



TUGAS & EVALUASI

Soal Tugas & Evaluasi

1. Apa yang dimaksud dengan inheritance serta sebutkan macam macam inheritance

Jawaban

Inheritance atau pewaeisan pada object oriented programming merupakan konsep yang mewariskan / mrnurunkan suatu karakteristik class ke class lain

- a. Single inheritance
- b. Hierarchical inheritance



TUGAS & EVALUASI

Soal Tugas & Evaluasi

2. Apa yg dimaksud dari generalisasi dan spesialisasi dalam inheritance?

Buatlah ilustrasi dari kedua konsep tersebut

Jawaban

Generalisasi adalah proses membuat kelas yang lebih umum (superclass) dari beberapa kelas yang lebih spesifik (subclass). Dalam generalisasi, kita mengambil elemen-elemen yang umum atau serupa dari beberapa kelas turunan dan menggabungkannya menjadi satu kelas induk yang lebih umum.

- **Contoh:** Misalnya ada kelas Mobil dan SepedaMotor, yang keduanya memiliki atribut dan metode yang serupa, seperti mesin dan berjalan(). Maka, kita bisa membuat kelas induk yang lebih umum, seperti Kendaraan, yang menyatukan elemen-elemen umum tersebut.

Spesialisasi adalah kebalikan dari generalisasi. Dalam spesialisasi, kita membuat kelas yang lebih khusus (subclass) dari kelas yang lebih umum (superclass). Proses ini melibatkan penambahan atau pengubahan perilaku untuk menyesuaikan dengan kebutuhan spesifik dari subclass tersebut.

Contoh: Dari kelas Kendaraan, kita bisa membuat kelas-kelas turunan seperti Mobil dan SepedaMotor, yang memiliki atribut atau metode khusus, misalnya AC untuk Mobil dan roda untuk SepedaMotor.

Source Code



TUGAS & EVALUASI

Soal Tugas & Evaluasi

3. Terdapat tiga jenis film : animasi documenter, dan, pendek. Setiap jenis film memiliki atribut umum serta atribut khusus untuk masing – masing jenis film. Buatlah kelas dan atribut yang sesuai serta buatlah method untuk menampilkan informasi tentang masin-masing film

PETUNJUK :

Class film sebagai class induk:

Class animasi, documenter, dan pendek class anak

Source Code

```
class Film {  
  
    String judul;  
    int durasi; // dalam menit  
    int tahunRilis;  
    String sutradara;  
  
    public Film(String judul, int durasi, int tahunRilis,  
String sutradara) {  
        this.judul = judul;  
        this.durasi = durasi;  
        this.tahunRilis = tahunRilis;  
        this.sutradara = sutradara;  
    }  
  
    public void tampilkanInfo() {  
        System.out.println("Judul: " + judul);  
        System.out.println("Durasi: " + durasi + " menit");  
        System.out.println("Tahun Rilis: " + tahunRilis);  
        System.out.println("Sutradara: " + sutradara);  
    }  
}  
  
class Animasi extends Film {  
    private String studioAnimasi;  
  
    public Animasi(String judul, int durasi, int tahunRilis,  
String sutradara, String studioAnimasi) {  
        super(judul, durasi, tahunRilis, sutradara);  
        this.studioAnimasi = studioAnimasi;  
    }  
}
```



TUGAS & EVALUASI

```
@Override
public void tampilkanInfo() {
    super.tampilkanInfo(); // Memanggil method
    tampilkanInfo() dari kelas induk
    System.out.println("Studio Animasi: " +
studioAnimasi);
}
}

class Dokumenter extends Film {
    private String topikDokumenter;

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

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Topik Dokumenter: " +
topikDokumenter);
    }
}

class Pendek extends Film {
    private int jumlahPenghargaan;

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

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Jumlah Penghargaan: " +
jumlahPenghargaan);
    }
}

public class Main {
    public static void main(String[] args) {
```



