TUGAS

PEMROGRAMAN BERBASIS OBJEK

MUHAMMAD RAYHAN ZAMZAMI

KELAS TI 1-H

244107020027

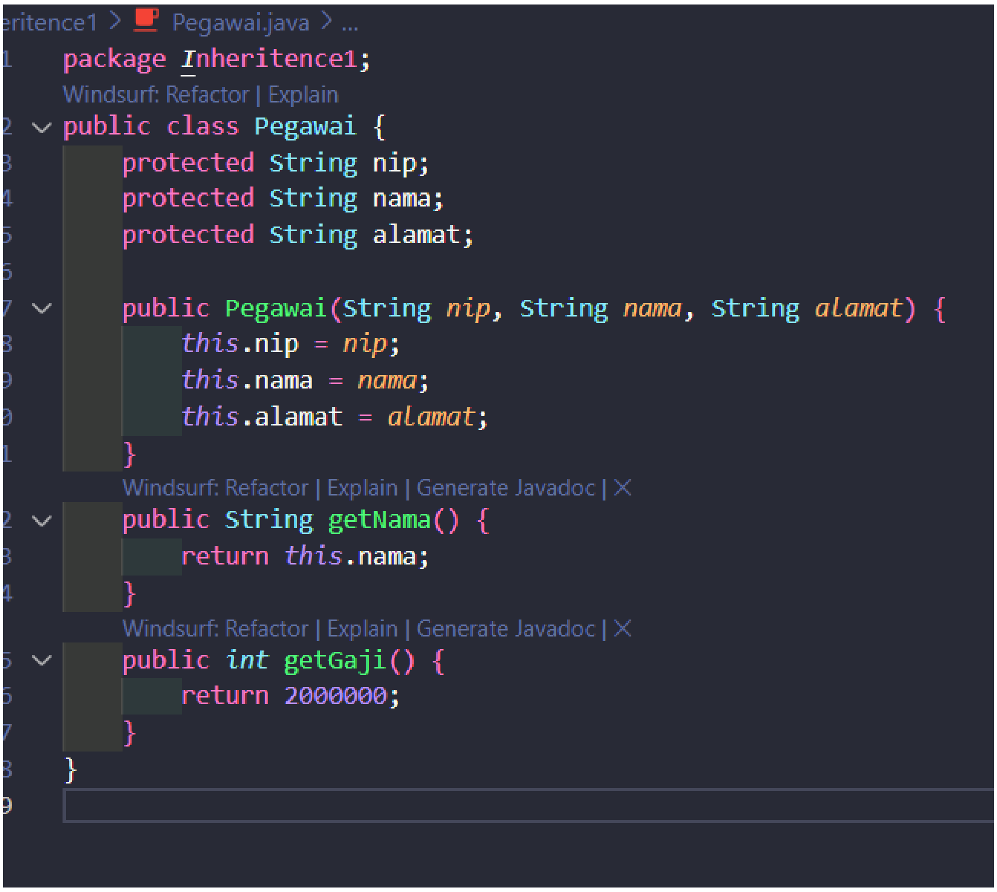
TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG 2024

