

# LAPORAN PRAKTIKUM



## **PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X**

**Nama** : Adam Fadhila Insaan  
**NPM** : 06.2023.1.07720  
**Pertemuan** : 3





### MEKANISME PRAKTIKUM

- 1) Buat Sebuah **Project Java Baru** pada IntelliJ IDEA dengan nama Project:  
**"PertemuanX\_NPM AKHIR"**  
Ganti "X" menjadi Pertemuan yang sedang berlangsung.
  - 2) Pada saat Praktikum, Jawabanlah Soal Pertanyaan yang memiliki Label **WAJIB** terlebih dahulu Pada Lembar "**Laporan Praktikum**".
  - 3) Segala Bentuk **Soal yang memiliki Jawaban** berupa **Kode Program**, maka kode program tersebut harus disimpan pada **File java Project** yang telah dibuat.
  - 4) Setiap **File Java** yang dibuat harus mencantumkan Pertanyaan pada bagian atas (baris pertama)
  - 5) Simpan **File Laporan Praktikum** yang berupa **DOCX** menjadi **FILE PDF** kemudian ubah nama file PDF menjadi:  
**"PertemuanX\_NPM AKHIR.pdf"**
  - 6) Upload File **Laporan Praktikum [PDF]** pada form yang sudah disediakan.
- 

### TUGAS PRAKTIKUM

1. Jelaskan perbedaan antara **Override** dan **Overload**. [Wajib]
2. Jelaskan fungsi dari kata kunci super dalam konsep inheritance. [Wajib]
3. Buatlah class **User** dengan atribut username, email, dan saldo. Kemudian buatlah sub-class **Driver** dan **Customer** yang mewarisi class **User**.  
  
Pada class **Driver**, tambahkan atribut kendaraan, dan nopol.  
  
Pada class **Customer**, tambahkan atribut bergabung (boolean) untuk menandai apakah pelanggan adalah pelanggan baru atau bukan.  
  
**Override** metode **tampilkanInfo()** di setiap sub-class untuk menampilkan informasi tambahan sesuai dengan class-nya.  
  
Buat objek dari class **Driver** dan **Customer**, lalu tampilkan informasi mereka menggunakan **tampilkanInfo()**. [Wajib]



**SOAL PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBJEK**  
**PERIODE IX**  
Laboratorium Rekayasa Perangkat Lunak, ITATS

---

4. Dari soal nomor 3, tambahkan method **hitungTarif()** yang menerapkan **Overload** untuk menghitung tarif dalam berbagai situasi berikut:
- Berdasarkan jarak dan tarif per kilometer.
  - Dengan tambahan parameter untuk potongan harga.
  - Dengan tambahan parameter jumlah penumpang.
5. Terdapat dua jenis tenaga medis di rumah sakit: **Dokter** dan **Perawat**. Setiap tenaga medis memiliki atribut umum serta atribut khusus untuk masing-masing jenis tenaga medis. Buatlah kelas dan atribut yang sesuai dalam dua package berbeda serta buatlah method untuk menampilkan informasi tentang masing-masing tenaga medis!

PETUNJUK:

- Package rumahsakit untuk class **TenagaMedis** sebagai Class Induk.
- Package jenispegawai untuk class **Dokter** dan **Perawat** sebagai Class Anak.



# **TUGAS PRAKTIKUM**

## **Soal Praktikum**

1. Jelaskan perbedaan antara Override dan Overload. [Wajib]

## **Jawaban**

Override adalah situasi dimana situasi objek dapat meneruskan sifat keobjek lain tetapi dapat berperilaku berbeda asalkan kedua objek tersebut memiliki hubungan

Overload adalah situasi dimana beberapa method sekaligus dapat memiliki nama yang sama tetapi fungsi berbeda



# **TUGAS PRAKTIKUM**

## **Soal Praktikum**

2. Jelaskan fungsi dari kata kunci super dalam konsep inheritance. [Wajib]

## **Jawaban**

Pada inheritance, kelas di ambil mewariskan method dan atribut kepada kelas lain



# TUGAS PRAKTIKUM

## Soal Praktikum

3. Buatlah class User dengan atribut username, email, dan saldo. Kemudian buatlah sub-class Driver dan Customer yang mewarisi class User.

Pada class Driver, tambahkan atribut kendaraan, dan nopol.

