

RELASI CLASS

1. KOMPETENSI

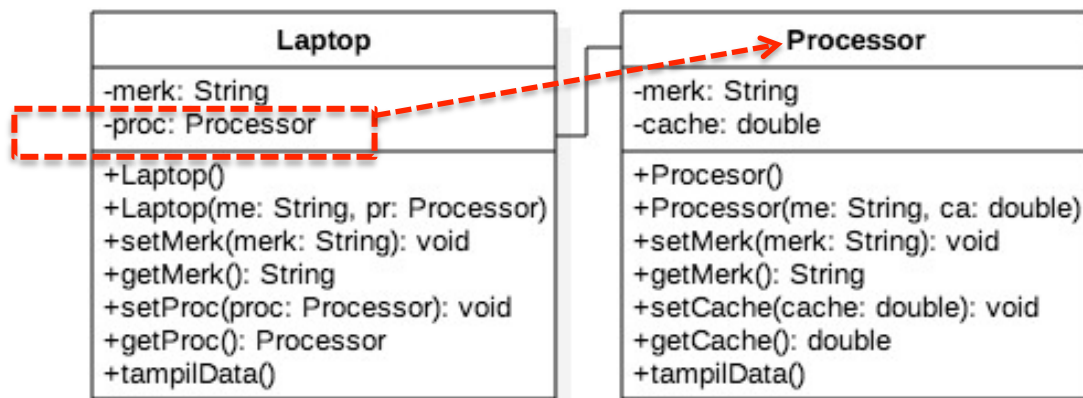
Setelah menempuh materi percobaan ini, mahasiswa mampu:

1. Memahami konsep relasi class
2. Mengimplementasikan relasi has-a dalam program

2. PENDAHULUAN

Dalam kasus yang lebih kompleks, dalam suatu sistem akan ditemukan lebih dari satu class yang saling memiliki keterkaitan antara class satu dengan yang lain. Dalam percobaan-percobaan sebelumnya, mayoritas kasus yang sudah dikerjakan hanya focus pada satu class saja. Pada jobsheet ini akan dilakukan percobaan yang melibatkan beberapa class yang saling berelasi.

Misalnya terdapat class **Laptop** yang memiliki atribut berupa **merk** dan **prosesor**. Jika diperhatikan lebih detil, maka atribut **prosesor** sendiri di dalamnya memiliki data berupa merk, nilai cache memori, dan nilai clock-nya. Artinya, ada class lain yang namanya **Processor** yang memiliki atribut **merk**, **cache** dan **clock**, dan atribut **prosesor** yang ada di dalam class **Laptop** itu merupakan objek dari class **Processor** tersebut. Sehingga terlihat antara class **Laptop** dan class **Processor** memiliki relasi memiliki (has-a).

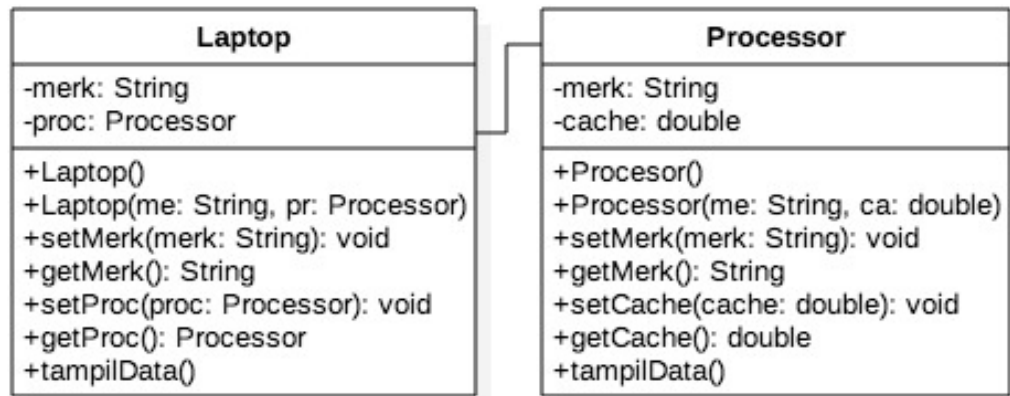


Jenis relasi has-a ini yang akan dicontohkan pada percobaan di jobsheet ini. Apabila dilihat lebih detil lagi, relasi tersebut disebut juga dengan agregasi. Terdapat banyak tipe relasi lain yaitu dependensi, komposisi, generalisasi (dibahas di minggu ke-10, materi inheritance), dan realisasi (dibahas minggu ke-12, materi interface). Diperlukan inisiatif mandiri dari tiap mahasiswa untuk memperdalam jenis relasi lain terutama yang tidak dibahas di matakuliah ini ☺