Inheritence 1

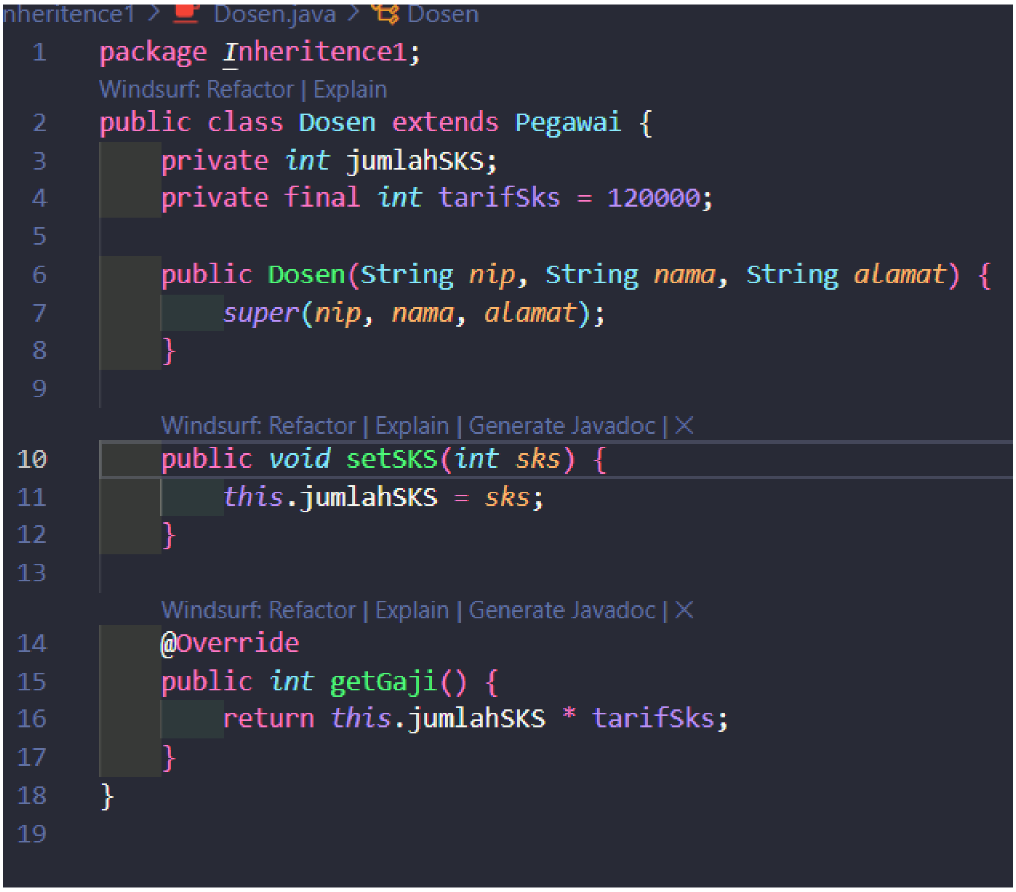
Class Pegawai



eritence1 > Pegawai.java > ...

```java
package Inheritence1;
Windsurf: Refactor | Explain
public class Pegawai {
    protected String nip;
    protected String nama;
    protected String alamat;

    public Pegawai(String nip, String nama, String alamat) {
        this.nip = nip;
        this.nama = nama;
        this.alamat = alamat;
    }
    Windsurf: Refactor | Explain | Generate Javadoc | ×
    public String getNama() {
        return this.nama;
    }
    Windsurf: Refactor | Explain | Generate Javadoc | ×
    public int getGaji() {
        return 2000000;
    }
}
```

