

**LAPORAN HASIL PRAKTIKUM 09**  
**PEMROMGRAMAN BERBASIS OBJEK**



**ATHAULLA HAFIZH**

**244107020030**

**TI 2A**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

### 3. Praktikum

#### Percobaan 1

##### Class Hewan (Hewan.java)

```
public abstract class Hewan {  
    private int umur;  
  
    protected Hewan() {  
        this.umur = 0;  
    }  
  
    public void bertambahUmur() {  
        this.umur += 1;  
    }  
  
    public abstract void bergerak();  
}
```

##### Class Kucing (Kucing.java)

```
public class Kucing extends Hewan {  
    @Override  
    public void bergerak() {  
        System.out.println("Berjalan dengan KAKI, \nTap..tap..\n");  
    }  
}
```

##### Class Ikan (Ikan.java)

```
public class Ikan extends Hewan {  
    @Override  
    public void bergerak() {  
        System.out.println("Berenang dengan SIRIP,  
        \nwush..wush..\n");  
    }  
}
```

### Class Orang (Orang.java)

```
public class Orang {  
    private String nama;  
    private Hewan hewanPeliharaan;  
  
    public Orang(String nama) {  
        this.nama = nama;  
    }  
  
    public void peliharaHewan(Hewan hewanPeliharaan) {  
        this.hewanPeliharaan = hewanPeliharaan;  
    }  
  
    public void ajakPeliharaanJalanJalan() {  
        System.out.println("Namaku " + this.nama);  
        System.out.println("Hewan peliharaanku berjalan dengan cara:"  
);  
        this.hewanPeliharaan.bergerak();  
        System.out.println("-----  
---");  
    }  
}
```

### Class Program1 (Program1.java)

```
public class Program1 {  
    public static void main(String[] args) {  
        Kucing kucingKampung = new Kucing();  
        Ikan lumbaLumba = new Ikan();  
  
        Orang ani = new Orang("Ani");  
        Orang budi = new Orang("Budi");  
  
        ani.peliharaHewan(kucingKampung);  
        budi.peliharaHewan(lumbaLumba);  
    }  
}
```

```
        ani.ajakPeliharaanJalanJalan();  
        budi.ajakPeliharaanJalanJalan();  
    }  
}
```

### Jawaban Diskusi :

**Pertanyaan:** Bolehkah class turunan tidak mengimplementasikan *abstract method*?

**Jawaban:** Tidak boleh, jika class turunan itu adalah class *konkret* (bukan *abstract*). Error kompilasi akan terjadi karena class tersebut gagal memenuhi "kontrak" dari abstract class induknya.

### Output

```
Namaku Ani  
Hewan peliharaanku berjalan dengan cara:  
Berjalan dengan KAKI, "Tap..tap.."  
-----  
Namaku Budi  
Hewan peliharaanku berjalan dengan cara:  
Berenang dengan SIRIP, "wush..wush.."  
-----
```

## Percobaan 2

### Interface ICumlaude (ICumlaude.java)

```
public interface ICumlaude {  
    public abstract void lulus();  
    public abstract void meraihIPKTinggi();  
}
```

### Class Mahasiswa (Mahasiswa.java)

```
public class Mahasiswa {  
    protected String nama;  
  
    public Mahasiswa(String nama) {  
        this.nama = nama;  
    }  
  
    public void kuliahDiKampus() {  
        System.out.println("Aku mahasiswa, namaku " + this.nama);  
        System.out.println("Aku berkuliah di kampus.");  
    }  
}
```

### Class Sarjana (Sarjana.java)

```
public class Sarjana extends Mahasiswa implements ICumlaude {  
    public Sarjana(String nama) {  
        super(nama);  
    }  
  
    @Override  
    public void lulus() {  
        System.out.println("Aku sudah menyelesaikan SKRIPSI");  
    }  
  
    @Override  
    public void meraihIPKTinggi() {  
        System.out.println("IPK-ku lebih dari 3,51");  
    }  
}
```

