

LAPORAN PRAKTIKUM

Pertemuan 3

Prinsip Perancangan Class



Disusun Oleh :

Ibrohim Huzaimi

19102005

Merlinda Wibowo, S.T., M.Phil.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM

PURWOKERTO

2021

I. Tujuan

Mahasiswa diharapkan mampu memahami tentang perancangan kelas yang baik serta menerjemahkannya dalam bahasa pemrograman.

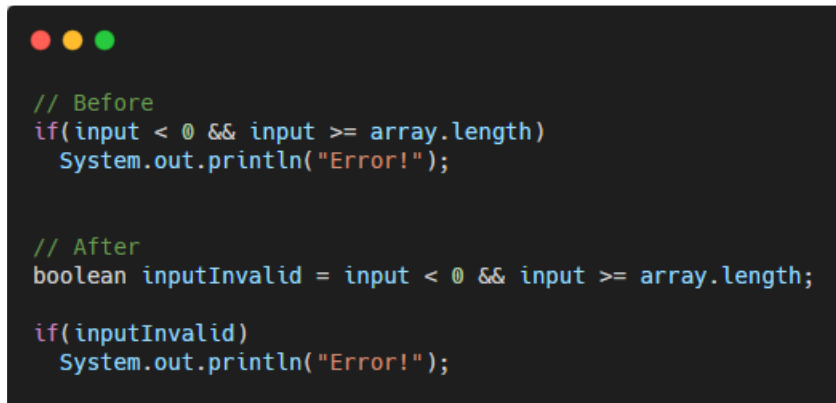
II. Dasar Teori

Empat prinsip utama dalam Pemrograman Berorientasi Objek:

1. Abstraction

Abstraksi berarti menggunakan hal-hal sederhana untuk mewakili hal-hal rumit.

Perhatikan contoh sederhana berikut:



```
// Before
if(input < 0 && input >= array.length)
    System.out.println("Error!");

// After
boolean inputInvalid = input < 0 && input >= array.length;

if(inputInvalid)
    System.out.println("Error!");
```

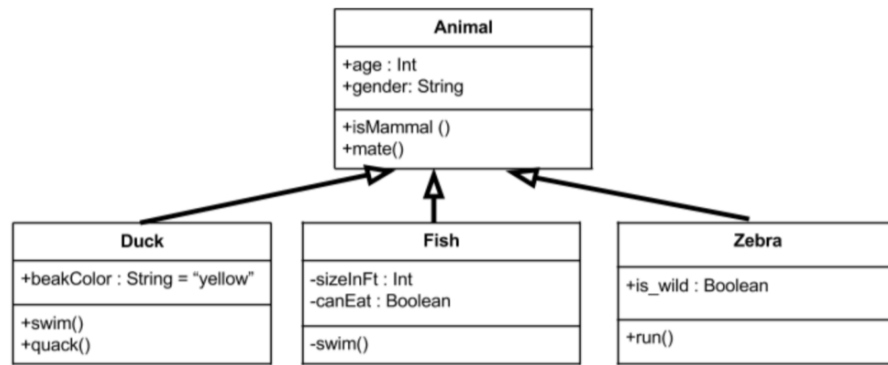
Pada contoh di atas, pada if pertama, orang lain yang belum paham tentang kode kita mungkin akan bertanya-tanya kondisi apa yang sedang dicek di if tersebut. Sedangkan pada if kedua, jika kita menunjukkan kode kita pada orang awam pun, orang itu akan paham bahwa kode tersebut sedang mengecek apakah input yang diterima program valid atau invalid.

2. Encapsulation

Enkapsulasi berarti menyembunyikan data-data kompleks di balik modifier Private dan menyediakan sedikit method Public sebagai “jalan” untuk mengakses data-data Private tersebut. Bayangkan sebuah TV. Anggaplah tombol-tombol di depan TV yang dapat kita tekan adalah method Public milik TV tersebut, sedangkan proses kompleks yang terjadi di belakang layar saat kita menekan tombol-tombol tersebut adalah method dan atribut Private. Sebagai pemilik TV, kita tahu apa fungsi dari masing-masing tombol, tetapi kita tidak tahu dan tidak peduli apa proses kompleks yang terjadi di belakang layar saat kita menekan tombol tersebut.

