**LAPORAN PRAKTIKUM 11**

“INTERFACE”

Dosen Pengampu : Moh. Shohibul Wafa,M.Kom



Disusun Oleh :

Dio Ananda Maulana Wijaya (4123031)

PROGRAM STUDI SISTEM INFORMASI

FAKULTAS SAINS DAN TEKNOLOGI

UNIVERSITAS PESANTREN TINGGI DARUL ULUM

2024

**Latihan**

package modul11;

/\*\*

\*

\* @author Dio Ananda

\*/

//D LATIHAN

// Kelas SmartPhone yang meng-extend Phone dan mengimplementasikan Camera dan CardReader

public class SmartPhone extends Phone implements Camera, CardReader {

private String lens;

public SmartPhone(String phoneNumber, String lens) {

super(phoneNumber);

this.lens = lens;

}

@Override

public void captureImage() {

System.out.println("Capturing image with lens: " + lens);

}

@Override

public void readCard() {

System.out.println("Reading card...");

}

@Override

public void call() {

System.out.println("SmartPhone calling from " + phoneNumber);

}

@Override

public void receiveCall() {

System.out.println("SmartPhone receiving call on " + phoneNumber);

}

public static void main(String[] args) {

SmartPhone myPhone = new SmartPhone("123-456-7890", "12MP");

myPhone.call();

myPhone.receiveCall();

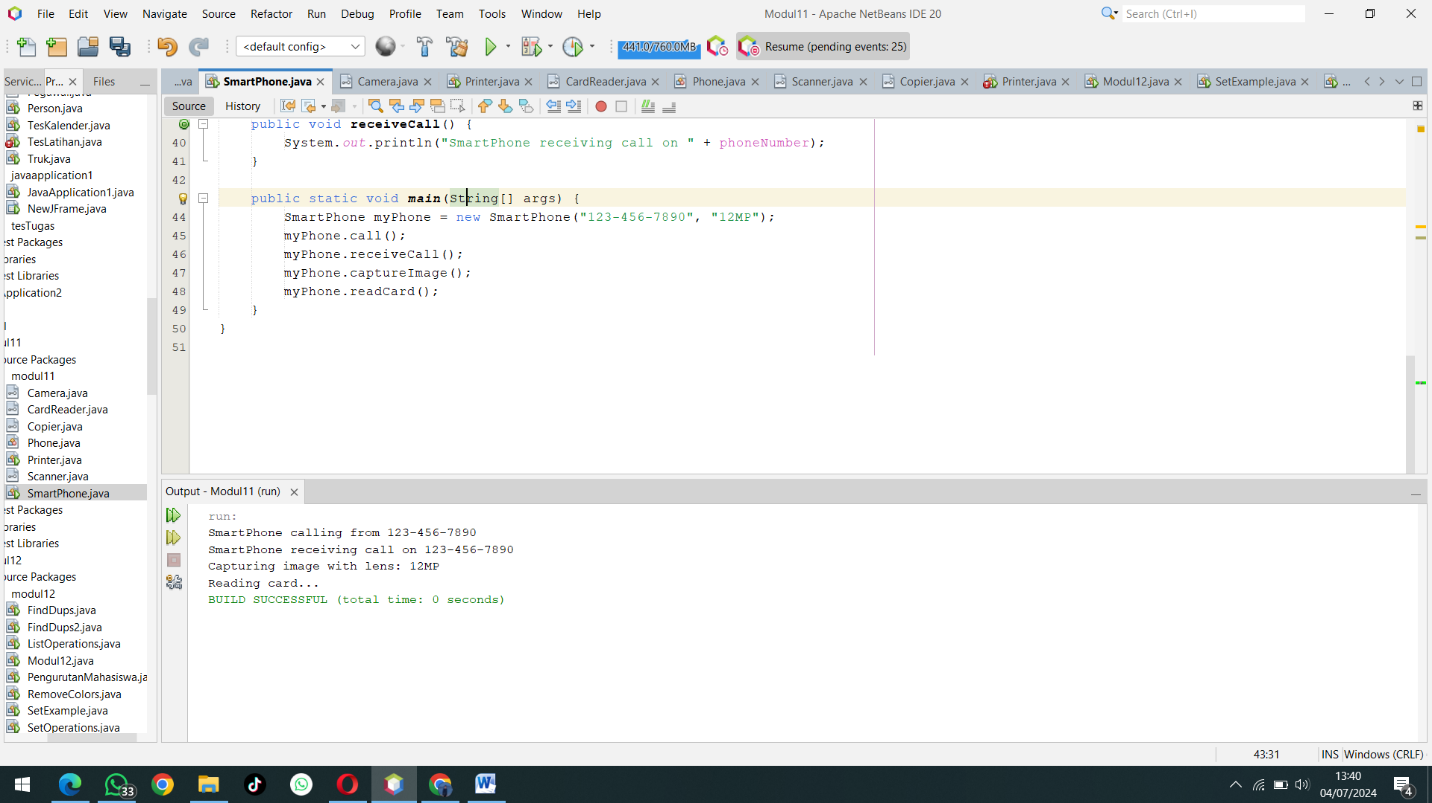
myPhone.captureImage();