### Class PascaSarjana (PascaSarjana.java)

```
public class Pascasarjana extends Mahasiswa implements ICumlaude {  
    public Pascasarjana(String nama) {  
        super(nama);  
    }  
  
    @Override  
    public void lulus() {  
        System.out.println("Aku sudah menyelesaikan TESIS");  
    }  
  
    @Override  
    public void meraihIPKTinggi() {  
        System.out.println("IPK-ku lebih dari 3,71");  
    }  
}
```

### Class Rektor (Rektor.java)

```
public class Rektor {

    public void beriSertifikatCumlaude(ICumlaude mahasiswa) {

        System.out.println("Saya REKTOR, memberikan sertifikat  
cumlaude.");

        System.out.println("Selamat! silahkan perkenalkan diri  
Anda..");

        mahasiswa.lulus();

        mahasiswa.meraihIPKTinggi();

        System.out.println("-----  
---");

    }

}
```

### Class Program2 (Program2.java)

```
public class Program2 {

    public static void main(String[] args) {

        Rektor pakRektor = new Rektor();

        Mahasiswa mahasiswaBiasa = new Mahasiswa("Charlie");
        Sarjana sarjanaCumlaude = new Sarjana("Dini");
        PascaSarjana masterCumlaude = new PascaSarjana("Elok");

        pakRektor.beriSertifikatCumlaude(mahasiswaBiasa);
        pakRektor.beriSertifikatCumlaude(sarjanaCumlaude);
        pakRektor.beriSertifikatCumlaude(masterCumlaude);

    }

}
```

### **Jawaban Diskusi :**

**a. Mengapa error di langkah 9?**

Karena mahasiswaBiasa (tipe Mahasiswa) tidak meng-implement ICumlaude, sehingga tidak bisa dikirim ke method beriSertifikatCumlaude yang parameternya bertipe ICumlaude.

**b. Bisakah sarjanaCumlaude.kuliahDiKampus() dipanggil di Program?**

Ya, bisa. sarjanaCumlaude bertipe Sarjana, dan Sarjana extends Mahasiswa, jadi ia mewarisi method kuliahDiKampus().

**c. Bisakah mahasiswa.kuliahDiKampus() dipanggil di Rektor?**

Tidak bisa (langsung). Parameter mahasiswa di method beriSertifikatCumlaude bertipe ICumlaude. Tipe ICumlaude hanya tahu method lulus() dan meraihIPKTinggi() .

**d. Modifikasilah method beriSertifikatCumlaude() pada class Rektor**

Sudah saya modif pada kode diatas



### Percobaan 3

#### Interface IBerprestasi (IBerprestasi.java)

```
public interface IBerprestasi {  
    public abstract void menjuaraiKompetisi();  
    public abstract void membuatPublikasiIlmiah();  
}
```

#### Class PascaSarjana (PascaSarjana.java)

```
public class PascaSarjana extends Mahasiswa implements ICumlaude,  
IBerprestasi {  
    public PascaSarjana(String nama) {  
        super(nama);  
    }  
  
    @Override  
    public void lulus() {  
        System.out.println("Aku sudah menyelesaikan TESIS");  
    }  
  
    @Override  
    public void meraihIPKTinggi() {  
        System.out.println("IPK-ku lebih dari 3,71");  
    }  
  
    @Override  
    public void menjuaraiKompetisi() {  
        System.out.println("Saya telah menjuarai kompetisi  
INTERNASIONAL");  
    }  
  
    @Override  
    public void membuatPublikasiIlmiah() {  
        System.out.println("Saya menerbitkan artikel di jurnal  
INTERNASIONAL");  
    }  
}
```

```
}  
  
}
```

### Class Rektor (Rektor.java)

```
public class Rektor {  
  
    public void beriSertifikatCumlaude(ICumlaude mahasiswa) {  
        System.out.println("Saya REKTOR, memberikan sertifikat  
cumlaude.");  
  
        System.out.println("Selamat! Bagaimana Anda bisa  
cumlaude?");  
  
        ((Mahasiswa) mahasiswa).kuliahDiKampus();  
  
        mahasiswa.lulus();  
        mahasiswa.meraihIPKTINGGI();  
  
        System.out.println("-----  
---");  
    }  
}
```