3. Inheritance

Inheritance (arti: Pewarisan) merupakan konsep menurunkan atribut dan method milik sebuah class ke class lain. Perhatikan contoh class diagram berikut:



Pada contoh di atas, class **Animal** disebut Parent class atau Superclass, sedangkan class **Duck**, **Fish**, dan **Zebra** disebut Child class atau Subclass. Ketiga Child class tersebut akan memiliki SEMUA atribut dan method milik Parent class.

Dalam merancang class, jika kita menemukan ada beberapa class dengan atribut dan method yang konteksnya sama, lebih baik menggunakan konsep Pewarisan saja. Salah satu keuntungannya adalah jika kita sewaktu-waktu ingin mengubah kode kita, kita hanya perlu mengubah kode kita satu kali saja, sedangkan jika kita menuliskan class berbeda tanpa ada hubungan Inheritance, jika kita ingin melakukan perubahan kita harus merubah isi dari semua class tersebut satu persatu.

4. Polymorphism

Polimorfisme merupakan konsep pembuatan method berbeda dengan nama yang sama. Polimorfisme dibagi menjadi dua, yaitu:

a. Overloading

Overloading adalah ketika di sebuah class ada 2 atau lebih method dengan nama sama namun masing-masing memiliki parameter berbeda. Walau namanya sama, program tetap mengerti harus menjalankan method yang mana berdasarkan parameter yang dimasukan saat pemanggilan method. Contoh sederhana method overloading:

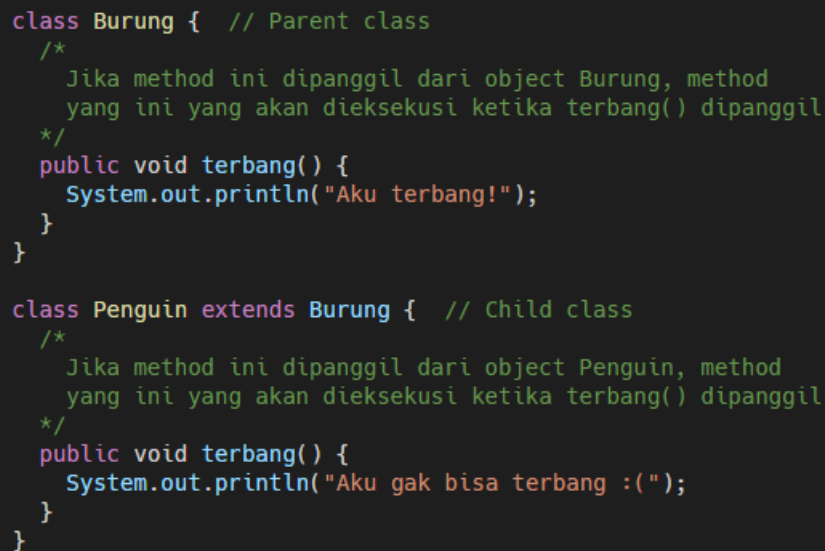
```
class Hero {
    // Akan dieksekusi jika tipe data parameter adalah Armor
    public void equip(Armor armor) {}

    // Akan dieksekusi jika tipe data parameter adalah Weapon
    public void equip(Weapon weapon) {}
}
```

b. Overriding

Overriding adalah ketika sebuah Child class memiliki method dengan nama yang sama dengan Parent class-nya, namun dengan isi yang berbeda (parameter boleh sama). Nantinya program akan memprioritaskan method milik Child class ini jika ternyata ketahuan Child class memiliki method yang di-Override dari Parent-nya.

