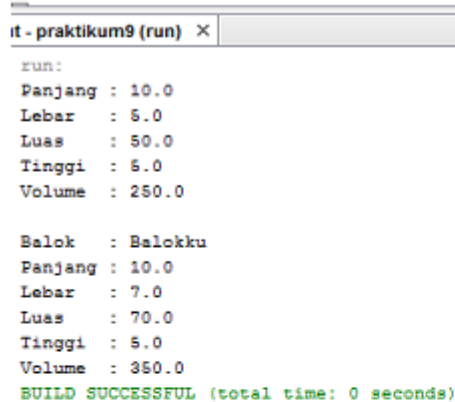


Praktikum 8 : Polymorphism

Contoh :

Hasil Program :



```
Run:
Panjang : 10.0
Lebar   : 5.0
Luas    : 50.0
Tinggi  : 5.0
Volume  : 250.0

Balok   : Balokku
Panjang : 10.0
Lebar   : 7.0
Luas    : 70.0
Tinggi  : 5.0
Volume  : 350.0
BUILD SUCCESSFUL (total time: 0 seconds)
```

Code Program :

class PersegiPanjang.java:

```
public class PersegiPanjang {
    double p,l,luas;
    void hitungLuas(){
        luas=p*l;
    }
    void cetak(){
        System.out.println("Panjang\t: "+p);
        System.out.println("Lebar\t: "+l);
        System.out.println("Luas\t: "+luas);
    }
}
```

class Balok.java:

```
public class Balok extends PersegiPanjang{
    double t,volume;
    void hitungVolume(){
        volume=luas*t;
    }
    void cetak(){
        super.cetak();
        System.out.println("Tinggi\t: "+t);
    }
}
```

```

        System.out.println("Volume\t: "+volume);
    }
    void cetak(String nama){
        System.out.println("Balok\t: "+nama);
        cetak();
    }
}

```

class BalokDemo.java:

```

public class BalokDemo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Balok a=new Balok();
        a.p=10;
        a.l=5;
        a.t=5;
        a.hitungLuas();
        a.hitungVolume();
        a.cetak();
        System.out.println("");
        a.l=7;
        a.hitungLuas();
        a.hitungVolume();
        a.cetak("Balokku");
    }

}

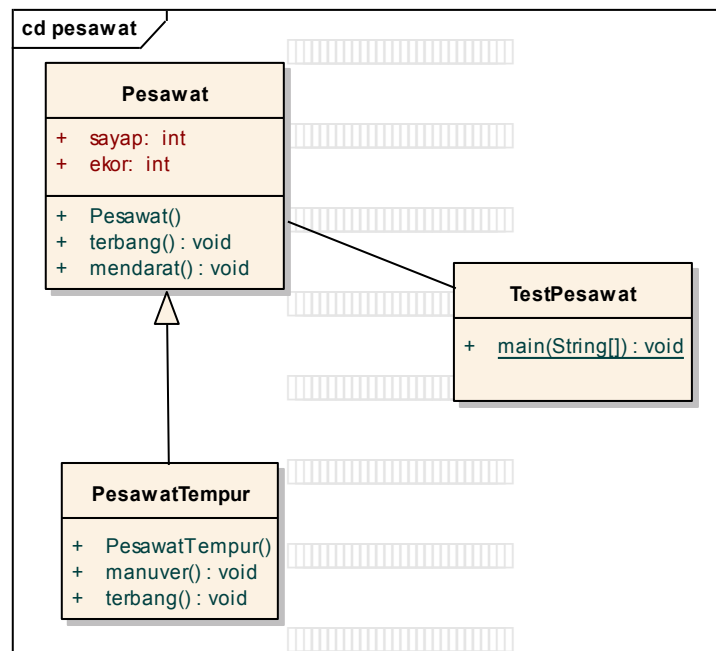
```

Latihan 1 :

1. Kembangkan class Matematika dan MatematikaDemo
2. Lakukan overloading pada Method yang ada (pertambahan, pengurangan, perkalian, pembagian) . Method baru adalah bertipe data double (pecahan) dan memiliki 3 parameter
3. Uji di kelas MatematikaDemo dengan parameter pecahan: 12.5, 28.7, 14.2
4. Misalnya:
pertambahan(12.5, 28.7, 14.2)

