

MAKALAH, PERCOBAAN, LATIHAN, DAN TUGAS
MODUL PRAKTIKUM 5

Disusun sebagai salah satu tugas
mata kuliah PBO I



Patricia Joanne
140810160065

Dikumpulkan tanggal
26 September 2017

PROGRAM STUDI S-1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN

2017

Tugas Pendahuluan

1. Apa yang dimaksud dengan inheritance?
 2. Buatlah contoh kasus yang menerapkan konsep inheritance!
 3. Adakah perbedaan cara mengakses member class milik parent dan member class milik sendiri? Jelaskan melalui contoh! (Silahkan memanfaatkan jawaban soal nomor 2)
 4. Apa yang dimaksud dengan konsep single inheritance?
 5. Apa yang dimaksud dengan konsep multi level inheritance?
 6. Ada berapa modifier untuk pengontrolan akses? Jelaskan masing-masing!
 7. Apakah kegunaan kata kunci super? Jelaskan!
 8. Apakah yang dimaksud dengan konstruktor tidak diwariskan?
-

1. Pengertian dari **inheritance** adalah dimana sebuah objek dapat mempunyai objek turunan. Dalam hal pemrograman makan dapat diimplementasikan dengan sebuah class yang mempunyai class turunan. Class yang mempunyai class turunan disebut sebagai Parent Class atau Base Class, sedangkan class yang menjadi turunan dari Parent Class disebut sebagai Subclass atau Child Class. Secara sederhana misal Hewan adalah parent class maka child class bisa berupa ayam, kucing, bebek dll. Suatu subclass dapat mewarisi sebagian atau keseluruhan sifat dari parent classnya, dengan kata lain sebuah subclass hanya merupakan perluasan dari parent class.

2. --- Parent ---

```
public class Induk {  
    String hartaInduk;  
  
    protected void setHartaInduk(String h) {  
        this.hartaInduk = h;  
    }  
}
```

--- Child ---

```
public class Anak extends Induk{ //extend menyatakan turunan dari induk
    String hartaAnak; //field atau tipe datanya

    protected void setHartaAnak(String h) {
        this.hartaAnak = h;
    }
}
```

3. Pengaksesan member yang ada di parent class dari subclassnya tidak jauh berbeda dengan pengaksesan member subclass itu sendiri. Contoh: Suatu parent class dapat tidak mewariskan sebagian member-nya kepada subclassnya. Sejauh mana suatu member dapat diwariskan ke class lain, ataupun suatu member dapat diakses dari class lain, sangat berhubungan dengan access control (kontrol pengaksesan).

Dalam inheritance, member kelas yang diwariskan hanyalah variabel dan method saja. Sedangkan konstruktor parent class tidak diwariskan ke subclass. Tetapi ketika suatu obyek anak dibuat dalam artian ketika konstruktor anak dijalankan maka konstruktor parent class dijalankan terlebih dahulu dan selanjutnya menyelesaikan konstruktor anak.

4. Konsep single inheritance hanya memperbolehkan suatu subclass mempunyai satu parent class. Dengan konsep single inheritance ini, masalah pewarisan akan dapat diamati dengan mudah.

5. Perbedaan konsep multi level inheritance dengan konsep single inheritance yaitu memperbolehkan suatu subclass mempunyai subclass lagi.

6. Modifier akses berguna untuk memberikan hak akses kepada user. Terdiri dari:

- DEFAULT: Class member hanya dapat digunakan hanya pada class di paket yang sama. Tanpa penulisan modifier akses.
- PUBLIC: Class member dapat diakses darimana saja, baik di dalam paket maupun di luar paket
- PRIVATE: Class member hanya dapat diakses di dalam class itu sendiri atau class yang telah didefinisikan.

- **PROTECTED:** Class member dapat diakses di dalam package yang sama dan sub classnya. Subclass dapat berada di paket yang berbeda subclassnya akan mewarisi member variabel.

Note	Private	Default/package	Protected	Public
Same Class	v	v	v	v
Same Package		v	v	v
Different package (sub class)			v	v
Different package (non – sub class)				v

7. Subclass juga dapat memanggil constructor secara eksplisit dari superclass terdekat. Hal ini dilakukan dengan pemanggil constructor super. Pemanggil constructor super dalam constructor dari subclass akan menghasilkan eksekusi dari superclass constructor yang bersangkutan, berdasar dari argumen sebelumnya.