myPhone.readCard();

}

}

**Output :**

****

1. Pewarisan (Inheritance)

* SmartPhone extends Phone:

SmartPhone mewarisi atribut dan metode dari kelas Phone. Ini berarti bahwa SmartPhone memiliki semua fungsionalitas dari Phone dan dapat menambah atau mengubah fungsionalitas tersebut. Misalnya, jika Phone memiliki metode call() dan receiveCall(), SmartPhone dapat menggunakan atau mengoverride metode ini.

1. Implementasi Interface

* SmartPhone implements Camera dan CardReader:

SmartPhone harus menyediakan implementasi untuk metode yang dideklarasikan dalam Camera dan CardReader. Ini berarti kelas SmartPhone harus mendefinisikan bagaimana captureImage() dan readCard() bekerja. Dengan demikian, SmartPhone dapat berfungsi sebagai kamera dan pembaca kartu, di samping fungsionalitas telepon.

1. Atribut dan Konstruktor

* Atribut lens:

Atribut lens digunakan untuk menyimpan informasi tentang lensa kamera. Ini menunjukkan bahwa SmartPhone memiliki fitur kamera dengan spesifikasi tertentu.

* Konstruktor:

Konstruktor SmartPhone(String phoneNumber, String lens) memanggil konstruktor superclass Phone dengan parameter phoneNumber dan menginisialisasi atribut lens.

1. Metode yang Di-Override

* call() dan receiveCall():

Metode ini di-override dari kelas Phone. SmartPhone memberikan implementasi khusus untuk metode ini yang sesuai dengan fungsionalitas tambahan perangkat.

* captureImage() dan readCard():

Metode ini diimplementasikan dari interface Camera dan CardReader. Implementasi metode ini menunjukkan bagaimana SmartPhone menangani operasi kamera dan pembaca kartu.

1. Metode main

* Pembuatan Instance dan Pengujian:

Di dalam metode main, instance dari SmartPhone dibuat dengan nomor telepon dan lensa spesifik.

* Metode call(), receiveCall(), captureImage(), dan readCard() dipanggil untuk menguji fungsionalitas yang ada di dalam SmartPhone.

1. **Tugas**

package modul11;

/\*\*

\*

\* @author Dio Ananda

\*/

//E TUGAS

public class Printer implements Scanner, Copier {

public void scanImage() {

System.out.println("Scanning image...");

}

@Override

public void copyImage() {

System.out.println("Copying image...");

}

public void printImage() {

System.out.println("Printing image...");

}

public static void main(String[] args) {

Printer myPrinter = new Printer();

myPrinter.scanImage();