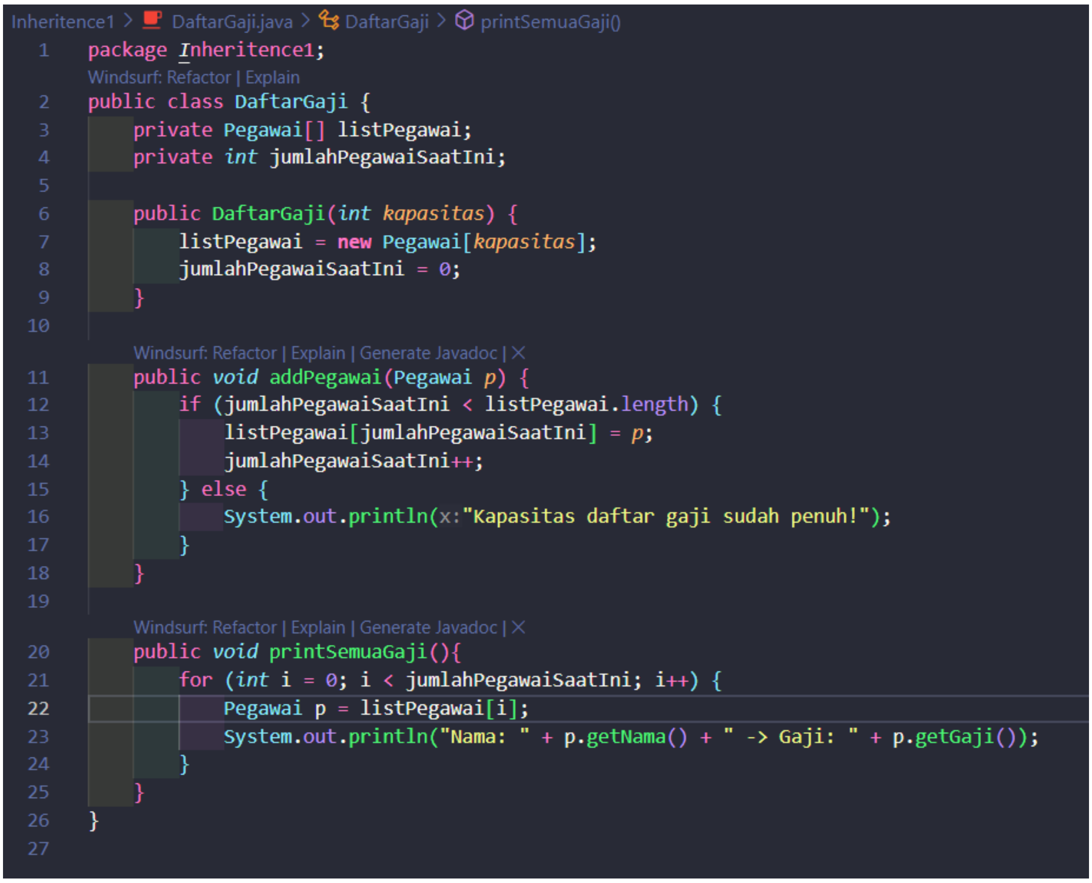
Class Dosen



nheritence1 > Dosen.java > Dosen

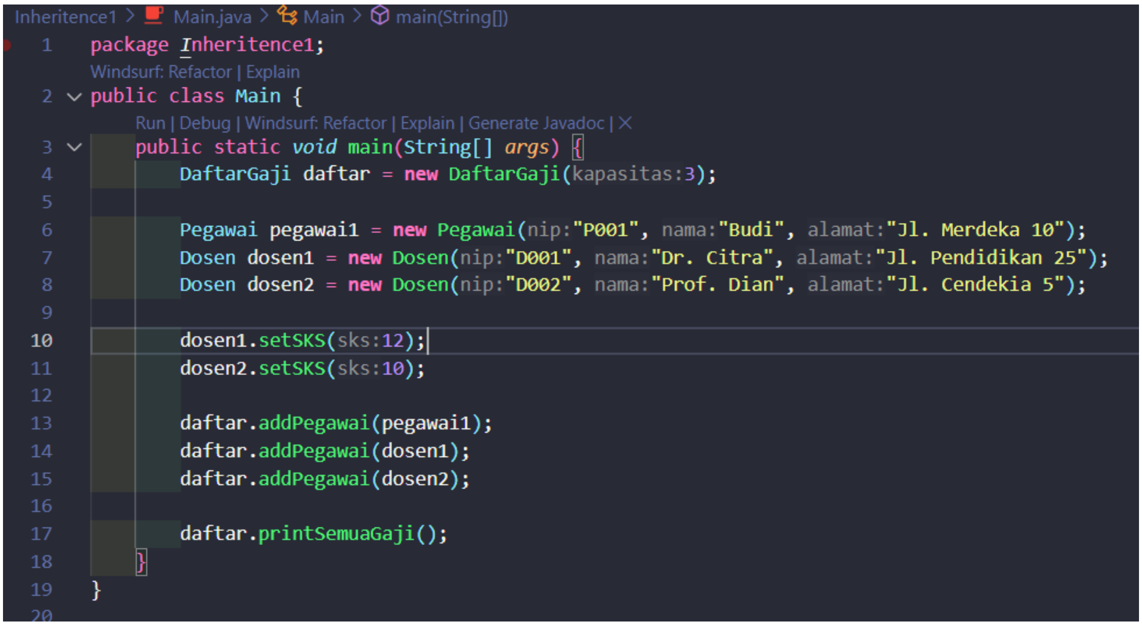
```java
package Inheritence1;
Windsurf: Refactor | Explain
public class Dosen extends Pegawai {
    private int jumlahSKS;
    private final int tarifSks = 120000;

    public Dosen(String nip, String nama, String alamat) {
        super(nip, nama, alamat);
    }

    Windsurf: Refactor | Explain | Generate Javadoc | ×
    public void setSKS(int sks) {
        this.jumlahSKS = sks;
    }

    Windsurf: Refactor | Explain | Generate Javadoc | ×
    @Override
    public int getGaji() {
        return this.jumlahSKS * tarifSks;
    }
}
```

Class DaftarGaji


Inheritence1 > DaftarGaji.java > DaftarGaji > printSemuaGaji()
Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×
Windsurf: Refactor | Explain | Generate Javadoc | ×

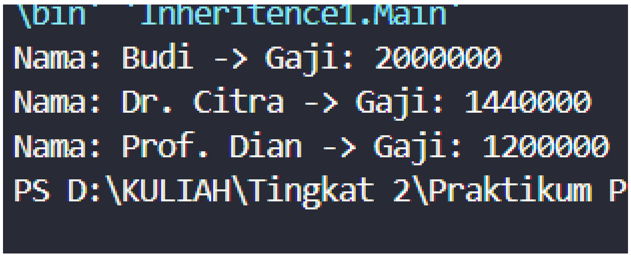
```java
package Inheritence1;
public class DaftarGaji {
    private Pegawai[] listPegawai;
    private int jumlahPegawaiSaatIni;

    public DaftarGaji(int kapasitas) {
        listPegawai = new Pegawai[kapasitas];
        jumlahPegawaiSaatIni = 0;
    }

    public void addPegawai(Pegawai p) {
        if (jumlahPegawaiSaatIni < listPegawai.length) {
            listPegawai[jumlahPegawaiSaatIni] = p;
            jumlahPegawaiSaatIni++;
        } else {
            System.out.println(x:"Kapasitas daftar gaji sudah penuh!");
        }
    }

    public void printSemuaGaji(){
        for (int i = 0; i < jumlahPegawaiSaatIni; i++) {
            Pegawai p = listPegawai[i];
            System.out.println("Nama: " + p.getNama() + " -> Gaji: " + p.getGaji());
        }
    }
}