Ada beberapa hal yang harus diingat ketika menggunakan pemanggil constructor super:

- Pemanggil super() harus dijadikan pernyataan pertama dalam constructor.
- Pemanggil super() hanya dapat digunakan dalam definisi constructor.
- Termasuk constructor this() dan pemanggil super() tidak boleh terjadi dalam constructor yang sama.
- Pemakaian lain dari super adalah untuk menunjuk anggota dari superclass (seperti reference this).

8. Dalam konsep pewarisan constructor kelas induk tidak akan diwariskan ke kelas induk. Untuk memfasilitasi pemanggilan fungsi constructor kelas induk oleh kelas anak maka Java menyediakan mekanisme rujukan ke kelas induk dengan menggunakan kata kunci super. Dengan menggunakan kata kunci ini, maka kita dapat mengakses semua variabel kelas induk walaupun variabel tersebut tidak diwariskan.

Bab 5

Sifat Pewarisan

Percobaan 1: Penyembunyian informasi

```
public class Pegawai {  
    private String nama;  
    public double gaji;  
}
```

```
public class Manajer extends Pegawai {  
    public String departemen;  
    public void IsiData(String n, String d){  
        nama=n;  
        departemen=d;  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Pegawai.java  
  
E:\DOCS\task.bbr\Programming\Java>javac Manajer.java  
Manajer.java:4: error: nama has private access in Pegawai  
    nama=n;  
    ^  
1 error
```

Percobaan 2: Menggunakan kata kunci super

```
class Parent {  
    public int x = 5;  
}  
  
class Child extends Parent {  
    public int x = 10;  
    public void Info(int x){  
        System.out.println("Patricia Joanne 140810160065");  
        System.out.println("Nilai x sebagai parameter = "+x);  
        System.out.println("Data member x di class Child = "+this.x);  
        System.out.println("Data member x di class Parent = "+super.x);  
    }  
}  
  
public class NilaiX {  
    public static void main(String args[]){  
        Child tes = new Child();  
        tes.Info(20);  
    }  
}
```

```

E:\DOCS\task.bbr\Programming\Java>javac NilaiX.java

E:\DOCS\task.bbr\Programming\Java>java NilaiX
Patricia Joanne 140810160065
Nilai x sebagai parameter = 20
Data member x di class Child = 10
Data member x di class Parent = 5

```

Percobaan 3: Konstruktor tidak diwariskan

```

class Parent {
}

public class Childx extends Parent {
    int x;
    public Childx(){
        x = 5;
        super();
    }
}

```

```

E:\DOCS\task.bbr\Programming\Java>javac Childx.java
Childx.java:9: error: call to super must be first statement in constructor
    super();
    ^
1 error

```

Seharusnya pemanggilan konstruktor untuk parent class harus dilakukan di baris pertama pada konstruktor subclassnya

```

class Parent {
}

public class Childx extends Parent {
    int x;
    public Childx(){
        super();
        x = 5;
    }
}

```

```

E:\DOCS\task.bbr\Programming\Java>javac Childx.java

E:\DOCS\task.bbr\Programming\Java>java Childx
Error: Main method not found in class Childx, please define the main method as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application

```

Latihan 1: Tempatkan class Base dan class Class1 di direktori yang sama. Apa yang terjadi ketika Class1.java dikompile dan dijalankan jika sebelumnya Base.java belum dikompile? Jelaskan!

```
//Base.java
package Base;

class Base {
    protected void amethod() {
        System.out.println("Patricia Joanne 140810160065");
        System.out.println("amethod");
    }
    //end of amethod
}
//end of class Base
```

```
//Class1.java
package Class1;

public class Class1 extends Base {
    public static void main(String argv[]){
        Base b = new Base();
        b.amethod();
    }
    //end of main
}
//end of Class1
```

```
E:\DOCS\task.bbr\Programming\Java>javac Base.java
E:\DOCS\task.bbr\Programming\Java>javac Class1.java
Class1.java:4: error: cannot find symbol
public class Class1 extends Base {
                        ^
  symbol: class Base
Class1.java:6: error: cannot find symbol
    Base b = new Base();
    ^
  symbol:   class Base
  location: class Class1
Class1.java:6: error: cannot find symbol
    Base b = new Base();
                  ^
  symbol:   class Base
  location: class Class1
3 errors
```