Pada class Customer, tambahkan atribut bergabung (boolean) untuk menandai apakah pelanggan adalah pelanggan baru atau bukan.

Override metode tampilkanInfo() di setiap sub-class untuk menampilkan informasi tambahan sesuai dengan class-nya.

Buat objek dari class Driver dan Customer, lalu tampilkan informasi mereka menggunakan tampilkanInfo(). [Wajib]

## Source Code

```
class User{
    String username;
    String email;
    int saldo;

    public User(String username, String email, int saldo){
        this.username = username;
        this.email = email;
        this.saldo = saldo;
    }

    public void tampilkanInfo(){
        System.out.println("Username: " + username);
        System.out.println("Email: " + email);
        System.out.println("Saldo: " + saldo);
    }
}

class Driver extends User {
    String kendaraan;
    int nopol;

    public Driver(String username, String email, int saldo,
String kendaraan, int nopol) {
        super (username, email, saldo);
        this.kendaraan = kendaraan;
        this.nopol = nopol;
    }
}
```



# TUGAS PRAKTIKUM

```
}

@Override
public void tampilkanInfo() {
    super.tampilkanInfo();
    System.out.println("Kendaraan: " + kendaraan);
    System.out.println("Nopol: " + nopol);
}
}

class Customer extends User{
    boolean bergabung;
    public Customer(String username, String email, int saldo,
boolean bergabung){
        super(username, email, saldo);
        this.bergabung = bergabung;
    }

    public void tampilkanInfo(){
        super.tampilkanInfo();
        String status = bergabung ? "Pelanggan Baru" :
"Pelanggan Lama";
        System.out.println("Status: " + status);
    }
}

public class Main {
    public static void main(String[] args) {
        User user1 = new User( "Adam",
"Adaminsaan24@gmail.com", 20000);
        Driver user2 = new Driver("Rafi",
"Rafilalala@gmail.com", 20000, "Avanza", 23741);
        Customer user3 = new Customer(user1.username,
user1.email, 10000, true);
        Customer user4 = new Customer("Nina",
"ninaCantik@gmail.com", 230000, false);
        user1.tampilkanInfo();
        user2.tampilkanInfo();
        user3.tampilkanInfo();
        user4.tampilkanInfo();
    }
}
```

## Penjelasan

Yang pertama kita buat class bernama user dengan atribut nama, email, saldo lalu dilanjutkan dengan konstruktor nya. Setelah itu membuat class driver dan di extends dari class user lalu menambahkan atribut kendaraan dan, nopol. Setelah itu membuat class lagi dengan nama customer di extends dari class user, lalu



# TUGAS PRAKTIKUM

ditambahkan atribut bergabung untuk menanyakan apakah pelanggan baru atau lama. Lalu membuat objek dan menginisialisasi dan ditampilkan

## Output

```
Username: Adam
Email: Adaminsaan24@gmail.com
Saldo: 20000
Username: Rafi
Email: Rafilalala@gmail.com
Saldo: 20000
Kendaraan: Avanza
Nopol: 23741
Username: Adam
Email: Adaminsaan24@gmail.com
Saldo: 10000
Status: Pelanggan Baru
Username: Nina
Email: ninaCantik@gmail.com
Saldo: 230000
Status: Pelanggan Lama
```





# TUGAS PRAKTIKUM

## Soal Praktikum

4. Dari soal nomor 3, tambahkan method `hitungTarif()` yang menerapkan

Overload untuk menghitung tarif dalam berbagai situasi berikut:

- Berdasarkan jarak dan tarif per kilometer.
- Dengan tambahan parameter untuk potongan harga.
- Dengan tambahan parameter jumlah penumpang.

## Source Code

```
class User {
    String username;
    String email;
    int saldo;

    public User(String username, String email, int saldo) {
        this.username = username;
        this.email = email;
        this.saldo = saldo;
    }

    public void tampilkanInfo() {
        System.out.println("Username: " + username);
        System.out.println("Email: " + email);
        System.out.println("Saldo: " + saldo);
    }
}

class Driver extends User {
    String kendaraan;
    int nopol;

    public Driver(String username, String email, int saldo,
String kendaraan, int nopol) {
        super(username, email, saldo);
        this.kendaraan = kendaraan;
        this.nopol = nopol;
    }

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Kendaraan: " + kendaraan);
        System.out.println("Nopol: " + nopol);
    }
}
```



