

LAPORAN PRAKTIKUM



PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X

Nama : Michael Agung
NPM : 06.2023.1.07651
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. Perusahaan pengiriman "Paket Mabur" mengelola pengiriman paket domestik dan internasional. Setiap paket memerlukan penanganan berbeda, terutama terkait biaya dan regulasi.

Buatlah class **Paket** dengan atribut resi, berat, dan tujuan, lalu method **tampilkanInfoPaket()**. Kemudian buatlah sub-class **PaketDomestik** dan **PaketInternasional** yang mewarisi class **Paket**.

Pada class **PaketDomestik**, tambahkan atribut wilayah, dan opsiPengiriman(seperti same day, hemat, reguler, cargo).

Pada class **PaketInternasional**, tambahkan atribut negara, dan biayaCukai.

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

Buat objek dari class **PaketDomestik** dan **PaketInternasional**, 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 **hitungBiayaPengiriman()** di class induk yang menerapkan *Overload* untuk menghitung biaya pengiriman dalam berbagai situasi berikut:
- Berdasarkan berat paket dan tarif dasar.
 - Dengan tambahan parameter untuk biaya asuransi.
 - Dengan tambahan parameter untuk potongan ongkos kirim.
5. Terdapat dua jenis karyawan di sebuah pabrik: **Karyawan Administrasi** dan **Karyawan Produksi**. Setiap karyawan memiliki atribut umum serta atribut khusus sesuai peran mereka di pabrik. Buatlah kelas dan atribut yang sesuai dalam dua package berbeda serta buatlah method untuk menampilkan informasi tentang masing-masing karyawan!

PETUNJUK:

- Package karyawan untuk class **KaryawanUmum** sebagai Class Induk.
- Package jenispegawai untuk class **KaryawanAdministrasi** dan **KaryawanProduksi** sebagai Class Anak.



TUGAS PRAKTIKUM

Soal Praktikum

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

Jawaban

method overriding terjadi saat sebuah subclass mendefinisikan ulang atau mengganti implementasi method yang diwarisi dari superclass-nya. Method ini memiliki nama, jumlah parameter, dan tipe parameter yang sama dengan method yang ada di superclass. Tujuannya adalah untuk memberikan perilaku spesifik di subclass yang mungkin berbeda dengan method yang ada di superclass.

Sedangkan method overloading adalah ketika sebuah kelas memiliki beberapa method dengan nama yang sama, tetapi dengan parameter yang berbeda, baik dari segi jumlah, tipe, atau urutan parameter. Overloading memungkinkan kita menggunakan nama method yang sama untuk berbagai kebutuhan yang berbeda, tergantung dari parameter yang diberikan.

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 PRAKTIKUM

Soal Praktikum

2. Jelaskan fungsi dari kata kunci super dalam konsep inheritance.

Jawaban

-inheritance merupakan suatu konsep yang mewariskan/menurunkan suatu karakteristik class ke class yang lain

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 PRAKTIKUM

Soal Praktikum

3. Perusahaan pengiriman "Paket Mabur" mengelola pengiriman paket domestik dan internasional. Setiap paket memerlukan penanganan berbeda, terutama

terkait biaya dan regulasi. Buatlah class Paket dengan atribut resi, berat, dan tujuan, lalu method tampilkanInfoPaket(). Kemudian buatlah sub-class PaketDomestik dan PaketInternasional yang mewarisi class Paket. Pada class PaketDomestik, tambahkan atribut wilayah, dan opsiPengiriman(seperti same day, hemat, reguler, cargo). Pada class PaketInternasional, tambahkan atribut negara, dan biayaCukai.

Override metode tampilkanInfoPaket() di setiap sub-class untuk menampilkan informasi tambahan sesuai dengan class-nya. Buat objek dari class PaketDomestik dan PaketInternasional, lalu tampilkan informasi mereka menggunakan tampilkanInfo().

Jawaban

Ketik jawaban disini ...

