**Nama : Febby Novanti Azhara**

**NIM : 1182002002**

**Pemrograman Berorientasi Objek – TUGAS 3**

Buatlah sebuah program berisi contoh:

1. Kelas abstrak.

2. Interface

3. Beri rangkuman cara kerja program.

Jawab :

1. Kelas Abstrak

Kelas abstrak merupakan kelas yang masih dalam bentuk abstrak. Karena bentuknya masih abstrak, dia tidak bisa dibuat langsung menjadi objek. Sebuah kelas agar dapat disebut kelas abstrak setidaknya memiliki satu atau lebih method abstrak. Method abstrak adalah method yang tidak memiliki implementasi atau tidak ada bentuk konkritnya.

Berikut contoh program kelas abstrak :

Kelas komponen.java

package app.klsabstrak;

public abstract class komponen {

    //Method Abstrak

    abstract void bahankulit();

    abstract void bahanfla();

    abstract void proses\_pembuatan();

}

Kelas membuatkue.java

package app.klsabstrak;

public class membuatkue extends komponen{

    public static void main(String[] args){

        //Membuat Instance/Objek dari Class Mambuat\_Kue

        membuatkue kue = new membuatkue();

        kue.bahankulit();

        kue.bahanfla();

        kue.proses\_pembuatan();

    }

    @Override

    void bahankulit() {

        String bahan1 = "Tepung Terigu";

        String bahan2 = "Mentega";

        String bahan3 = "Telur";

        System.out.println("====== RESEP PIE SUSU ======");

        System.out.println("====== BAHAN-BAHAN ======");

        System.out.println("1."+bahan1);

        System.out.println("2."+bahan2);

        System.out.println("3."+bahan3);

    }

    @Override

    void bahanfla() {

        String bahanfla1 = "Susu Full Cream";

        String bahanfla2 = "Kuning Telur";

        String bahanfla3 = "Susu Kental Manis";

        System.out.println("====== ALAT-ALAT ======");

        System.out.println("1."+bahanfla1);

        System.out.println("2."+bahanfla2);

        System.out.println("3."+bahanfla3);

    }

    @Override

    void proses\_pembuatan() {

        System.out.println("====== PROSES ======");

        System.out.println("1.Membuat Kulit pie dengan mencapur bahan bahan kulit");

        System.out.println("2.Adonkan adonan kulit hingga kalis");

        System.out.println("3.Tempatkan adonan kulit diatas teflon(ratakan membentuk bulat teflon)");

        System.out.println("4.Tusuk - tusuk kulit yang sudah di ratakan di teflon dengan garpu");

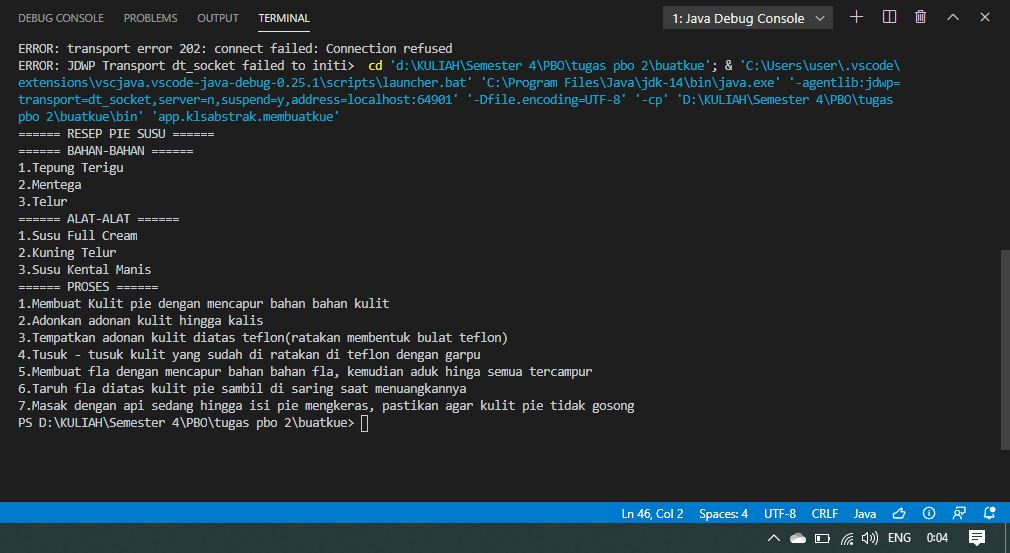
        System.out.println("5.Membuat fla dengan mencapur bahan bahan fla, kemudian aduk hinga semua tercampur");

        System.out.println("6.Taruh fla diatas kulit pie sambil di saring saat menuangkannya");

        System.out.println("7.Masak dengan api sedang hingga isi pie mengkeras, pastikan agar kulit pie tidak gosong");

    }

}

Berikut hasil run dari kodingan di atas :

2. Interface

Interface terdiri dari dua kata yaitu *Inter*(antar) dan *Face*(muka), jadi interface merupakan antramuka yang berfungsi sebagai penghubung antara sesuatu yang abstrak dengan sesuatu yang nyata.