### Class Program2 (Program2.java)

```
public class Program2 {  
    public static void main(String[] args) {  
        Rektor pakRektor = new Rektor();  
  
        Sarjana sarjanaCumlaude = new Sarjana("Dini");  
        PascaSarjana masterCumlaude = new PascaSarjana("Elok");  
  
        System.out.println("--- Pemberian Sertifikat MAWAPRES ---");  
  
        pakRektor.beriSertifikatMawapres(sarjanaCumlaude);  
        pakRektor.beriSertifikatMawapres(masterCumlaude);  
    }  
}
```

```
}  
}
```

### Jawaban Diskusi :

**a. Mengapa error di langkah 9?**

Karena mahasiswaBiasa (tipe Mahasiswa) tidak meng-implement ICumlaude, sehingga tidak bisa dikirim ke method beriSertifikatCumlaude yang parameternya bertipe ICumlaude.

**b. Bisakah sarjanaCumlaude.kuliahDiKampus() dipanggil di Program?**

Ya, bisa. sarjanaCumlaude bertipe Sarjana, dan Sarjana extends Mahasiswa, jadi ia mewarisi method kuliahDiKampus().

**c. Bisakah mahasiswa.kuliahDiKampus() dipanggil di Rektor?**

Tidak bisa (langsung). Parameter mahasiswa di method beriSertifikatCumlaude bertipe ICumlaude. Tipe ICumlaude hanya tahu method lulus() dan meraihIPKtinggi() .

**d. Modifikasilah method beriSertifikatCumlaude() pada class Rektor agar hasil eksekusi class Program**

Sudah saya tambahkan sekalian diatas

### Output

```
Saya REKTOR, memberikan sertifikat MAWAPRES.  
Selamat! Bagaimana Anda bisa berprestasi?  
Saya telah menjuarai kompetisi INTERNASIONAL  
Saya menerbitkan artikel di jurnal INTERNASIONAL
```

## Jawaban Diskusi :

Penambahan yang saya lakukan sebagai berikut:

```
public void beriSertifikatMawapres(IBerprestasi mahasiswa) {  
    System.out.println("Saya REKTOR, memberikan sertifikat  
MAWAPRES.");  
  
    System.out.println("Selamat! Bagaimana Anda bisa  
berprestasi?");  
  
    mahasiswa.menjuaraiKompetisi();  
    mahasiswa.membuatPublikasiIlmiah();  
  
    System.out.println("-----  
---");  
}  
}
```

## Output

```
--- Pemberian Sertifikat CUMLAUDE ---  
Saya REKTOR, memberikan sertifikat cumlaude.  
Selamat! Bagaimana Anda bisa cumlaude?  
Aku mahasiswa, namaku Dini  
Aku berkuliah di kampus.  
Aku sudah menyelesaikan SKRIPSI  
IPK-ku lebih dari 3,51  
-----  
Saya REKTOR, memberikan sertifikat cumlaude.  
Selamat! Bagaimana Anda bisa cumlaude?  
Aku mahasiswa, namaku Elok  
Aku berkuliah di kampus.  
Aku sudah menyelesaikan TESIS  
IPK-ku lebih dari 3,71  
-----  
--- Pemberian Sertifikat MAWAPRES ---  
Saya REKTOR, memberikan sertifikat MAWAPRES.  
Selamat! Bagaimana Anda bisa berprestasi?  
Saya telah menjuarai kompetisi NASIONAL  
Saya menerbitkan artikel di jurnal NASIONAL  
-----  
Saya REKTOR, memberikan sertifikat MAWAPRES.  
Selamat! Bagaimana Anda bisa berprestasi?  
Saya telah menjuarai kompetisi INTERNASIONAL  
Saya menerbitkan artikel di jurnal INTERNASIONAL  
-----
```