Contoh sederhana method overriding:



```
class Burung { // Parent class
    /*
     * Jika method ini dipanggil dari object Burung, method
     * yang ini yang akan dieksekusi ketika terbang() dipanggil
     */
    public void terbang() {
        System.out.println("Aku terbang!");
    }
}

class Penguin extends Burung { // Child class
    /*
     * Jika method ini dipanggil dari object Penguin, method
     * yang ini yang akan dieksekusi ketika terbang() dipanggil
     */
    public void terbang() {
        System.out.println("Aku gak bisa terbang :(");
    }
}
```

(Inheritance dan Polymorphism akan dibahas lebih lanjut di modul selanjutnya)

Beberapa prinsip-prinsip perancangan kelas yang perlu diperhatikan, antara lain:

1. Constructor. Method ini digunakan untuk inisialisasi atau mempersiapkan data untuk objek.
2. Visibilitas Bagi Atribut dan Method. Diperlukan kontrol akses untuk mengatur siapa saja yang dapat mengakses atau mengubah nilai dari atribut atau method dalam kelas tersebut sehingga penyalahgunaan atribut atau method dapat dihindari. Kontrol akses yang sering digunakan dalam perancangan suatu kelas sudah ada pada modul minggu lalu.
3. Fungsi Accessor dan Mutator. Fungsi accessor merupakan fungsi untuk mendapatkan properti dari suatu objek dan mengembalikan nilai atau value dari suatu atribut (get). Sedangkan fungsi mutator mengubah properti dari suatu objek dan mengubah nilai atau value dari sebuah atribut (set).

4. Method dan Operator Overloading. Method-method tersebut harus dapat dibedakan antara satu dengan yang lain dalam jumlah dan atau tipe datanya.

5. Melewatkan Argumen/Parameter ke Method. Cara melewati argumen/parameter ini dapat dilakukan dengan melewati secara nilai (pass by value) dan melewati secara referensi (pass by reference).

```
// VALUE
int x = 7;
int y = x;

System.out.println(x); // Output: 7
System.out.println(y); // Output: 7

y = 9;

System.out.println(x); // Output: 7
System.out.println(y); // Output: 9

// REFERENCE
int[] x = {2, 3, 5, 7};
int[] y = x;

for(int e : x)
    System.out.print(e + " "); // Output: 2, 3, 5, 7
System.out.println();

for(int e : y)
    System.out.print(e + " "); // Output: 2, 3, 5, 7
System.out.println();

y[0] = 1;

for(int e : x)
    System.out.print(e + " "); // Output: 1, 3, 5, 7
System.out.println();

for(int e : y)
    System.out.print(e + " "); // Output: 1, 3, 5, 7
System.out.println();
```

Pass by value diterapkan pada argumen bertipe data primitif dan prosesnya hanya menyalin isi memori (yang telah dialokasikan untuk suatu variabel), dan kemudian menyampaikan salinan tersebut kepada method yang bersangkutan (isi memory merupakan data sesungguhnya yang akan dioperasikan), karena yang disampaikan hanya salinan dari isi memori, maka perubahan yang terjadi pada variabel akibat proses didalam method tidak mempengaruhi nilai variabel asalnya

di dalam memori. Sedangkan, pass by reference diterapkan pada argumen bertipe data array atau objek dan isi memori pada variabel array atau objek merupakan penunjuk alamat memori yang mengandung data sesungguhnya yang akan dioperasikan.

6. Responsibility Driven Design. Semua fungsi dan method yang ada harus mencerminkan perilaku lengkap yang dimiliki kelas tersebut. Semua fungsi/method tersebut bertanggung jawab terhadap maintenance atribut yang dimiliki kelas.

III. Praktikum

A. GUIDED

Buatlah package baru dengan format com.nama.praktikumpbo.pertemuan3.guided, lalu ikuti langkah langkah di bawah:

DASAR ARRAY

Untuk membuat array kosong di Java, pertama kita tuliskan dulu tipe data yang nantinya akan ditampung array yang kita buat. Setelah itu, jangan lupa berikan kurung siku terbuka dan tertutup []. Setelah itu, tulis nama yang akan kita gunakan untuk mereferensikan array tersebut. Terakhir, kita harus menuliskan tipe data array tersebut lagi dan kali ini di dalam kurung siku kita tuliskan jumlah elemen (isi) maksimum yang dapat ditampung oleh array tersebut.

