

PBO TOPIC 9

PRE LAB 1

Soal 1

Benar

Poin 15,00 dari 15,00

► Tandai pertanyaan

Pada inheritance, class yang merupakan versi umum dari suatu class disebut...

- ☐ slave class
- ☐ child class
- ☐ subclass
- ☒ superclass



Jawaban anda benar.

Pada inheritance, class yang merupakan versi spesialisasi dari suatu class disebut dengan **subclass** (atau class turunan). Sedangkan class yang merupakan versi umum dari suatu class disebut dengan **superclass** (atau class dasar).

Soal 2

Benar

Poin 15,00 dari 15,00

Tandai pertanyaan

Pada inheritance, class yang merupakan versi spesialisasi dari suatu class disebut...

- ☐ superclass
- ☐ master class
- ☐ parent class
- ☒ subclass



Jawaban anda benar.

Pada inheritance, class yang merupakan versi spesialisasi dari suatu class disebut dengan **subclass** (atau class turunan). Sedangkan class yang merupakan versi umum dari suatu class disebut dengan **superclass** (atau class dasar).

Soal 3

Benar

Poin 15,00 dari 15,00

Tandai pertanyaan

Keyword apa yang menunjukkan bahwa suatu class mewarisi class lain?

- ☒ extends
- ☐ derived
- ☐ specialized
- ☐ based



Jawaban anda benar.

Inheritance memungkinkan kita untuk mengekstensi kemampuan dari sebuah class dengan membuat class lain yang merupakan versi spesialisasi dari class tersebut. keyword *extends* menandakan bahwa class ini mengekstensi class lain (sebuah *superclass*). Nama dari *superclass* dituliskan setelah keyword *extends*.

Soal 4

Benar

Poin 15,00 dari 15,00

Tandai pertanyaan

Berikut adalah baris pertama dari definisi sebuah class:

```
public class Truck extends Vehicle
```

Apa nama superclass dan subclassnya?

Nama subclass:

Truck



Nama superclass:

Vehicle



Halaman sebelumnya

Halaman selanjutnya

No 5

```
public class MobilListrik extends Mobil {  
    private int lamaRecharge;
```

```
    public MobilListrik(){  
        setPabrikan("");  
        setModel("");  
        setWarna("");  
        setKecepatanMax(0);  
        this.lamaRecharge = 0;  
    }
```

```
    public MobilListrik(String pabrikan, String model, String warna, int kecepatanMax, int lamaRecharge){  
        setPabrikan(pabrikan);  
        setModel(model);  
        setWarna(warna);  
        setKecepatanMax(kecepatanMax);  
        this.lamaRecharge = lamaRecharge;  
    }
```

```
    public void setLamaRecharge(int lamaRecharge)  
    {  
        this.lamaRecharge = lamaRecharge;  
    }
```

```
public int getLamaRecharge()
{
    return lamaRecharge;
}
}
```

No 6

```
public class Lingkaran extends Bangun{
    private double radius;

    public void setRadius(double radius)
    {
        this.radius = radius;
        double luas = Math.PI * radius * radius;
        setLuas(luas);
    }

    public double getRadius()
    {
        return radius;
    }
}
```

PRELAB 2

Soal 1

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Berikut ini manakah yang merupakan panggilan eksplisit ke constructor default pada superclass?

- ☐ default()
- ☐ class()
- ☒ super()
- ☐ base()



Jawaban anda benar.

Kita dapat menuliskan statement `super` yang memanggil constructor superclass, tetapi hanya pada constructor subclass. Kita tidak dapat memanggil constructor superclass pada method-method lain.

Soal 2

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Dalam constructor subclass, jika terdapat statement `super` yang memanggil constructor superclass, maka statement tersebut harus ditulis sebagai?

- ☐ Statement terakhir
- ☐ Tidak dituliskan dimanapun
- ☐ Di antara header constructor dan kurung kurawal
- ☒ Statement pertama



Jawaban anda benar.