3. PERCOBAAN 1

A. TAHAP PERCOBAAN

1. Perhatikan diagram class di bawah ini:



2. Buka project di Netbeans dan buat paket **relasiclass.percobaan1**
3. Buat class **Processor** di dalam paket tersebut

```
1 package relasiclass.percobaan1;
2
3 public class Processor {
4     private String merk;
5     private double cache;
6
7     public Processor(){
8     }
9
10    public Processor(String me, double ca){
11        merk = me;
12        cache = ca;
13    }
14
15    public String getMerk() {
16        return merk;
17    }
18
19    public void setMerk(String merk) {
20        this.merk = merk;
21    }
22
23    public double getCache() {
24        return cache;
25    }
26
27    public void setCache(double cache) {
28        this.cache = cache;
29    }
30
31    public void tampilData(){
32        System.out.println("Merk Prosesor = "+merk);
33        System.out.println("Cache Memory = "+cache);
34    }
35 }
```

4. Buat class **Laptop** juga di dalam paket tersebut

```
1 package relasiclass.percobaan1;
2
3 public class Laptop {
4     private String merk;
5     private Processor proc;
6
7     public Laptop(){
8     }
9
10    public Laptop(String me, Processor pr){
11        merk = me;
12        proc = pr;
13    }
14
15    public String getMerk() {
16        return merk;
17    }
18
19    public void setMerk(String merk) {
20        this.merk = merk;
21    }
22
23    public Processor getProc() {
24        return proc;
25    }
26
27    public void setProc(Processor proc) {
28        this.proc = proc;
29    }
30
31    public void tampilData(){
32        System.out.println("Merk Laptop = "+merk);
33        proc.tampilData();
34    }
35 }
```

5. Di paket yang sama, buat klas **Main**.

```
1 package relasiclass.percobaan1;
2
3 public class Main {
4     public static void main(String[] args) {
5         //instansiasi menggunakan konstruktor berparameter
6         Processor p = new Processor("Intel Core2Duo", 3);
7         Laptop l = new Laptop("Dell Vostro", p);
8         l.tampilData();
9
10        //contoh lain instansiasi menggunakan konstruktor default
11        Processor p1 = new Processor();
12        p1.setMerk("Intel Core i3");
13        p1.setCache(4);
14        Laptop l1 = new Laptop();
15        l1.setMerk("HP Probook");
16        l1.setProc(p1);
17    }
18 }
19 }
```

6. Compile dan run class Main, sehingga akan didapatkan output seperti di bawah ini.

```
run:
Merk Laptop = Dell Vostro
Merk Prosesor = Intel Core2Duo
Cache Memory = 3.0
BUILD SUCCESSFUL (total time: 2 seconds)
```

B. PERTANYAAN

Berdasarkan percobaan 1 di atas, jawab pertanyaan-pertanyaan yang terkait, di bawah ini:

1. Di dalam class **Processor** dan class **Laptop**, terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut?
2. Di dalam class **Processor** dan class **Laptop**, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut?
3. Perhatikan class **Laptop**, diantara 2 atribut yang dimiliki (**merk** dan **proc**), atribut manakah yang *bertipe object*?
4. Perhatikan class **Laptop**, pada baris manakah yang menunjukkan bahwa class **Laptop** *memiliki relasi* dengan class **Processor**?
5. Perhatikan pada baris ke-33 class **Laptop**. Apakah guna dari sintaks **proc.tampilData()** ?
6. Perhatikan class **Main**, pada baris ke-7. Di baris tersebut terlihat dilewatkan **p** pada saat instansiasi **new Laptop ("Dell Vostro", p);**. Apakah **p** tersebut? Misalkan pada baris ke-7 tersebut diubah menjadi:

```
Laptop l = new Laptop("Dell Vostro", new
Processor("Intel Core2Duo",3));
```

bagaimanakah hasil program saat dijalankan, apakah ada perubahan?