```

Class Main


Inheritence1 > Main.java > Main > main(String[])
Windsurf: Refactor | Explain
Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | ×

```java
package Inheritence1;
public class Main {
    public static void main(String[] args) {
        DaftarGaji daftar = new DaftarGaji(kapasitas:3);

        Pegawai pegawai1 = new Pegawai(nip:"P001", nama:"Budi", alamat:"Jl. Merdeka 10");
        Dosen dosen1 = new Dosen(nip:"D001", nama:"Dr. Citra", alamat:"Jl. Pendidikan 25");
        Dosen dosen2 = new Dosen(nip:"D002", nama:"Prof. Dian", alamat:"Jl. Cendekia 5");

        dosen1.setSKS(sks:12);
        dosen2.setSKS(sks:10);

        daftar.addPegawai(pegawai1);
        daftar.addPegawai(dosen1);
        daftar.addPegawai(dosen2);

        daftar.printSemuaGaji();
    }
}
```

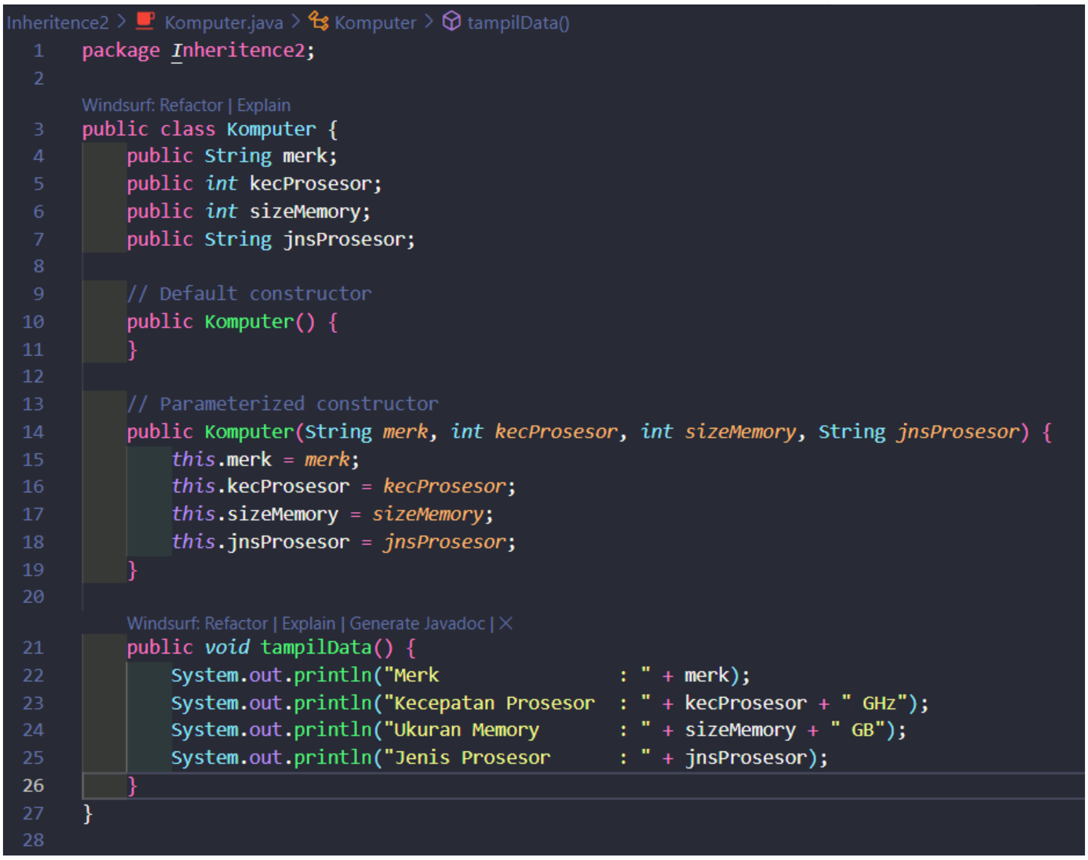
Output



```
\bin' 'Inheritence1.Main'
Nama: Budi -> Gaji: 2000000
Nama: Dr. Citra -> Gaji: 1440000
Nama: Prof. Dian -> Gaji: 1200000
PS D:\KULIAH\Tingkat 2\Praktikum P
```