Statement `super` yang memanggil constructor superclass haruslah **statement pertama** dalam constructor subclass. Ini karena constructor superclass harus dieksekusi sebelum kode dalam constructor subclass dieksekusi.

no 3

```
Car car = new Car(5);
RaceCar fastCar = new RaceCar(5);
car.display();
car.addFuel();
car.display();
fastCar.display();
fastCar.addFuel();
fastCar.display();
```

Jawaban: 5 6 10 11



Constructor class RaceCar memanggil constructor class Car dengan memberikan argument berupa dua kali nilai argument yang diterimanya. Sehingga, statement:

RaceCar fastCar = new RaceCar(5);

akan membuat sebuah object RaceCar dengan menginisialisasi field fuel dari superclass yang terasosiasi dengan object tersebut dengan nilai 10.

Karena class RaceCar mengekstensi class Car, maka semua method public dari class Car diwarisi oleh class RaceCar. Ini berarti object dari class RaceCar dapat memanggil method-method public dari class Car.

No 4

```
public class Point3D extends Point2D {
    private int z;

    public Point3D()
    {
        super(0, 0);
        this.z = 0;
    }

    public Point3D(int x, int y, int z) {
        super(x, y);
        this.z = z;
    }

    public void setZ(int z)
    {
        this.z = z;
    }

    public int getZ()
```

```
{  
    return z;  
}  
}
```

PRELAB 3

Soal 1

Benar

Poin 15,00 dari 15,00

Tandai pertanyaan

Jika terdapat method di dalam subclass yang mempunyai signature yang sama dengan method di superclass, maka method tersebut disebut melakukan ____ method di superclass.

☒ overriding



☐ overloading

☐ composition

☐ error

Jawaban anda benar.

Jika sebuah method dalam subclass mempunyai **signature yang sama** seperti sebuah method dalam superclass, method subclass ini disebut **meng-overriding** method superclass.

Soal 2

Benar

Poin 15,00 dari 15,00

Tandai pertanyaan

Jika terdapat method di dalam subclass yang mempunyai nama yang sama dengan superclass tetapi mempunyai signature yang berbeda, maka method di dalam subclass tersebut disebut melakukan ___ terhadap method di superclassnya.

- ☐ Composition
- ☒ Overloading
- ☐ Error
- ☐ Overriding




Jawaban anda benar.

Jika dua method **memiliki nama sama namun signature yang berbeda**, mereka disebut **ter-overloading**. Method overloading dapat dilakukan dalam class yang sama atau untuk method yang berada di superclass dan method lain yang berada di dalam subclass.


```
public void setId (int theId)
{
    this.id = theId;
}
}
```

Misalkan kita ingin menambahkan method `getFood` pada class `Student` yang meng-`overriding` method `getFood` pada class `Person`. Header method `getFood` manakah dari header-header method di bawah yang akan secara benar meng-`overriding` method `getFood` pada class `Person`?

- ☒ `public String getFood()` 
- ☐ `public void getFood()`
- ☐ `public String getFood(int quantity)`
- ☐ `public void getFood(int quantity)`

Jawaban anda benar.

3.

```
public class Truk extends Mobil
{
    public void m1()
    {
        System.out.println("Truk method 1");
    }
}
```

Asumsikan terdapat deklarasi-deklarasi variabel berikut:

```
Mobil myMobil = new Mobil();
Truk myTruk = new Truk();
```

Apa output dari statement-statement berikut?

- | | | |
|------------------|----------------|---|
| 1. myMobil.m1(); | Mobil method 1 | ✓ |
| 2. myMobil.m2(); | Mobil method 2 | ✓ |
| 3. myTruk.m1(); | Truk method 1 | ✓ |
| 4. myTruk.m2(); | Mobil method 2 | ✓ |

4.

