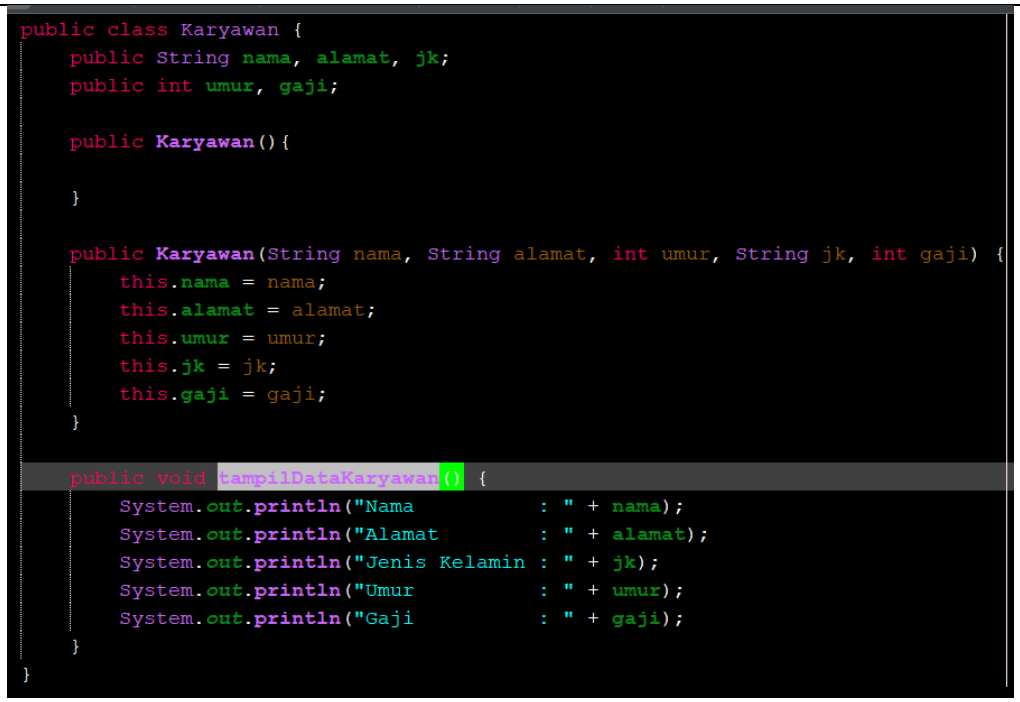
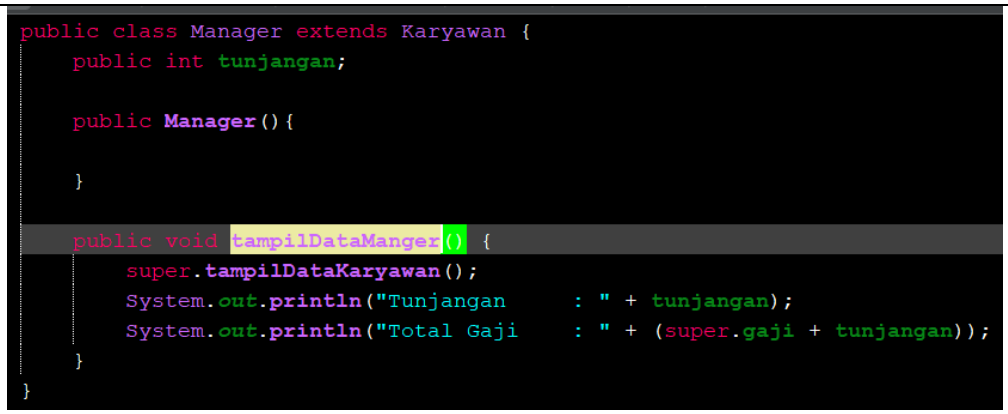
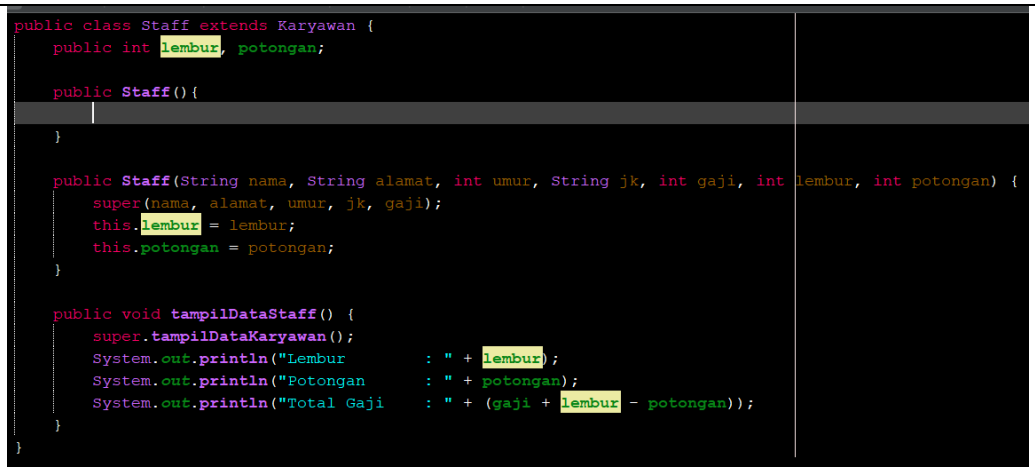
OBJECT ORIENTED PROGRAM