7. Perhatikan class **Main**, jika pada baris ke-17 ditambahkan sintaks:

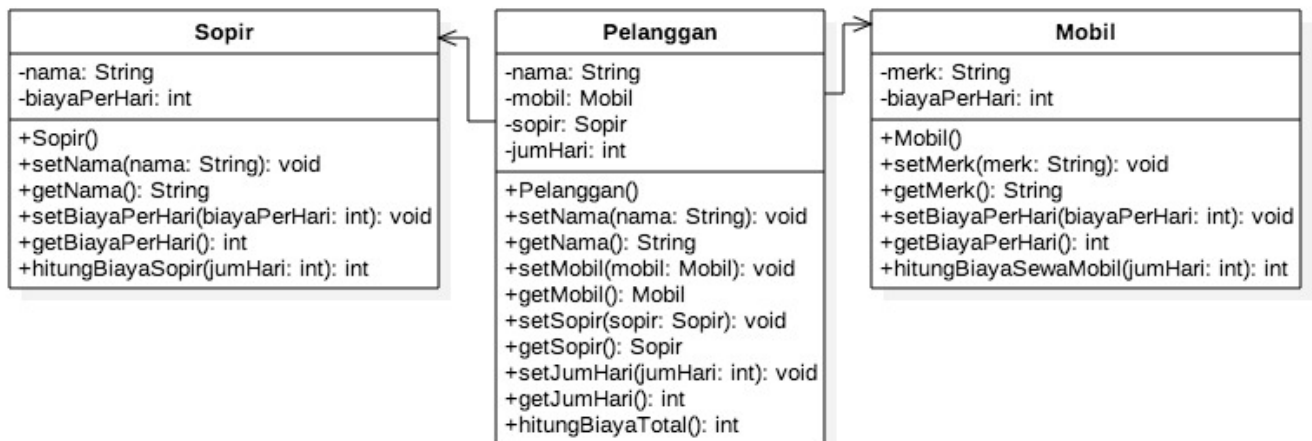
```
System.out.println("Merk          Laptop          =
"+l1.getMerk());
System.out.println("Merk          Processor          =
"+l1.getProc().getMerk());
System.out.println("Cache          Memory          =
"+l1.getProc().getCache());
```

Bagaimanakah perubahan yang terjadi saat class **Main** dijalankan? Jadi, untuk apakah sintaks **l1.getProc().getMerk()** yang ada di dalam method **main** tadi?

4. PERCOBAAN 2

A. TAHAP PERCOBAAN

1. Perhatikan diagram class di bawah ini:



Dalam sistem rental mobil, pelanggan bisa menyewa mobil sekaligus sopirnya. Biaya sopir dan biaya sewa mobil dihitung perhari.

2. Buat paket **relasiclass.percobaan2**.

3. Buat class **Mobil** di dalam paket tersebut.

```
1 package relasiclass.percobaan2;
2
3 public class Mobil {
4     private String merk;
5     private int biayaPerHari;
6
7     public Mobil() {
8     }
9     public String getMerk() {
10         return merk;
11     }
12     public void setMerk(String merk) {
13         this.merk = merk;
14     }
15     public int getBiayaPerHari() {
16         return biayaPerHari;
17     }
18     public void setBiayaPerHari(int biayaPerHari) {
19         this.biayaPerHari = biayaPerHari;
20     }
21     public int hitungBiayaSewaMobil(int jumHari){
22         return biayaPerHari*jumHari;
23     }
24 }
```

4. Buat class **Sopir** di dalam paket tersebut.

```

1  package relasiclass.percobaan2;
2
3  public class Sopir {
4      private String nama;
5      private int biayaPerHari;
6
7      public Sopir() {
8      }
9      public String getNama() {
10         return nama;
11     }
12     public void setNama(String nama) {
13         this.nama = nama;
14     }
15     public int getBiayaPerHari() {
16         return biayaPerHari;
17     }
18     public void setBiayaPerHari(int biayaPerHari) {
19         this.biayaPerHari = biayaPerHari;
20     }
21     public int hitungBiayaSopir(int jumHari){
22         return biayaPerHari*jumHari;
23     }
24 }

