RELASI CLASS

1. KOMPETENSI

Setelah menempuh materi percobaan ini, mahasiswa mampu:

1. Memahami konsep relasi class
2. Mengimplementasikan relasi has-a dalam program
3. 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 Proceessor tersebut. Sehingga terlihat antara class Laptop dan class Processor memiliki relasi memiliki (has-a).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Laptop |  |  | ^Processor |
|  | -merk: Strinq r |  |  | -merk: String |
| 1  1 | -proc: Processor |  |  | -cache: double |
|  | +LaptopO |  | | -t-Procesor() |
|  | +Laptop(me: String, pr: Processor) |  | | +Processor(me: String, ca: double) |
|  | +setMerk(merk: String): void |  | | +setMerk(merk: String): void |
|  | +getMerkO: String |  | | +getMerkO: String |
|  | +setProc(proc: Processor): void |  | | +setCache(cache: double): void |
|  | +getProcO: Processor |  | | ♦getCacheO: double |
|  | +tampilDataQ |  | | +tampilDataQ |

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 benyak 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.

3.

Laptop

-merk: String -proc: Processor

+LaptopO

+Laptop(me: String, pr: Processor) +setMerk(merk: String): void +getMerkO: String +setProc(proc: Processor): void +getProcO: Processor +tampilData()

Processor

-merk: String -cache: double

♦ProcesorO

+Processor(me: String, ca: double) +setMerk(merk: String): void ♦getMerkO: String +setCache(cache: double): void +getCacheO: double ♦tampilDataO

Buka project di Netbeans dan buat paket relasiclass.percobaanl Buat class Processor di dalam paket tersebut

1

2

**3**

**4**

**5**

6

**7**

1. **9**

**10**

11

12

13

**14**

**15**

1. **17 18**

**19**

1. 21 22

**23**

**24**

**25**

**26**

**27**

**28**

**29**

**30**

**31**

**32**

**33**

**34**

**35**

□

*B*

*B*

El

**package relasiclass.percobaanl;**

**public class Processor { private String merk; private| double cache;**

**public Processor(){**

>

**public Processor(String me, double ca){**

**merk = me; cache = ca;**

>

**public String getMerkO { return merk;**

>

**public void setMerk(String merk) { this.merk = merk;**

>

**public double getCacheO { return cache;**

>

**public void setCache(double cache) { this.cache = cache;**

>

**public void tampilData(){**

**System.out.println("Merk Prosesor = "+merk); System.out.printlnC'Cache Memory = "+cache);**

**package relasiclass.percobaanl;**

5.

**5**

6

1. $
2. 9

**10** **0 11** 12

**13**

**14**

1. **13**

**16**

**17**

**18**

**19**

1. 21 22
2. **13**

**24**

**25**

**26**

**27**

**28**

**29**

**30**

1. **13** 12..

**33**

**"34“**

**35**

1

2

**3**

1. **□**

**5**

JiL.

**7**

“8

**9**

10

11

**12**

**13**

**14**

**15**

**16**

**17**

1. **- 19**

**def au**

**public class** Laptop **{ private String merk; private Processor proc;**

**public Laptop!){**

}

**public Laptop(String ms, Processor pr){**

**merk = me; proc = pr;**

>

**public String getMerkO { return merk;**

>

**public void setMerk(String merk) { this.merk = merk;**

>

**public Processor getProcO { return proc;**

>

**public void setProc(Processor proc) { this.proc = proc;**

>

**public void tampilData!){**

Syste\*.oi/t^prlrttLni,,MerkJLaptQp="+fl«rk);

**proc.tampilData!);**

*J*

>

Di paket yang sama, buat klas Main.

**package relasiclass.percobaanl; public class** Main **{**

**public static void main(String[] args) {**

**//instansiasi menggunakan konstruktor berparameter**

JP\_rjD\_c£\_s5Qt\_\_p\_\_=\_jdew\_\_Pj:Q££ss\_arX,\_'Jn\_t£\_l\_\_CQr\_e.2D\_uo\_,'+\_\_lJ!;

**Laptop l = new LaptopC'Dell Vostro", p); [-.“TampiTITatTO"; J**

**//contoh lain instansiasi menggunakan konstruktor Processor pi** = **new Processor!); pl.setMerkC'Intel Core i3"); pi.setCache(4);**

**Laptop 11 = new Laptop!);**

**H.setMerkC'HP Probook"); ll.setProc(pl);**

U|

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 atributyang dimiliki (merk dan proc), atribut manakah yang bertipe object?
4. Perhatikan class Laptop, pada baris manakah yang menunjukan 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 1 = new Laptop("Dell Vostro", new

Processor("Intel Core2Duo",3));

bagaimanakah hasil program saat dijalankan, apakah ada perubahan?

