

Percobaan 1

Pertanyaan:

1. Di dalam class Processor dan class Laptop , terdapat method setter dan getter untuk masing-masing atributnya. Apakah gunanya method setter dan getter tersebut ?

Jawab:

Agar merubah nilai atribut dari class Processor dan Laptop ketika tidak memasukkan nilai saat menginstansiasi class-class-nya dan agar dapat mengambil value dari atributnya ketika dibutuhkan.

2. Di dalam class Processor dan class Laptop, masing-masing terdapat konstruktor default dan konstruktor berparameter. Bagaimanakah beda penggunaan dari kedua jenis konstruktor tersebut ?

Jawab:

konstruktor default menjadikan atribut didalamnya tidak mempunyai value agar dapat memberi value-nya nanti, sedangkan konstruktor berparameter langsung memberikan value kepada atributnya dan tidak perlu di-set value-nya lagi.

3. Perhatikan class Laptop, di antara 2 atribut yang dimiliki (merk dan proc), atribut manakah yang bertipe object ?

Jawab:

proc, karena ketika mendeklarasikan tipe data-nya menggunakan nama class.

4. Perhatikan class Laptop, pada baris manakah yang menunjukan bahwa class Laptop memiliki relasi dengan class Processor?

Jawab:

- private Processor proc;
- method setProc() dan getProc()
- proc.info;
- 5. Perhatikan pada class Laptop , Apakah guna dari sintaks proc.info() ?

Jawab:

menjalankan method info() di dalam class Processor yaitu print merk dan cache memory.

6. Pada class MainPercobaan1, terdapat baris kode:

Laptop I = new Laptop("Thinkpad", p);.

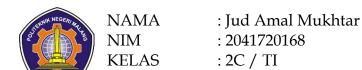
Apakah p tersebut?

Dan apakah yang terjadi jika baris kode tersebut diubah menjadi:

Laptop I = new Laptop("Thinkpad", new Processor("Intel i5",3));

Bagaimanakah hasil program saat dijalankan, apakah ada perubahan?

Jawab:



MATERI

: Relasi Kelas

p adalah instansiasi dari objek Processor; tidak ada perubahan, karena sama saja memasukkan objek Processor ke dalam class Laptop.

Percobaan 2

Pertanyaan:

 Perhatikan class Pelanggan. Pada baris program manakah yang menunjukan bahwa class Pelanggan memiliki relasi dengan class Mobil dan class Sopir ?

Jawab:

- setter dan getter class Sopir
- setter dan getter class Mobil
- method hitungBiayaMobil dan hitungBiayaSopir dalam method hitungBiayaTotal
- 2. Perhatikan method hitungBiayaSopir pada class Sopir, serta method hitungBiayaMobil pada class Mobil. Mengapa menurut Anda method tersebut harus memiliki argument hari?

Jawab:

karena pembayaran rental mobil dan sewa sopir dihitung per hari.

Perhatikan kode dari class Pelanggan. Untuk apakah perintah mobil.hitungBiayaMobil(hari) dan sopir.hitungBiayaSopir(hari) ?

Jawab:

menghitung harga sewa mobil dan sopir dikalikan hari.

4. Perhatikan class MainPercobaan2. Untuk apakah sintaks p.setMobil(m) dan p.setSopir(s) ?

Jawab :

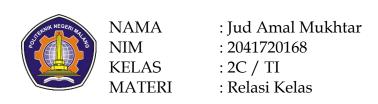
menjadikan objek m dan s, mobil dan sopir dalam class Pelanggan p.

Perhatikan class MainPercobaan2. Untuk apakah proses p.hitungBiayaTotal() tersebut ?

Jawab:

menghitung total biaya sewa mobil dan sopir selama 2 hari yang telah dideklarasikan sebelumnya.

6. Perhatikan class MainPercobaan2, coba tambahkan pada baris terakhir dari method main 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?



Jawab:

untuk mengambil value dari merk dalam class Mobil.