Jobsheet 6 – Inheritance 2

AYU JOVITA WIDYADHARI
2241720219/04/2I

1st Practicum

Class	Code
Karyawan	
Manager	
Staff	

```java
public class Karyawan {
    public String nama, alamat, jk;
    public int umur, gaji;

    public Karyawan(){

    }

    public Karyawan(String nama, String alamat, int umur, String jk, int gaji) {
        this.nama = nama;
        this.alamat = alamat;
        this.umur = umur;
        this.jk = jk;
        this.gaji = gaji;
    }

    public void tampilDataKaryawan() {
        System.out.println("Nama           : " + nama);
        System.out.println("Alamat         : " + alamat);
        System.out.println("Jenis Kelamin : " + jk);
        System.out.println("Umur           : " + umur);
        System.out.println("Gaji           : " + gaji);
    }
}
```

```java
public class Manager extends Karyawan {
    public int tunjangan;

    public Manager(){

    }

    public void tampilDataManger() {
        super.tampilDataKaryawan();
        System.out.println("Tunjangan      : " + tunjangan);
        System.out.println("Total Gaji     : " + (super.gaji + tunjangan));
    }
}
```

```java
public class Staff extends Karyawan {
    public int lembur, potongan;

    public Staff(){

    }

    public Staff(String nama, String alamat, int umur, String jk, int gaji, int lembur, int potongan) {
        super(nama, alamat, umur, jk, gaji);
        this.lembur = lembur;
        this.potongan = potongan;
    }

    public void tampilDataStaff() {
        super.tampilDataKaryawan();
        System.out.println("Lembur         : " + lembur);
        System.out.println("Potongan       : " + potongan);
        System.out.println("Total Gaji     : " + (gaji + lembur - potongan));
    }
}
```

Main class

```java
public class Inheritance1 {
    public static void main(String[] args) {
        //TODO code app logic is here
        // Objek Manager
        System.out.println(x: "========== Data Manager ----------");
        Manager M = new Manager();
        M.nama = "Viona";
        M.alamat = "Jl. Ijen";
        M.umur = 24;
        M.jk = "Perempuan";
        M.gaji = 3000000;
        M.tunjangan = 1000000;
        M.tampilDataManger();

        // Objek Staff
        System.out.println(x: "========== Data Staff ----------");
        Staff S = new Staff();
        S.nama = "Leo";
        S.alamat = "Jl. MT Haryono";
        S.umur = 25;
        S.jk = "Laki-Laki";
        S.nama = "Leo";
        S.gaji = 2000000;
        S.lembur = 500000;
        S.potongan = 250000;
        S.tampilDataStaff();
```