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

System.out.println("Merk Laptop =

"+11.getMerk());

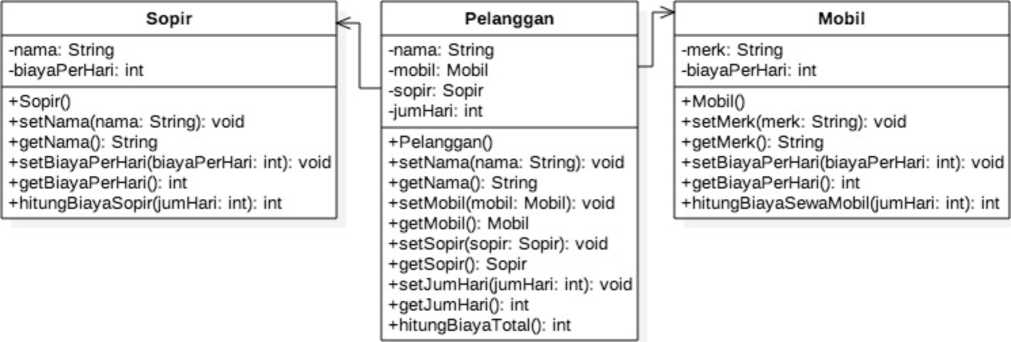
System.out.println("Merk Processor =

"+11.getProc().getMerk());

System.out.println("Cache Memory =

"+11.getProc().getCache());

Bagaimanakah perubahan yangterjadi saat class Main dijalankan? Jadi, untuk apakah sintaks 11 .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

2.

3.

sopirnya. Biaya sopir dan biaya sewa mobil dihitung perhari. Buat paket relasiclass.percobaan2.

Buat class Mobil di dalam paket tersebut.

1

2

**3**

**4**

**5**

6

image2

**9 0** 10

11 L 12 0

**13**

1. **-**
2. **0**

**16**

1. **L**
2. **0**

**19**

1. -

2~r~0

22

**23 -**

**2“4**

**package relasiclass.percobaan2;**

**public class Mobil {**

**private String merk; private int biayaPerHari;**

**public Mobil() {**

>

**public String getMerkO { return merk;**

>

**public void setMerk(String merk) { this.merk = merk;**

>

**public int getBiayaPerHari!) {**

**return biayaPerHari;**

>

**public void setBiayaPerHarilint biayaPerHari) this.biayaPerHari = biayaPerHari;**

# >

**public int hitungBiayaSewaMobil!int jumHariR**

**return biayaPerHari\*]umHari;**

>

{

4. Buat class Sopir di dalam paket tersebut.

**package relasiclass.percobaan2;**

**public class Sopir {**

1

2

**3**

**4**

**5**

6

**7**

1. 9

10

11

**12**

**13**

**14**

**15**

**16**

**17**

**18**

**19**

1. 21 22

**23**

**24**

**private String nama; private int biayaPerHari;**

**1^ public SopirO {**

**E public String getNama() {**

**return nama;**

L >

**B public void setNama(String nama) { this.nama = nama;**

L >

**E public int getBiayaPerHari{) {**

**return biayaPerHari;**

L >

**E public void setBiayaPerHari{int biayaPerHari)**

{

**this.biayaPerHari = biayaPerHari;**

**B public int** hitungBiayaSopir**(int jumHariK**

**return biayaPerHari\*jumHari;**

L >

}'

5. Buat class Pelanggan di dalam paket tersebut.

**package relasiclass.percobaan2;**

1

2

3

4

5

6

7

1. **T**
2. L

9 f n 12

1. El

14

1. L
2. El

17

1. -
2. El

20

21 -

1. El

23

1. L
2. El

26

1. -
2. El

29

1. -
2. El

**32**

1. **L**
2. **El**

**35**

36""-

37

**public class Pelanggan { private String nama; private Mobil mobil; private Sopir sopir; private int jumHari; public PelangganO {**