Source Code

```
class Paket {
    protected String resi;
    protected double berat;
    protected String tujuan;

    public Paket(String resi, double berat, String tujuan) {
        this.resi = resi;
        this.berat = berat;
        this.tujuan = tujuan;
    }

    public void tampilkanInfoPaket() {
        System.out.println("Resi: " + resi);
        System.out.println("Berat: " + berat + " kg");
        System.out.println("Tujuan: " + tujuan);
    }
}
```



TUGAS PRAKTIKUM

```
}  
}  
  
class PaketDomestik extends Paket {  
    private String wilayah;  
    private String opsiPengiriman;  
  
    public PaketDomestik(String resi, double berat, String  
tujuan, String wilayah, String opsiPengiriman) {  
        super(resi, berat, tujuan);  
        this.wilayah = wilayah;  
        this.opsiPengiriman = opsiPengiriman;  
    }  
  
    @Override  
    public void tampilkanInfoPaket() {  
        super.tampilkanInfoPaket();  
        System.out.println("Wilayah: " + wilayah);  
        System.out.println("Opsi Pengiriman: " +  
opsiPengiriman);  
    }  
}  
  
class PaketInternasional extends Paket {  
    private String negara;  
    private double biayaCukai;  
  
    public PaketInternasional(String resi, double berat, String  
tujuan, String negara, double biayaCukai) {  
        super(resi, berat, tujuan);  
        this.negara = negara;  
        this.biayaCukai = biayaCukai;  
    }  
  
    @Override  
    public void tampilkanInfoPaket() {  
        super.tampilkanInfoPaket();  
        System.out.println("Negara: " + negara);  
        System.out.println("Biaya Cukai: " + biayaCukai);  
    }  
}
```



TUGAS PRAKTIKUM

```
    }  
}  
  
public class Pertemuan3_07651 {  
    public static void main(String[] args) {  
        PaketDomestik paketDomestik = new  
PaketDomestik("RESI001", 2.5, "Jakarta", "Jawa", "Same Day");  
        System.out.println("Informasi Paket Domestik:");  
        paketDomestik.tampilkanInfoPaket();  
  
        PaketInternasional paketInternasional = new  
PaketInternasional("RESI002", 5.0, "Tokyo", "Jepang", 250.0);  
        System.out.println("\nInformasi Paket Internasional:");  
        paketInternasional.tampilkanInfoPaket();  
    }  
}
```

Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini



TUGAS PRAKTIKUM

Soal Praktikum

4. Dari soal nomor 3, tambahkan method `hitungBiayaPengiriman()` di class induk yang menerapkan Overload untuk menghitung biaya pengiriman dalam berbagai situasi berikut:

- Berdasarkan berat paket dan tarif dasar.
- Dengan tambahan parameter untuk biaya asuransi.
- Dengan tambahan parameter untuk potongan ongkos kirim.

Jawaban

Ketik jawaban disini ...

Source Code

```
class Paket {  
    protected String resi;  
    protected double berat;  
    protected String tujuan;  
    private static final double TARIF_DASAR = 5000.0;  
  
    public Paket(String resi, double berat, String tujuan) {  
        this.resi = resi;  
        this.berat = berat;  
        this.tujuan = tujuan;  
    }  
  
    public double hitungBiayaPengiriman() {  
        return berat * TARIF_DASAR;  
    }  
  
    public double hitungBiayaPengiriman(double biayaAsuransi) {  
        return (berat * TARIF_DASAR) + biayaAsuransi;  
    }  
  
    public double hitungBiayaPengiriman(double biayaAsuransi,  
    double potongan) {
```



TUGAS PRAKTIKUM

```
        return (berat * TARIF_DASAR) + biayaAsuransi - potongan;
    }

    public void tampilkanInfoPaket() {
        System.out.println("Resi: " + resi);
        System.out.println("Berat: " + berat + " kg");
        System.out.println("Tujuan: " + tujuan);
    }
}

class PaketDomestik extends Paket {
    private String wilayah;
    private String opsiPengiriman;

    public PaketDomestik(String resi, double berat, String
tujuan, String wilayah, String opsiPengiriman) {
        super(resi, berat, tujuan);
        this.wilayah = wilayah;
        this.opsiPengiriman = opsiPengiriman;
    }

    @Override
    public void tampilkanInfoPaket() {
        super.tampilkanInfoPaket();
        System.out.println("Wilayah: " + wilayah);
        System.out.println("Opsi Pengiriman: " +
opsiPengiriman);
    }
}

class PaketInternasional extends Paket {
    private String negara;
    private double biayaCukai;

    public PaketInternasional(String resi, double berat, String
tujuan, String negara, double biayaCukai) {
        super(resi, berat, tujuan);
        this.negara = negara;
        this.biayaCukai = biayaCukai;
    }
}
```



