

**Nama : Sakti Hermawan**

**Kelas : TI 6**

**NIM : 2023150072**

## **A. Rancangan Class, Atribut, dan Method**

### **1. Class Supir**

java

CopyEdit

```
class Supir {  
    String nip;  
    String nama;  
  
    // Constructor  
    Supir(String nip, String nama) {  
        this.nip = nip;  
        this.nama = nama;  
    }  
}
```

### **2. Class Mobil**

java

CopyEdit

```
class Mobil {  
    String nomorPolisi;  
    String merkMobil;  
    String warna;  
    boolean tersedia;  
  
    // Constructor  
    Mobil(String nomorPolisi, String merkMobil, String warna) {  
        this.nomorPolisi = nomorPolisi;  
        this.merkMobil = merkMobil;  
        this.warna = warna;  
    }  
}
```

```
        this.tersedia = true;
    }

    void setTersedia(boolean status) {
        this.tersedia = status;
    }
}
```

### 3. Class Rental

java

CopyEdit

```
import java.util.Date;
```

```
class Rental {
    Supir supir;
    Mobil mobil;
    Date tanggalPinjam;
    Date tanggalKembali;

    // Constructor
    Rental(Supir supir, Mobil mobil) {
        this.supir = supir;
        this.mobil = mobil;
        this.tanggalPinjam = new Date(); // otomatis tanggal hari ini
        this.mobil.setTersedia(false); // mobil sedang dipinjam
    }

    void kembalikanMobil() {
        this.tanggalKembali = new Date(); // tanggal pengembalian
        this.mobil.setTersedia(true); // mobil tersedia kembali
    }
}
```



**Ringkasan :**

Class	Atribut Utama	Method Utama
Supir	nip, nama	-
Mobil	nomorPolisi, merkMobil, warna, tersedia	setTersedia()
Rental	supir, mobil, tanggalPinjam, tanggalKembali	kembalikanMobil()

**B. Kode Program dari hasil analisa**

// Abstract class Person

```
abstract class Person {  
    String nama;  
  
    Person(String nama) {  
        this.nama = nama;  
    }  
  
    abstract void tampilkanInfo();  
}
```

// Supir mewarisi Person

```
class Supir extends Person {  
    String nip;  
  
    Supir(String nip, String nama) {  
        super(nama);  
        this.nip = nip;  
    }  
  
    @Override  
    void tampilkanInfo() {
```

```
        System.out.println("NIP: " + nip);  
        System.out.println("Nama: " + nama);  
    }  
}
```

```
// Kelas Mobil
```

```
class Mobil {  
    String nomorPolisi;  
    String merkMobil;  
    String warna;  
    boolean tersedia;  
  
    Mobil(String nomorPolisi, String merkMobil, String warna) {  
        this.nomorPolisi = nomorPolisi;  
        this.merkMobil = merkMobil;  
        this.warna = warna;  
        this.tersedia = true;  
    }  
  
    void setTersedia(boolean status) {  
        this.tersedia = status;  
    }  
  
    void tampilkanInfo() {  
        System.out.println("Nomor Polisi: " + nomorPolisi);  
        System.out.println("Merk: " + merkMobil);  
        System.out.println("Warna: " + warna);  
        System.out.println("Tersedia: " + tersedia);  
    }  
}
```

```
// Interface RentalAction
```

```
interface RentalAction {
```

```
    void pinjam();
```

```
    void kembalikan();
```

```
}
```

```
// Class Rental mengimplementasikan RentalAction
```

```
import java.util.Date;
```

```
class Rental implements RentalAction {
```

```
    Supir supir;
```

```
    Mobil mobil;
```

```
    Date tanggalPinjam;
```

```
    Date tanggalKembali;
```

```
    Rental(Supir supir, Mobil mobil) {
```

```
        this.supir = supir;
```

```
        this.mobil = mobil;
```

```
    }
```

```
@Override
```

```
public void pinjam() {
```

```
    if (mobil.tersedia) {
```

```
        tanggalPinjam = new Date();
```

```
        mobil.setTersedia(false);
```

```
        System.out.println("Mobil berhasil dipinjam pada " + tanggalPinjam);
```

```
    } else {
```

```
        System.out.println("Mobil tidak tersedia!");
```

```
    }
```

```
}
```

```

@Override

public void kembalikan() {
    if (!mobil.tersedia) {
        tanggalKembali = new Date();
        mobil.setTersedia(true);
        System.out.println("Mobil dikembalikan pada " + tanggalKembali);
    } else {
        System.out.println("Mobil sudah tersedia sebelumnya.");
    }
}
}

```

// Main class untuk menjalankan program

```

public class Main {
    public static void main(String[] args) {
        Supir supir1 = new Supir("12345", "Budi");
        Mobil mobil1 = new Mobil("AB123CD", "Toyota Avanza", "Hitam");

        Rental rental1 = new Rental(supir1, mobil1);

        supir1.tampilkanInfo();
        mobil1.tampilkanInfo();

        System.out.println("\n--- Proses Peminjaman ---");
        rental1.pinjam();

        System.out.println("\n--- Info Mobil Setelah Dipinjam ---");
        mobil1.tampilkanInfo();

        System.out.println("\n--- Proses Pengembalian ---");
        rental1.kembalikan();
    }
}

```

```
        System.out.println("\n--- Info Mobil Setelah Dikembalikan ---");  
        mobil1.tampilkanInfo();  
    }  
}
```

**Penjelasan Konsep yang Digunakan:**

- **Inheritance:** Supir mewarisi Person.
- **Abstract Class:** Person tidak bisa dibuat langsung, hanya bisa diwarisi.
- **Interface:** RentalAction mendefinisikan kontrak method pinjam() dan kembalikan().