

PERTEMUAN KE – 9

kelas Abstrak

A. TUJUAN

Dapat menjelaskan, mendefinisikan, membuat dan menggunakan kelas abstrak serta dapat menjelaskan membuat dan menggunakan interface pada aplikasi

B. TEORI SINGKAT

Abstraksi mengacu pada kemampuan untuk membuat kelas abstrak di OOP. Sebuah kelas abstrak adalah salah satu yang tidak bisa dipakai. Semua fungsi lain dari kelas masih ada, dan ladang-ladanganya, metode, dan konstruktor semua diakses dengan cara yang sama. Anda hanya tidak bisa membuat sebuah instance dari kelas abstrak.

Jika kelas abstrak dan tidak dapat dipakai, kelas tidak memiliki banyak digunakan kecuali subclass. Ini biasanya bagaimana kelas abstrak terjadi selama tahap desain. Sebuah kelas induk berisi fungsi umum dari kumpulan anak kelas, tapi kelas induk itu sendiri terlalu abstrak untuk digunakan sendiri.

Abstrak Kelas:

Gunakan kata kunci `abstract` untuk menyatakan `abstract class`. Kata kunci muncul dalam deklarasi kelas di suatu tempat sebelum kata kunci kelas.

C. PRAKTIK

Praktik 1: membuat Karyawan

```
public abstract class Karyawan
{
    private String name;
    private String address;
    private int number;
    public Karyawan(String name, String address, int number)
    {
        System.out.println("Constructing Karyawan");
        this.name = name;
        this.address = address;
        this.number = number;
    }
    public double computePay()
    {
        System.out.println("Inside Karyawan computePay");
        return 0.0;
    }
    public void mailCheck()
```

```

    {
        System.out.println("Mailing a check to " + this.name
            + " " + this.address);
    }
    public String toString()
    {
        return name + " " + address + " " + number;
    }
    public String getName()
    {
        return name;
    }
    public String getAddress()
    {
        return address;
    }
    public void setAddress(String newAddress)
    {
        address = newAddress;
    }
    public int getNumber()
    {
        return number;
    }
}

```

Perhatikan bahwa tidak ada yang berbeda dalam kelas Karyawan ini. Kelas abstrak, tetapi masih memiliki tiga atribut, tujuh metode, dan satu konstruktor.

Praktik 2. Menggunakan kelas Karyawan

```

public class AbstractDemo
{
    public static void main(String [] args)
    {
        Karyawan e = new Karyawan("George W.", "Houston, TX", 43);

        System.out.println("\n Panggil mailCheck pakai Karyawan");
        e.mailCheck();
    }
}

```

Ketika dikompilasi akan didapatkan error sebagai berikut:

```

Karyawan.java:46: Karyawan is abstract; cannot be instantiated
    Karyawan e = new Karyawan("George W.", "Houston, TX", 43);
                    ^

```

1 error

Praktik 3. Extend kelas Abstrak

```

public class Gaji extends Karyawan
{
    private double Gaji; //Annual Gaji
    public Gaji(String name, String address, int number, double Gaji)

```

```

{
    super(name, address, number);
    setGaji(Gaji);
}
public void mailCheck()
{
    System.out.println("Within mailCheck Gaji");
    System.out.println("Mailing check to " + getName()
        + " with Gaji " + Gaji);
}
public double getGaji()
{
    return Gaji;
}
public void setGaji(double newGaji)
{
    if(newGaji >= 0.0)
    {
        Gaji = newGaji;
    }
}
public double computePay()
{
    System.out.println("Computing Gaji pay for " + getName());
    return Gaji/52;
}
}

```

Praktik 4. Menggunakan Gaji

Di sini, kita tidak bisa instantiate Karyawan baru, tetapi jika kita instantiate objek Gaji baru, objek Gaji akan mewarisi tiga bidang dan tujuh metode dari karyawan.

```

public class AbstractDemo
{
    public static void main(String [] args)
    {
        Gaji s = new Gaji("Mohd Mohtashim", "Ambehta, UP", 3, 3600.00);
        Karyawan e = new Gaji("John Adams", "Boston, MA", 2, 2400.00);

        System.out.println("Panggil mailCheck pakai Gaji");
        s.mailCheck();

        System.out.println("\n Panggil mailCheck pakai Karyawan");
        e.mailCheck();
    }
}

```

D. LATIHAN

- Latihan diberikan oleh dosen pengampu pada saat praktikum.
- Dikerjakan di laboratorium pada jam praktikum.

E. TUGAS

- *Tugas diberikan oleh dosen pengampu pada akhir praktikum.*
- *Dikerjakan di rumah dan dilampirkan pada laporan.*