



TUGAS & EVALUASI

Soal Tugas & Evaluasi

1. Apa itu Access Modifier dan buatlah ilustrasi Batasan Akses dari setiap Access Modifier!

Jawaban

Access Modifier di dalam OOP akan menentukan apakah class lain dapat menggunakan field atau pun meminta izin untuk mengakses field tersebut yang berisi data dari suatu class

Modifier	Class	Package	Subclass	Global
Public	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Protected	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Private	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Default	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

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

2. Jelaskan perbedaan antara Association, Aggregation, dan Composition dalam hubungan antar class. Berikan contoh kasus untuk setiap jenis relasi tersebut.

Jawaban

- Association adalah hubungan umum antara dua kelas yang independen satu sama lain, tanpa kepemilikan. Kedua kelas dapat eksis secara mandiri. Contoh kasus Hubungan antara kelas Teacher dan Student. Seorang Teacher mengajar banyak Student, dan seorang Student dapat diajar oleh banyak Teacher. Meski ada hubungan, Teacher dan Student bisa ada tanpa satu sama lain.

- Aggregation adalah hubungan "has-a" yang menunjukkan kepemilikan lemah antara kelas. Kelas yang satu memiliki kelas lain, tetapi kelas yang dimiliki masih dapat eksis tanpa kelas yang memiliki. Contoh Kasus Hubungan antara kelas Library dan Book. Sebuah Library memiliki banyak Book, tetapi Book masih bisa ada meski tidak ada di dalam Library.

- Composition adalah hubungan "has-a" yang kuat antara kelas. Jika kelas yang memiliki dihancurkan, maka kelas yang dimiliki juga ikut dihancurkan. Ini menunjukkan kepemilikan penuh. Contoh kasus Hubungan antara kelas House dan Room. Sebuah House memiliki Room, dan jika House dihapus, maka Room-nya juga ikut terhapus.

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

3. Buatlah sebuah class AkunBank yang berisi atribut data- data yang ada di akun seorang nasabah bank dengan menerapkan konsep Encapsulation, kemudian analisis modifier apa saja yang dipakai pada setiap atribut dan tentukan method Getter dan Setter-nya!

Jawaban

Ketik jawaban disini ...

Source Code

```
public class AkunBank {  
  
    private String namaNasabah;  
    private String nomorRekening;  
    private double saldo;  
  
    public AkunBank(String namaNasabah, String nomorRekening,  
double saldo) {  
        this.namaNasabah = namaNasabah;  
        this.nomorRekening = nomorRekening;  
        this.saldo = saldo;  
    }  
  
    public String getNamaNasabah() {  
        return namaNasabah;  
    }  
  
    public void setNamaNasabah(String namaNasabah) {  
        this.namaNasabah = namaNasabah;  
    }  
}
```



TUGAS & EVALUASI

```
public String getNomorRekening() {
    return nomorRekening;
}

public void setNomorRekening(String nomorRekening) {
    this.nomorRekening = nomorRekening;
}

public double getSaldo() {
    return saldo;
}

public void setSaldo(double saldo) {
    this.saldo = saldo;
}

public void tampilkanInformasi() {
    System.out.println("Nama Nasabah: " + namaNasabah);
    System.out.println("Nomor Rekening: " + nomorRekening);
    System.out.println("Saldo: " + saldo);
}

public static void main(String[] args) {
    AkunBank akun = new AkunBank("Budi Santoso",
    "1234567890", 5000000.0);

    akun.tampilkanInformasi();

    akun.setNamaNasabah("Budi Setiawan");

    akun.setSaldo(6000000.0);

    System.out.println("\nSetelah perubahan:");
    akun.tampilkanInformasi();
}
}
```

Penjelasan



TUGAS & EVALUASI

AkunBank menerapkan konsep encapsulation dengan menyembunyikan atributnya menggunakan modifier private, sehingga akses langsung dari luar class tidak dimungkinkan. Atribut yang ada, seperti namaNasabah, nomorRekening, dan saldo, hanya dapat diakses melalui method getter dan setter yang disediakan. Dengan cara ini, kita menjaga integritas data dan mencegah perubahan yang tidak diinginkan. Class ini juga memiliki method untuk menampilkan informasi akun dan contoh penggunaan di dalam method main, yang menunjukkan bagaimana objek AkunBank diinisialisasi, diubah, dan informasi yang ditampilkan ke layar.

