

Nama: Arditya Adjie Rosandi

NIM : 20230801274

Pemrograman Berorientasi Objek



1. Program Mobil

```
java Salin kode  
  
class Mobil {  
    String aktifitas;  
    String warna;  
    int kecepatan;  
}
```

Mendefinisikan kelas **Mobil** dengan tiga atribut:


- aktifitas: sebuah String yang digunakan untuk menyimpan status aktifitas mobil, misalnya "PARKIR".
- warna: sebuah String untuk menyimpan warna mobil.
- kecepatan: sebuah int untuk menyimpan kecepatan mobil.

```
java Salin kode  
  
public Mobil (String aktifitas, String warna, int kecepatan){  
    this.aktifitas = aktifitas;  
    this.warna = warna;  
    this.kecepatan = kecepatan;  
}
```

Mendefinisikan konstruktor dengan parameter aktifitas, warna, dan kecepatan. Konstruktor ini digunakan untuk menginisialisasi objek Mobil dengan nilai-nilai tertentu ketika objek dibuat.

- this.aktifitas = aktifitas; -> Menggunakan kata kunci this untuk membedakan antara atribut kelas dan parameter metode. Atribut aktifitas dari objek diset dengan nilai dari parameter aktifitas.
- this.warna = warna; -> Atribut warna diset dengan nilai dari parameter warna.
- this.kecepatan = kecepatan; -> Atribut kecepatan diset dengan nilai dari parameter kecepatan.


java

 Salin kode

```
public Mobil() {  
}
```

Mendefinisikan konstruktor tanpa parameter. Ini adalah konstruktor default yang memungkinkan pembuatan objek Mobil tanpa memberikan nilai awal untuk atribut.

java


 Salin kode

```
void cekKecepatan() {  
    if (kecepatan == 0)  
        aktifitas = "PARKIR";  
}
```

Mendefinisikan metode cekKecepatan yang memeriksa nilai kecepatan:

- Jika kecepatan bernilai 0, maka aktifitas diatur menjadi "PARKIR".
- Metode ini digunakan untuk memastikan bahwa jika mobil tidak bergerak (kecepatan = 0), maka status aktifitas mobil adalah "PARKIR".

java

 Salin kode

```
void cetakAtribut() {  
    System.out.println("AKTIFITAS = " + aktifitas);  
    System.out.println("WARNA = " + warna);  
    System.out.println("KECEPATAN = " + kecepatan);  
}
```

Mendefinisikan metode cetakAtribut untuk mencetak nilai dari aktifitas, warna, dan kecepatan ke layar.

- `System.out.println("AKTIFITAS = " + aktifitas);` -> Mencetak nilai aktifitas mobil.
- `System.out.println("WARNA = " + warna);` -> Mencetak nilai warna mobil.
- `System.out.println("KECEPATAN = " + kecepatan);` -> Mencetak nilai kecepatan mobil.

java

Salin kode

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

Mendefinisikan kelas **ClassMobil** yang memiliki metode utama main. Metode main adalah titik masuk program, yang akan dieksekusi ketika program dijalankan.

java

Salin kode

```
Mobil mobilku = new Mobil();
```

Membuat objek mobilku dari kelas Mobil menggunakan konstruktor tanpa parameter. Ini berarti mobilku dibuat dengan atribut aktifitas, warna, dan kecepatan yang belum diinisialisasi (atau null dan 0 sebagai nilai default).

java

Salin kode

```
mobilku.kecepatan = 0;  
mobilku.warna = "MERAH";
```

Mengatur nilai atribut kecepatan dan warna untuk objek mobilku:

- kecepatan diatur ke 0.
- warna diatur ke "MERAH".


java

Salin kode

```
mobilku.cekKecepatan();
```

Memanggil metode cekKecepatan() pada objek mobilku. Karena kecepatan mobil diatur ke 0, maka metode ini akan mengubah aktifitas mobil menjadi "PARKIR".


java

 Salin kode

```
        mobilku.cetakAtribut();  
    }  
}
```

Memanggil metode cetakAtribut() pada objek mobilku, yang mencetak nilai dari aktifitas, warna, dan kecepatan ke layar.

makefile


 Salin kode

```
AKTIFITAS = PARKIR  
WARNA = MERAH  
KECEPATAN = 0
```

Karena kecepatan diatur ke 0, metode cekKecepatan() mengubah aktifitas menjadi "PARKIR". Metode cetakAtribut() kemudian mencetak nilai-nilai dari atribut mobilku.

2. Program BintangFilm


java

 Salin kode

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

Mendefinisikan kelas **ClassBintangFilm** yang berisi metode utama main. Metode main adalah titik masuk program, yang akan dieksekusi saat program dijalankan.

java


 Salin kode

```
BintangFilm siA = new BintangFilm("BUDI", true);  
BintangFilm siB = new BintangFilm("HANI", false);
```

Membuat dua objek BintangFilm:

- siA dengan nama "BUDI" dan pria bernilai true, artinya siA adalah seorang pria.
- siB dengan nama "HANI" dan pria bernilai false, artinya siB adalah seorang wanita.

java


 Salin kode

```
        System.out.println("siA => " + siA.perolehNama() + ", " + siA.perolehJenisKelamin()  
        System.out.println("siB => " + siB.perolehNama() + ", " + siB.perolehJenisKelamin()  
    }  
}
```

Menampilkan atribut dari objek siA dan siB menggunakan metode perolehNama dan perolehJenisKelamin.