Buatlah sebuah package baru di dalam package guided dan beri nama array. Buat class baru di dalam package array tersebut dan beri nama ArrayBasic. Salin kode berikut:

```
public class ArrayBasic {
    public static void main(String[] args) {
        // tipeData[] namaArray = new tipeData[panjangArray];
        int[] numbers = new int[10];

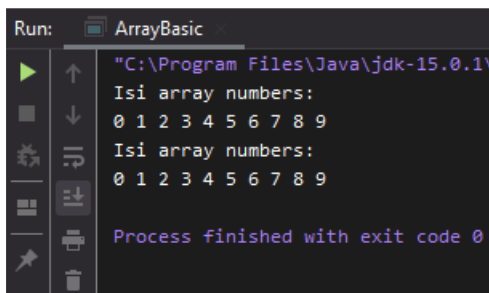
        // Array diisi dengan bilangan 0 sampai 9
        for(int i=0; i<10; i++)
            numbers[i] = i;

        // Output seluruh isi array (Cara 1)
        System.out.println("Isi array numbers:");
        for(int i=0; i<numbers.length; i++)
            System.out.print(numbers[i] + " ");
        System.out.println();

        // Output seluruh isi array (Cara 2)
        System.out.println("Isi array numbers:");
        for(int number : numbers)
            System.out.print(number + " ");
        System.out.println();
    }
}
```

Pada contoh di atas, array kita beri nama numbers. dan array numbers ini dapat menampung sejumlah 10 bilangan bulat. Untuk mencetak isi array, kita bisa memanfaatkan for loop. Ada 2 variasi for loop, anda bebas ingin menggunakan yang mana.

Hasil running kodenya:

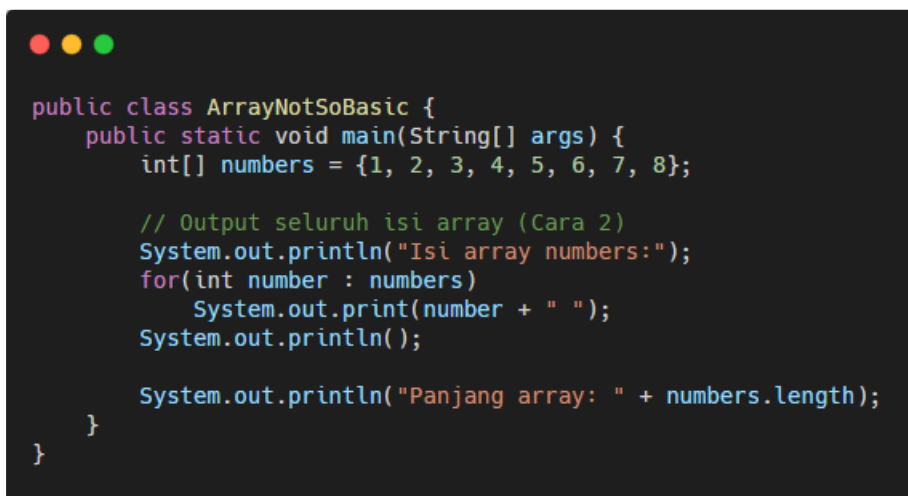


```
Run: ArrayBasic
"C:\Program Files\Java\jdk-15.0.1\
Isi array numbers:
0 1 2 3 4 5 6 7 8 9
Isi array numbers:
0 1 2 3 4 5 6 7 8 9
Process finished with exit code 0
```

ARRAY DENGAN NILAI YANG SUDAH DITENTUKAN DARI AWAL

Untuk membuat sebuah array yang sudah langsung memiliki nilai sejak ia dibuat, pertama tuliskan tipe datanya terlebih dahulu diikuti dengan 2 kurung siku []. Setelah itu jangan lupa tulis nama array-nya. Lalu, tulis = {}. Nah, untuk memasukan sesuatu ke array ini, anda tinggal mengetikan saja apa yang anda masukan ke array tersebut ke dalam {} tadi. Gunakan tanda koma (,) sebagai pemisah antara elemen/nilai index satu dengan index lainnya.