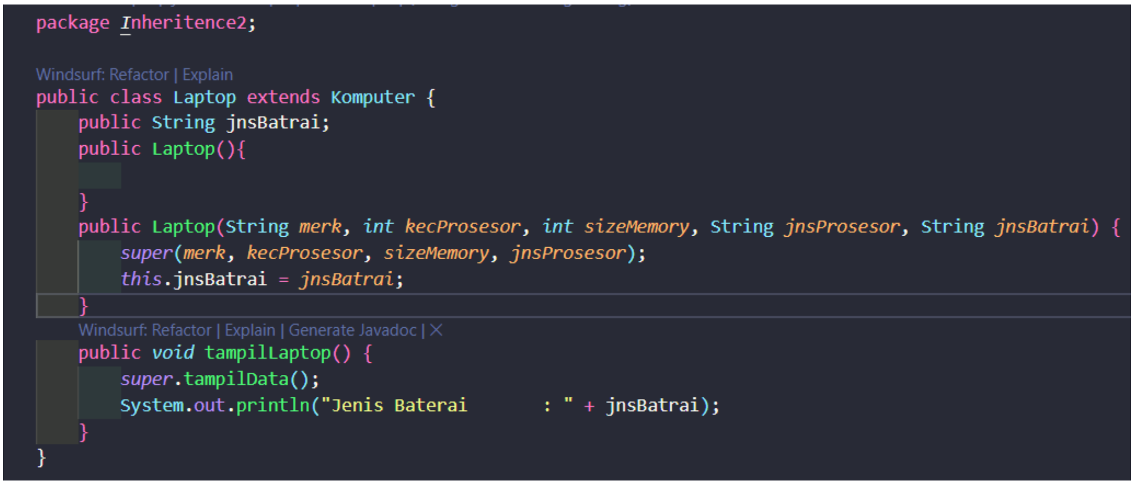
Inheritence 2

Class Komputer


Inheritence2 > Komputer.java > Komputer > tampilData()
Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×

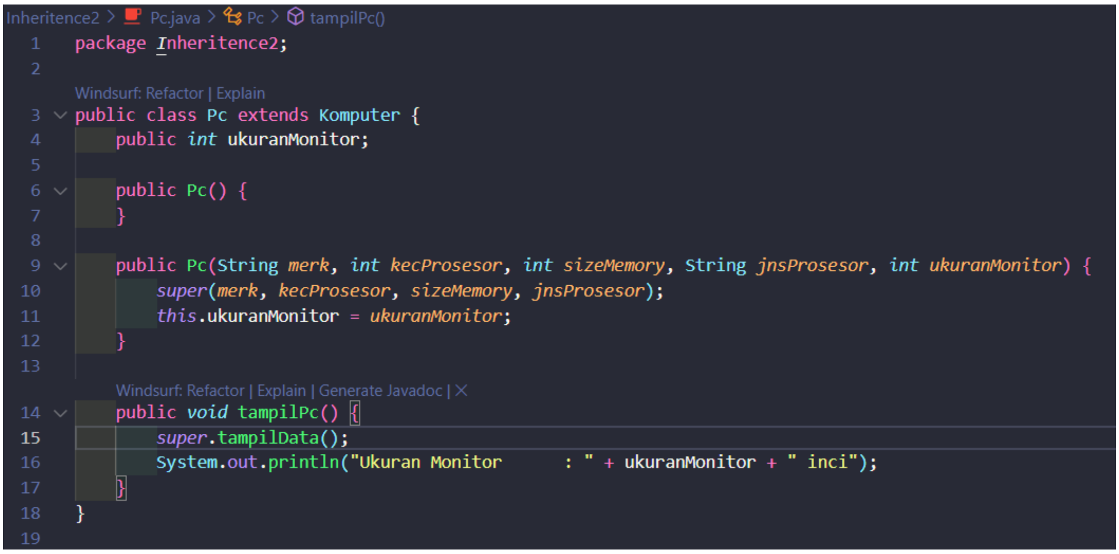
```java
package Inheritence2;

public class Komputer {
    public String merk;
    public int kecProsesor;
    public int sizeMemory;
    public String jnsProsesor;

    // Default constructor
    public Komputer() {
    }

    // Parameterized constructor
    public Komputer(String merk, int kecProsesor, int sizeMemory, String jnsProsesor) {
        this.merk = merk;
        this.kecProsesor = kecProsesor;
        this.sizeMemory = sizeMemory;
        this.jnsProsesor = jnsProsesor;
    }

    public void tampilData() {
        System.out.println("Merk                : " + merk);
        System.out.println("Kecepatan Prosesor  : " + kecProsesor + " GHz");
        System.out.println("Ukuran Memory       : " + sizeMemory + " GB");
        System.out.println("Jenis Prosesor      : " + jnsProsesor);
    }
}

