***LAPORAN TUGAS PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK***

|  |  |
| --- | --- |
| **Pertemuan Ke-:**  **09** | |
| **Pembahasan:**  Abstract dan Interface | |
| **NIM:**  1841720165 | **Dosen Pengampu:**  Septian Enggar Saputra S.PD.M. T. |
| **Nama Mahasiswa:**  Thariq Alfa Benriska | **Nilai:**  97 |

**TUJUAN PRAKTIKUM (10 points)**

1. Menjelaskan maksud dan tujuan penggunaan Abstract Class,

2. Menjelaskan maksud dan tujuan penggunaan Interface,

3. Menerapkan Abstract Class dan Interface di dalam pembuatan program.

**JAWABAN PERTANYAAN (30 points)**

*Petunjuk: jawaban dari pertanyaan yang terdapat pada modul 6 dituliskan pada bagian kolom kotak yang telah disediakan*

**Percobaan 1.**

15. Pertanyaan diskusi: Bolehkah apabila sebuah class yang meng-extend suatu abstract class tidak mengimplementasikan method abstract yang ada di class induknya? Buktikan!

Jawab: Tidak boleh

**Percobaan 2 ( Pertanyaan Diskusi )**

a. Mengapa pada langkah nomor 9 terjadi error? Jelaskan!

Jawab : Karena pada objek pakRektor memanggil Mahasiswa. Sedangkan calss MAhasiswa itu sendiri tidak ter-implements dengan Interface cumlaude. Dan pada class Rektor membutuhkan Interface cumlaude

b. Dapatkah method kuliahDiKampus() dipanggil dari objek sarjanaCumlaude di class Program? Mengapa demikian?

Jawab : Bisa, karena pada class Sarjana sudah di inansiasi menjadi objek sarjana Cumlaude di class Program. Dan class SArjana sudah ter-extends dengan class Mahasiswa

c. Dapatkah method kuliahDiKampus() dipanggil dari parameter mahasiswa di method beriSertifikatCumlaude() pada class Rektor? Mengapa demikian?

Jawab : Tidak boleh, sebab pada object pakRektor memanggil method beriSertifikatCumlaude() untuk membuat inputan sebuah objek yang sudah terinstansi. Bukan untuk memanggil method.

d. Modifikasilah method beriSertifikatCumlaude() pada class Rektor agar hasil eksekusi class Program menjadi seperti berikut ini

Jawab :

Class Sarjana

public class Sarjana extends Mahasiwa implements Icumlaude, Iberprestasi {

public Sarjana(String nama) {

super(nama);

}

@Override

public void lulus() {

super.kuliahDikampus();

System.out.println("Aku sudah menyelesaikan Skripsi");

}

@Override

public void meraihIKPTingggi() {

System.out.println("IPK-KU lebih dari 3,51");

}

Class PascaSarjana

public class PascaSarjana extends Mahasiwa implements Icumlaude, Iberprestasi {

public PascaSarjana(String nama) {

super(nama);

}

@Override

public void lulus() {

super.kuliahDikampus();

System.out.println("Aku sudah menyelasaikan TESIS");

}

@Override

public void meraihIKPTingggi() {

System.out.println("IPK-KU lebih dari 3,71");

}

**Percobaan 2 ( Pertanyaan Diskusi )**

a. Pertanyaan diskusi

Apabila Sarjana Berprestasi harus menjuarai kompetisi NASIONAL dan menerbitkan artikel di jurnal NASIONAL, maka modifikasilah class-class yang terkait pada aplikasi Anda agar di class Program objek pakRektor dapat memberikan sertifikat mawapres pada objek sarjanaCumlaude.

Jawab :

public class Program {

public static void main(String[] args) {

Rektor pak = new Rektor();

Mahasiwa mahasiswabiasa = new Mahasiwa("Charie");

Sarjana sarjanacumlaude = new Sarjana("Dini");

PascaSarjana masterCumlude = new PascaSarjana("elok");

pak.beriSertifikatMawapres(sarjanacumlaude);

pak.beriSertifikatMawapres(masterCumlude);