5. public class Manager extends Pegawai {

private double bonus;

```
public Manager(String nama, double gajiPokok, double bonus){
    super(nama, gajiPokok);
    this.bonus = bonus;
}
```

```
@Override
public double getGaji()
{
    return super.getGaji() + bonus;
}
}
```

PRELAB 4

1.

```
public class BujurSangkar extends PersegiPanjang {  
    public BujurSangkar(double sisi){  
        super(sisi, sisi);  
  
        setLuas(getPanjang() * getLebar());  
    }  
}
```

PRELAB 5

Soal 1

Benar

Poin 25,00 dari 25,00

Tandai pertanyaan

Perhatikan potongan definisi-definisi class berikut (hanya header-header class yang ditampilkan):

```
public class Vehicle { ... }  
public class Car extends Vehicle { ... }  
public class SUV extends Car { ... }
```

Manakah dari statement-statement di bawah yang merupakan statement-statement yang valid (tidak menghasilkan error)? (Pilih satu atau lebih)

☒ Vehicle v = new Car();



☒ SUV s = new SUV();



☒ Car c = new SUV();



☒ Vehicle v = new SUV();



☐ Car c = new Vehicle();

☐ SUV s = new Car();

Soal 2

Benar

Poin 25,00 dari 25,00

Tandai pertanyaan

Operator manakah di bawah ini yang bisa digunakan untuk menguji apakah suatu variabel referensi mereferensikan object dari suatu class?

- ☐ typeof
- ☐ isclass
- ☐ isinstance
- ☒ instanceof

Jawaban anda benar.

Operator `instanceof` dapat kita gunakan untuk menguji apakah sebuah object adalah sebuah instance dari suatu class tertentu. Ekspresi dengan operator `instanceof` adalah ekspresi Boolean yang mengembalikan `true` jika object yang direferensikan oleh suatu variabel adalah sebuah instance dari suatu class.

3 . Prelab 9.5 no 3

```
public class UjiBangun
{
    public static void main(String[] args)
    {
        // Buat array empat elemen bertipe Bangun.
        Bangun[] obj = new Bangun[4];

        // Isi array dengan object-object berupa subclass dari Bangun.
        obj[0] = new PersegiPanjang(7.5, 5.0);
        obj[1] = new Lingkaran(4.5);
        obj[2] = new BujurSangkar(12.5);
        obj[3] = new SegitigaSiku(10.0, 7.5);

        // Cetak Daftar Bangun
        System.out.println("Daftar Bangun:");

        // Loop untuk mencetak informasi setiap object
        for (
            int i = 0; i < obj.length; i++
        )
        {
```

```

System.out.println();
// Cetak Judul nomor bangun.
System.out.println("Bangun " + (i + 1) + ": ");
// Cetak luas bangun
System.out.println(
"Luas = " + obj[i].getLuas()
);
    System.out.print("Jenis: ");

    // Statement if untuk menguji class dari setiap object
    // dan menampilkan informasi setiap object.

    // Case 1: Jika object adalah dari class PersegiPanjang
    if (
obj[i] instanceof PersegiPanjang
)
    {
        // Cetak Persegi Panjang
        System.out.println("Persegi Panjang");
        // Casting ke tipe PersegiPanjang
        PersegiPanjang p =
(PersegiPanjang) obj[i];

        // Cetak panjang dan lebar
        System.out.println(
"panjang = " + p.getPanjang() + ", " + "lebar= " + p.getLebar()
);
    }
    // Case 2: Jika object adalah dari class BujurSangkar
    else if (
obj[i] instanceof BujurSangkar
)
    {
        // Cetak Bujur Sangkar
        System.out.println("Bujur Sangkar");
        // Casting ke tipe BujurSangkar
        BujurSangkar b =
(BujurSangkar) obj[i];

        // Cetak sisi
        System.out.println(
"panjang = " + b.getPanjang() + ", " + "lebar=" + b.getPanjang()
);
    }
    // Case 3: Jika object adalah dari class Lingkaran
    else if (
obj[i] instanceof Lingkaran