```

Class Laptop


Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×

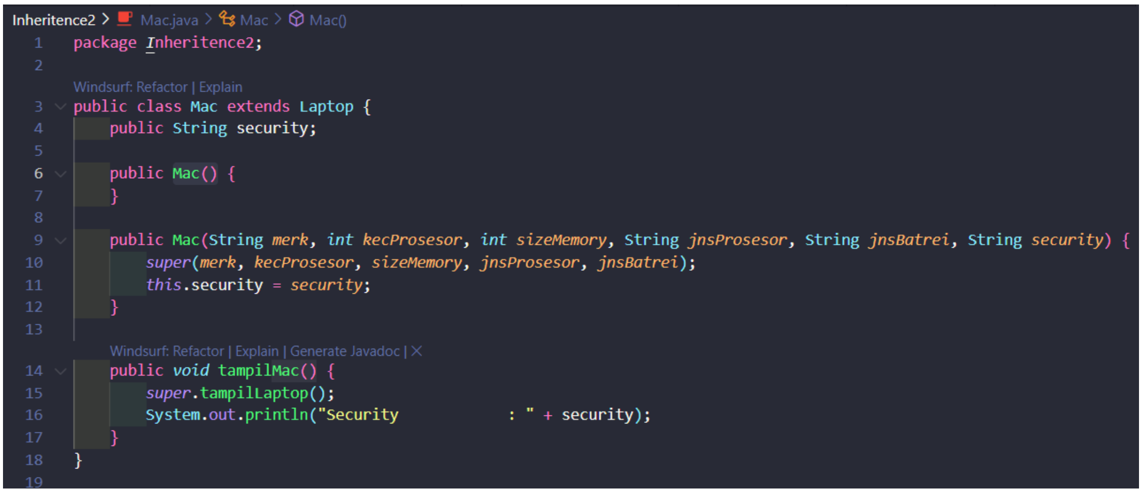
```java
package Inheritence2;

public class Laptop extends Komputer {
    public String jnsBatrai;
    public Laptop(){
        
    }
    public Laptop(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrai) {
        super(merk, kecProsesor, sizeMemory, jnsProsesor);
        this.jnsBatrai = jnsBatrai;
    }
    public void tampilLaptop() {
        super.tampilData();
        System.out.println("Jenis Baterai       : " + jnsBatrai);
    }
}
```

Class Pc


Inheritence2 > Pc.java > Pc > tampilPc()
Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×

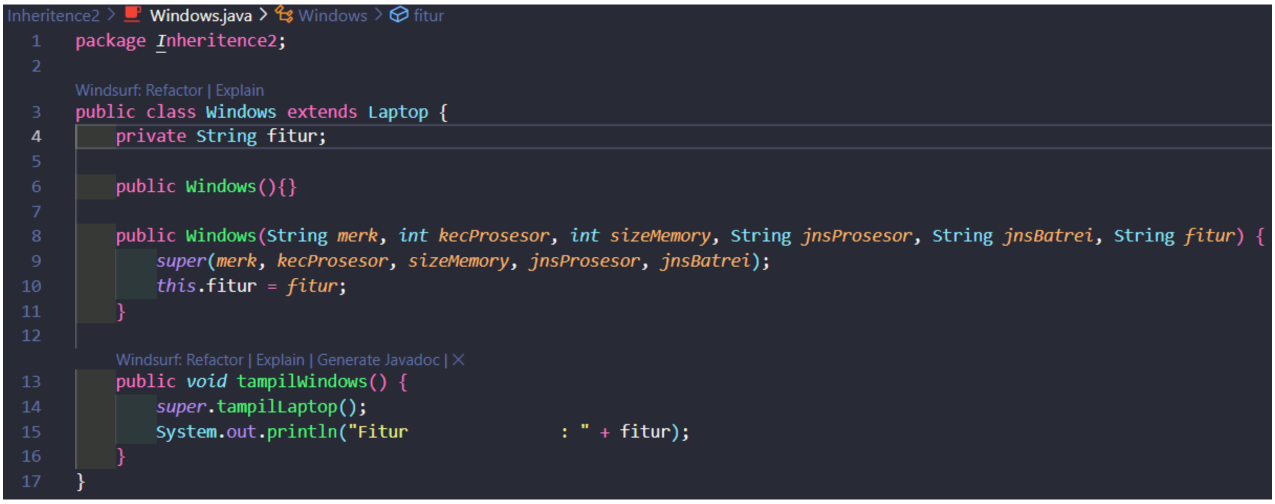
```java
package Inheritence2;

public class Pc extends Komputer {
    public int ukuranMonitor;

    public Pc() {
    }

    public Pc(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, int ukuranMonitor) {
        super(merk, kecProsesor, sizeMemory, jnsProsesor);
        this.ukuranMonitor = ukuranMonitor;
    }

    public void tampilPc() {
        super.tampilData();
        System.out.println("Ukuran Monitor      : " + ukuranMonitor + " inci");
    }
}
```