# TUGAS PRAKTIKUM

```
}

class Customer extends User {
    boolean bergabung;

    public Customer(String username, String email, int saldo,
boolean bergabung) {
        super(username, email, saldo);
        this.bergabung = bergabung;
    }

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        String status = bergabung ? "Pelanggan Baru" :
"Pelanggan Lama";
        System.out.println("Status: " + status);
    }

    // Menghitung tarif berdasarkan jarak dan tarif per
kilometer
    public int hitungTarif(int jarak, int tarifPerKm) {
        return jarak * tarifPerKm;
    }

    // Menghitung tarif dengan tambahan parameter potongan
harga
    public int hitungTarif(int jarak, int tarifPerKm, int
potongan) {
        int total = jarak * tarifPerKm;
        return total - potongan;
    }

    // Menghitung tarif dengan tambahan parameter jumlah
penumpang
    public int hitungTarif(int jarak, int tarifPerKm, int
potongan, int jumlahPenumpang) {
        int total = jarak * tarifPerKm;
        int tarifDenganPenumpang = total * jumlahPenumpang;
        return tarifDenganPenumpang - potongan;
    }
}

public class Main {
    public static void main(String[] args) {
        User user1 = new User("Adam",
"Adaminsaan24@gmail.com", 20000);
        Driver user2 = new Driver("Rafi",
"Rafilalala@gmail.com", 20000, "Avanza", 23741);
        Customer user3 = new Customer(user1.username,
user1.email, 10000, true);
        Customer user4 = new Customer("Nina",
"ninaCantik@gmail.com", 230000, false);

        user1.tampilkanInfo();
    }
}
```



# TUGAS PRAKTIKUM

```
user2.tampilkanInfo();
user3.tampilkanInfo();
user4.tampilkanInfo();

int tarif1 = user3.hitungTarif(10, 2000);
int tarif2 = user3.hitungTarif(10, 2000, 5000);
int tarif3 = user3.hitungTarif(10, 2000, 5000, 3);

System.out.println(" ");

System.out.println("Tarif (jarak dan tarif per km): "
+ tarif1);
System.out.println("Tarif (dengan potongan): " +
tarif2);
System.out.println("Tarif (dengan potongan dan jumlah
penumpang): " + tarif3);

}
}
```

## Penjelasan

Kurang lebih sama seperti jawaban nomor 3 tetapi disini ditambahkan method `hitungTarif()` yang menerapkan

Overload untuk menghitung tarif dalam berbagai situasi yaitu

- Berdasarkan jarak dan tarif per kilometer.
- Dengan tambahan parameter untuk potongan harga.
- Dengan tambahan parameter jumlah penumpang.

## Output



# TUGAS PRAKTIKUM

Email: Adaminsaan24@gmail.com

Saldo: 20000

Username: Rafi

Email: Rafilalala@gmail.com

Saldo: 20000

Kendaraan: Avanza

Nopol: 23741

Username: Adam

Email: Adaminsaan24@gmail.com

Saldo: 10000

Status: Pelanggan Baru

Username: Nina

Email: ninaCantik@gmail.com

Saldo: 230000

Status: Pelanggan Lama

Tarif (jarak dan tarif per km): 20000

Tarif (dengan potongan): 15000

Tarif (dengan potongan dan jumlah penumpang): 55000



# TUGAS PRAKTIKUM

## Soal Praktikum

Terdapat dua jenis tenaga medis di rumah sakit: Dokter dan Perawat. Setiap tenaga medis memiliki atribut umum serta atribut khusus untuk masing-masing jenis tenaga medis. Buatlah kelas dan atribut yang sesuai dalam dua package berbeda serta buatlah method untuk menampilkan informasi tentang masing-masing tenaga medis!

PETUNJUK:

- Package rumahsakit untuk class TenagaMedis sebagai Class Induk.
- Package jenispegawai untuk class Dokter dan Perawat sebagai Class Anak.

## Jawaban

Ketik jawaban disini ...