Di package yang sama, buatlah class baru lagi dan beri nama ArrayNotSoBasic, lalu salin kode berikut:



```
public class ArrayNotSoBasic {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8};

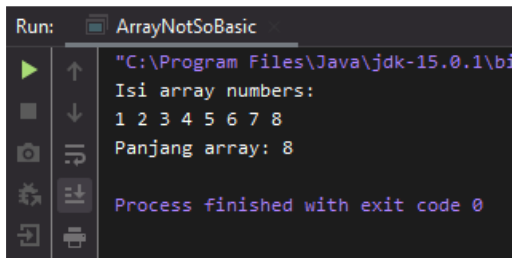
        // Output seluruh isi array (Cara 2)
        System.out.println("Isi array numbers:");
        for(int number : numbers)
            System.out.print(number + " ");
        System.out.println();

        System.out.println("Panjang array: " + numbers.length);
    }
}
```

Pada contoh di atas, kita membuat array dengan nama numbers yang menampung bilangan bulat 1, 2, 3, 4, 5, 6, 7, 8. Jika anda membuat array dengan cara seperti ini, anda tidak perlu menuliskan berapa panjang array-nya, karena panjang array akan menyesuaikan, tergantung anda memasukan berapa nilai atau object ke dalam array tersebut saat anda membuatnya.

Untuk mengecek panjang array, anda bisa menuliskan nama array tersebut, lalu menambahkan .length setelahnya. Pada contoh di atas, karena nama array-nya adalah numbers, jadi kita menuliskan numbers.length

Hasil running kodenya:

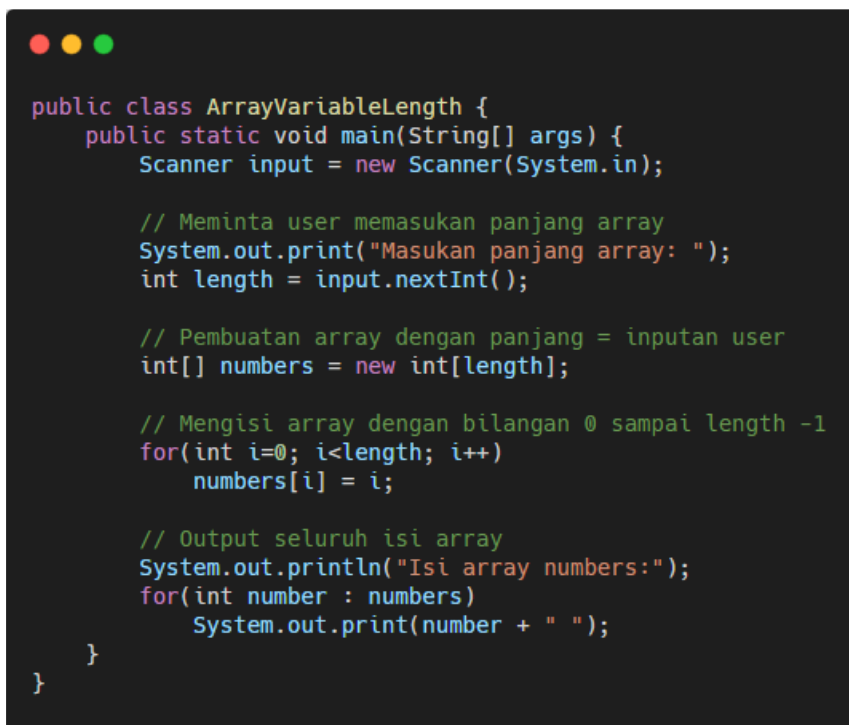


```
Run: ArrayNotSoBasic
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"
Isi array numbers:
1 2 3 4 5 6 7 8
Panjang array: 8
Process finished with exit code 0
```