```

```

)
{
    // Cetak Lingkaran
    System.out.println("Lingkaran");
    // Casting ke tipe Lingkaran
    Lingkaran l =
(Lingkaran) obj[i];

    // Cetak radius
    System.out.println(
"radius = " + l.getRadius()
);
}
// Case 4: Jika object adalah dari class SegitigaSiku
else if (
obj[i] instanceof SegitigaSiku
)
{
    // Cetak Segitiga Siku
    System.out.println("Segitiga Siku");
    // Casting ke tipe SegitigaSiku
    SegitigaSiku s =
(SegitigaSiku) obj[i];

    // Cetak alas dan tinggi
    System.out.println(
"alas = " + s.getAlas() + ", " + "tinggi = " + s.getTinggi()
);
}
}
}
}

```

PRELAB 6

Soal 1

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Apa nama class yang secara langsung atau tidak langsung diwarisi oleh semua class?

- ☐ Java
- ☒ Object
- ☐ Root
- ☐ Super

Jawaban anda benar.

Semua class dalam Java secara langsung atau tidak langsung mewarisi dari sebuah class bernama `Object`. Class yang tidak ditulis dengan keyword `extends` untuk mewarisi dari class lain, secara otomatis di-ekstensi oleh Java untuk mewarisi dari class `Object`.

2 .

```
public class RekeningBank
{
    private String nomorRekening;
    private double saldo;

    public RekeningBank(String nomorRekening, double saldo)
    {
        this.nomorRekening = nomorRekening;
        this.saldo = saldo;
    }

    public String getNomorRekening()
    {
        return nomorRekening;
    }

    public double getSaldo()
    {
        return saldo;
    }

    // [1] Tambahkan method toString di bawah.
    public String toString(){
        return this.nomorRekening + ", " + this.saldo;
    }
}
```



```
}

// [2] Tambahkan method equals di bawah.
public boolean equals(RekeningBank rekening){
    if(rekening.nomorRekening == this.nomorRekening && rekening.saldo ==
this.saldo){
        return true;
    }

    return false;
}
}
```

PRELAB 7

Soal 1

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Pernyataan manakan dari pernyataan-pernyataan di bawah ini yang **benar** mengenai class abstract?

- ☐ Class abstract tidak dapat memiliki field
- ☐ Class abstract tidak dapat memiliki method abstract
- ☐ Class abstract tidak dapat digunakan sebagai superclass
- ☒ Class abstract tidak dapat diinstansiasi



Jawaban anda benar.

Class abstract tidak untuk diinstansiasi, tetapi berfungsi sebagai superclass untuk class-class lain. Class abstract dapat dipandang sebagai class yang belum lengkap, dan subclass-subclass yang mengekstensinya bertugas untuk melengkapinya.:

Soal 2

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Pernyataan-pernyataan berikut manakah yang benar jika di dalam sebuah class terdapat method abstract? (Pilih satu atau lebih)

☐ Subclass yang mengekstensi class tersebut harus meng-overloading method-method abstract yang terdapat dalam class tersebut.

☒ Class tersebut tidak dapat diinstansiasi

☒ Subclass yang mengekstensi class tersebut harus meng-overriding method-method abstract yang terdapat dalam class tersebut.

☒ Class tersebut harus dideklarasikan sebagai class abstract

Jawaban anda benar.

1. Class yang memiliki method abstract harus dideklarasikan sebagai class abstract.
2. Jika terdapat method abstract di dalam sebuah class, method tersebut harus di-overriding oleh sebuah method dalam subclass yang mengekstensi class tersebut. Jika tidak terdapat method yang meng-overriding, maka compiler akan memberikan pesan error. Method abstract digunakan untuk memastikan subclass mengimplementasikan (menuliskan kode-kode untuk) method tersebut.

3 .

```
public class Kucing extends Hewan{
    public Kucing(){
        super(4);
    }

    @Override
    public void bersuara(){
        System.out.println("Meong...");
    }
}
```

PRELAB 8

Soal 1

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Ketika kita membuat sebuah class yang menggunakan sebuah interface, kita menuliskan keyword _____ dan nama interface pada header dari class tersebut.

