**LAPORAN PRAKTIKUM PEMROGRAMAN BEORIENTASI OBJEK**

“Tugas 04 - Inheritance lanjutan plymorphism*”*

A yellow and black logo

Description automatically generated

ms

**Andika Haikal Syahputra**

**4523210016**

Dosen:

Adi Wahyu Pribadi , S.Si., M.Kom

**FAKULTAS TEKNIK INFORMATIKA**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024/2025**

1. **Implementasi Macam - macam Inheritance**
2. **Single**

* App.java

| public class App {  public static void main(String[] args) {  Mobil m1 = new Mobil("Hino");    Sedan s1 = new Sedan("Vios");    m1.klakson();  s1.klakson();  }  } |
| --- |

* Mobil.java

| ublic class Mobil {  protected String nama;  public Mobil(String nama) {  this.nama = nama;  }  public void klakson() {  System.out.println("Teet");  }  } |
| --- |

* Sedan.java

| public class Sedan extends Mobil {  public Sedan(String nama) {  super(nama);  }  @Override  public void klakson() {  System.out.println("Tiin");  }  } |
| --- |

* Output

|  |
| --- |

1. **Multilevel**

* App.java

| public class App {  public static void main(String[] args) {  Binatang kambing = new Binatang("kambing");  Kucing mudreg = new Kucing("mudreg");  Cemeng mocha = new Cemeng("mocha");  kambing.bersuara();  mudreg.bersuara();  mocha.bersuara();  }  } |
| --- |

* Binatang.java

| class Binatang{  protected String nama;  public Binatang (String nama){  this.nama = nama;  }  public void bersuara(){  System.out.println("mbek");  }  } |
| --- |

* Cemeng.java

| public class Cemeng extends Kucing {  public Cemeng(String nama) {  super(nama);  }  @Override  public void bersuara() {  System.out.println("Ngeeeong");  }  } |
| --- |

* Kucing.java

| public class Kucing extends Binatang {  private String nama;  public Kucing(String nama) {  super(nama);  }  public void bersuara() {  System.out.println("Meong");  }  } |
| --- |

* Output

|  |
| --- |

1. **Hierarchical**

* Dosen.java

| class Dosen extends Karyawan {  private String NIDN;  public Dosen(String kodeKaryawan, String NIDN, String nama) {  super(kodeKaryawan, nama);  this.NIDN = NIDN;  }  public void setNIDN(String NIDN) {  this.NIDN = NIDN;  }  public void getNIDN() {  System.out.println("NIDN: " + this.NIDN);  }  public void ngajar() {  System.out.println(this.nama + ": sedang mengajar");  }  @Override  public void getInfo() {  super.getInfo();  System.out.println("NIDN: " + this.NIDN);  }  } |
| --- |

* Karyawan.java

| class Karyawan {  protected String kodeKaryawan;  protected String nama;  public Karyawan(String kodeKaryawan, String nama) {  this.kodeKaryawan = kodeKaryawan;  this.nama = nama;  }  public void absenPagi() {  System.out.println(this.nama + ": absen pagi");  }  public void kerja() {  System.out.println(this.nama + ": sedang bekerja");  }  public void absenPulang() {  System.out.println(this.nama + ": absen pulang");  }  public void getInfo() {  System.out.println("Kode Karyawan: " + this.kodeKaryawan);  System.out.println("Nama: " + this.nama);  }  } |
| --- |

* Main.java

| public class Main {  public static void main(String[] args) throws Exception {  Karyawan Andika = new Karyawan("23210016", "Andika Haikal S");  Andika.getInfo();  Andika.absenPagi();  Andika.kerja();  Andika.absenPulang();    System.out.println();    Karyawan Andre = new Karyawan("23210020", "Andre");  Andre.getInfo();  Andre.absenPagi();  Andre.kerja();  Andre.absenPulang();    System.out.println();  Dosen Rina = new Dosen("2323", "2121", "Rina");  Rina.getInfo();  Rina.absenPagi();  Rina.kerja();  Rina.ngajar();  Rina.absenPulang();  System.out.println();    Dosen buion = new Dosen ("112233", "1133", "Bu Ion");  buion.getInfo();  buion.absenPagi();  buion.kerja();  buion.ngajar();  buion.absenPulang();  }  } |
| --- |

* Output

|  |
| --- |

**II. Implementasi Polymorphism**

1. **Method OverloadingStatic Polymorphism**

* MobilSport.java

| public class Main {  public static void main(String[] args) {  MobilSport jaguar = new MobilSport("Jaguar");  for (int i = 1; i <= 5; i++) {  jaguar.gigi(i);  }    for (int t = 1; t <= 5; t++) {  jaguar.gigi(5, t);  }  }  } |
| --- |

* Main.java

| public class Main {  public static void main(String[] args) {  MobilSport jaguar = new MobilSport("Jaguar");  for (int i = 1; i <= 5; i++) {  jaguar.gigi(i);  }    for (int t = 1; t <= 5; t++) {  jaguar.gigi(5, t);  }  }  } |
| --- |

* Output

|  |
| --- |

1. **Method Overriding Dynamic Polymorphism**

* App.java

| public class App {  public static void main(String[] args) {  BangunDatar bd = new BangunDatar();  bd.luas();  bd.keliling();  Lingkaran lk = new Lingkaran(15);  System.out.println("Luas lingkaran: " + lk.luas());  System.out.println("keliling lingkaran: " + lk.keliling());  Persegi pj = new Persegi(10);  System.out.println("Luas Bujur Sangkar: " + pj.luas());  System.out.println("keliling Bujur Sangkar: " + pj.keliling());  Segitiga sg = new Segitiga(10, 8);  System.out.println("Luas Segitiga: " + sg.luas());    sg.keliling();  }  } |
| --- |

* BangunDatar.java

| public class BangunDatar {  float luas(){  System.out.println("Menghitung luas bangun datar");  return 0;  }    float keliling(){  System.out.println("Menghitung keliling bangun datar");  return 0;  }  } |
| --- |

* Lingkaran.java

| public class Lingkaran extends BangunDatar {  private int r;  public Lingkaran(int r) {  this.r = r;  }    @Override  public float luas(){  return (float) (Math.PI \* r \* r);  }    @Override  public float keliling(){  return (float) (2 \* Math.PI \* r);  }    } |
| --- |

* Persegi.java

| public class Persegi extends BangunDatar{  private int sisi;    public Persegi(int sisi) {  this.sisi = sisi;  }    @Override  public float luas() {  return this.sisi \* this.sisi;  }    @Override  public float keliling(){  return this.sisi \* 4;  }  } |
| --- |

* Segitiga.java

| public class Segitiga extends BangunDatar{  private int alas;  private int tinggi;  public Segitiga(int alas, int tinggi) {  this.alas = alas;  this.tinggi = tinggi;  }    @Override  public float luas(){  return (this.alas \* this.tinggi) / 2;  }  } |
| --- |

* Output

|  |
| --- |

Kesimpulannya untuk memahami Praktikum Pemprograman Beorientasi Objek dalam memahami JAVA dan Inheritance lanjutan plymorphism.

TERIMAKASIH