## Source Code

Package rumahSakit

```
package rumahSakit;

import jenispegawai.Dokter;
import jenispegawai.Perawat;

public class Main {
    public static void main(String[] args) {
        Dokter dokter1 = new Dokter("Dr. Adam", 45, "D123",
        "Ortopedi", 20);
        Perawat perawat1 = new Perawat("Nina", 30, "P456",
        "Malam", "ICU");

        System.out.println("Informasi Dokter:");
        dokter1.tampilkanInfo();

        System.out.println("\nInformasi Perawat:");
        perawat1.tampilkanInfo();
    }
}
```



# TUGAS PRAKTIKUM

```
package rumahSakit;

public class TenagaMedis {
    protected String nama;
    protected int umur;
    protected String id;

    public TenagaMedis(String nama, int umur, String id) {
        this.nama = nama;
        this.umur = umur;
        this.id = id;
    }

    public void tampilkanInfo() {
        System.out.println("Nama: " + nama);
        System.out.println("Umur: " + umur);
        System.out.println("ID: " + id);
    }
}
```

```
package jenisPegawai
```

```
import rumahSakit.TenagaMedis;

public class Perawat extends TenagaMedis {
    private String shift;
    private String unitKerja;

    public Perawat(String nama, int umur, String id, String
shift, String unitKerja) {
        super(nama, umur, id);
        this.shift = shift;
        this.unitKerja = unitKerja;
    }

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Shift: " + shift);
        System.out.println("Unit Kerja: " + unitKerja);
    }
}
```

```
package jenispegawai;

import rumahSakit.TenagaMedis;

public class Dokter extends TenagaMedis {
    private String spesialisasi;
    private int tahunPengalaman;

    public Dokter(String nama, int umur, String id, String
spesialisasi, int tahunPengalaman) {
        super(nama, umur, id);
    }
}
```



# TUGAS PRAKTIKUM

```
this.spesialisasi = spesialisasi;  
this.tahunPengalaman = tahunPengalaman;  
}  
  
@Override  
public void tampilkanInfo() {  
    super.tampilkanInfo();  
    System.out.println("Spesialisasi: " + spesialisasi);  
    System.out.println("Tahun Pengalaman: " +  
tahunPengalaman);  
}
```

## Penjelasan

Tulis Penjelasan disini ...

## Output

Masukan screenshot output disini



# TUGAS PRAKTIKUM

## Soal Praktikum

Ketik soal disini ...

## Jawaban

Ketik jawaban disini ...

## Source Code

Tulis kode program dikotak ini...

1 kotak dan 1 Penjelasan untuk 1 Class

## Penjelasan

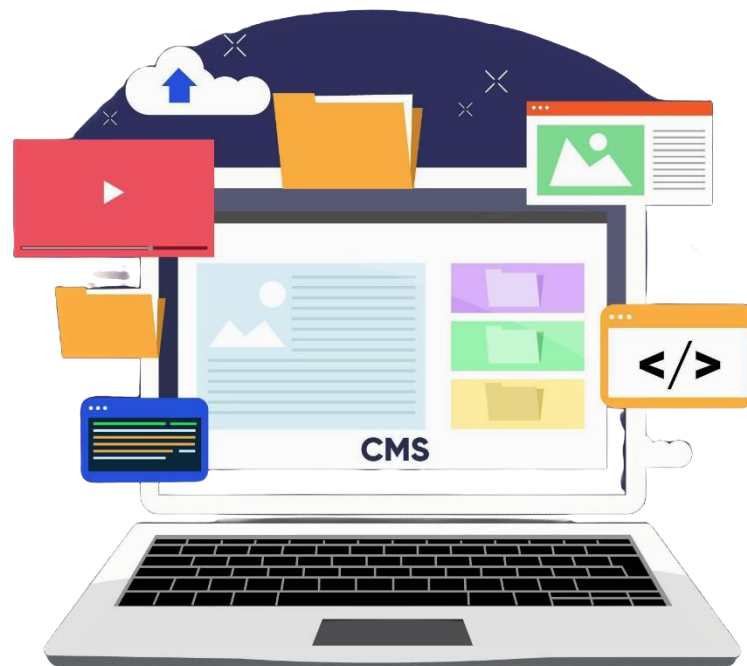
Tulis Penjelasan disini ...