Pembuatan package menjadi penyebab error karena dalam pewarisan sifat dalam 1 direktori tidak perlu menggunakan package. Solusi:

```
//Base.java
//package Base;

class Base {
    protected void amethod() {
        System.out.println("Patricia Joanne 140810160065");
        System.out.println("amethod");
    }
    //end of amethod
}
//end of class Base
```

```
//Class1.java
//package Class1;

public class Class1 extends Base {
    public static void main(String argv[]) {
        Base b = new Base();
        b.amethod();
    }
    //end of main
}
//end of Class1
```

```
E:\DOCS\task.bbr\Programming\Java>javac Base.java
E:\DOCS\task.bbr\Programming\Java>javac Class1.java
E:\DOCS\task.bbr\Programming\Java>java Class1
Patricia Joanne 140810160065
amethod
```

Latihan 2: Aturan overriding

- Berdasarkan kode di bawah ini, akses modifier (public, protected atau private) apa yang diijinkan di tambahkan sebelum myMethod() baris 3?
- Jika baris 3 seperti kode di bawah (apa adanya tanpa perubahan) keywords apa yang diijinkan ditambahkan sebelum myMethod baris 8?


```

class HumptyDumpty {
    void myMethod() {}
}
class HankyPanky extends HumptyDumpty {
    void myMethod() {}
}

```

Berdasarkan Source code di atas, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 3 yaitu Private. Sedangkan, akses modifier yang diijinkan untuk ditambahkan sebelum myMethod() di baris 8 yaitu Public dan Protected.

Latihan 3:

- Apa yang terjadi bila kedua kode di bawah ini dikompile dan dijalankan dalam satu direktori? Jelaskan!
- Bagaimana solusi supaya tidak terjadi error?

```

package MyPackage;

class P1 {
    void aFancyMethod() {
        System.out.println("What a fancy method");
    }
}

```

```

public class P2 extends P1 {
    public static void main(String argv[]) {
        P2 p2 = new P2();
        p2.aFancyMethod();
    }
}

```

```

E:\DOCS\task.bbr\Programming\Java>javac P1.java
E:\DOCS\task.bbr\Programming\Java>javac P2.java
P2.java:1: error: cannot access P1
public class P2 extends P1 {
                        ^
bad class file: .\P1.class
class file contains wrong class: MyPackage.P1
Please remove or make sure it appears in the correct subdirectory of the classpath.

```

Seharusnya packagenya dihapus saja atau dibuat folder dengan nama MyPackage dan P1.java dimasukkan ke dalam folder tersebut. Jadi yang benar seperti ini.

```

class P1 {
    void aFancyMethod() {
        System.out.println("Patricia Joanne 140810160065");
        System.out.println("What a fancy method");
    }
}

```

E:\DOCS\task.bbr\Programming\Java>javac P1.java

E:\DOCS\task.bbr\Programming\Java>javac P2.java

E:\DOCS\task.bbr\Programming\Java>java P2

Patricia Joanne 140810160065

What a fancy method

Latihan 4: Mengimplementasikan UML class diagram dalam program untuk package perbankan

```

package perbankan;

public class PenyimpananUang extends Tabungan {
    private double tingkatBunga;
    public PenyimpananUang(int saldo, double tingkatBunga) {
        super(saldo);
        this.tingkatBunga = tingkatBunga;
    }
    public double cekUang() {
        return saldo+(saldo*tingkatBunga);
    }
}

```

```

package perbankan;

public class Tabungan {
    protected int saldo;
    public Tabungan(int saldo) {
        this.saldo = saldo;
    }
    public int getSaldo() {
        return saldo;
    }
    public void simpanUang(int jumlah) {
        saldo+=jumlah;
    }
    public boolean ambilUang(int jumlah) {
        saldo-=jumlah;
        if(saldo==0) return true;
        else {
            saldo+=jumlah;
            return false;
        }
    }
}