Result

```
========== Data Manager ----------
Nama           : Viona
Alamat         : Jl. Ijen
Jenis Kelamin  : Perempuan
Umur           : 24
Gaji           : 3000000
Tunjangan      : 1000000
Total Gaji     : 4000000
========== Data Staff ----------
Nama           : Leo
Alamat         : Jl. MT Haryono
Jenis Kelamin  : Laki-Laki
Umur           : 25
Gaji           : 2000000
Lembur         : 500000
Potongan       : 250000
Total Gaji     : 2250000
--------------------------------
BUILD SUCCESS
```

Question

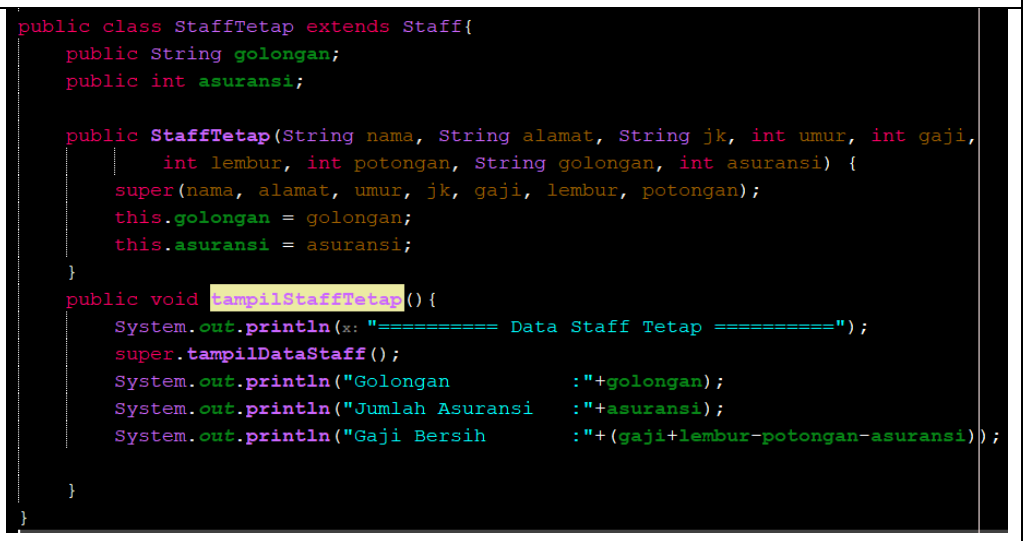
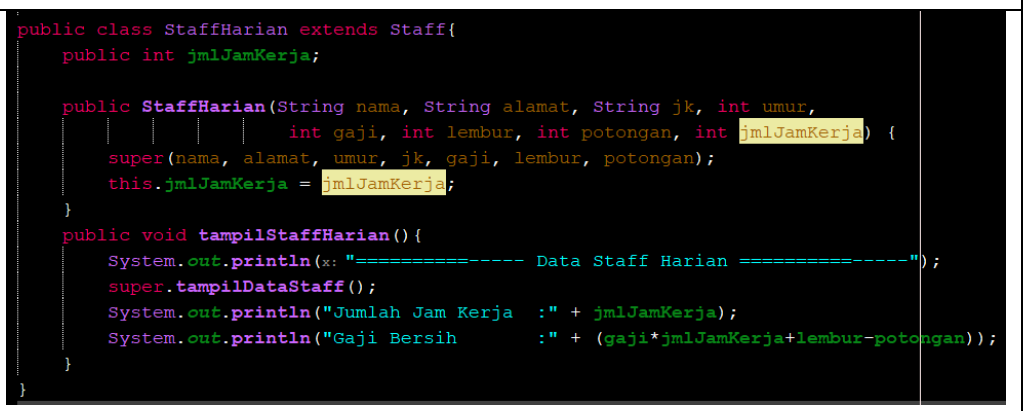
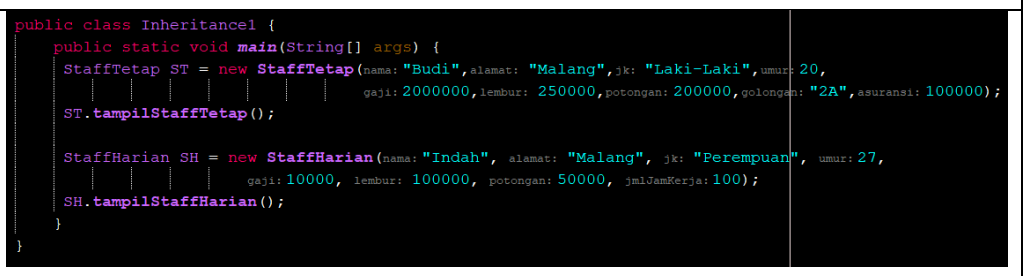
1. Sebutkan class mana yang termasuk super class dan sub class dari percobaan 1 diatas!
   Jawaban:
   - Super class : Karyawan class
     Karena, semua property dan method dari kelas Karyawan diwarisi oleh kelas Manager dan kelas Staff.
   - Sub class: Manger class dan Staff class
     Kedua kelas tersebut mewarisi property dan method dari super class dan dapat memiliki property atau method tambahan tergantung kebutuhan masing-masing kelas.
2. Kata kunci apakah yang digunakan untuk menurunkan suatu class ke class yang lain?
   Jawaban:
   kata kunci yang digunakan untuk menurunkan suatu kelas ke kelas lain yakni menambahkan syntax 'extends'.
   Cara penggunaan: sub class 'extends' super class.
3. Perhatikan kode program pada class Manager, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Karyawan!
   Jawaban:
   Atribut yang dimiliki kelas Manager:
   - Nama, alamat, umur, jk (jenis kelamin), gaji, tunjangan.