## Output

Masukan screenshot output disini



# TUGAS DAN EVALUASI



## PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X

Nama : Adam Fadhila Insaan  
NPM : 06.2023.1.07720  
Modul : 6





# **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

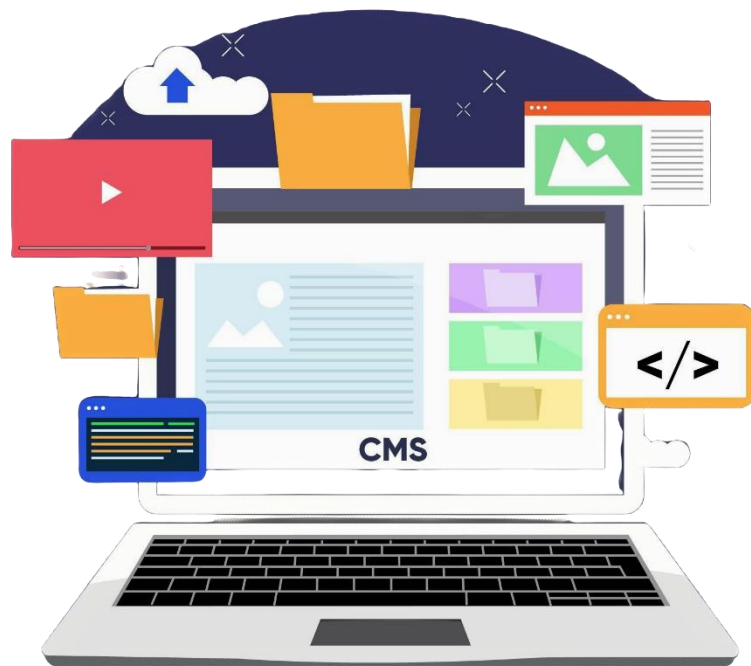
## Output

```
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): |
```



# TUGAS DAN EVALUASI



## PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X

Nama : Adam Fadhila Insaan  
NPM : 06.2023.1.07720  
Modul : 7





# **TUGAS & EVALUASI**

## **Soal Tugas & Evaluasi**

1. Apa yang dimaksud package

## **Jawaban**

Package adalah suatu folde yang mengelompokkan suatu class, interface, dan file-file lainnya ke dalam satu folder



# **TUGAS & EVALUASI**

## **Soal Tugas & Evaluasi**

2. Sebutkan keuntungan dan syarat-syarat dalam menerapkan konsep package

## **Jawaban**

Keuntungan :

- a. Mengelompokkan class dan file java lainnya yang membuat aplikasi kita lebih mudah dipelihara
- b. Mencegah terjadinya konflik penamaan yang sama pada suatu class

Syarat:

- a. Pada Bahasa java penamaan package disarankan menggunakan huruf kecil
- b. Penamaan package menggambarkan tujuan dari class dibungkusnya
- c. Penamaan tidak boleh sama(unique) antar package

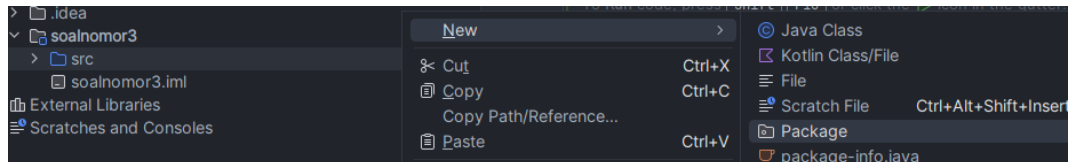


# TUGAS & EVALUASI

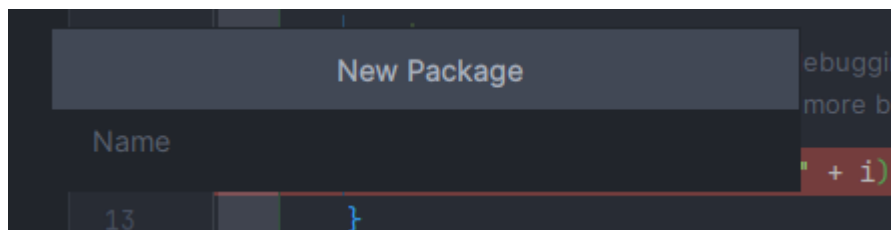
## Soal Tugas & Evaluasi

3. Sebutkan dan berikan screenshot langkah ;angkah dalam membuat sebuah package baru

## Jawaban



Klik kanan dari 'src' modul awal yang kalian buat, lalu ketik new dan pilih yang package



Lalu kasih nama package yang kalian inginkan



# TUGAS & EVALUASI

## Soal Tugas & Evaluasi

4. Buatlah 3 class dalam package yang anda buat dari soal no 3 lalu akses class yang ada pada package tersebut dengan menggunakan metode:
  1. Metode import all
  2. Metode import each
  3. Metode fulley qualified name

## Jawaban

Ketik jawaban disini ...