```

5. Buat class **Pelanggan** di dalam paket tersebut.

```

1  package relasiclass.percobaan2;
2
3  public class Pelanggan {
4      private String nama;
5      private Mobil mobil;
6      private Sopir sopir;
7      private int jumHari;
8      public Pelanggan() {
9      }
10     public String getNama() {
11         return nama;
12     }
13     public void setNama(String nama) {
14         this.nama = nama;
15     }
16     public Mobil getMobil() {
17         return mobil;
18     }
19     public void setMobil(Mobil mobil) {
20         this.mobil = mobil;
21     }
22     public Sopir getSopir() {
23         return sopir;
24     }
25     public void setSopir(Sopir sopir) {
26         this.sopir = sopir;
27     }
28     public int getJumHari() {
29         return jumHari;
30     }
31     public void setJumHari(int jumHari) {
32         this.jumHari = jumHari;
33     }
34     public int hitungBiayaTotal(){
35         return mobil.hitungBiayaSewaMobil(jumHari)+sopir.hitungBiayaSewaMobil(jumHari);
36     }
37 }

```

6. Buat class **Main** di dalam paket tersebut.

```
1 package relasiclass.percobaan2;
2
3 public class Main {
4     public static void main(String[] args) {
5         Mobil m = new Mobil();
6         m.setMerk("Avanza");
7         m.setBiayaPerHari(350000);
8         Sopir s = new Sopir();
9         s.setNama("John Doe");
10        s.setBiayaPerHari(200000);
11        Pelanggan p = new Pelanggan();
12        p.setNama("Jane Doe");
13        p.setMobil(m);
14        p.setSopir(s);
15        p.setJumHari(2);
16        System.out.println("Biaya Total = "+p.hitungBiayaTotal());
17    }
18 }
```

7. Compile dan Run class Main, dan perhatikan hasilnya!

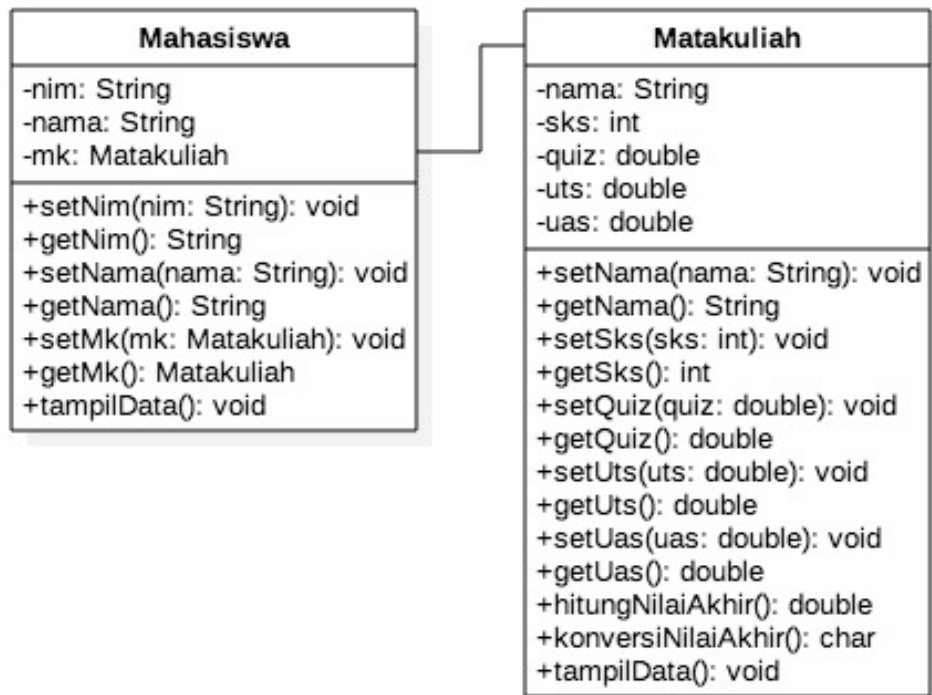
```
run:
Biaya Total = 1100000
BUILD SUCCESSFUL (total time: 1 second)
```