Berikut contoh program interface :

Kelas Phone.java

package app.interfacecoba;

public interface Phone {

    int MAX\_VOLUME = 100;

    int MIN\_VOLUME = 0;

    void powerOn();

    void powerOff();

    void volumeUp();

    void volumeDown();

}

Kelas PhoneUser.java

package app.interfacecoba;

public class PhoneUser {

    private Phone phone;

    public PhoneUser(Phone phone) {

        this.phone = phone;

    }

    void turnOnThePhone(){

        this.phone.powerOn();

    }

    void turnOffThePhone(){

        this.phone.powerOff();

    }

    void makePhoneLouder(){

        this.phone.volumeUp();

    }

    void makePhoneSilent(){

        this.phone.volumeDown();

    }

}

Kelas Samsung.java

package app.interfacecoba;

public class Samsung implements Phone {

    private int volume;

    private boolean isPowerOn;

    public Samsung() {

        // set volume awal

        this.volume = 50;

    }

    @Override

    public void powerOn() {

        isPowerOn = true;

        System.out.println("Handphone menyala...");

        System.out.println("Selamat datang di SAMSUNG");

        System.out.println("Android version 29");

    }

    @Override

    public void powerOff() {

        isPowerOn = false;

        System.out.println("Handphone dimatikan");

    }

    @Override

    public void volumeUp() {

        if (isPowerOn) {

            if (this.volume == MAX\_VOLUME) {

                System.out.println("Volume FULL!!");

                System.out.println("sudah " + this.getVolume() + "%");

            } else {

                this.volume += 10;

                System.out.println("Volume sekarang: " + this.getVolume());

            }

        } else {

            System.out.println("Nyalakan dulu donk HP-nya!!");

        }

    }

    @Override

    public void volumeDown() {

        if (isPowerOn) {

            if (this.volume == MIN\_VOLUME) {

                System.out.println("Volume = 0%");

            } else {

                this.volume -= 10;

                System.out.println("Volume sekarang: " + this.getVolume());

            }

        } else {

            System.out.println("Nyalakan dulu donk HP-nya!!");

        }

    }

    public int getVolume() {

        return this.volume;

    }

}

Kelas Main.java

package app.interfacecoba;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        // membuat objek HP

        Phone samsungA50 = new Samsung();

        // membuat objek user

        PhoneUser dian = new PhoneUser(samsungA50);

        // kita coba nyalakan HP-nya

        dian.turnOnThePhone();

        // biar enak, kita buat dalam program

        Scanner scan = new Scanner(System.in);

        String aksi;

        while (true) {

            System.out.println("=== APLIKASI INTERFACE ===");

            System.out.println("[1] Nyalakan HP");

            System.out.println("[2] Matikan HP");

            System.out.println("[3] Perbesar Volume");

            System.out.println("[4] Kecilkan Volume");

            System.out.println("[0] Keluar");

            System.out.println("--------------------------");

            System.out.print("Pilih aksi> ");

            aksi = scan.nextLine();

            if(aksi.equalsIgnoreCase("1")){

                dian.turnOnThePhone();

            } else if (aksi.equalsIgnoreCase("2")){

                dian.turnOffThePhone();

            } else if (aksi.equalsIgnoreCase("3")){

                dian.makePhoneLouder();

            } else if (aksi.equalsIgnoreCase("4")){

                dian.makePhoneSilent();

            } else if (aksi.equalsIgnoreCase("0")){

                System.exit(0);

            } else {

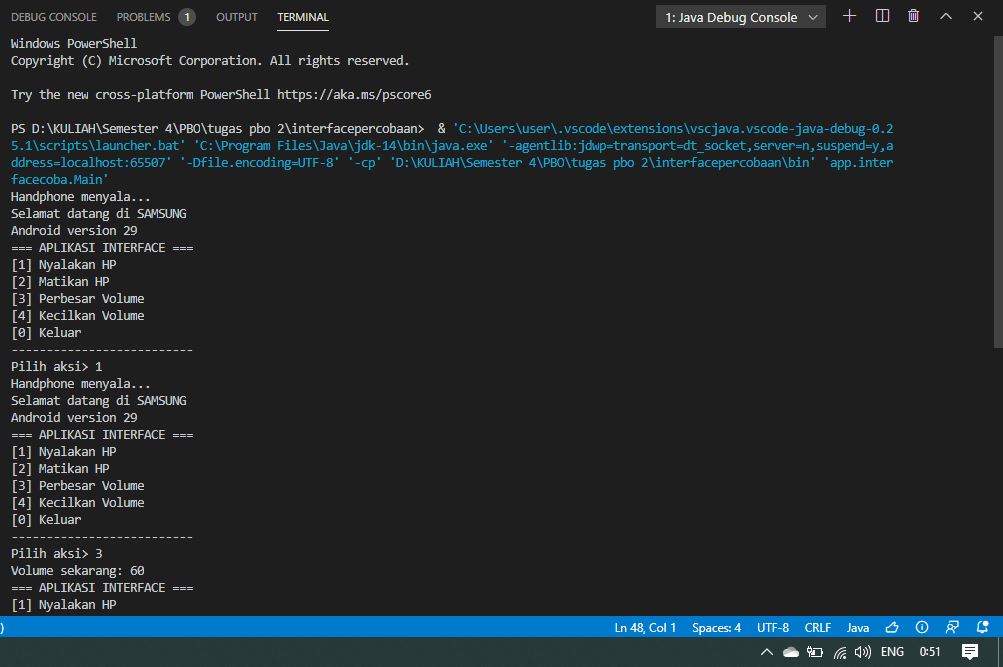
                System.out.println("Kamu memilih aksi yang salah!");