```

```

import perbankan.Tabungan;
import perbankan.PenyimpananUang;

public class Test {
    public static void main(String args[]){
        System.out.println("Patricia Joanne 140810160065");
        PenyimpananUang tabungan = new PenyimpananUang(5000, 8.5/100);
        System.out.println("Uang yang ditabung: 5000");
        System.out.println("Tingkat bunga sekarang: 8.5%");
        System.out.println("Total uang sekarang: "+tabungan.cekUang());
    }
}

```

E:\DOCS\task.bbr\Programming\Java>javac Test.java

E:\DOCS\task.bbr\Programming\Java>java Test

```

Patricia Joanne 140810160065
Uang yang ditabung: 5000
Tingkat bunga sekarang: 8.5%
Total uang sekarang: 5425.0

```

Latihan 5: Konstruktor tidak diwariskan

```

class Basee {
    Basee(int i){
        System.out.println("Patricia Joanne 140810160065");
        System.out.println("Base constructor");
    }
    Basee() {
    }
}

```

```

public class Sup extends Basee {
    public static void main(String argv[]){
        Sup s = new Sup();
        //baris 1
    }
    Sup() {
        //baris 2
        super(1);
    }
    public void derived(){
        //baris 3
    }
}

```

```
E:\DOCS\task.bbr\Programming\Java>javac Basee.java
E:\DOCS\task.bbr\Programming\Java>javac Sup.java
E:\DOCS\task.bbr\Programming\Java>java Sup
Patricia Joanne 140810160065
Base constructor
```

Latihan 6: Konstruktor tidak diwariskan

```
private class Basen {
    Basen() {
        int i = 100;
        System.out.println(i);
    }
}
```

```
public class Pri extends Basen {
    static int i = 200;
    public static void main(String args[]){
        System.out.println("Patricia Joanne 140810160065");
        Pri p = new Pri();
        System.out.println(i);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Basen.java
Basen.java:1: error: modifier private not allowed here
private class Basen {
    ^
1 error
```

Class yang diprivate tidak bisa digunakan untuk file class lainnya. Seharusnya privatenya dihapus saja.

```
class Basen {
    Basen() {
        int i = 100;
        System.out.println(i);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Pri.java
E:\DOCS\task.bbr\Programming\Java>java Pri
Patricia Joanne 140810160065
100
200
```

Latihan 7: Apa yang tampil di layar bila kode di bawah ini dijalankan?

```
class X {  
    Y b = new Y();  
    X() {  
        System.out.print("X");  
    }  
}
```

```
class Y {  
    Y() {  
        System.out.print("Y");  
    }  
}
```

```
public class Z extends X {  
    Y y = new Y();  
    Z() {  
        System.out.print("Z");  
    }  
    public static void main(String[] args) {  
        new Z();  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Z.java
```

```
E:\DOCS\task.bbr\Programming\Java>java Z  
YXYZ
```

Latihan 8: Kompilasi dan jalankan program berikut! Apa yang terjadi? Jelaskan!

```
public class Hope {  
    public static void main(String argv[]) {  
        Hope h = new Hope();  
    }  
    protected Hope() {  
        for(int i=0; i<10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Hope.java
```

```
E:\DOCS\task.bbr\Programming\Java>java Hope
```

```
0  
1  
2  
3  
4  
5  
6  
7  
8  
9
```