>

**public Spring getNamaU { return nama;**

>

**public void setNama(String nama) { this.nama = nama;**

>

**public Mobil getMobiK) { return mobil;**

>

**public void setMobil(Mobil mobil) { this.mobil = mobil;**

>

**public Sopir getSopirU { return sopir;**

>

**public void setSopir(Sopir sopir) { this.sopir = sopir;**

>

**public int getJumHariO { return jumHari;**

>

**public void setJumHari(int jumHari) { this.jumHari = jumHari;**

>

**p\_ubl\_ic\_\_int\_hitungBiayaTqta\(){**

**return mobil.hitungBiayaSewaMobil(jumHari)+sopir.hitungBiayaSewaMobil(jumHari);**

>

1

2

**3**

1. **E**

**5**

6

**7**

1. 9

10

11

12

**13**

**14**

**15**

**16**

1. **-**

**18**

**package relasiclass.percobaan2; public class Main {**

**public static void main(String[) args) {**

**Mobil m = new MobilO; m.setMerkC'Avanza"); m.setBiayaPerHari(350000);**

**Sopir s = new SopirO; s.setNamaC'John Doe"); s.setBiayaPerHari(200000);**

**Pelanggan p = new PelangganO; p.setNamal"Jane Doe"); p.setMobil(m); p.setSopir(s); p.setJumHari(2);**

**System.out.printlnC'Biaya Total = "+p.hitungBiayaTotal());**

>

}

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 menunjukan 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) danp. 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 () .getMerkO) ;

Jadi untuk apakah sintaks p. getMobil () . getMerk () yang ada di dalam method main tersebut?

5. LATIHAN

1. Perhatikan diagram class di bawah ini!

|  |  |  |  |
| --- | --- | --- | --- |
| Mahasiswa |  |  | Matakuliah |
| -nim: String -nama: String -mk: Matakuliah |  |  | -nama: String -sks: int -quiz: double -uts: double -uas: double |
| +setNim(nim: String): void +getNim(): String +setNama(nama: String): void +getNamaO: String +setMk(mk: Matakuliah): void +getMk(): Matakuliah +tampilData(): void |  | |
|  | | +setNama(nama: String): void +getNamaO: String +setSks(sks: int): void +getSksO: int  +setQuiz(quiz: double): void |
|  |  | | +getQuiz(): double +setUts(uts: double): void +getUtsO: double +setUas(uas: double): void +getUasO: double +hitungNilaiAkhir0: double +konversiNilaiAkhir(): char +tampilData(): void |

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:

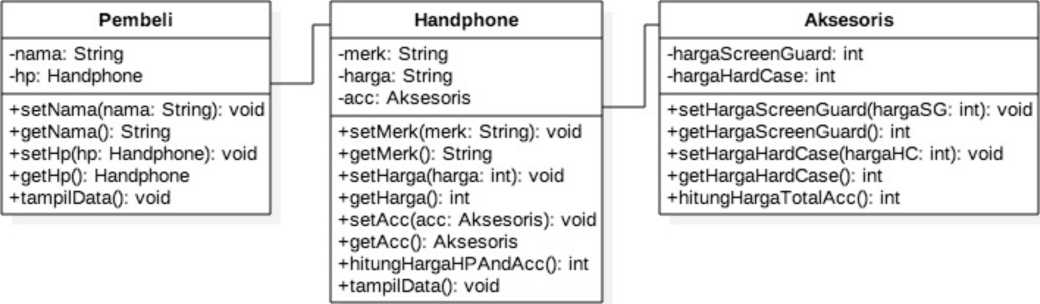
Nilai akhir > 80 -> A

75 < Nilai akhir <= 80 -> B+

65 < Nilai akhir <= 75 -> B 60 < Nilai akhir <= 65 -> C 50 < Nilai akhir <= 60 -> D Nilai akhir <= 50 -> E

Buat program sehingga akan didapatkan tampilan output: Nama Pembeli : John Doe

2. Perhatikan diagram class di bawah ini!



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