Output

```
Nama Nasabah: Budi Santoso  
Nomor Rekening: 1234567890  
Saldo: 5000000.0  
  
Setelah perubahan:  
Nama Nasabah: Budi Setiawan  
Nomor Rekening: 1234567890  
Saldo: 6000000.0
```



TUGAS & EVALUASI

Soal Tugas & Evaluasi

4. Buat dua class, Mahasiswa dan MataKuliah, yang menerapkan relasi Asosiasi Tidak Berarah. Implementasikan method yang memungkinkan seorang mahasiswa untuk mendaftar ke mata kuliah dan mata kuliah menerima pendaftaran mahasiswa.

Jawaban

Ketik jawaban disini ...

Source Code

```
import java.util.ArrayList;
import java.util.List;

class Mahasiswa {
    private String nama;
    private String nim;
    private List<MataKuliah> mataKuliahList;

    public Mahasiswa(String nama, String nim) {
        this.nama = nama;
        this.nim = nim;
        this.mataKuliahList = new ArrayList<>();
    }

    public String getNama() {
        return nama;
    }

    public String getNim() {
        return nim;
    }

    public void mendaftarMataKuliah(MataKuliah mataKuliah) {
        mataKuliah.tambahMahasiswa(this);
        mataKuliahList.add(mataKuliah);
    }
}
```



TUGAS & EVALUASI

```
        public void tampilkanMataKuliah() {
            System.out.println("Mata Kuliah yang diambil oleh " +
nama + ":");
            for (MataKuliah mk : mataKuliahList) {
                System.out.println("- " + mk.getNamaMataKuliah());
            }
        }
    }

class MataKuliah {
    private String namaMataKuliah;
    private List<Mahasiswa> mahasiswaList;

    public MataKuliah(String namaMataKuliah) {
        this.namaMataKuliah = namaMataKuliah;
        this.mahasiswaList = new ArrayList<>();
    }

    public String getNamaMataKuliah() {
        return namaMataKuliah;
    }

    public void tambahMahasiswa(Mahasiswa mahasiswa) {
        if (!mahasiswaList.contains(mahasiswa)) {
            mahasiswaList.add(mahasiswa);
        }
    }

    public void tampilkanMahasiswa() {
        System.out.println("Mahasiswa yang terdaftar di " +
namaMataKuliah + ":");
        for (Mahasiswa mhs : mahasiswaList) {
            System.out.println("- " + mhs.getNama());
        }
    }
}

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



TUGAS & EVALUASI

```
Mahasiswa mhs1 = new Mahasiswa("Alice", "101");
Mahasiswa mhs2 = new Mahasiswa("Bob", "102");

MataKuliah mk1 = new MataKuliah("Pemrograman Dasar");
MataKuliah mk2 = new MataKuliah("Struktur Data");

mhs1.mendaftarMataKuliah(mk1);
mhs1.mendaftarMataKuliah(mk2);
mhs2.mendaftarMataKuliah(mk1);

mhs1.tampilkanMataKuliah();
mhs2.tampilkanMataKuliah();

mk1.tampilkanMahasiswa();
mk2.tampilkanMahasiswa();
}
}
```

Penjelasan

Class Mahasiswa dan MataKuliah menerapkan relasi asosiasi tidak berarah, di mana mahasiswa dapat mendaftar ke mata kuliah dan mata kuliah dapat mencatat pendaftaran mahasiswa. Class Mahasiswa memiliki method untuk mendaftar ke mata kuliah, yang juga menambahkan mahasiswa ke dalam daftar mata kuliah tersebut. Sebaliknya, class MataKuliah menyimpan daftar mahasiswa yang terdaftar dan menyediakan method untuk menampilkan daftar tersebut. Dalam contoh penggunaan, kita menunjukkan bagaimana mahasiswa mendaftar ke berbagai mata kuliah dan menampilkan informasi tersebut.

Output