TUGAS PRAKTIKUM

```
}

@Override
public void tampilkanInfoPaket() {
    super.tampilkanInfoPaket();
    System.out.println("Negara: " + negara);
    System.out.println("Biaya Cukai: " + biayaCukai);
}

@Override
public double hitungBiayaPengiriman() {
    return super.hitungBiayaPengiriman() + biayaCukai;
}
}

public class Paket_07651 {
    public static void main(String[] args) {
        PaketDomestik paketDomestik = new
PaketDomestik("RESI001", 2.5, "Jakarta", "Jawa", "Same Day");
        System.out.println("Informasi Paket Domestik:");
        paketDomestik.tampilkanInfoPaket();
        System.out.println("Biaya Pengiriman (tanpa asuransi
atau potongan): " + paketDomestik.hitungBiayaPengiriman());
        System.out.println("Biaya Pengiriman (dengan asuransi
5000): " + paketDomestik.hitungBiayaPengiriman(5000));
        System.out.println("Biaya Pengiriman (dengan asuransi
5000 dan potongan 2000): " +
paketDomestik.hitungBiayaPengiriman(5000, 2000));

        PaketInternasional paketInternasional = new
PaketInternasional("RESI002", 5.0, "Tokyo", "Jepang", 250.0);
        System.out.println("\nInformasi Paket Internasional:");
        paketInternasional.tampilkanInfoPaket();
        System.out.println("Biaya Pengiriman (termasuk biaya
cukai): " + paketInternasional.hitungBiayaPengiriman());
        System.out.println("Biaya Pengiriman (dengan asuransi
10000 dan potongan 5000): " +
paketInternasional.hitungBiayaPengiriman(10000, 5000));
    }
}
```



TUGAS PRAKTIKUM

Penjelasan

Program ini terdiri dari kelas Paket sebagai superclass yang menyimpan atribut dasar paket dan menyediakan metode hitungBiayaPengiriman() yang overload, memungkinkan perhitungan biaya berdasarkan berat, tambahan asuransi, atau potongan ongkos kirim. Subclass PaketDomestik menambahkan atribut wilayah dan opsiPengiriman, sedangkan PaketInternasional menambahkan atribut negara dan biayaCukai. Masing-masing subclass meng-override tampilkanInfoPaket() untuk menampilkan informasi khusus sesuai jenis pengiriman. Pada kelas Main, program membuat objek PaketDomestik dan PaketInternasional, menampilkan detail paket, serta menghitung biaya pengiriman dengan variasi parameter.

Output

Masukan screenshot output disini



TUGAS PRAKTIKUM

Soal Praktikum

5. Terdapat dua jenis karyawan di sebuah pabrik: Karyawan Administrasi dan Karyawan Produksi. Setiap karyawan memiliki atribut umum serta atribut khusus sesuai peran mereka di pabrik. Buatlah kelas dan atribut yang sesuai dalam dua package berbeda serta buatlah method untuk menampilkan informasi tentang masing-masing karyawan!

PETUNJUK: - -

Package karyawan untuk class KaryawanUmum sebagai Class Induk.

Package jenispegawai untuk class KaryawanAdministrasi dan

KaryawanProduksi sebagai Class Anak.

Jawaban

Ketik jawaban disini ...