TUGAS & EVALUASI

```
Animasi animasi = new Animasi("Upin & Ipin", 90, 2016,
"Adam", "Les O Amigos Production");
Dokumenter dokumenter = new Dokumenter("Penghijauan
Dunia", 60, 2020, "Nina", "Lingkungan Hidup");
Pendek pendek = new Pendek("Singkat Tapi Penuh Makna",
15, 2023, "Rafi", 5);

System.out.println("Informasi Film Animasi:");
animasi.tampilkanInfo();
System.out.println("\nInformasi Film Dokumenter:");
dokumenter.tampilkanInfo();
System.out.println("\nInformasi Film Pendek:");
pendek.tampilkanInfo();
}
}
```

Penjelasan

Membuat class induk yaitu film dengan attribute judul, durasi, tahun, sutradara, lalu membuat konstruktor nya juga dan membuat fungsi tampilan, setelah itu membuat class anak animas dengan atribut studio animasi lalu menggunakan konstruktornya dengan parameter class induk, dan ditambahkan atribut dari class anak, lalu memanggil konstruktor class induk dengan menggunakan super dengan parameter class induk

Membuat class dokkumenter dengan atribut topik documenter dan membuat kosntruktor dengan parameter class induk dan tambahkan dengan atribut dari documenter tersebut, dan membuat mamnggil konstruktor dengan cara menegtikkan super dan ditambahkan parameter class induk

Membuat class pendek dengan atribut jumlah penghargaan dan membuat kosntruktor dengan parameter class induk dan tambahkan dengan atribut dari pendek tersebut, dan membuat mamnggil konstruktor dengan cara menegtikkan super dan ditambahkan parameter class induk

Output



TUGAS & EVALUASI

Informasi Film Animasi:

Judul: Upin & Ipin

Durasi: 90 menit

Tahun Rilis: 2016

Sutradara: Adam

Studio Animasi: Les O Amigos Production

Informasi Film Dokumenter:

Judul: Penghijauan Dunia

Durasi: 60 menit

Tahun Rilis: 2020

Sutradara: Nina

Topik Dokumenter: Lingkungan Hidup

Informasi Film Pendek:

Judul: Singkat Tapi Penuh Makna

Durasi: 15 menit

Tahun Rilis: 2023

Sutradara: Rafi

Jumlah Penghargaan: 5



TUGAS & EVALUASI

Soal Tugas & Evaluasi

4. Apa yang dimaksud dengan polimorfisme statis dan polimorfisme dinamis?

Jawaban

1. Polimorfisme statis adalah polimorfisme ini juga dikenal dengan sebagai compile time polymorphism karena method ini dipanggil selama kompilasi objek, implementasi polimorfisme ini adalah overloading
2. Polimorfisme dinamis, polimorfisme ini disebut juga runtime polymorphism, hal ini karena method ini dipanggil selama objek dijalankan, implementasi dari polimorfisme ini yaitu overriding



TUGAS & EVALUASI

Soal Tugas & Evaluasi

5. Dalam sebuah restoran terdapat 4 jenis pegawai, yaitu pegawai tetap, pegawai part time, 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 jamnya, sedangkan pegawai magang hanya berasal dari gaji tetap tanpa tunjangan dan bonus, berdasarkan kasus tersebut implementasikan metode overloading dalam Bahasa Java

Source Code

```
class Pegawai {  
    public double hitungGaji(double gajiPokok, double  
tunjangan, double bonus) {  
        return gajiPokok + tunjangan + bonus; // untuk Pegawai  
Tetap  
    }  
  
    public double hitungGaji(double jamKerja, double  
tarifPerJam) {  
        return jamKerja * tarifPerJam; // untuk Pegawai  
PartTime  
    }  
  
    public double hitungGaji(double gajiTetap) {  
        return gajiTetap; // untuk Pegawai Magang  
    }  
}  
  
class PegawaiTetap extends Pegawai {  
    private double gajiPokok;  
    private double tunjangan;  
    private double bonus;  
  
    public PegawaiTetap(double gajiPokok, double tunjangan,  
double bonus) {  
        this.gajiPokok = gajiPokok;  
        this.tunjangan = tunjangan;  
        this.bonus = bonus;  
    }  
}
```




