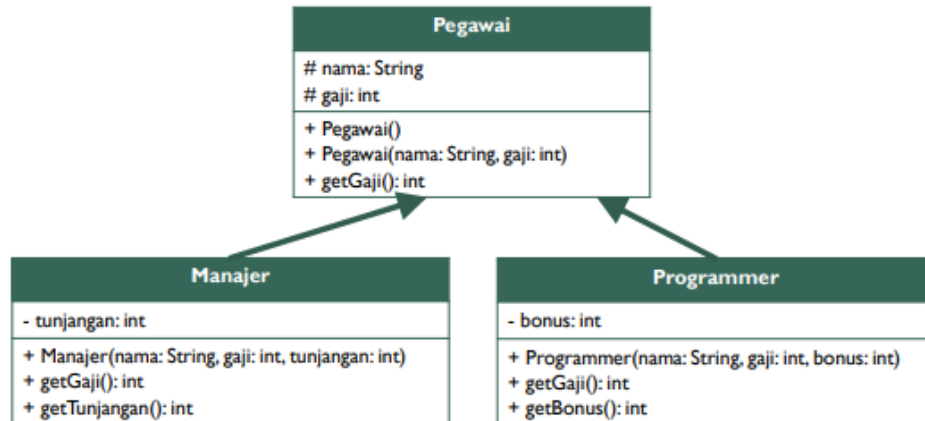




NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : PEMROGRAMAN BERBASIS OBJEK (TEORI)

EXERCISE 1

Buatlah program dari class diagram dibawah:



KODE PROGRAM :

```
public class Pegawai {
    protected String nama;
    protected int gaji;

    public Pegawai() {
    }

    public Pegawai(String nama, int gaji) {
        this.nama = nama;
        this.gaji = gaji;
    }

    public int getGaji() {
        return gaji;
    }
}
```

```
public class Manajer extends Pegawai {
    private int tunjangan;

    public Manajer(int tunjangan) {
        this.tunjangan = tunjangan;
    }

    public Manajer(String nama, int gaji, int tunjangan) {
        super(nama, gaji);
        this.tunjangan = tunjangan;
    }

    @Override
    public int getGaji() {
        return gaji;
    }

    public int getTunjangan() {
        return tunjangan;
    }
}
```

```
public class Programmer extends Pegawai {
    private int bonus;

    public Programmer(int bonus) {
        this.bonus = bonus;
    }

    public Programmer(String nama, int gaji, int bonus) {
        super(nama, gaji);
        this.bonus = bonus;
    }

    @Override
    public int getGaji() {
        return gaji;
    }

    public int getBonus() {
        return bonus;
    }
}
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : PEMROGRAMAN BERBASIS OBJEK (TEORI)

Kemudian buat class Bayaran dan TestBayaran dibawah ini untuk pengetesan:

```
12 public class Bayaran {
13     public int hitungBayaran(Pegawai pg) {
14         int uang = pg.getGaji();
15
16         if(pg instanceof Manajer){
17             uang += ((Manajer)pg).getTunjangan();
18         }
19         else if(pg instanceof Programmer){
20             uang += ((Programmer)pg).getBonus();
21         }
22         return uang;
23     }
24 }
```

```
12 public class TestBayaran {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         Manajer man = new Manajer("Agus", 800, 50);
20         Programmer prog = new Programmer("Budi", 600, 30);
21         Bayaran hr = new Bayaran();
22
23         System.out.println("Bayaran Manajer : "+hr.hitungBayaran(man));
24         System.out.println("Bayaran Programmer : "+hr.hitungBayaran(prog));
25     }
26 }
```

HASIL RUNNING PROGRAM