Source Code

```
class KaryawanUmum {
    protected String nama;
    protected int id;
    protected double gaji;

    public KaryawanUmum(String nama, int id, double gaji) {
        this.nama = nama;
        this.id = id;
        this.gaji = gaji;
    }

    public void tampilkanInfoKaryawan() {
        System.out.println("Nama: " + nama);
        System.out.println("ID: " + id);
        System.out.println("Gaji: " + gaji);
    }
}
```



TUGAS PRAKTIKUM

```
}  
}  
  
class KaryawanAdministrasi extends KaryawanUmum {  
    private String departemen;  
  
    public KaryawanAdministrasi(String nama, int id, double gaji, String departemen) {  
        super(nama, id, gaji);  
        this.departemen = departemen;  
    }  
  
    @Override  
    public void tampilkanInfoKaryawan() {  
        super.tampilkanInfoKaryawan();  
        System.out.println("Departemen: " + departemen);  
    }  
}  
  
class KaryawanProduksi extends KaryawanUmum {  
    private String shiftKerja;  
  
    public KaryawanProduksi(String nama, int id, double gaji, String shiftKerja) {  
        super(nama, id, gaji);  
        this.shiftKerja = shiftKerja;  
    }  
  
    @Override  
    public void tampilkanInfoKaryawan() {  
        super.tampilkanInfoKaryawan();  
        System.out.println("Shift Kerja: " + shiftKerja);  
    }  
}  
  
public class Karyawan_07651 {  
    public static void main(String[] args) {  
        KaryawanAdministrasi admin = new  
KaryawanAdministrasi("Alice", 101, 5000000, "Keuangan");  
    }  
}
```



TUGAS PRAKTIKUM

```
System.out.println("Informasi Karyawan Administrasi:");
admin.tampilkanInfoKaryawan();

        KaryawanProduksi produksi = new KaryawanProduksi("Bob",
102, 4500000, "Shift Malam");
        System.out.println("\nInformasi Karyawan Produksi:");
        produksi.tampilkanInfoKaryawan();
    }
}
```

Penjelasan

Program ini memiliki kelas induk `KaryawanUmum` yang menyimpan atribut dasar seperti nama, id, dan gaji untuk semua karyawan di pabrik. Kelas `KaryawanAdministrasi` dan `KaryawanProduksi` adalah subclass dari `KaryawanUmum`, masing-masing menambahkan atribut khusus untuk departemen atau shift kerja. Pada Main, objek dari kedua kelas dibuat untuk menampilkan informasi lengkap sesuai tipe karyawan.

Output

```
F:\workspace\storage\7a57c0d2c9b04b0e54b2c0eb1e50d5ee\reunat.java\Jut_ws\Praktikum
Informasi Karyawan Administrasi:
Nama: Alice
ID: 101
Gaji: 5000000.0
Departemen: Keuangan

Informasi Karyawan Produksi:
Nama: Bob
ID: 102
Gaji: 4500000.0
Shift Kerja: Shift Malam
PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 3> |
```



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

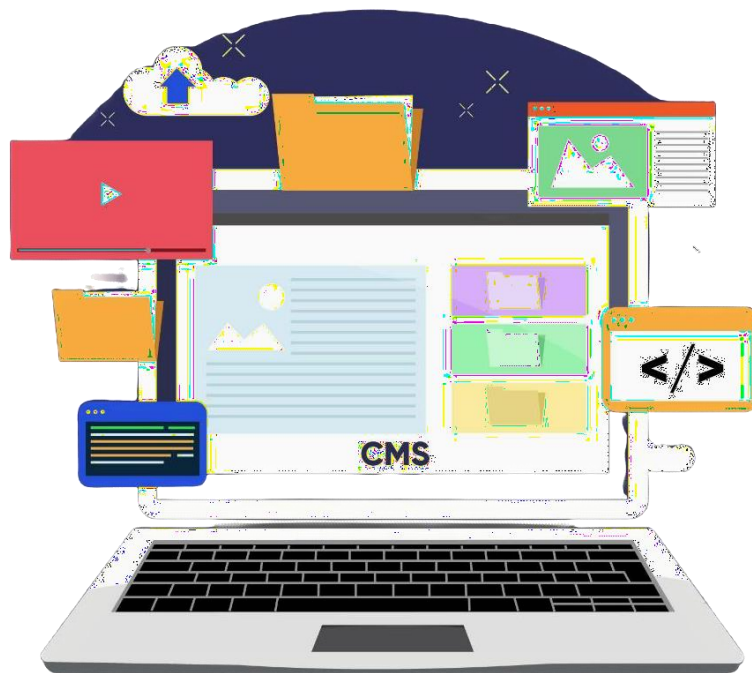
Penjelasan

Tulis Penjelasan disini ...