TUGAS & EVALUASI

```
@Override
public double hitungGaji(double gajiPokok, double
tunjangan, double bonus) {
    return super.hitungGaji(gajiPokok, tunjangan, bonus);
}

public void tampilkanGaji() {
    double gaji = hitungGaji(gajiPokok, tunjangan, bonus);
    System.out.println("Gaji Pegawai Tetap: " + gaji);
}
}

class PegawaiPartTime extends Pegawai {
    private double jamKerja;
    private double tarifPerJam;

    public PegawaiPartTime(double jamKerja, double
tarifPerJam) {
        this.jamKerja = jamKerja;
        this.tarifPerJam = tarifPerJam;
    }

    @Override
    public double hitungGaji(double jamKerja, double
tarifPerJam) {
        return super.hitungGaji(jamKerja, tarifPerJam);
    }

    public void tampilkanGaji() {
        double gaji = hitungGaji(jamKerja, tarifPerJam);
        System.out.println("Gaji Pegawai Part-Time: " + gaji);
    }
}

class PegawaiMagang extends Pegawai {
    private double gajiTetap;

    public PegawaiMagang(double gajiTetap) {
        this.gajiTetap = gajiTetap;
    }

    @Override
    public double hitungGaji(double gajiTetap) {
        return super.hitungGaji(gajiTetap);
    }

    public void tampilkanGaji() {
        double gaji = hitungGaji(gajiTetap);
        System.out.println("Gaji Pegawai Magang: " + gaji);
    }
}
```



TUGAS & EVALUASI

```
}  
  
public class Main {  
    public static void main(String[] args) {  
  
        PegawaiTetap pegawaiTetap = new PegawaiTetap(5000000,  
1000000, 500000);  
        PegawaiPartTime pegawaiPartTime = new  
PegawaiPartTime(120, 20000); // 120 jam kerja, 20000 per jam  
        PegawaiMagang pegawaiMagang = new  
PegawaiMagang(3000000);  
  
        pegawaiTetap.tampilkanGaji();  
        pegawaiPartTime.tampilkanGaji();  
        pegawaiMagang.tampilkanGaji();  
    }  
}
```

Penjelasan

Membuat overloadin class pegawai

Membuat public double hitung gaji dengan parameter gaji pokok, tunjangan, bonus ini untuk pegawai tetap

Membuat public double hitung gaji dengan parameter jam kerja dan tarif per jam ini untuk pegawai parttime

Membuat public double hitung gaji dengan parameter gaji tetap untuk pegawai magang

Lalu membuat class pegawai tetap dengan attribute gaji pokok, tunjangan, bonus lalu membuat konstruktornya dan di override hitung gajinya, lalu membuat void tampilan,

Ini berlaku juga untuk part time dan magang, lanjut ke main membuat objek dan di inisialisasi lalu di panggil

Output



TUGAS & EVALUASI

```
D:\ngoding\bin\java.exe "-javaagent:D:\ngoding\bin\javaagent.jar"
Gaji Pegawai Tetap: 6500000.0
Gaji Pegawai Part-Time: 2400000.0
Gaji Pegawai Magang: 3000000.0

Process finished with exit code 0
```



TUGAS & EVALUASI

Soal Tugas & Evaluasi

6. 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 sistem. Berdasarkan kasus tersebut implementasikan overriding dalam Bahasa Java

Source Code

```
import java.util.Scanner;

class SistemInformasiAkademik {

    static class Pengguna {
        String username;
        String password;

        public void login() {
            System.out.println("Login untuk pengguna sistem informasi akadem.");
        }
    }

    static class Admin extends Pengguna {

        @Override
        public void login() {
            Scanner input = new Scanner(System.in);

            System.out.print("Masukkan username admin: ");
            username = input.nextLine();
            System.out.print("Masukkan password admin: ");
            password = input.nextLine();

            if (username.equals("adam") && password.equals("adam123")) {
                System.out.println("Login berhasil sebagai Admin!");
            } else {
                System.out.println("Username atau password salah untuk Admin.");
            }
        }
    }

    static class Dosen extends Pengguna {

        @Override
```



