



**unindra**  
universitas indraprasta pgri

**Abstract**

**Abstract**

**PERTEMUAN 15**

# ABSTRACT

- **Abstraksi** adalah proses menyembunyikan detail implementasi , dan hanya menampilkan fungsionalitas kepada pengguna.
- Abstrak secara umum dapat kita artikan sebagai gambaran tentang sesuatu hal namun belum begitu jelas gambarannya. Sehingga masih butuh suatu proses lagi supaya gambaran tentang sesuatu tersebut menjadi lebih jelas (konkrit).
- Kelas Abstrak tidak berbeda dengan kelas- kelas lainnya yaitu memiliki class members (method dan variabel). Sebuah class adalah abstrak jika salah satu methodnya di deklarasikan abstract.
- Method abstrak adalah method yang ditentukan dari dalam kelas tapi tidak disertai definisinya .

# TUJUAN ABSTRACT

Sebuah kelas dapat dideklarasikan sebagai kelas abstrak. Tujuannya :

1. Agar suatu kelas *tidak dapat di instansiasikan* sebagai sebuah objek dan hanya dapat diturunkan.
2. Agar satu kelas lain dapat memperluasnya dengan jalan menjadi subclass darinya

Beberapa hal yang perlu diperhatikan adalah sebagai berikut:

- Class abstrak tidak dapat dibuatkan instant atau objeknya menggunakan keyword new.
- Sebuah class dapat dideklarasikan sebagai class abstrak walaupun tidak memiliki method abstrak.

# Deklarasi ABSTRACT

Untuk membuat sebuah kelas atau method menjadi abstrak maka menggunakan keyword **abstract** setelah modifier pada deklarasi kelas atau method. Contoh kelas abstrak

```
public abstract class Unggas  
{  
  
}
```

## Method Abstract

- Method abstrak adalah method yang memiliki definisi namun tidak memiliki implementasi
- Method di dalam abstract class boleh berupa method abstrak ataupun bukan.
- Untuk membuat abstract method, hanya menuliskan deklarasi method tanpa body dan gunakan keyword abstract

# Contoh Program ABSTRACT

```
public abstract class bangun {
    String warna ;
    void setWarna (String warna) {
        this.warna=warna ;
    }
    String getWarna () {
        return this.warna ;
    }
    abstract float getArea ();
}
```

saved

```
public class segitiga extends bangun {
    private float dasar ;
    private float tinggi ;

    public segitiga (int dasar,int tinggi){
        this.dasar = dasar ;
        this.tinggi = tinggi ;
    }
    float getArea () {
        return 0.5f * dasar * tinggi ;
    }
}
```

```
public class lingkaran extends bangun {
    private float radius ;

    public lingkaran (float radius) {
        this.radius = radius ;
    }

    float getArea () {
        return (float) (Math.PI * radius * radius);
    }
}
```

```
public class panggilabstract {
    public static void main (String [] xx) {
        bangun s1 = new segitiga (4,5 );
        bangun s2 = new lingkaran (10) ;

        System.out.println ("Luas Segitiga " + s1.getArea ());
        System.out.println ("Luas Lingkaran " + s2.getArea ());
    }
}
```

# LATIHAN

1. Buatlah class dengan konsep PBO (analisis dari tiap materi) dan abstract, terdiri dari class bernama IdentitasKaryawan1 dan class method main bernama class PanggilKaryawan.

Adapun ketentuan dari class IdentitasKaryawan1 sebagai berikut :

- Ada 5 atribut (no\_karyawan, nama\_karyawan, jabatan, golongan, tunjangan\_golongan, gaji\_pokok)
- Ada 1 method untuk total gaji karyawan (gaji\_pokok + tunjangan\_golongan)
- Ada 1 method untuk mencetak data karyawan
- Adapun persyaratan tunjangan tetap, Golongan A = Rp 500.000, Golongan B = Rp. 250.000, selain itu tunjangan yang di dapat Rp. 100.000
- Adapun persyaratan tunjangan, honorer Golongan A = Rp 300.000, Golongan B = Rp. 200.000, selain itu tunjangan tidak dapat
- Buat 1 class abstract untuk mendeklarasikan method abstract yang di panggil di IdentitasKaryawan
- Buat class IdentitasKaryawan2 merupakan turunan dari class abstract untuk mendeklarasikan 1 method total gaji karyawan lembur (total gaji karyawan + lembur ) untuk Karyawan Tetap
- Buat class IdentitasKaryawan3 merupakan turunan dari class abstract untuk mendeklarasikan 1 method total gaji karyawan lembur (total gaji karyawan + lembur) untuk Karyawan Honorer