- siA.perolehNama() mengembalikan nama "BUDI", dan siA.perolehJenisKelamin() mengembalikan "PRIA".
- siB.perolehNama() mengembalikan nama "HANI", dan siB.perolehJenisKelamin() mengembalikan "WANITA".

makefile

 Salin kode


```
siA => BUDI, PRIA  
siB => HANI, WANITA
```

Pada **output** ini:

- siA memiliki nama "BUDI" dan jenis kelamin "PRIA".
- siB memiliki nama "HANI" dan jenis kelamin "WANITA".

3. Program Calculator


java

 Salin kode

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

Kelas **ProgramCalculator** berisi metode utama main, yang merupakan titik masuk program.

java

 Salin kode

```
    Calculator calc = new Calculator();
```

Membuat objek Calculator bernama **calc**.

java

Salin kode

```
calc.isiOperan1(10.0);  
calc.isiOperan2(4.0);
```

Mengatur operan1 dengan nilai 10.0 dan operan2 dengan nilai 4.0 menggunakan metode isiOperan1 dan isiOperan2.

java

Salin kode

```
System.out.println("Operan1 + Operan2 = " + calc.tambah());  
System.out.println("Operan1 - Operan2 = " + calc.kurang());  
System.out.println("Operan1 * Operan2 = " + calc.kali());  
System.out.println("Operan1 / Operan2 = " + calc.bagi());  
System.out.println("Operan1 ^ Operan2 = " + calc.pangkat());  
}  
}
```

Mencetak hasil operasi matematika antara operan1 dan operan2 menggunakan metode yang telah dibuat (tambah, kurang, kali, bagi, dan pangkat).

Salin kode

```
Operan1 + Operan2 = 14.0  
Operan1 - Operan2 = 6.0  
Operan1 * Operan2 = 40.0  
Operan1 / Operan2 = 2.5  
Operan1 ^ Operan2 = 10000.0
```

Output ini menunjukkan hasil operasi matematika yang dilakukan antara operan1 dan operan2.

4. Program BangunDatar

java

Salin kode

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

Kelas ProgramBangunDatar berisi metode main, yang merupakan titik masuk program. Program ini akan meminta pengguna untuk memilih jenis bangun datar yang luasnya akan dihitung.

java

Salin kode

```
Scanner input = new Scanner(System.in);
```

Membuat objek Scanner untuk menerima input dari pengguna.

java

Salin kode

```
System.out.println("PILIH BANGUN DATAR UNTUK MENGHITUNG LUAS");
System.out.println("\n1. LUAS PERSEGI");
System.out.println("2. LUAS PERSEGI PANJANG");
System.out.println("3. LUAS LINGKARAN");
System.out.println("4. LUAS SEGITIGA");
System.out.print("\nMASUKKAN PILIHAN : ");
int pilihan = input.nextInt();
```

Menampilkan daftar pilihan bangun datar dan meminta input pengguna untuk memilih salah satu.

java

Salin kode

```
BangunDatar(double sisi) {
    this.sisi = sisi;
}
```


java

Salin kode

```
double luasPersegi() {
    return sisi * sisi;
}
```


Konstruktor ini digunakan untuk inisialisasi persegi dengan mengisi atribut sisi dengan nilai yang diberikan sebagai parameter.

java

 Salin kode

```
BangunDatar(double panjang, double lebar) {  
    this.panjang = panjang;  
    this.lebar = lebar;  
}
```


java

 Salin kode

```
double luasPersegiPanjang() {  
    return panjang * lebar;  
}
```


Konstruktor ini digunakan untuk inisialisasi persegi panjang dengan mengisi atribut panjang dan lebar dengan nilai yang diberikan sebagai parameter.

java

 Salin kode

```
BangunDatar(double jariJari, boolean isLingkaran) {  
    this.jariJari = jariJari;  
}
```

java

 Salin kode

```
double luasLingkaran() {  
    return Math.PI * jariJari * jariJari;  
}
```

Konstruktor ini digunakan untuk inisialisasi lingkaran dengan mengisi atribut jariJari. Parameter tambahan isLingkaran hanya berfungsi sebagai penanda dan tidak digunakan dalam logika program.

java

Salin kode

```
BangunDatar(double alas, double tinggi, boolean isSegitiga) {  
    this.alas = alas;  
    this.tinggi = tinggi;  
}
```

java

Salin kode

```
double luasSegitiga() {  
    return 0.5 * alas * tinggi;  
}
```

Konstruktor ini digunakan untuk inisialisasi segitiga dengan mengisi atribut alas dan tinggi. Parameter tambahan isSegitiga hanya berfungsi sebagai penanda dan tidak digunakan dalam logika program.

java

Salin kode


```
switch (pilihan) {
```

Menggunakan struktur switch untuk memproses pilihan pengguna dan menghitung luas bangun datar sesuai pilihan.

Pilihan:

- **Pilihan 1 (Persegi):**
 - Meminta pengguna untuk memasukkan panjang sisi persegi, lalu menghitung dan menampilkan luas persegi.
- **Pilihan 2 (Persegi Panjang):**
 - Meminta pengguna memasukkan panjang dan lebar persegi panjang, lalu menghitung dan menampilkan luas persegi panjang.
- **Pilihan 3 (Lingkaran):**
 - Meminta pengguna memasukkan jari-jari lingkaran, lalu menghitung dan menampilkan luas lingkaran.
- **Pilihan 4 (Segitiga):**
 - Meminta pengguna memasukkan panjang alas dan tinggi segitiga, lalu menghitung dan menampilkan luas segitiga.
- **Pilihan Tidak Valid:**
 - Menampilkan pesan jika pilihan yang dimasukkan tidak valid.

yaml

 Salin kode

LUAS PERSEGI : 25.0

Jika pengguna memilih opsi 1 dan memasukkan sisi persegi 5