Tugas 1: Mengimplementasikan UML class diagram dalam program untuk package perbankan

```
package bank;

public class Tabungan {
    int saldo;
    public Tabungan(int iniSaldo){
        this.saldo = iniSaldo;
    }
}
```

```
package bank;

public class PengambilanUang extends Tabungan {
    private int proteksi;
    public PengambilanUang(int saldo){
        super(saldo);
    }
    public PengambilanUang(int saldo, int proteksi){
        super(saldo);
        this.proteksi=proteksi;
    }
    public int getSaldo(){
        return saldo;
    }
    public boolean ambilUang(int jumlah){
        int temp=saldo;
        saldo-=jumlah;
        if(saldo<proteksi){
            saldo=temp;
            return false;
        }
        else return true;
    }
}
```

```
import bank.*;

public class Tugas1{
    public static void main(String args[]){
        System.out.println("Patricia Joanne 140810160065\n");
        PengambilanUang tabungan = new PengambilanUang(5000, 1000);
        System.out.println("Uang yang ditabung: 5000");
        System.out.println("Uang yang diproteksi: 1000");
        System.out.println("-----");
        System.out.println("Uang yang akan diambil: 4500 " +tabungan.ambilUang(4500));
        System.out.println("Saldo sekarang: " +tabungan.getSaldo());
        System.out.println("-----");
        System.out.println("Uang yang akan diambil: 2500 " +tabungan.ambilUang(2500));
        System.out.println("Saldo sekarang: " +tabungan.getSaldo());
    }
}
```

```

E:\DOCS\task.bbr\Programming\Java>javac Tugas1.java

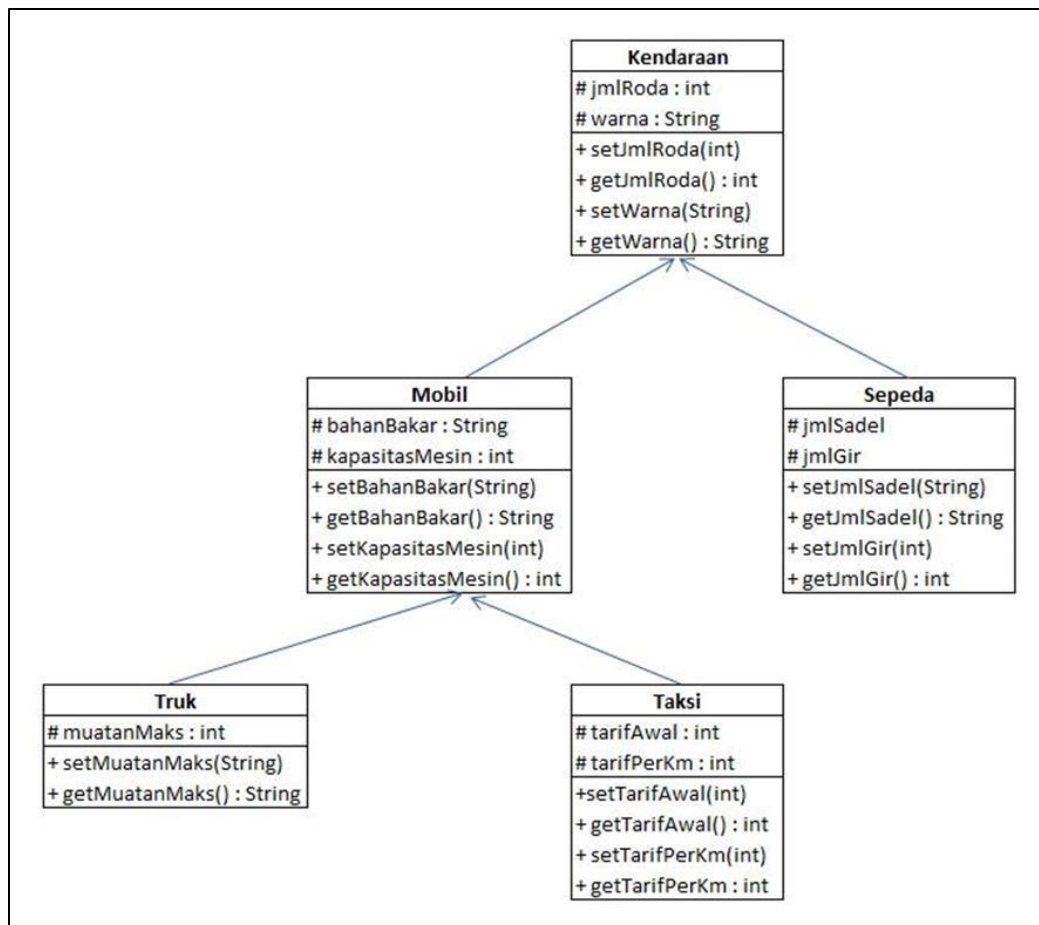
E:\DOCS\task.bbr\Programming\Java>java Tugas1
Patricia Joanne 140810160065

Uang yang ditabung: 5000
Uang yang diproteksi: 1000
-----
Uang yang akan diambil: 4500 false
Saldo sekarang: 5000
-----
Uang yang akan diambil: 2500 true
Saldo sekarang: 2500

```

Tugas 2: Mengimplementasikan UML class diagram dalam program

a. Buatlah kelas-kelas berdasarkan UML class diagram yang ada!