Output

Masukan screenshot output disini

TUGAS DAN EVALUASI



PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X

Nama : Michael Agung Y.P
NPM : 06.2023.1.07657
Modul : 6





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();  
    }  
  
}
```

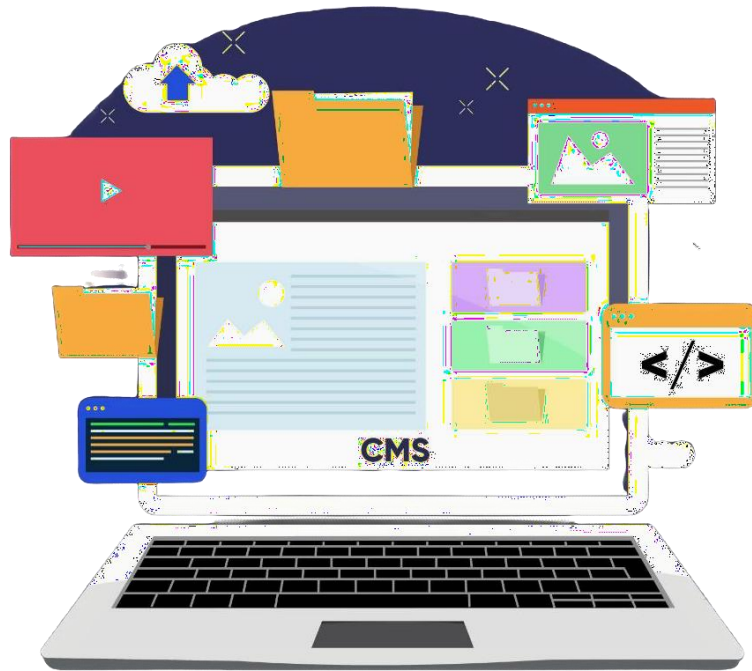
Penjelasan

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
```

TUGAS DAN EVALUASI



PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK PERIODE X

Nama : Michael Agung Y.P
NPM : 06.2023.1.07657
Modul : 7





TUGAS & EVALUASI

Soal Tugas & Evaluasi

1. Apa yang dimaksud dengan package ?

Jawaban

Package adalah sejenis folder untuk 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 :

- 1) Mengelompokkan class dan file java lainnya yang membuat aplikasi kita lebih mudah di pelihara
- 2) Mencegah terjadinya konflik penamaan yang sama pada suatu class
- 3) Mengatur hak akses kepemilikan (enkapsulasi)

Syarat – Syarata :

- 1) pada Bahasa java penamaan package disarankan menggunakan huruf kecil.
- 2) penamaan package menggambarkan tujuan dari class yang di bungkusnya.
- 3) Penamaan tidak boleh sama (unique) antar package.
- 4) Menuliskan keyword package pada bari paling atas tepat sebelum keyword import



TUGAS & EVALUASI

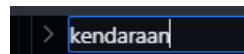
Soal Tugas & Evaluasi

3. Sebutkan dan berikan screenshot Langkah Langkah dalam membuat sebuah package baru!

Jawaban

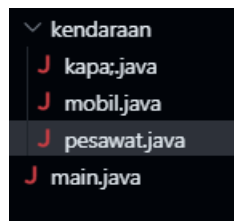
1. pertama kita membuat folder / package baru.

2. berinama packge kendaraan



3. lalu klik enter

4. buat sebuah class baru didalam kendaraan yang memiliki ketertarikan dengan nama package.





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
- metode import All
 - metode import each
 - metode full qualified name

Jawaban

Source Code

```
package kendaraan;

public class kapal {
    String nama;
    String jenis;

    public kapal(String nama, String jenis){
        this.nama = nama;
        this.jenis = jenis;
    }

    public void tampilkanKapal(){
        System.out.println("KAPAL");
        System.out.println("Nama kapal : " + nama);
        System.out.println("Jenis kapal : " + jenis);
        System.out.println("");
    }
}
```

Penjelasan

Class Kapal berada di package kendaraan.