Percobaan 3

Pertanyaan:

 Di dalam method info() pada class KeretaApi, baris this.masinis.info() dan this.asisten.info() digunakan untuk apa ?

Jawab:

menambahkan method info() dari class Pegawai dari objek masinis dan asisten.

2. Buatlah main program baru dengan nama class MainPertanyaan pada package yang sama. Tambahkan kode berikut pada method main()!

Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants"); KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis); System.out.println(keretaApi.info());

Jawab:

```
1  ...5 lines

package percobaan3;

8  /**...4 lines */
public class MainPertanyaan {

public static void main(String[] args) {
    Pegawai masinis = new Pegawai("1234", "Spongebob Squarepants");
    KeretaApi keretaApi = new KeretaApi("Gaya Baru", "Bisnis", masinis);
    System.out.println(keretaApi.info());
}

19 }
```

3. Apa hasil output dari main program tersebut ? Mengapa hal tersebut dapat terjadi ?

Jawab:

```
run:

Exception in thread "main" java.lang.NullPointerException

at percobaan3.KeretaApi.info(KeretaApi.java:67)

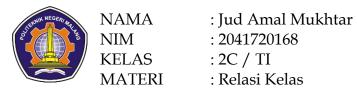
at percobaan3.MainPertanyaan.main(MainPertanyaan.java:17)

C:\Users\Asus-PC\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1

BUILD FAILED (total time: 10 seconds)
```

karena pada info() terdapat asisten.info() yang padahal belum dideklarasikan

4. Perbaiki class KeretaApi sehingga program dapat berjalan ! Jawab :



```
public String info() {
    String info = "";
    info += "Nama: " + this.nama + "\n";
    info += "Kelas: " + this.kelas + "\n";
    info += "Masinis: " + this.masinis.info() + "\n";
    //info += "Asisten: " + this.asisten.info() + "\n";
    return info;
}

run:
Nama: Gaya Baru
Kelas: Bisnis
Masinis: Nip: 1234
Nama: Spongebob Squarepants

BUILD SUCCESSFUL (total time: 1 second)
```

Percobaan 4

Pertanyaan:

 Pada main program dalam class MainPercobaan4, berapakah jumlah kursi dalam Gerbong A ?

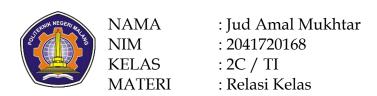
Jawab : 10 kursi.

2. Perhatikan potongan kode pada method info() dalam class Kursi. Apa maksud kode tersebut ?

```
...
if (this.penumpang != null) {
  info += "Penumpang: " + penumpang.info() + "\n";
}
...
```

Jawab:

jika objek penumpang ber-value null atau tidak ada isinya maka if clause tidak dijalankan



3. Mengapa pada method setPenumpang() dalam class Gerbong, nilai nomor dikurangi dengan angka 1?

Jawab:

karena mengisi mengikuti nilai urutan array.

4. Instansiasi objek baru budi dengan tipe Penumpang, kemudian masukkan objek baru tersebut pada gerbong dengan gerbong.setPenumpang(budi, 1). Apakah yang terjadi ?

Jawab:

objek p terganti dengan objek budi

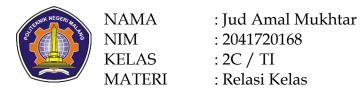
5. Modifikasi program sehingga tidak diperkenankan untuk menduduki kursi yang sudah ada penumpang lain!
Jawab:

Main

```
public class MainPercobaan4 {
   public static void main(String[] args) {
        Penumpang p = new Penumpang("12345", "Mr. Krab");
        Penumpang budi = new Penumpang("67890", "Budi");
        Gerbong gerbong = new Gerbong("A", 10);
        gerbong.setPenumpang(p, 1);
        System.out.println(gerbong.info());
        gerbong.setPenumpang(budi, 1);
        System.out.println(gerbong.info());
    }
}
```

