



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Apa yang dimaksud dengan Inheritance serta sebutkan macam-macam Inheritance!

Jawaban

Inheritance adalah pewarisan pada object oriented programming yang merupakan suatu konsep pewarisan atau menurunkan suatu karakteristik class ke class yang lain.

Macam – macam inheritance

1. Single inheritance : merupakan suatu pewarisan Tunggal Ketika sebuah super class mewariskan satu sub class.
2. Hierarchical inheritance : Ketika sebuah Super Class mewariskan ke dua atau lebih sebuah Sub class

Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Apa maksud dari Generalisasi dan Spesialisasi dalam Inheritance? Buatlah ilustrasi dari kedua konsep tersebut!

Jawaban

- Generalisasi adalah proses menggabungkan fitur-fitur yang umum dari beberapa kelas menjadi kelas induk yang lebih umum. Contoh: Kita memiliki class kendaraan, nah kita bisa menggeneralisasi atribut dan perilaku umum mereka dalam satu kelas induk yang lebih umum.
- Spesialisasi adalah ini melibatkan penciptaan kelas-kelas yang lebih spesifik atau lebih terperinci dari kelas yang lebih umum. Contoh jika kita memiliki berbagai jenis kendaraan seperti Mobil, Motor, dan Sepeda, kita bisa menggeneralisasi atribut dan perilaku yang umum dari ketiganya (seperti warna, kecepatan, dan kemampuan untuk berjalan) dalam satu kelas induk yang lebih umum, misalnya kelas Kendaraan.)

Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Terdapat tiga jenis film: Animasi, Dokumenter, dan Pendek. Setiap jenis film memiliki atribut umum serta atribut khusus untuk masing-masing jenis film. Buatlah class dan atribut yang sesuai serta buatlah method untuk menampilkan informasi tentang masing-masing film!

PETUNJUK:

Class Film sebagai Class Induk;

Class Animasi, Dokumenter dan Pendek sebagai Class Anak;

Jawaban

Ketik jawaban disini ...

Source Code

```
// Class induk: Film
class Film {
    // Atribut umum untuk semua jenis film
    protected String judul;
    protected String sutradara;
    protected int tahunRilis;

    public Film(String judul, String sutradara, int tahunRilis)
    {
        this.judul = judul;
        this.sutradara = sutradara;
        this.tahunRilis = tahunRilis;
    }

    public void tampilkanInfo() {
        System.out.println("Judul: " + judul);
        System.out.println("Sutradara: " + sutradara);
        System.out.println("Tahun Rilis: " + tahunRilis);
    }
}
```



TUGAS & EVALUASI

```
}

// Class Anak: Animasi
class Animasi extends Film {

    private String studio;
    private String teknikAnimasi;

    public Animasi(String judul, String sutradara, int
tahunRilis, String studio, String teknikAnimasi) {
        super(judul, sutradara, tahunRilis);
        this.studio = studio;
        this.teknikAnimasi = teknikAnimasi;
    }

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Studio: " + studio);
        System.out.println("Teknik Animasi: " + teknikAnimasi);
    }
}

class Dokumenter extends Film {
    private String topik;
    private String durasi;

    public Dokumenter(String judul, String sutradara, int
tahunRilis, String topik, String durasi) {
        super(judul, sutradara, tahunRilis);
        this.topik = topik;
        this.durasi = durasi;
    }

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Topik: " + topik);
        System.out.println("Durasi: " + durasi);
    }
}
```



TUGAS & EVALUASI

```
    }  
}  
  
class Pendek extends Film {  
    private int durasi;  
  
    public Pendek(String judul, String sutradara, int  
tahunRilis, int durasi) {  
        super(judul, sutradara, tahunRilis);  
        this.durasi = durasi;  
    }  
  
    @Override  
    public void tampilkanInfo() {  
        super.tampilkanInfo();  
        System.out.println("Durasi: " + durasi + " menit");  
    }  
}  
  
public class Film_07651{  
    public static void main(String[] args) {  
        Film film1 = new Animasi("Up", "Pete Docter", 2009,  
"Pixar", "3D CGI");  
        Film film2 = new Dokumenter("Our Planet", "David  
Attenborough", 2019, "Alam dan Lingkungan", "8 episode x 50  
menit");  
        Film film3 = new Pendek("The Neighbor's Window",  
"Marshall Curry", 2019, 20);  
  
        System.out.println("Informasi Film Animasi:");  
        film1.tampilkanInfo();  
        System.out.println("\nInformasi Film Dokumenter:");  
        film2.tampilkanInfo();  
        System.out.println("\nInformasi Film Pendek:");  
        film3.tampilkanInfo();  
    }  
}
```