   Atribut yang diwarisi dari kelas Karyawan:
   - Nama, alamat, umur, jk, dan gaji.
4. Jelaskan kata kunci super pada potongan program dibawah ini yang terdapat pada class Manager!

```
System.out.println("Total Gaji       ="+(super.gaji+tunjangan));
```

   Jawaban:
   Syntax super digunakan untuk merujuk ke super class. Pada kode diatas 'super.gaji' merujuk pada variable 'gaji' yang diwarisi oleh kelas Karyawan. Sedangkan 'super' digunakan untuk mengakses variable dari superclass, dengan begitu program bisa membedakan antara variable gaji dari kelas Manager dengan kelas Karyawan.
5. Program pada percobaan 1 diatas termasuk dalam jenis inheritance apa? Jelaskan alasannya!
   Jawaban:
   Percobaan 1 merupakan jenis "single inheritance".
   Karena, pada percobaan diatas bisa kita lihat terdapat tiga tingkat kelas dalam hirarki: Karyawan class (superclass), Manager class dan Staff Class sebagai sub class-nya. Konsep Single inheritance sendiri hanya memperbolehkan suatu subclass mempunyai satu parent class. Jadi, Manager class dan Staff class yang merupakan subclass masing masing memiliki satu parent class yakni kelas Karyawan.