ARRAY KOSONG DENGAN PANJANG BERVARIASI

Selain 2 cara di atas, ada cara lain lagi yang bisa anda gunakan untuk membuat array. Pada cara ini, panjang array akan ditentukan oleh variable, yang mana berarti kita bisa membuat program yang memiliki array dengan panjang yang berbeda tergantung inputan user!

Masih di package yang sama, buatlah class baru dan beri nama `ArrayVariableLength`, lalu salin kode berikut:



```
public class ArrayVariableLength {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Meminta user memasukan panjang array
        System.out.print("Masukan panjang array: ");
        int length = input.nextInt();

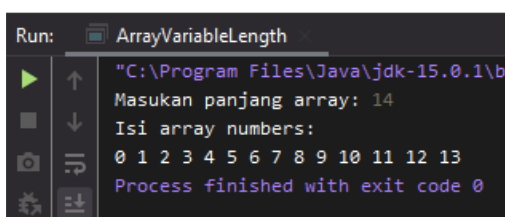
        // Pembuatan array dengan panjang = inputan user
        int[] numbers = new int[length];

        // Mengisi array dengan bilangan 0 sampai length -1
        for(int i=0; i<length; i++)
            numbers[i] = i;

        // Output seluruh isi array
        System.out.println("Isi array numbers:");
        for(int number : numbers)
            System.out.print(number + " ");
    }
}
```

Prinsipnya sama seperti metode pembuatan array pertama, hanya saja kita memasukan sebuah variable ke dalam [], bukan angka.

Hasil running kodenya:



```
Run: ArrayVariableLength
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"
Masukan panjang array: 14
Isi array numbers:
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Process finished with exit code 0
```

Sekarang, kita akan mencoba merapikan kode ArrayVariableLength ini. Menggunakan prinsip Enkapsulasi yang kita pelajari di awal pertemuan ini, jika ada serangkaian kode yang memiliki konteks atau tujuan yang sama, alangkah baiknya jika kita buat rangkaian kode tersebut menjadi sebuah fungsi. Mari kita coba terapkan prinsip tersebut! Source kode ArrayVariableLength setelah dirapikan:

```
public class ArrayVariableLength {
    public static void isiArray(int[] array) {
        for(int i=0; i<array.length; i++)
            array[i] = i;
    }

    public static void lihatIsiArray(int[] array) {
        System.out.println("Isi array numbers:");
        for(int number : array)
            System.out.print(number + " ");
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Meminta user memasukan panjang array
        System.out.print("Masukan panjang array: ");
        int length = input.nextInt();

        // Pembuatan array dengan panjang = inputan user
        int[] numbers = new int[length];

        isiArray(numbers);
        lihatIsiArray(numbers);
    }
}
```

PENGUNAAN ARRAY DALAM PBO

Buatlah package baru di luar package array tetapi masih di dalam package guided. Namai package tersebut restaurant. Anda berada di jalan yang benar jika package anda terlihat seperti ini: com.nama.praktikumpbo.pertemuan3.guided.restaurant

Di dalam package tersebut, buat class baru bernama Item, lalu salin kode berikut:

```
public class Item {
    private String nama;
    private int harga;
    private String jenis;

    public Item(String nama, int harga, String jenis) {
        this.nama = nama;
        this.harga = harga;
        this.jenis = jenis;
    }

    public String getNama() {
        return nama;
    }

    public int getHarga() {
        return harga;
    }

    public String getJenis() {
        return jenis;
    }

    public String showDetail() {
        // Contoh output: Mendoan (Makanan) - Rp 1000
        return getNama() + " (" + getJenis() + ") - Rp " + getHarga();
    }
}
```