Gerbong

```
public void setPenumpang(Penumpang penumpang, int nomor) {
    if (this.arrayKursi[nomor - 1].getPenumpang() == nutt) {
        this.arrayKursi[nomor - 1].setPenumpang(penumpang);
    }etse{
        System.out.println("maaf kursi telah terisi");
    }
}
```



Hasil

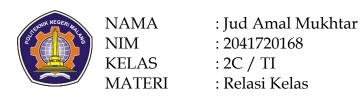
```
Nomor: 1
Penumpang:
Ktp: 12345
Nama: Mr. Krab
Nomor: 2
Nomor: 4
Nomor: 5
Nomor: 8
Nomor: 9
Nomor: 10
maaf kursi telah terisi
Nomor: 1
Penumpang:
Ktp: 12345
Nama: Mr. Krab
Nomor: 2
Nomor: 3
Nomor: 6
```



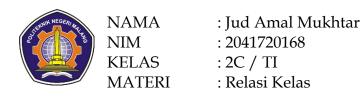
Tugas

Main

Panitia

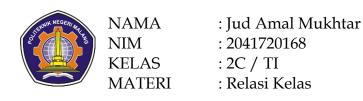


```
public void setPengawasTurnamen(Pengawas pengawas, int tim){
              arrTurnamen[tim-1].setPengawas(pengawas);
          public String getNama() {
          public void setNama(String nama) {
              this.nama = nama;
          public String getJabatan() {
          public void setJabatan(String jabatan) {
          public String info(){
54
              info+="====== Turnamen GOH ======\n\n";
              info+=this.nama+",\n sebagai "+this.jabatan+"\n\n";
              info+="T U R N A M E N\n";
              for(int i=0;i<arrTurnamen.length;i++){</pre>
Q.
                   if(arrTurnamen[i].getPengawas()!=nutt){
                       info+=arrTurnamen[i].info();
```



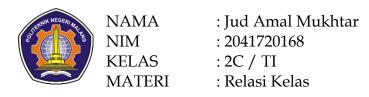
Pengawas

```
...5 lines
/**...4 lines */
public class Pengawas {
   public Pengawas(){
    public Pengawas(String nama, int umur) {
        this.nama = nama;
    public String getNama() {
    public void setNama(String nama) {
    public int getUmur() {
    public void setUmur(int umur) {
         this.umur = umur;
    public String info(){
```



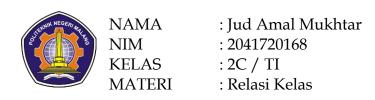
Peserta

```
...5 lines
/**...4 lines */
public class Peserta {
    private String nama;
    private int umur;
public Peserta(){
    public Peserta(String nama, int umur) {
        this.nama = nama;
        this.umur = umur;
    public String getNama() {
    public void setNama(String nama) {
    public int getUmur() {
    public void setUmur(int umur) {
    public String info(){
        info+=this.nama+", "+this.umur+" tahun\n";
```



```
Turnamen
       package Turnamen;
       /**...4 lines */
       public class Turnamen {
   private int noTim;
            private Pengawas pengawas=new Pengawas();
            private Pesenta[] arrPesenta=new Pesenta[3];
 Q,
                for(int i=0;i<arrPeserta.length;i++){</pre>
                    arrPeserta[i]=new Peserta();
            public int getNoTim() {
 24
            public void setNoTim(int noTim) {
            public void setPengawas(Pengawas pengawas) {
           public String getPengawas(){
35
                return this.pengawas.getNama();
```

```
public void setPeserta(Peserta peserta) {
    for(int i=0;i<arrPeserta.length;i++){</pre>
        if(arrPeserta[i].getNama()==nutt){
            arrPeserta[i]=peserta;
public String info(){
    String info="";
info+="----\n";
    info+="
    info+="----\n";
    info+="Pengawas = "+this.pengawas.info();
    info+="Peserta :\n";
    for(int i=0;i<arrPeserta.length;i++){</pre>
        info+=(j+1)+". "+arrPeserta[i].info();
```



Hasil