Class Mac


Inheritence2 > Mac.java > Mac > Mac()
Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×

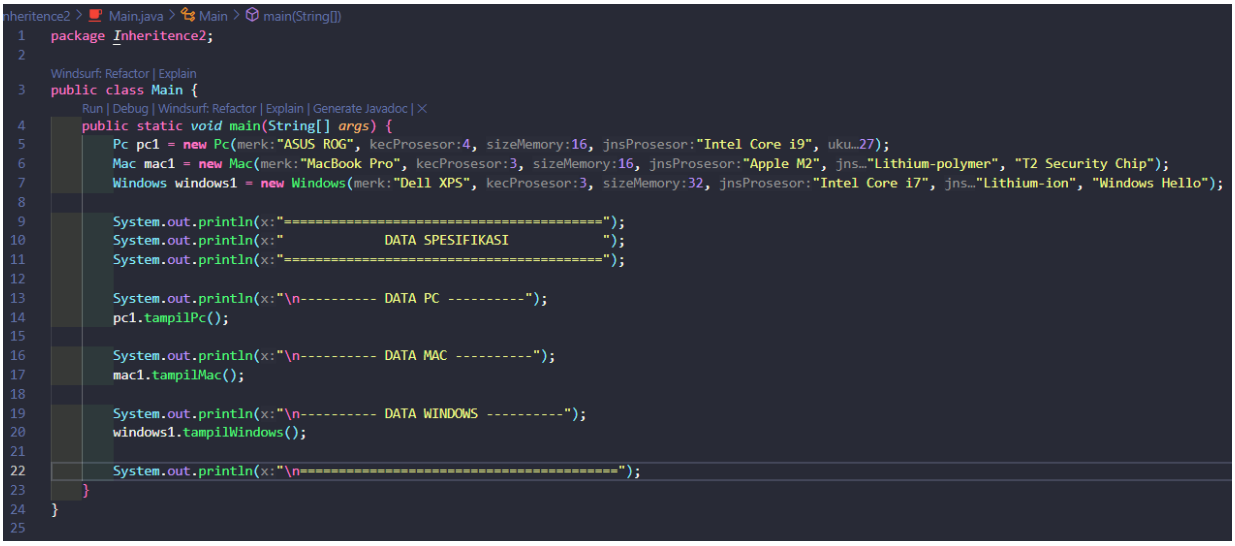
```java
package Inheritence2;

public class Mac extends Laptop {
    public String security;

    public Mac() {
    }

    public Mac(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrei, String security) {
        super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrei);
        this.security = security;
    }

    public void tampilMac() {
        super.tampilLaptop();
        System.out.println("Security            : " + security);
    }
}
```

Class Windows


Inheritence2 > Windows.java > Windows > fitur
Windsurf: Refactor | Explain
Windsurf: Refactor | Explain | Generate Javadoc | ×