}

}

**KODE PROGRAM DAN PENJELASAN TIAP METHODNYA (30 points)**

Interface

|  |
| --- |
| Nama interface: Penilaian |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public interface Penilaian {  public abstract void pindahGigi();  public abstract void putarKemudi();  public abstract void injakPedalGas();  public abstract void injakPedalRem();  public abstract void getNama();  } |
| Penjelasan:  Merupakan interface penilaian yang berikisan abstract method |

|  |
| --- |
| Nama Class: Pengemudi |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public abstract class Pengemudi implements Penilaian{  protected String nama;    public Pengemudi(String nama){  this.nama = nama;    }    public void getNama(){  System.out.println("Nama Pengemudi : " + this.nama);  }  public abstract void pindahGigi();  public abstract void putarKemudi();  public abstract void injakPedalGas();  public abstract void injakPedalRem();  } |
| Penjelasan:  Konstruktor public Pengemudi(String nama) = untuk memberi nilai variabel nama.  public void getNama() = method unntuk menampilkan nilai nama.  public abstract void pindahGigi(); merupakan method abstract  public abstract void putarKemudi();merupakan method abstract  public abstract void injakPedalGas();merupakan method abstract  public abstract void injakPedalRem();merupakan method abstract |

|  |
| --- |
| Nama Class: PengemudiAmateur |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public class PengemudiAmateur extends Pengemudi implements Penilaian{  public PengemudiAmateur(String nama) {  super(nama);  }  @Override  public void pindahGigi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara memindahkan gigi : Acak");  }  @Override  public void putarKemudi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara putar kemudi : Kurang menaati aturan yang ada");  }  @Override  public void injakPedalGas() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal gas : Asal ");  }  @Override  public void injakPedalRem() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal rem : Asal");  }  } |
| Penjelasan:  public PengemudiAmateur(String nama) = konstruktor untuk memberi nama yang merupakan turunan dari pengemudi  public void pindahGigi(); = method untuk menempilkan cara pindah gigi pengemudi amatir  public void putarKemudi();= method untuk menempilkan cara putar kemudi pengemudi amatir  public void injakPedalGas(); = method untuk menempilkan cara injak pedal gas pengemudi amatir  public void injakPedalRem(); = method untuk menempilkan cara injak pedal rem pengemudi amatir |

|  |
| --- |
| Nama Class: PengemudiPro |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public class PengemudiPro extends Pengemudi implements Penilaian{  public PengemudiPro(String nama) {  super(nama);  }  @Override  public void pindahGigi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara memindahkan gigi : Berurutan dan halus");  }  @Override  public void putarKemudi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara putar kemudi : Taat aturan dan undang-undang");  }  @Override  public void injakPedalGas() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal gas : Berekualitas");  }  @Override  public void injakPedalRem() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal rem : Mulus");  }  } |
| Penjelasan:  public PengemudiAmateur(String nama) = konstruktor untuk memberi nama yang merupakan turunan dari pengemudi  public void pindahGigi(); = method untuk menempilkan cara pindah gigi pengemudi profesional  public void putarKemudi();= method untuk menempilkan cara putar kemudi pengemudi profesional  public void injakPedalGas(); = method untuk menempilkan cara injak pedal gas pengemudi profesional  public void injakPedalRem(); = method untuk menempilkan cara injak pedal rem pengemudi profesional |

|  |
| --- |
| Nama Class: Cetak |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public class Cetak {  public void Cetak(Penilaian x){  x.getNama();  x.pindahGigi();  x.putarKemudi();  x.injakPedalGas();  x.injakPedalRem();  System.out.println("---------------------------------------");  }  } |
| Penjelasan:  public void Cetak(Penilaian x) = method untuk menampilkan beberapa method yang ada dalam method cetak dengan ditentukan dari tipe data penilaian dan bervariabel x |

|  |
| --- |
| Nama Class: Main |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public class Main {  public static void main(String[] args) {  PengemudiPro pPro = new PengemudiPro("Endg");  PengemudiAmateur pAmateur = new PengemudiAmateur("Sikis");    Cetak c = new Cetak();  c.Cetak(pPro);  c.Cetak(pAmateur);    }  } |
| Penjelasan: |

Abstract

|  |
| --- |
| Nama Class: Pengemudi |
| package Abstract.PengemudiAbstract;  public abstract class Pengemudi {  protected String nama;    public Pengemudi(String nama){  this.nama = nama;    }    public void getNama(){  System.out.println("Nama Pengemudi : " + this.nama);  }  public abstract void pindahGigi();  public abstract void putarKemudi();  public abstract void injakPedalGas();  public abstract void injakPedalRem();  } |
| Penjelasan:  Konstruktor public Pengemudi(String nama) = untuk memberi nilai variabel nama.  public void getNama() = method unntuk menampilkan nilai nama.  public abstract void pindahGigi(); merupakan method abstract  public abstract void putarKemudi();merupakan method abstract  public abstract void injakPedalGas();merupakan method abstract  public abstract void injakPedalRem();merupakan method abstract |