- ☐ listener
- ☐ extends
- ☒ implements
- ☐ interface



Jawaban anda benar.

Untuk mendeklarasikan sebuah class yang mengimplementasikan sebuah interface, kita menuliskan keyword `implements` yang diikuti nama interface yang ingin diimplementasikan setelah nama class pada header class tersebut.

Soal 2

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Hal apa yang harus dilakukan oleh class yang mengimplementasi sebuah interface?

- ☐ Tidak menuliskan constructor
- ☒ Menuliskan kode untuk method-method yang dirinci oleh interface dengan signature dan tipe return yang sama persis
- ☐ Meng-overloading semua method yang berada di interface
- ☐ Dideklarasikan sebagai class abstract



Jawaban anda benar.

Interface harus diimplementasikan oleh class. Class yang mengimplementasikan sebuah interface harus meng-overriding method-method yang dirinci oleh interface tersebut.

Soal 3

Benar

Poin 20,00 dari 20,00

Tandai pertanyaan

Pernyataan-pernyataan berikut manakah yang benar mengenai perbedaan interface dengan class abstract? (Pilih satu atau lebih)

- ☒ Interface tidak dapat memiliki method non-abstract sedangkan class abstract dapat memiliki method non-abstract ✓
- ☒ Interface tidak memiliki constructor sedangkan class abstract memiliki constructor ✓
- ☐ Interface dapat diinstansiasi sedangkan class abstract tidak dapat diinstansiasi
- ☒ Sebuah class dapat mengimplementasikan lebih dari satu interface sedangkan sebuah class hanya dapat mengekstensi satu class abstract ✓

Jawaban anda benar.

Interface dalam bentuk dasarnya mirip dengan class abstract yang hanya terdiri dari method-method abstract. Di dalam class abstract kita dapat menuliskan method abstract maupun method non-abstract. Namun di dalam interface, kita hanya bisa menuliskan method-method abstract. **Method non-abstract tidak diperbolehkan ditulis di dalam interface.**

4 .

```
public class Buku implements BarangRitel {
    private String judul;
    private String pengarang;
    private double hargaRitel;

    public Buku(String judul, String pengarang, double hargaRitel){
        this.judul = judul;
        this.pengarang = pengarang;
        this.hargaRitel = hargaRitel;
    }

    public String getJudul(){
        return judul;
    }

    public String getPengarang(){
        return pengarang;
    }

    @Override
    public double getHargaRitel(){
        return hargaRitel;
    }
}
```

LAB ACT 9

1 .

```
public class PegawaiHonoror extends Pegawai {
    private double honorPerJam, jamKerja;

    public PegawaiHonoror(String nama, double honorPerJam, double jamKerja){
        super(nama, honorPerJam);

        this.honorPerJam = honorPerJam;
        this.jamKerja = jamKerja;
    }

    @Override
    public double getGaji(){
        return honorPerJam * jamKerja;
    }
}
```

2 .

```
public class RekeningMinMax extends RekeningBank {
    private double saldoMax, saldoMin;

    public RekeningMinMax(String rekening, double saldoAwal){
        super(rekening, saldoAwal);
        this.saldoMax = saldoAwal;
        this.saldoMin = saldoAwal;
    }

    public double getSaldoMax(){
        return saldoMax;
    }

    public double getSaldoMin(){
        return saldoMin;
    }

    @Override
    public void deposit(double jumlah)
    {
        setSaldo(getSaldo() + jumlah);

        if(getSaldo() > this.saldoMax){
            saldoMax = getSaldo();
        }
    }
}
```

```
}

@Override
public void withdraw(double jumlah)
{
    if (getSaldo() >= jumlah)
    {
        setSaldo(getSaldo() - jumlah);

        if(getSaldo() < this.saldoMin){
            saldoMin = getSaldo();
        }
    }
    else
    {
        System.out.println("Dana tidak mencukupi.");
    }
}
}
```