B. PERTANYAAN

1. Perhatikan class **Pelanggan**. Pada baris program manakan yang menunjukkan bahwa class **Pelanggan** memiliki relasi dengan class **Mobil** dan class **Sopir**?
2. Perhatikan method **hitungBiayaSopir** pada class **Sopir**, serta method **hitungBiayaSewaMobil** pada class **Mobil**. Mengapa menurut Anda method tersebut harus memiliki argument **jumHari**?
3. Perhatikan baris program ke-35 dari class **Pelanggan**. Untuk apakah perintah **mobil.hitungBiayaSewaMobil(jumHari)** dan **sopir.hitungBiayaSopir(jumHari)**?
4. Perhatikan class **Main** pada baris ke-13 dan ke-14. Untuk apakah sintaks **p.setMobil(m)** dan **p.setSopir(s)**?
5. Perhatikan class **Main** pada baris ke-16. Untuk apakah proses baris baris tersebut?
6. Perhatikan class **Main**, coba tambahkan pada baris ke-17 sintaks di bawah ini, dan amati perubahan saat di-run!
System.out.println(p.getMobil().getMerk());
Jadi untuk apakah sintaks **p.getMobil().getMerk()** yang ada di dalam method **main** tersebut?

5. LATIHAN

1. Perhatikan diagram class di bawah ini!



Buat programnya sehingga akan didapatkan output:

NIM : 1213000045
Nama : John Doe
Nama Matakuliah : PBO
Nilai Quiz : 67
Nilai UTS : 75
Nilai UAS : 85
Nilai Akhir : 77
Nilai Huruf : B+

Aturan perhitungan nilai akhir = 25% Quiz + 35% UTS + 40% UAS

Aturan konversi nilai:

Nilai akhir > 80 → A

75 < Nilai akhir <= 80 → B+

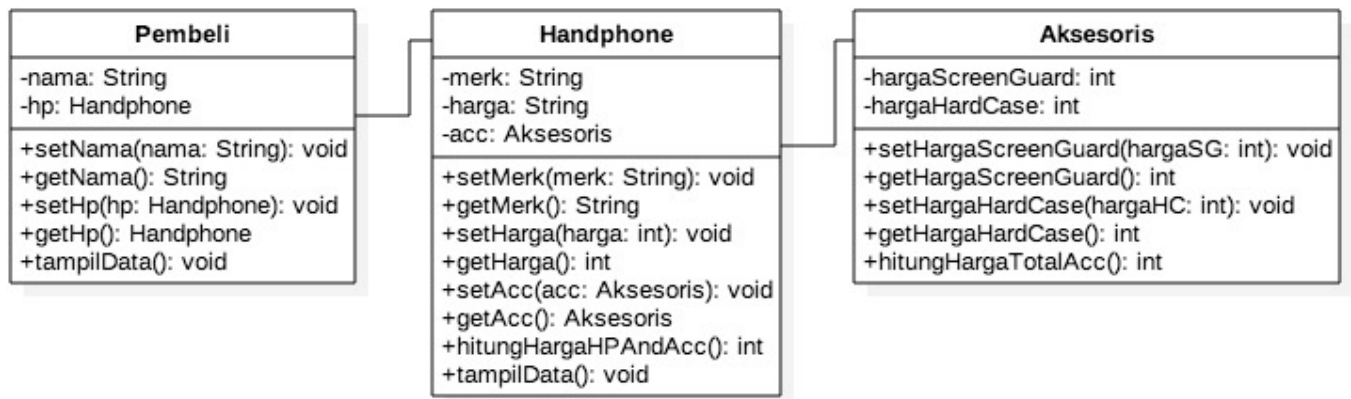
65 < Nilai akhir <= 75 → B

60 < Nilai akhir <= 65 → C

50 < Nilai akhir <= 60 → D

Nilai akhir <= 50 → E

2. Perhatikan diagram class di bawah ini!



Buat program sehingga akan didapatkan tampilan output:

Nama Pembeli : John Doe

Data Handphone yang dibeli:

Handphone : Samsung J7 Prime

Harga : 3400

Dengan harga aksesoris:

Harga Screen Guard : 60

Harga Hard Case : 80

#####

Harga Total : 3540

Catatan:

- Method `hitungHargaHPAndAcc()` akan menjumlahkan antara harga HP dan harga total aksesoris
- Method `hitungHargaTotalAcc()` akan menjumlahkan antara harga hard case dan harga screen guard