- memiliki atribut nama dan jenis;



TUGAS & EVALUASI

- memiliki konstruktor yang menginisialisasi atribut nama dan jenis.
- memiliki method tampilkanKapal() untuk menampilkan di layar

Source Code

```
package kendaraan;

public class mobil {
    String merek;
    String bahanBakar;

    public mobil(String merek, String bahanBakar){
        this.merek = merek;
        this.bahanBakar = bahanBakar;
    }

    public void tampilkanMobil(){
        System.out.println("MOBIL");
        System.out.println("Merek Mobil : " + merek);
        System.out.println("Bahan Bakar : " + bahanBakar);
        System.out.println("");
    }
}
```

Penjelasan

Sama seperti class kapal hanya berbeda di atribut.

Source Code

```
package kendaraan;

public class pesawat {
    String nama;
    String penerbangan;

    public pesawat (String nama, String penerbangan){
        this.nama = nama;
        this.penerbangan = penerbangan;
    }
}
```



TUGAS & EVALUASI

```
}

    public void tampilkanPesawat(){
        System.out.println("PESAWAT");
        System.out.println("Nama Pesawat : " + nama);
        System.out.println("Penerbangan : " + penerbangan);
        System.out.println("");
    }
}
```

Penjelasan

Sama seperti class kapal berbeda di atribut

Source Code

```
import kendaraan.*;

public class main {
    public static void main(String[] args) {
        kapal kapal1 = new kapal("Moshi", "Kapal Laut");
        mobil mobil1 = new mobil("BMW", "Bensin");
        pesawat pesawat1 = new pesawat("Etihad Airways",
        "Jakarta - Dubai");

        kapal1.tampilkanKapal();
        mobil1.tampilkanMobil();
        pesawat1.tampilkanPesawat();
    }
}
```

Penjelasan

- Disini kita menggunakan metode import ALL dengan menggunakan import namapackage.*
- membuat objek kapal1, mobil1, dan pesawat1.



TUGAS & EVALUASI

- memanggil method `tampilkanKapal()`, `tampilkanMobil()`, dan `tampilkanPesawat` untuk menampilkan dilayar

Source Code

```
import kendaraan.kapal;
import kendaraan.mobil;
import kendaraan.pesawat;

public class main {
    public static void main(String[] args) {
        kapal kapal1 = new kapal("Moshi", "Kapal Laut");
        mobil mobil1 = new mobil("BMW", "Bensin");
        pesawat pesawat1 = new pesawat("Etihad Airways",
        "Jakarta - Dubai");

        kapal1.tampilkanKapal();
        mobil1.tampilkanMobil();
        pesawat1.tampilkanPesawat();
    }
}
```

Penjelasan

- Disini kita menggunakan metode import each dengan menggunakan `import namapackage.namafilename`.

- membuat objek `kapal1`, `mobil1`, dan `pesawat1`.

- memanggil method `tampilkanKapal()`, `tampilkanMobil()`, dan `tampilkanPesawat` untuk menampilkan dilayar

Source Code

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



TUGAS & EVALUASI

```
kendaraan.kapal kapal1 = new kendaraan.kapal("Moshi",  
"Kapal Laut");  
kendaraan.mobil mobil1 = new kendaraan.mobil("BMW",  
"Bensin");  
kendaraan.pesawat pesawat1 = new  
kendaraan.pesawat("Etihad Airways", "Jakarta - Dubai");  
  
    kapal1.tampilkanKapal();  
    mobil1.tampilkanMobil();  
    pesawat1.tampilkanPesawat();  
}  
}
```

Penjelasan

- Disini kita menggunakan metode fully qualified dengan menggunakan import namapackage.*
- membuat objek kapal1, mobil1, dan pesawat1.
- memanggil method tampilkanKapal(), tampilkanMobil(), dan tampilkanPesawat untuk menampilkan dilayar

Output

```
KAPAL  
Nama kapal : Moshi  
Jenis kapal : Kapal Laut  
  
MOBIL  
Merek Mobil : BMW  
Bahan Bakar : Bensin  
  
PESAWAT  
Nama Pesawat : Etihad Airways  
Penerbangan : Jakarta - Dubai
```