Penjelasan



TUGAS & EVALUASI

Dalam kode Java di atas, kita membuat sebuah class induk Film yang memiliki atribut umum seperti judul, sutradara, dan tahunRilis, serta method tampilkanInfo() untuk menampilkan informasi film secara umum. Kemudian, tiga class anak (Animasi, Dokumenter, dan Pendek) mewarisi class Film dan menambahkan atribut khusus yang relevan dengan masing-masing jenis film, seperti studio dan teknik animasi untuk Animasi, topik dan durasi untuk Dokumenter, serta durasi untuk Pendek. Setiap class anak mengoverride method tampilkanInfo() untuk menampilkan informasi yang lebih spesifik sesuai dengan jenis filmnya.

Output

```
User (workspace) age (4432100)
Informasi Film Animasi:
Judul: Up
Sutradara: Pete Docter
Tahun Rilis: 2009
Studio: Pixar
Teknik Animasi: 3D CGI

Informasi Film Dokumenter:
Judul: Our Planet
Sutradara: David Attenborough
Tahun Rilis: 2019
Topik: Alam dan Lingkungan
Durasi: 8 episode x 50 menit

Informasi Film Pendek:
Judul: The Neighbor's Window
Sutradara: Marshall Curry
Tahun Rilis: 2019
Durasi: 20 menit
```



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Apa yang dimaksud dengan Polimorfisme, Polimorfisme Statis dan Polimorfisme Dinamis?

Jawaban

- Polimorfisme adalah prinsip bahwa suatu class dapat memiliki banyak bentuk metode yang berbeda walaupun memiliki nama yang sama.
- Polimorfisme Statis adalah Kompiler waktu Metode ini biasanya di panggil Ketika kompilasi objek, implementasi Polimorfisme ini adalah OverLoading
- Polimorfisme Dinamis adalah metode ini biasanya di sebut juga Runtime Polymorphism, hal ini karena metode ini di panggil selama objek di jalankan. Implementasinya adalah Overriding

Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Dalam sebuah Restoran terdapat 3 jenis pegawai, yaitu **Pegawai Tetap**, **Pegawai Part-Time**, dan **Pegawai Magang**. Ketiga pegawai tersebut memiliki kriteria gaji yang berbeda, untuk pegawai tetap gajinya berasal dari gaji pokok, tunjangan khusus, dan bonus, pegawai part-time berasal dari jumlah jam kerja dan tarif per jam nya, sedangkan pegawai magang hanya berasal dari gaji tetap tanpa tunjangan dan bonus. Berdasarkan kasus tersebut Implementasikan metode **Overloading** dalam bahasa Java!

Jawaban

Ketik jawaban disini ...

Source Code

```
class Pegawai {  
  
    public double hitungGaji(double gajiPokok, double  
tunjanganKhusus, double bonus) {  
        return gajiPokok + tunjanganKhusus + bonus;  
    }  
  
    public double hitungGaji(double jamKerja, double  
tarifPerJam) {  
        return jamKerja * tarifPerJam;  
    }  
  
    public double hitungGaji(double gajiTetap) {  
        return gajiTetap;  
    }  
}  
  
public class Restoran_07651 {  
    public static void main(String[] args) {  
        Pegawai pegawai = new Pegawai();  
    }  
}
```




TUGAS & EVALUASI

```
        double gajiPegawaiTetap = pegawai.hitungGaji(5000000,
2000000, 1000000);
        System.out.println("Gaji Pegawai Tetap: " +
gajiPegawaiTetap);

        double gajiPegawaiPartTime = pegawai.hitungGaji(40,
50000);
        System.out.println("Gaji Pegawai Part-Time: " +
gajiPegawaiPartTime);

        double gajiPegawaiMagang = pegawai.hitungGaji(2000000);
        System.out.println("Gaji Pegawai Magang: " +
gajiPegawaiMagang);
    }
}
```