|  |
| --- |
| Nama Class: PengemudiAmateur |
| package Abstract.PengemudiAbstract;  /\*\*  \*  \* @author Macair4  \*/  public class PengemudiAmateur extends Pengemudi{  public PengemudiAmateur(String nama) {  super(nama);  }  @Override  public void pindahGigi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara memindahkan gigi : Acak");  }  @Override  public void putarKemudi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara putar kemudi : Kurang menaati aturan yang ada");  }  @Override  public void injakPedalGas() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal gas : Asal ");  }  @Override  public void injakPedalRem() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal rem : Asal");  }    } |
| Penjelasan:  public PengemudiAmateur(String nama) = konstruktor untuk memberi nama yang merupakan turunan dari pengemudi  public void pindahGigi(); = method untuk menempilkan cara pindah gigi pengemudi amatir  public void putarKemudi();= method untuk menempilkan cara putar kemudi pengemudi amatir  public void injakPedalGas(); = method untuk menempilkan cara injak pedal gas pengemudi amatir  public void injakPedalRem(); = method untuk menempilkan cara injak pedal rem pengemudi amatir |

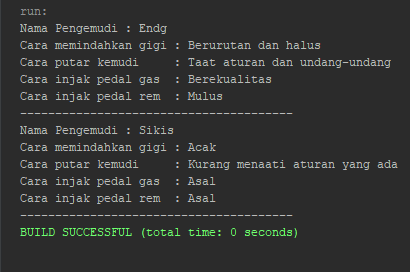
|  |
| --- |
| Nama Class: PengemudiPro |
| package Abstract.PengemudiAbstract;  /\*\*  \*  \* @author Macair4  \*/  public class PengemudiPro extends Pengemudi{  public PengemudiPro(String nama) {  super(nama);  }  @Override  public void pindahGigi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara memindahkan gigi : Berurutan dan halus");  }  @Override  public void putarKemudi() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara putar kemudi : Taat aturan dan undang-undang");  }  @Override  public void injakPedalGas() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal gas : Berekualitas");  }  @Override  public void injakPedalRem() {  // throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.  System.out.println("Cara injak pedal rem : Mulus");  }  } |
| Penjelasan:  public PengemudiAmateur(String nama) = konstruktor untuk memberi nama yang merupakan turunan dari pengemudi  public void pindahGigi(); = method untuk menempilkan cara pindah gigi pengemudi profesional  public void putarKemudi();= method untuk menempilkan cara putar kemudi pengemudi profesional  public void injakPedalGas(); = method untuk menempilkan cara injak pedal gas pengemudi profesional  public void injakPedalRem(); = method untuk menempilkan cara injak pedal rem pengemudi profesional |

|  |
| --- |
| Nama Class: Cetak |
| package Abstract.PengemudiAbstract;  /\*\*  \*  \* @author Macair4  \*/  public class Cetak {  public void Cetak(Pengemudi x){  x.getNama();  x.pindahGigi();  x.putarKemudi();  x.injakPedalGas();  x.injakPedalRem();  System.out.println("---------------------------------------"); |
| Penjelasan:  public void Cetak(Penilaian x) = method untuk menampilkan beberapa method yang ada dalam method cetak dengan ditentukan dari tipe data penilaian dan bervariabel x |

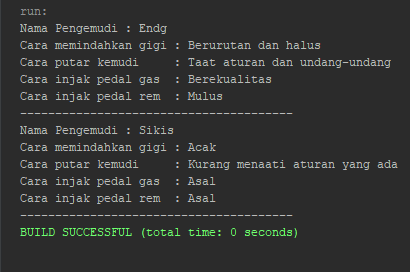
|  |
| --- |
| Nama Class: Main |
| package Abstract.pengemudiInterface;  /\*\*  \*  \* @author Macair4  \*/  public class Main {  public static void main(String[] args) {  PengemudiPro pPro = new PengemudiPro("Endg");  PengemudiAmateur pAmateur = new PengemudiAmateur("Sikis");    Cetak c = new Cetak();  c.Cetak(pPro);  c.Cetak(pAmateur);    }  } |
| Penjelasan: |

**HASIL (15 points)**

Interface



Abstract



**KESIMPULAN (15 points)**

Abstract Class dapat menggambarkan sesuatu yang bersifat umum, yang hanya bisa berfungsi setelah ia dideskripsikan ke dalam bentuk yang lebih spesifik dan sedangkan Interface bertindak seperti semacam kontrak / syarat yang HARUS dipenuhi bagi suatu class agar class tersebut dapat dianggap sebagai ‘sesuatu yang lain’.