b. Selanjutnya buatlah kelas Tes.java yang membuat obyek-obyek serta mengeset nilai variabel seperti pada Tabel 5.2. dan tampilkan data per obyek.

```
package transportasi;

public class Kendaraan{
    protected int jmlRoda;
    protected String warna;

    public void setJmlRoda(int roda){
        jmlRoda=roda;
    }
    public int getJmlRoda(){
        return jmlRoda;
    }
    public void setWarna(String warna){
        this.warna=warna;
    }
    public String getWarna(){
        return warna;
    }
}
```

```
package transportasi;

public class Mobil extends Kendaraan {
    protected String bahanBakar;
    protected int kapasitasMesin;

    public void setBahanBakar(String bahanBakar){
        this.bahanBakar=bahanBakar;
    }
    public String getBahanBakar(){
        return bahanBakar;
    }
    public void setKapasitasMesin(int kapasitasMesin){
        this.kapasitasMesin=kapasitasMesin;
    }
    public int getKapasitasMesin(){
        return kapasitasMesin;
    }
}
```



```

package transportasi;

public class Sepeda extends Kendaraan{
    protected int jmlSadel;
    protected int jmlGir;

    public void setJmlSadel(int jmlSadel){
        this.jmlSadel=jmlSadel;
    }
    public int getJmlSadel(){
        return jmlSadel;
    }
    public void setJmlGir(int jmlGir){
        this.jmlGir=jmlGir;
    }
    public int getJmlGir(){
        return jmlGir;
    }
}

```

```

package transportasi;

public class Truk extends Mobil{
    protected int muatanMaks;

    public void setMuatanMaks(int muatanMaks){
        this.muatanMaks=muatanMaks;
    }
    public int getMuatanMaks(){
        return muatanMaks;
    }
}

```

```

package transportasi;

public class Taksi extends Mobil{
    protected int tarifAwal;
    protected int tarifPerKm;
    public void setTarifAwal(int tarifAwal){
        this.tarifAwal=tarifAwal;
    }
    public int getTarifAwal(){
        return tarifAwal;
    }
    public void setTarifPerKm(int tarifPerKm){
        this.tarifPerKm=tarifPerKm;
    }
    public int getTarifPerKm(){
        return tarifPerKm;
    }
}

```



```

System.out.println("Taksi1\t"+taksi1.getJmlRoda()+"\t"+taksi1.getWarna()+"\t"+taksi1.getBahanBakar()+"\t\t"
+taksi1.getKapasitasMesin()+"\t\t"+taksi1.getTarifAwal()+"\t\t"+taksi1.getTarifPerKm());
System.out.println("-----");
System.out.println("Taksi2\t"+taksi2.getJmlRoda()+"\t"+taksi2.getWarna()+"\t"+taksi2.getBahanBakar()+"\t\t"
+taksi2.getKapasitasMesin()+"\t\t"+taksi2.getTarifAwal()+"\t\t"+taksi2.getTarifPerKm());
System.out.println("-----");
System.out.print("\t\t\tjmlSadel\tjmlGir\n");
System.out.println("-----");
System.out.println("Sepeda1\t"+sepeda1.getJmlRoda()+"\t"+sepeda1.getWarna()+"\t"
+sepeda1.getJmlSadel()+"\t\t"+sepeda1.getJmlGir());
System.out.println("-----");
System.out.println("Sepeda2\t"+sepeda2.getJmlRoda()+"\t"+sepeda2.getWarna()+"\t"
+sepeda2.getJmlSadel()+"\t\t"+sepeda2.getJmlGir());
}
}

```

E:\DOCS\task.bbr\Programming\Java>javac Tugas2.java

E:\DOCS\task.bbr\Programming\Java>java Tugas2

Patricia Joanne 140810160065

Obyek	jmlRoda	warna	bahanBakar	KapasitasMesin	muatanMaks	
Truk1	4	Kuning	Solar	1500	5000	
Truk2	6	Merah	Solar	2000	5000	
					tarifAwal	trfPerKm
Taksi1	4	Oranye	Bensin	1500	10000	5000
Taksi2	4	Biru	Bensin	1300	7000	3500
			jmlSadel	jmlGir		
Sepeda1	3	Hitam	1	2		
Sepeda2	2	Putih	2	5		