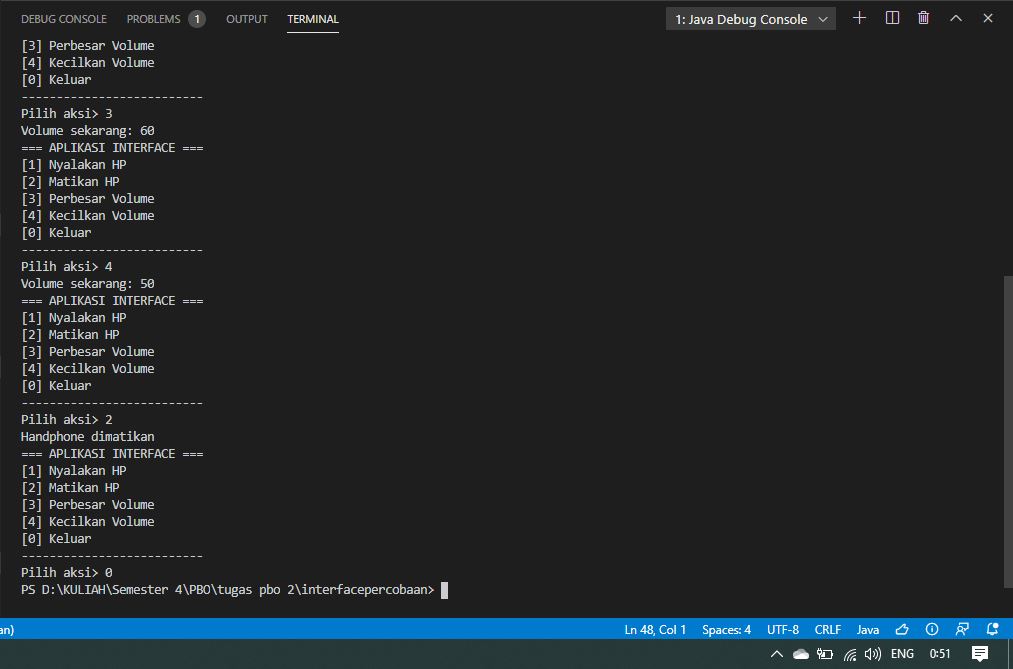
            }

        }

    }

}

Berikut hasil run program diatas :



3. Rangkuman Cara kerjanya

**Pada program Kelas Abstrak**

1. Pertama buka class *komponen.java* pada project kalian, untuk mengubahnya menjadi class abstrak, kita perlu menambahkan keyword *abstract*dibelakang nama classnya, lalu akan kita pasang beberapa atribut seperti method pada class tersebut. Jadi intinya method abstrak itu adalah sebuah method yang tidak tahu mau kita apakan nantinya, sebuah class dan method abstrak dibuat oleh seorang programmer sebagai acuan atau gambaran dari program yang ingin mereka buat.
2. Selanjutnya buka class membuatkue.java, lalu kita *extends*dan implementasikan semua method tersebut pada class tersebut. Beberapa method yang sudah di implementasikan akan terlihat seperti dibawah ini :

package WILDAN\_TECHNO\_ART;

public class membuat\_kue extends komponen{

public static void main(String[] args){

}

@Override

void bahan\_bahan() {

}

@Override

void peralatan() {

}

@Override

void proses\_pembuatan() {

}

}

1. Dengan adanya class dan method abstrak, si programmer jadi tahu statement-statement apa saya yang harus dibuatnya, kalian dapat mengisi statement pada method-method tersebut seperti kodingan program diatas.
2. Run pada kelas main.

**Pada program Interface**

1. Pertama – tama buat kelas Phone.java dengan kodigan seperti diatas.
2. Berikutnya, buat kelas PhoneUser.java
3. Berikutnya kita akan membuat class implementasi dari interface Phone. Kita coba buat satu saja dulu. Buat class baru dengan nama samsung, kemudian isi kodenya seperti diatas.
4. Terakhir, buatlah class Main dengan isi kodingan seperti diatas.
5. Sekarang coba jalankan class Main.

Sumber :

https://www.petanikode.com/java-oop-interface/

https://www.wildantechnoart.net/2017/11/class-abstract-dan-interface-pada-java.html