Penjelasan

Kode ini mendefinisikan kelas `Pegawai` yang memiliki tiga metode `hitungGaji`, dimana masing-masing metode menghitung gaji berdasarkan tipe pegawai yang berbeda. Metode pertama menghitung gaji untuk pegawai tetap, menerima parameter `gajiPokok`, `tunjanganKhusus`, dan `bonus`, kemudian menjumlahkan ketiganya. Metode kedua menghitung gaji pegawai part-time berdasarkan `jamKerja` dan `tarifPerJam` dengan mengalikan kedua parameter tersebut. Metode ketiga menghitung gaji pegawai magang yang hanya menerima satu parameter `gajiTetap`. Di dalam kelas `Pegawai_07651`, tiga objek gaji dihitung dengan memanggil metode `hitungGaji` yang sesuai, dan hasilnya ditampilkan ke layar: gaji pegawai tetap, gaji pegawai part-time, dan gaji pegawai magang.

Output

```
PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 3\Bab 6> & C:\V
\Code\User\workspaceStorage\44521c5d318b25b84c130772b80e2c70\redhat.java\jdt_ws\Bab
Gaji Pegawai Tetap: 8000000.0
Gaji Pegawai Part-Time: 2000000.0
Gaji Pegawai Magang: 2000000.0
PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 3\Bab 6>
```



TUGAS & EVALUASI

Soal Tugas & Evaluasi

Dalam Sistem Informasi Akademik terdapat beberapa jenis pengguna, yaitu **Admin**, **Dosen**, dan **Mahasiswa**. Ketika ingin menggunakan sistem informasi tersebut, pastinya pengguna perlu melakukan proses login. Setiap jenis pengguna memiliki proses login yang berbeda sesuai dengan peran mereka dalam system. Berdasarkan kasus tersebut Implementasikan metode **Overriding** dalam bahasa Java!

Jawaban

Source Code

```
class Pengguna {
    protected String Username;
    protected String Password;

    public Pengguna(String username, String password){
        this.Username = username;
        this.Password = password;
    }

    public void login() {
        System.out.println("Login sebagai pengguna umum");
    }
}

class Mahasiswa extends Pengguna{
    public Mahasiswa(String username, String password){
        super(username, password);
    }

    @Override
    public void login() {
```



TUGAS & EVALUASI

```
        System.out.println("Login sebagai Mahasiswa dengan Nama
" + Username);

    }
}

class Dosen extends Pengguna{
    public Dosen (String username, String password){
        super(username, password);
    }

    @Override
    public void login() {
        System.out.println("Login sebagai Dosen dengan nama " +
Username);
    }
}

class Admin extends Pengguna{
    public Admin (String username, String password){
        super(username, password);
    }

    @Override
    public void login() {
        System.out.println(" Login sebgai Admin dengan nama " +
Username);
    }
}

public class LoginUser_07651 {
    public static void main(String[] args) {
        Pengguna Mahasiswa = new Mahasiswa("A_07651", "a07651");
        Pengguna Dosen = new Dosen("A_07651", "a07651");
        Pengguna Admin = new Admin("A_07651", "a07651");

        Mahasiswa.login();
    }
}
```



TUGAS & EVALUASI

```
        Admin.login();  
        Dosen.login();  
    }  
  
}
```

Penjelasan

Pada kode di atas, atribut username dan password pada superclass Pengguna menggunakan modifier protected, sehingga dapat diakses langsung oleh subclass seperti Admin, Dosen, dan Mahasiswa tanpa perlu getter atau setter. Dengan protected, atribut tetap aman dari akses luar tetapi tetap tersedia untuk subclass, yang memungkinkan setiap subclass melakukan kustomisasi pada proses login sesuai perannya. Contohnya, metode login() di setiap subclass mengimplementasikan cara login yang berbeda sesuai dengan tipe pengguna (Admin, Dosen, atau Mahasiswa), sambil tetap menjaga aksesibilitas terbatas untuk keamanan.

Output

```
● PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 3\B  
w' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\agung\AppData  
Login sebagai Mahasiswa dengan Nama A_07651  
Login sebagai Admin dengan nama A_07651  
Login sebagai Dosen dengan nama A_07651  
○ PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 3\B
```