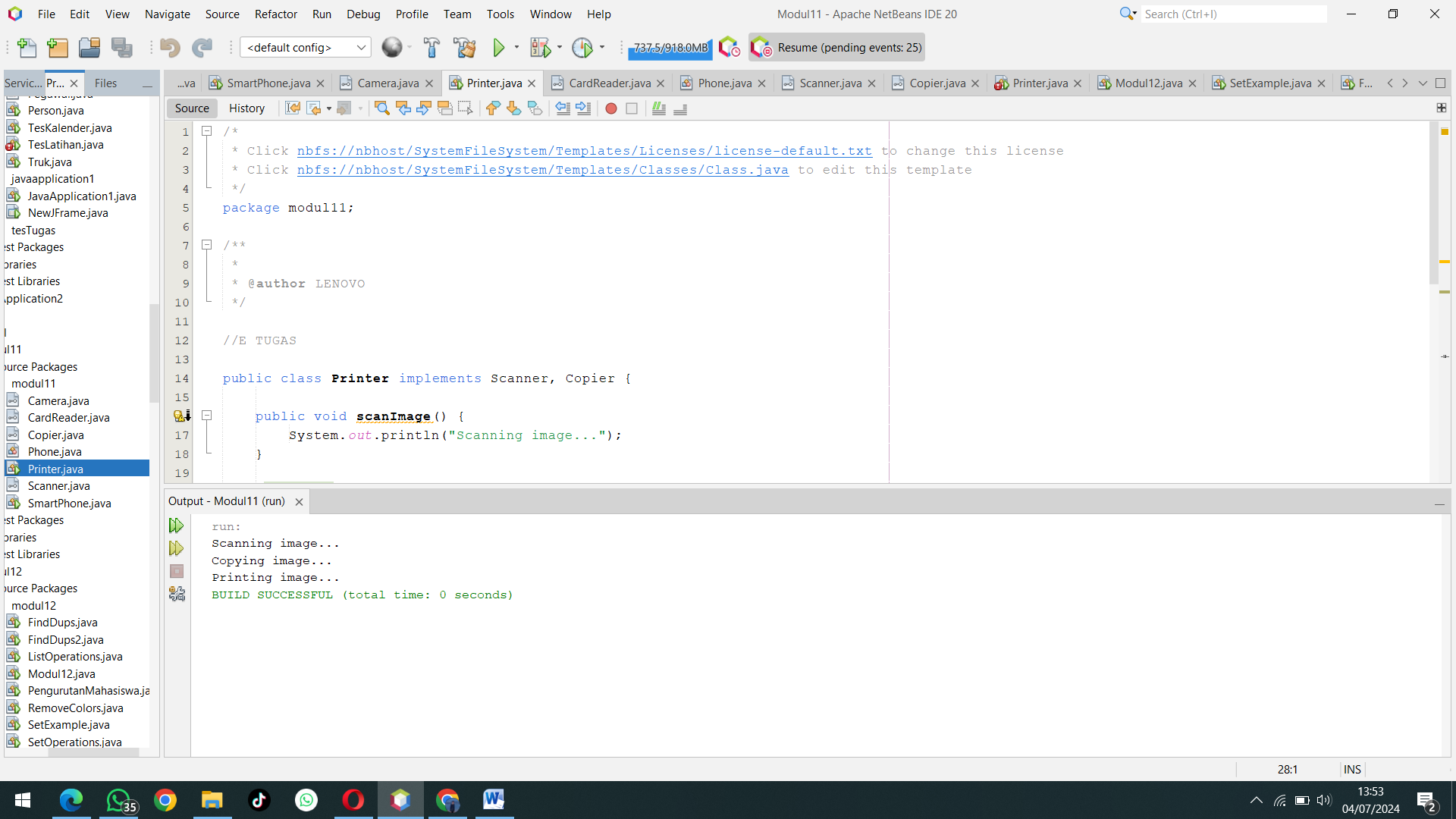
myPrinter.copyImage();

myPrinter.printImage();

}

}

**Output :**



**Penjelasan Kesalahan dan Perbaikannya**

1. Nama Interface:

* Kesalahan: Nama interface Scanner bertabrakan dengan nama kelas java.util.Scanner yang sudah ada dalam Java.
* Perbaikan: Mengganti nama interface Scanner menjadi ImageScanner untuk menghindari konflik nama.

1. Implementasi Metode dalam Interface:

* Kesalahan: Interface Scanner berisi implementasi metode scaneImage, yang tidak diperbolehkan di interface Java.
* Perbaikan: Menghapus implementasi dari interface dan hanya mendeklarasikan metode. Implementasi harus dilakukan di kelas yang mengimplementasikan interface tersebut.

1. Typo dalam Nama Metode:

* Kesalahan: Nama metode scaneImage di interface dan di kelas Printer tidak konsisten. Nama metode yang benar adalah scanImage.
* Perbaikan: Memperbaiki nama metode di interface dan kelas implementasi agar konsisten.

**Analisa**

* Interface: Interface di Java hanya berisi deklarasi metode tanpa implementasi (kecuali metode default atau statik yang diperkenalkan di Java 8). Interface digunakan untuk mendefinisikan kontrak yang harus diikuti oleh kelas yang mengimplementasikannya.
* Kelas yang Mengimplementasikan Interface: Kelas Printer mengimplementasikan interface Copier dan ImageScanner. Oleh karena itu, kelas Printer harus menyediakan implementasi untuk semua metode yang dideklarasikan dalam interface tersebut.
* Metode printImage: Kelas Printer menambahkan metode printImage yang tidak ada dalam interface. Ini valid dan menunjukkan bahwa kelas Printer memiliki fungsionalitas tambahan di luar yang didefinisikan oleh interface.