```
Coding\Code\user\workspace\storage\14c51f9490519ef37a702770ac25c3c2\runat.java\JUC_ws\bab_5_7\10e8369\win - Kampo
● Mata Kuliah yang diambil oleh Alice:
- Pemrograman Dasar
- Struktur Data
Mata Kuliah yang diambil oleh Bob:
- Pemrograman Dasar
Mahasiswa yang terdaftar di Pemrograman Dasar:
- Alice
- Bob
Mahasiswa yang terdaftar di Struktur Data:
- Alice
PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 2 java\Evaluasi 4 & 5\Bab 5> []
```




TUGAS & EVALUASI

Soal Tugas & Evaluasi

5. Buatlah class Mobil dan class Mesin yang menerapkan relasi Aggregation. Dalam program tersebut, Mobil memiliki sebuah method pasangMesin() yang menghubungkan class Mesin dengan Mobil sebelum mesin dapat dinyalakan.

Jawaban

Ketik jawaban disini ...

Source Code

```
class Mesin {
    private String tipeMesin;

    public Mesin(String tipeMesin) {
        this.tipeMesin = tipeMesin;
    }

    public String getTipeMesin() {
        return tipeMesin;
    }

    public void nyalakan() {
        System.out.println("Mesin " + tipeMesin + "
dinyalakan.");
    }
}

class Mobil {
    private String merek;
    private Mesin mesin;

    public Mobil(String merek) {
        this.merek = merek;
    }
}
```



TUGAS & EVALUASI

```
        public String getMerek() {
            return merek;
        }

        public void pasangMesin(Mesin mesin) {
            this.mesin = mesin;
            System.out.println("Mesin " + mesin.getTipeMesin() + "
dipasang pada mobil " + merek + ".");
        }

        public void nyalakanMesin() {
            if (mesin != null) {
                mesin.nyalakan();
                System.out.println("Mesin belum dipasang!");
            }
        }
    }

    public class Mobil1 {
        public static void main(String[] args) {
            Mesin mesin1 = new Mesin("V8");
            Mesin mesin2 = new Mesin("V6");

            Mobil mobil1 = new Mobil("Toyota");
            Mobil mobil2 = new Mobil("Honda");

            mobil1.pasangMesin(mesin1);
            mobil2.pasangMesin(mesin2);

            mobil1.nyalakanMesin();
            mobil2.nyalakanMesin();
        }
    }
```

Penjelasan

Class Mobil dan Mesin menerapkan relasi aggregation, di mana Mobil memiliki referensi ke objek Mesin. Class Mobil memiliki method pasangMesin() untuk



TUGAS & EVALUASI

menghubungkan mesin ke mobil dan method nyalakanMesin() untuk menyalakan mesin jika telah dipasang. Di dalam class Main, kita menciptakan objek Mesin dan Mobil, memasang mesin pada mobil, dan menyalakan mesin tersebut, yang menunjukkan interaksi antara kedua class tanpa ketergantungan yang ketat.

Output

```
Data\Roaming\Code\User\workspaceStorage\f4c51f949b5f9ef37a70277dac25c3c2\redhat.java
Mesin V8 dipasang pada mobil Toyota.
Mesin V6 dipasang pada mobil Honda.
Mesin V8 dinyalakan.
Mesin belum dipasang!
Mesin V6 dinyalakan.
Mesin belum dipasang!
PS C:\Users\agung\Pictures\Kuliah\Semester 3\Coding Java\Praktikum 2 java\Evaluasi 4
```