## Source Code

Main

```
1. //IMPORT ALL
import soalnomor3.*;

//IMPORT EACH
//import soalnomor3.myshop_a;
//import soalnomor3.myshop_b;
//import soalnomor3.myshop_c;

//
public class main {
    public static void main(String[] args) {
        myshopA shop_a = new myshopA();
        shop_a.displayA();

        myshopB shop_b = new myshopB();
        shop_b.displayB();

        myshopC shop_c = new myshopC();
        shop_c.displayC();

        //METODE IMPORT FULLEY QUALIFIED NAME
        //        soalnomor3.myshopA shop1 = new
        soalnomor3.myshopA();
        //        shop1.displayA();
        //
        //        soalnomor3.myshopB shop2 = new
        soalnomor3.myshopB();
        //        shop2.displayB();
        //
        //        soalnomor3.myshopA shop3 = new
        soalnomor3.myshopA();
        //        shop3.displayA();
```



# TUGAS & EVALUASI

```
}  
}
```

## Class A

```
package soalnomor3;  
  
public class myshopA {  
    public void displayA() {  
        System.out.println("Welcome to my shop");  
    }  
}
```

## Class B

```
package soalnomor3;  
  
public class myshopB {  
    public void displayB() {  
        System.out.println("What do you want?, you can choose  
anything");  
    }  
}
```

## Class C

```
package soalnomor3;  
  
public class myshopC {  
    public void displayC() {  
        System.out.println("I think you must try golda");  
    }  
}
```

## //import all

```
//IMPORT ALL  
import soalnomor3.*;  
  
//IMPORT EACH  
//import soalnomor3.myshop_a;  
//import soalnomor3.myshop_b;  
//import soalnomor3.myshop_c;  
  
//  
public class main {  
    public static void main(String[] args) {  
        myshopA shop_a = new myshopA();  
        shop_a.displayA();  
  
        myshopB shop_b = new myshopB();  
        shop_b.displayB();  
  
        myshopC shop_c = new myshopC();  
        shop_c.displayC();  
    }  
}
```



# TUGAS & EVALUASI

//import each

```
//IMPORT EACH
import soalnomor3.myshopA;
import soalnomor3.myshopB;
import soalnomor3.myshopC;

//
public class main {
    public static void main(String[] args) {
        myshopA shop_a = new myshopA();
        shop_a.displayA();

        myshopB shop_b = new myshopB();
        shop_b.displayB();

        myshopC shop_c = new myshopC();
        shop_c.displayC();
    }
}
```

//metode import fulley qualified name

```
public class main {
    public static void main(String[] args) {
        //      myshopA shop_a = new myshopA();
        //      shop_a.displayA();
        //
        //      myshopB shop_b = new myshopB();
        //      shop_b.displayB();
        //
        //      myshopC shop_c = new myshopC();
        //      shop_c.displayC();

        //      METODE IMPORT FULLEY QUALIFIED NAME
        soalnomor3.myshopA shop1 = new
soalnomor3.myshopA();
        shop1.displayA();

        soalnomor3.myshopB shop2 = new
soalnomor3.myshopB();
        shop2.displayB();

        soalnomor3.myshopA shop3 = new
soalnomor3.myshopA();
        shop3.displayA();
    }
}
```

## Penjelasan



# TUGAS & EVALUASI

Pertama membuat package main, lalu membuat package soalno3 yang berisi classA, classB, classC, lalu diimport di file main

## Output

//import all

```
D:\ngoding\bin\java.exe "-javaagent:D:\ngodin
Welcome to my shop
What do you want?, you can choose anything
I think you must try golda
```

//import each

```
D:\ngoding\bin\java.exe "-javaagent:D:\ngodin
Welcome to my shop
What do you want?, you can choose anything
I think you must try golda
```

//metode import fulley qualified name

```
D:\ngoding\bin\java.exe "-javaagent:D:\ngodin
Welcome to my shop
What do you want?, you can choose anything
I think you must try golda
```