

Praktikum 3: Overloading Constructor

```
class Mobil {
    String merek;
    String warna;
    String tahun;

    public Mobil(String merek) {
        this.merek = merek;
    }

    public Mobil(int tahun) {
        this.tahun = tahun;
    }

    public Mobil(String merek, String warna) {
        this.merek = merek;
        this.warna = warna;
    }

    public Mobil(String merek, int tahun) {
        this.merek = merek;
        this.tahun = tahun;
    }

    public Mobil(int tahun, String merek) {
        this.merek = merek;
        this.warna = warna;
    }

    public Mobil(String merek, String warna, int tahun) {
        this.merek = merek;
        this.warna = warna;
        this.tahun = tahun;
    }

    public Mobil(String merek, int tahun, String warna) {
        this.merek = merek;
        this.tahun = tahun;
        this.warna = warna;
    }

    public Mobil(int tahun, String merek, String warna) {
        this.tahun = tahun;
        this.merek = merek;
        this.warna = warna;
    }
}
```

```

        void lihatMerek() {
            System.out.println(merek);
        }

        void lihatWarna() {
            System.out.println(warna);
        }

        void lihatTahun() {
            System.out.println(tahun);
        }
    }

    class TestMobil {
        public static void main(String[] args) {
            Mobil mbl01 = new Mobil("Honda", "hitam");
            mbl01.lihatMerek();
            mbl01.lihatWarna();

            Mobil mbl02 = new Mobil("Suzuki", "merah", 2020);
            mbl02.lihatMerek();
            mbl02.lihatWarna();
            mbl02.lihatTahun();

            // meskipun memiliki jumlah parameter yang sama, tetapi
            // constructor ini memiliki signature yang berbeda dengan
            // constructor di atasnya, yaitu (int, String, String)
            // sedangkan constructor di atas memiliki
            // signature (String, String, int)
            Mobil mbl03 = new Mobil(2019, "Mitsubishi", "putih");
            mbl03.lihatMerek();
            mbl03.lihatWarna();
            mbl03.lihatTahun();
        }
    }
}

```

Seluruh constructor pada class Mobil di atas adalah overloading karena memiliki signature yang berbeda.

Di bawah ini adalah contoh class yang constructor-nya bukan merupakan overloading constructor karena setiap constructor-nya memiliki signature yang sama, yaitu (String, String).

```
class Mobil {  
    String merek;  
    String warna;  
  
    public Mobil(String merek, String warna) {  
        this.merek = merek;  
        this.warna = warna;  
    }  
  
    public Mobil(String warna, String merek) {  
        this.warna = warna;  
        this.merek = merek;  
    }  
}
```