Latihan 2 :

1. Buat class Pesawat dengan atribut : **sayap**, **ekor**, method : **Pesawat()**, **terbang()**, **mendarat()**.
2. Buat class PesawatTempur turunan dari Pesawat dengan method : **PesawatTempur()**, **manuver()**, **terbang()**.
3. Class diagram :



4. Panggil dari class **TestPesawat** dengan hasil sebagai berikut :

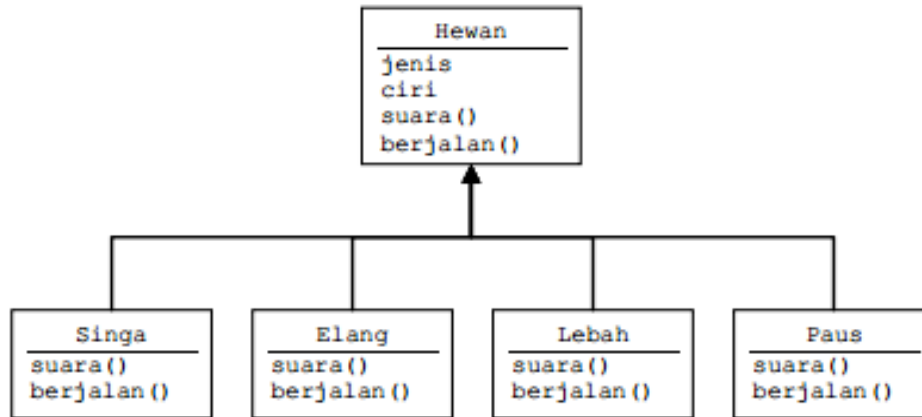
```
Command Prompt
D:\JavaMagelang>java TestPesawat
object Pesawat dibuat.....
object Pesawat dibuat.....
object pesawat tempur dibuat.....
...airbus...
terbang.....
mendarat.....
...f16...
terbang.....
terbang ala tempur.....
mendarat.....
manuver.....
```

The screenshot shows a Windows Command Prompt window titled 'Command Prompt'. The command 'java TestPesawat' has been executed. The output shows the creation of objects for 'Pesawat' and 'pesawat tempur', followed by a series of method calls and their outputs: '...airbus...', 'terbang...', 'mendarat...', '...f16...', 'terbang...', 'terbang ala tempur...', 'mendarat...', and 'manuver...'.

Overriding:

Latihan

- 1 Terdapat class-class hewan sebagai berikut :



Penjelasan dari masing-masing atribut dan method sebagai berikut :

jenis : apakah termasuk hewan mamalia, serangga atau burung

ciri : ciri dari hewan tersebut

suara() : suara hewan tersebut

berjalan() : cara hewan tersebut berjalan/bergerak

Buatlah class-class yang mengimplimentasikan gambar diatas !

- 2 Buatlah class TestHewan yang akan mengimplementasikan class-class di atas dengan ketentuan sebagai berikut :
 - Saat pertama kali dijalankan akan ditampilkan menu pilihan, yaitu nama-nama hewan tersebut serta menu untuk keluar dari program.
 - Setelah user memilih hewan yang diinginkan, maka tampilkan suara dan cara berjalan/bergerak dari hewan yang dipilih.
 - Jika user memilih Keluar, maka program selesai.
- 3 Setelah program anda berjalan dengan benar, cobalah untuk menambahi cara masing-masing hewan tersebut bernafas, yaitu apakah menggunakan paru-paru, insang, atau yang lainnya. Sehingga pada saat class TestHewan dijalankan akan menampilkan suara, cara berjalan/bergerak dan cara bernafas dari hewan yang dipilih. Buat method ini dengan teknik polymorphism. Buatlah kesimpulan sendiri tentang polymorphism.

Latihan 3 :

Implementasikan Class Diagram di bawah dengan menggunakan konsep overloading dan overriding !

