tunjukkan contoh polimorfism pada percobaan di atas

|  |
| --- |
| Fans fans3 = new KpopFans("Tomi"); |

1. Tunjukkan contoh inheritance pada percobaan di atas

|  |
| --- |
| public class KpopFans extends Fans  public class Singer extends Musician  public class Biduan extends Singer  public class Kpop extends Singer |