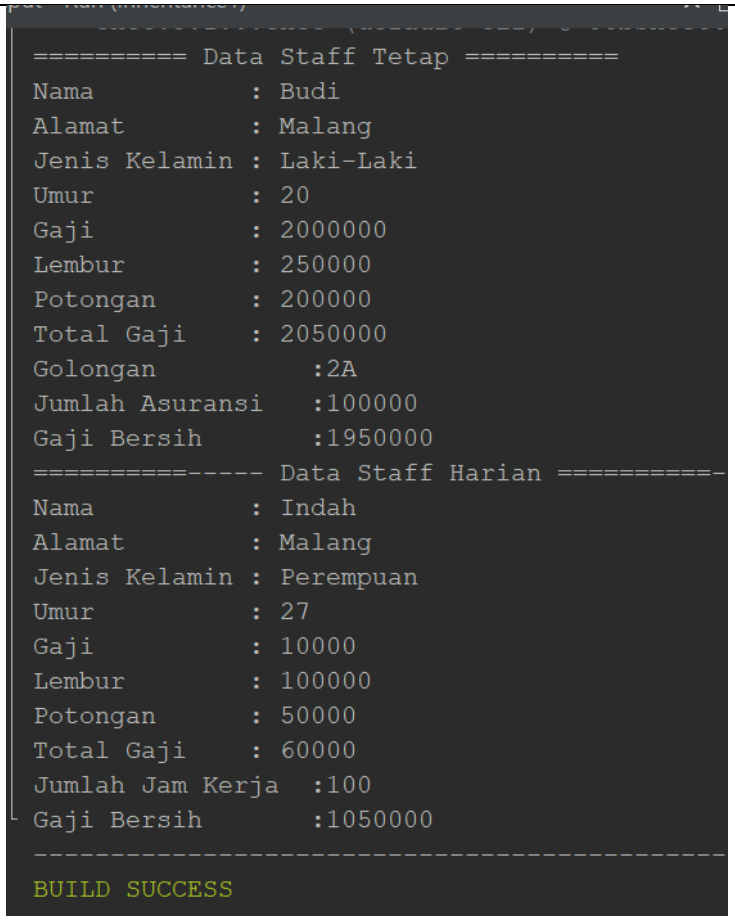
2nd Practicum

Class	Code
StaffTetap	
StaffHarian	
Main class	

```java
public class StaffTetap extends Staff{
    public String golongan;
    public int asuransi;

    public StaffTetap(String nama, String alamat, String jk, int umur, int gaji,
            int lembur, int potongan, String golongan, int asuransi) {
        super(nama, alamat, umur, jk, gaji, lembur, potongan);
        this.golongan = golongan;
        this.asuransi = asuransi;
    }
    public void tampilStaffTetap(){
        System.out.println(x: "========== Data Staff Tetap ==========");
        super.tampilDataStaff();
        System.out.println("Golongan           :"+golongan);
        System.out.println("Jumlah Asuransi    :"+asuransi);
        System.out.println("Gaji Bersih        :"+(gaji+lembur-potongan-asuransi));

    }
}
```

```java
public class StaffHarian extends Staff{
    public int jmlJamKerja;

    public StaffHarian(String nama, String alamat, String jk, int umur,
                int gaji, int lembur, int potongan, int jmlJamKerja) {
        super(nama, alamat, umur, jk, gaji, lembur, potongan);
        this.jmlJamKerja = jmlJamKerja;
    }
    public void tampilStaffHarian(){
        System.out.println(x: "--------------- Data Staff Harian ----------------");
        super.tampilDataStaff();
        System.out.println("Jumlah Jam Kerja  :" + jmlJamKerja);
        System.out.println("Gaji Bersih       :" + (gaji*jmlJamKerja+lembur-potongan));
    }
}
```

```java
public class Inheritance1 {
    public static void main(String[] args) {
        StaffTetap ST = new StaffTetap(nama: "Budi",alamat: "Malang",jk: "Laki-Laki",umur: 20,
                gaji: 2000000,lembur: 250000,potongan: 200000,golongan: "2A",asuransi: 100000);
        ST.tampilStaffTetap();

        StaffHarian SH = new StaffHarian(nama: "Indah", alamat: "Malang", jk: "Perempuan", umur: 27,
                gaji: 10000, lembur: 100000, potongan: 50000, jmlJamKerja: 100);
        SH.tampilStaffHarian();
    }
}
```

Result	

Question

6. Berdasarkan class diatas manakah yang termasuk single inheritance dan mana yang termasuk multilevel inheritance?

   Jawab:

   a. Single inheritance: Karyawan class(super class) dengan Manager class dan Staff class (sub class)

   b. Multilevel inheritance: sub class yang memiliki sub class lagi

      Sub class Staff memiliki sub class lagi yang bernama StaffTetap dan StaffHarian class.

7. Perhatikan kode program class StaffTetap dan StaffHarian, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Staff!

   Jawab:

   - StaffTetap
     - Atribut: nama, alamat, jk, umur, gaji, lembur, potongan, golongan, asuransi
     - Diwarisi dari kelas Staff: nama, alamat, jk, umur, gaji, lembur dan potongan.
   - StaffHarian

- Atribut: nama, alamat, jk, umur, gaji, lembur, potongan, jmlJamKerja
- Diwarisi dari kelas Staff: nama, alamat, jk, umur, gaji, lembur dan potongan.