TUGAS & EVALUASI

```
public void login() {
    Scanner input = new Scanner(System.in);

    // Data login dosen
    System.out.print("Masukkan username dosen: ");
    username = input.nextLine();
    System.out.print("Masukkan password dosen: ");
    password = input.nextLine();

    if (username.equals("nina") &&
password.equals("nina123")) {
        System.out.println("Login berhasil sebagai
Dosen!");
    } else {
        System.out.println("Username atau password
salah untuk Dosen.");
    }
}

static class Mahasiswa extends Pengguna {

    @Override
    public void login() {
        Scanner input = new Scanner(System.in);

        System.out.print("Masukkan username mahasiswa: ");
        username = input.nextLine();
        System.out.print("Masukkan password mahasiswa: ");
        password = input.nextLine();

        if (username.equals("risal") &&
password.equals("risal123")) {
            System.out.println("Login berhasil sebagai
Mahasiswa!");
        } else {
            System.out.println("Username atau password
salah untuk Mahasiswa.");
        }
    }
}

public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    while (true) {

System.out.println("\n=====
=====");
        System.out.println("Selamat datang di Sistem
Informasi Akademik!");

System.out.println("=====
===");
        System.out.println("Pilih jenis pengguna untuk
```



TUGAS & EVALUASI

```
login:");
        System.out.println("1. Admin");
        System.out.println("2. Dosen");
        System.out.println("3. Mahasiswa");
        System.out.println("4. Keluar");
        System.out.print("Pilihan Anda (1/2/3/4): ");
        int pilihan = input.nextInt();
        input.nextLine();

        if (pilihan == 4) {
            System.out.println("Terima kasih telah
menggunakan sistem informasi akademik.");
            break;
        }

        Pengguna pengguna = null;

        switch (pilihan) {
            case 1:
                pengguna = new Admin();
                break;
            case 2:
                pengguna = new Dosen();
                break;
            case 3:
                pengguna = new Mahasiswa();
                break;
            default:
                System.out.println("Pilihan tidak valid!
Silakan pilih 1, 2, 3, atau 4.");
                continue;
        }

        pengguna.login();
    }
}
```

Penjelasan

Membuat public class system informasi akademik

Dan membuat lagi class pengguna dengan attrtibus string username, dan string password. Lalu buat static class extends pengguna dengan memanggil void login yang ada di class pengguna dan membuat objek scanner input. Dan mencetak username admin dan masukkan password admin , lalu membuat if else dengan kondisi username = adam dan password = adam123 maka mencetak login berhasil



TUGAS & EVALUASI

sebagai admin, jika salah mencetak username atau password salah, dosen dan mahasiswa sama seperti admin

Lanjut ke main membuat objek input, lalu membuat pengulangan while. Ketika true mencetak kalimat dan menginputkan, lalu membuat if jika pilihan == 4 maka mencetak kalimat tersebut

Membuat deklarasi pengguna yang bernilai null

Lalu membuat switch case untuk memilih 1-3. Lalu yang terakhir memanggil pengguna dan login

Output

```
Selamat datang di Sistem Informasi Akademik!
Pilih jenis pengguna untuk login:
1. Admin
2. Dosen
3. Mahasiswa
4. Keluar
Pilihan Anda (1/2/3/4): 1
Masukkan username admin: Adam
Masukkan password admin: adam123
Username atau password salah untuk Admin.

Selamat datang di Sistem Informasi Akademik!
Pilih jenis pengguna untuk login:
1. Admin
2. Dosen
3. Mahasiswa
4. Keluar
Pilihan Anda (1/2/3/4): |
```