```
run:
Bayaran Manajer : 850
Bayaran Programmer : 600
BUILD SUCCESSFUL (total time: 0 seconds)
```

PENJELASAN :

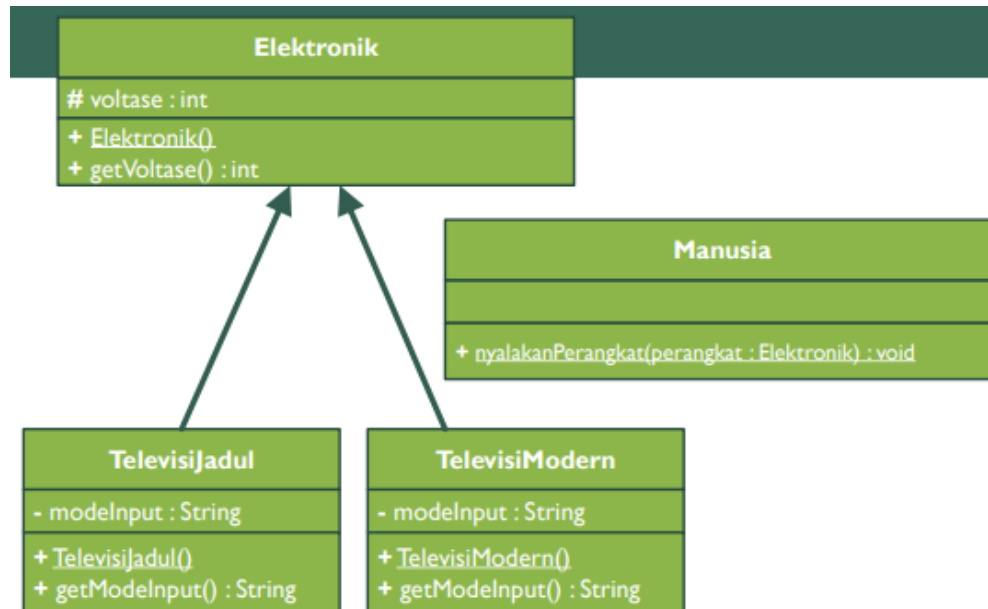
- Jadi, pada kode program ini terjadi inheritance antara class Pegawai dengan Manajer dan Pegawai dengan Programmer. Class Pegawai menjadi parent class, dan class Manajer dan class Programmer menjadi subclass / childclass.
- Pada class Manajer dan class Programmer terdapat method OverRiding (karena nama method, parameter, jenis pengembalian yang sama diantara kedua class)
- Untuk class Bayaran terdapat pernyataan instanceof untuk mengetahui tipe asal dari suatu polymorphic arguments



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : PEMROGRAMAN BERBASIS OBJEK (TEORI)

EXERCISE 2

Buatlah program dari class diagram dibawah:



KODE PROGRAM :

```
13 public class Elektronik {
14     protected int voltase;
15
16     public Elektronik() {
17         this.voltase = 220;
18     }
19
20     public int getVoltase() {
21         return voltase;
22     }
23 }
```

```
public class TelevisiJadul extends Elektronik {
    private String modelInput;

    public TelevisiJadul() {
        this.modelInput = "DVI";
    }

    public String getModelInput() {
        return modelInput;
    }
}
```

```
public class TelevisiModern extends Elektronik {
    private String modelInput;

    public TelevisiModern() {
        this.modelInput = "HDMI";
    }

    public String getModelInput() {
        return modelInput;
    }
}
```



NAMA : ANA QONITAH MUNAWWAROH
NIM : 2041720118
KELAS : 2C / 03
MATERI : PEMROGRAMAN BERBASIS OBJEK (TEORI)

Buat program sesuai dengan class diagram diatas, berikut class Manusia :

```
public class Manusia {  
    public void nyalakanPerangkat(Elektronik perangkat) {  
  
        if(perangkat instanceof TelevisiJadul){  
            System.out.println("Nyalakan televisi jadul dengan input : "+((TelevisiJadul) perangkat).getModeInput());  
            System.out.println("Voltase televisi : "+perangkat.getVoltase());  
        }  
        else if(perangkat instanceof TelevisiModern){  
            System.out.println("Nyalakan televisi modern dengan input: "+((TelevisiModern) perangkat).getModeInput());  
            System.out.println("Voltase televisi : "+perangkat.getVoltase());  
        }  
    }  
}
```

kemudian buat class TestElektronik seperti dibawah:

```
public class TestElektronik {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        Manusia indro = new Manusia();  
        TelevisiJadul tvjadul = new TelevisiJadul();  
        TelevisiModern tvmodern = new TelevisiModern();  
  
        indro.nyalakanPerangkat(tvjadul);  
        indro.nyalakanPerangkat(tvmodern);  
    }  
}
```

HASIL RUNNING PROGRAM

```
run:  
Nyalakan televisi jadul dengan input : DVI  
Voltase televisi : 220  
Nyalakan televisi modern dengan input: HDMI  
Voltase televisi : 220  
BUILD SUCCESSFUL (total time: 0 seconds)
```

PENJELASAN :

- Pada kode program ini terjadi inheritance antara class TelevisiJadul dan TelevisiModern dengan class Elektronik. Class Elektronik menjadi Parent class dan class TelevisiJadul dan Televisi Modern menjadi childclass.
- Pada class main akan memanggil method nyalakanPerangkat. Dan method nyalakanPerangkat berada di class Manusia, dan dimethod tersebut akan mencetak Inputan televisi dan Voltase