8. Apakah fungsi potongan program berikut pada class StaffHarian

```
super(nama,alamat,jk,umur,gaji,lembur,potongan);
```

- Digunakan untuk menginisialisasi atribut-atribut dari class **Staff** dengan nilai yang diberikan saat membuat objek dari class **StaffHarian**. Dengan menggunakan **super**, class **StaffHarian** memastikan bahwa atribut-atribut yang diwarisi dari **Staff** diinisialisasi dengan nilai yang sesuai.

Apakah fungsi potongan program berikut pada class StaffHarian

```
super.tampilDataStaff();
```

- Digunakan untuk memanggil metode **tampilDataStaff()** dari superclass (**Staff**). Dengan menggunakan **super**, class **StaffHarian** memanggil metode **tampilDataStaff()** yang ada di class **Staff**. Dengan cara ini, class **StaffHarian** dapat menampilkan data yang diwarisi dari class **Staff** bersama dengan data tambahan yang dimilikinya sendiri.

Jawab:

Dalam kedua kode tersebut, penggunaan **super** memungkinkan class **StaffHarian** untuk berinteraksi dengan fungsionalitas yang ada di class **Staff**, baik itu untuk inisialisasi nilai atribut atau untuk memanggil metode dari superclassnya.

9. Perhatikan kode program dibawah ini yang terdapat pada class StaffTetap

```
System.out.println("Gaji Bersih     ="+(gaji+lembur-potongan-asuransi));
```

Terlihat dipotongan program diatas atribut gaji, lembur dan potongan dapat diakses langsung. Kenapa hal ini bisa terjadi dan bagaimana class StaffTetap memiliki atribut gaji, lembur, dan potongan padahal dalam class tersebut tidak dideklarasikan atribut gaji, lembur, dan potongan?

Jawab:

Dalam potongan program tersebut, atribut **gaji**, **lembur**, dan **potongan** diakses langsung dalam class **StaffTetap**. Ini terjadi karena class **StaffTetap** adalah subclass dari class **Staff**, yang berarti **StaffTetap** mewarisi atribut-atribut public atau protected dari class **Staff**.

Assignment

```java
public class Komputer {
    protected String merk;
    protected int kecProsesor;
    protected int sizeMemory;
    protected String jnsProsesor;

    public Komputer() {
    }

    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("Size Memory: " + sizeMemory + " GB");
        System.out.println("Jenis Prosesor: " + jnsProsesor);
    }
}
```

```java
public class Laptop extends Komputer {
    private String jnsBaterai;

    public Laptop() {
    }

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

    public void tampilLaptop() {
        super.tampilData();
        System.out.println("Jenis Baterai: " + jnsBaterai);
    }
}
```

```java
public class Pc extends Komputer {
    private 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 + " inch");
    }
}
```

```java
public class Mac extends Laptop {
    private String security;

    public Mac() {
    }

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

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

```java
public class Windows extends Laptop {
    private String fitur;

    public Windows() {
    }

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

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

```java
public class Main {
    public static void main(String[] args) {
        Mac mac = new Mac(merk: "Macbook Air", kecProsesor: 2, sizeMemory:8, jnsProsesor: "Intel Core i5", jnsBaterai:"Lithium Polymer", security: "Touch ID");
        Windows windows = new Windows(merk: "Dell XPS", (int) 2.5, sizeMemory:16, jnsProsesor: "Intel Core i7", jnsBaterai:"Lithium Ion", fitur: "Windows Hello");

        Pc pc = new Pc(merk: "HP Pavilion", kecProsesor: 3, sizeMemory:12, jnsProsesor: "AMD Ryzen 5", ukuranMonitor: 27);

        System.out.println(x: "Data Mac:");
        mac.tampilMac();
        System.out.println(x: "\nData Windows:");
        windows.tampilWindows();
        System.out.println(x: "\nData PC:");
        pc.tampilPc();
    }
}
```