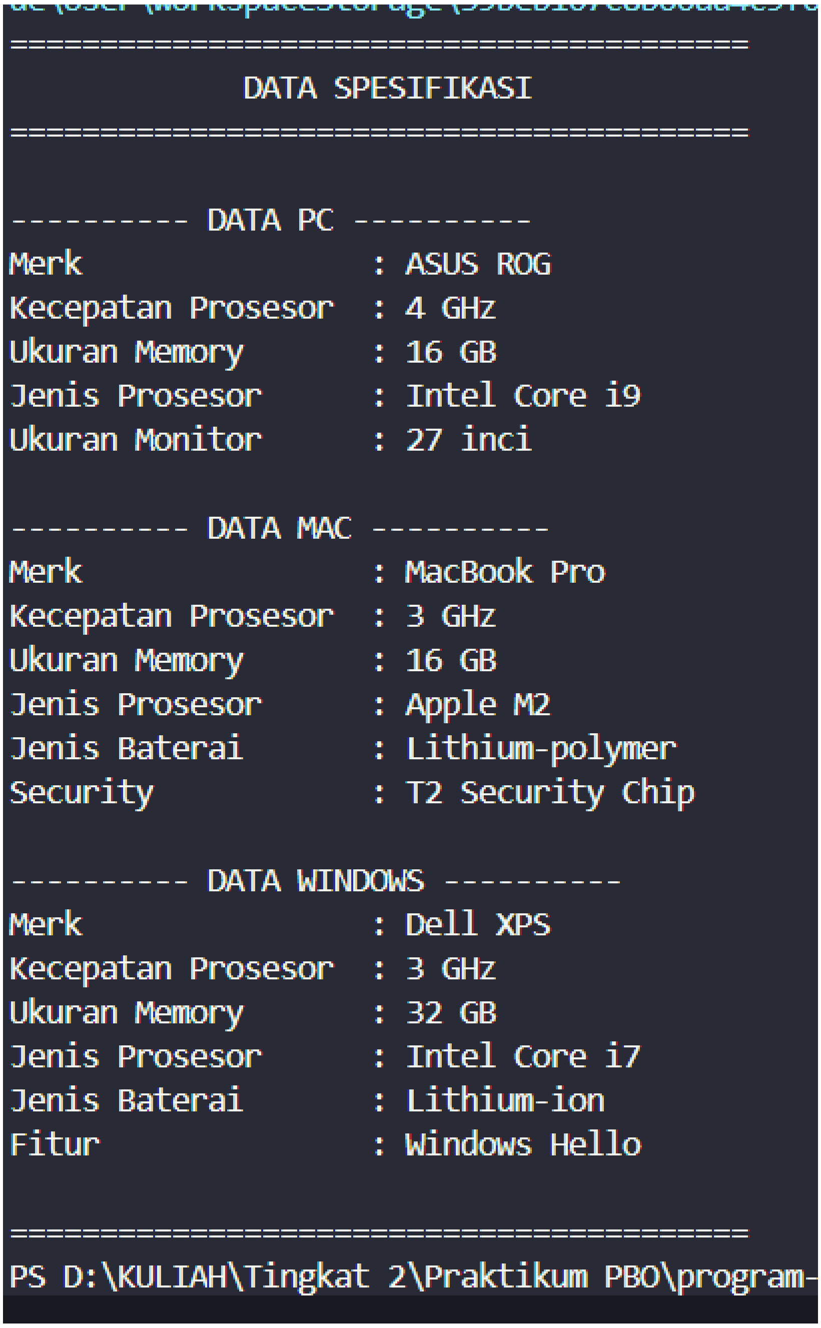
```java
package Inheritence2;

public class Windows extends Laptop {
    private String fitur;

    public Windows(){}

    public Windows(String merk, int kecProsesor, int sizeMemory, String jnsProsesor, String jnsBatrei, String fitur) {
        super(merk, kecProsesor, sizeMemory, jnsProsesor, jnsBatrei);
        this.fitur = fitur;
    }

    public void tampilWindows() {
        super.tampilLaptop();
        System.out.println("Fitur               : " + fitur);
    }
}
```

Class Main


nheritence2 > Main.java > Main > main(String[])
Windsurf: Refactor | Explain
Run | Debug | Windsurf: Refactor | Explain | Generate Javadoc | ×

```java
package Inheritence2;

public class Main {
    public static void main(String[] args) {
        Pc pc1 = new Pc(merk:"ASUS ROG", kecProsesor:4, sizeMemory:16, jnsProsesor:"Intel Core i9", uku…27);
        Mac mac1 = new Mac(merk:"MacBook Pro", kecProsesor:3, sizeMemory:16, jnsProsesor:"Apple M2", jns…"Lithium-polymer", "T2 Security Chip");
        Windows windows1 = new Windows(merk:"Dell XPS", kecProsesor:3, sizeMemory:32, jnsProsesor:"Intel Core i7", jns…"Lithium-ion", "Windows Hello");

        System.out.println(x:"=========================================");
        System.out.println(x:"             DATA SPESIFIKASI            ");
        System.out.println(x:"=========================================");

        System.out.println(x:"\n---------- DATA PC ----------");
        pc1.tampilPc();

        System.out.println(x:"\n---------- DATA MAC ----------");
        mac1.tampilMac();

        System.out.println(x:"\n---------- DATA WINDOWS ----------");
        windows1.tampilWindows();

        System.out.println(x:"\n=========================================");
    }
}
```

Output



```
=========================================
             DATA SPESIFIKASI
=========================================


---------- DATA PC ----------
Merk                    : ASUS ROG
Kecepatan Prosesor      : 4 GHz
Ukuran Memory           : 16 GB
Jenis Prosesor          : Intel Core i9
Ukuran Monitor          : 27 inci

---------- DATA MAC ----------
Merk                    : MacBook Pro
Kecepatan Prosesor      : 3 GHz
Ukuran Memory           : 16 GB
Jenis Prosesor          : Apple M2
Jenis Baterai           : Lithium-polymer
Security                : T2 Security Chip

---------- DATA WINDOWS ----------
Merk                    : Dell XPS
Kecepatan Prosesor      : 3 GHz
Ukuran Memory           : 32 GB
Jenis Prosesor          : Intel Core i7
Jenis Baterai           : Lithium-ion
Fitur                   : Windows Hello

=========================================
PS D:\KULIAH\Tingkat 2\Praktikum PBO\program-
```