Masih di package yang sama, buat class baru dan beri nama Pembeli. Salin kode berikut:

```

public class Pembeli {
    private String nama;
    private Item[] historyPemesanan = new Item[99];
    private int itemsOrdered = 0;

    public Pembeli(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void lakukanPemesanan(Item item) {
        historyPemesanan[itemsOrdered] = item;
        itemsOrdered++;
        System.out.println(item.getNama()
            + " berhasil ditambahkan ke history pemesanan sdr "
            + this.getNama());
    }

    public void lihatHistoryPemesanan() {
        if(itemsOrdered > 0) {
            System.out.println("History Pemesanan sdr "
                + this.getNama() + ":");
            for(int i=0; i<itemsOrdered; i++) {
                Item pesanan = historyPemesanan[i];
                System.out.println("- " + pesanan.getNama()
                    + " (" + pesanan.getJenis()
                    + ") - Rp " + pesanan.getHarga());
            }
            System.out.println();
        }
        else {
            System.out.println("Anda belum pernah memesan apa-apa!");
            System.out.println();
        }
    }

    public void kosongkanHistory() {
        System.out.println("Mereset history...");
        this.historyPemesanan = new Item[99];
        this.itemsOrdered = 0;
        System.out.println("History pemesanan berhasil dikosongkan!");
        System.out.println();
    }
}

```

Terakhir, buat class baru lagi bernama Main masih di package yang sama, lalu salin kode berikut:

```

import java.util.Scanner;

public class Main {
    private static void tampilkanMenu() {
        int nomorItem = 1;
        for(Item item : menu) {
            System.out.println(nomorItem + ". " + item.showDetail());
            nomorItem++;
        }
        System.out.println();
    }

    private static Item[] menu = {
        new Item("Mendoan", 1_000, "Makanan"),
        new Item("Es Teh", 2_500, "Minuman"),
        new Item("Aneka Sayur", 5_000, "Makanan"),
        new Item("Paha Ayam", 7_500, "Makanan"),
        new Item("Sate Ayam", 15_000, "Makanan")
    };

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        boolean keepLooping = true;

        System.out.print("Masukan nama: ");
        String namaPembeli = input.nextLine();
        Pembeli pembeli = new Pembeli(namaPembeli);

        System.out.println();
        do {
            System.out.println("==> Menu Utama <==");
            System.out.println("1) Pesan makanan");
            System.out.println("2) Lihat history pemesanan");
            System.out.println("3) Kosongkan history pemesanan");
            System.out.println("4) Keluar");
            System.out.print("> ");

            int nav = input.nextInt();
            System.out.println();

            switch(nav) {
                case 1: // Pesan makanan
                    tampilkanMenu();
                    System.out.print("Pilih makanan yang ingin dipesan: ");
                    int indexItem = input.nextInt() - 1;

                    // Jika user menginput yang aneh-aneh
                    if(indexItem < 0 || indexItem >= menu.length)
                        System.out.println("Error: Mohon masukan yang benar!");

                    // Jika user menginput yang benar
                    else
                        pembeli.lakukanPemesanan(menu[indexItem]);

                    System.out.println();
                    break;

                case 2: // Lihat history
                    pembeli.lihatHistoryPemesanan();
                    break;

                case 3: // Kosongkan history
                    pembeli.kosongkanHistory();
                    break;

                case 4: // Keluar
                    System.out.println("Terima kasih sudah mampir! ^^");
                    keepLooping = false;
                    break;

                default: // Jika user menginput yang aneh-aneh
                    System.out.println("Tolong hanya masukan angka 1-4 saja!");
                    System.out.println();
                    break;
            }
        } while(keepLooping);
    }
}

```

IV. UNGUIDED

Seperti biasa, buat package baru di dalam package pertemuan3 dan beri nama unguided. Anda berada di jalan yang benar jika package anda terlihat seperti ini: com.nama.praktikumpbo.pertemuan3.unguided ;)

1. Buatlah sebuah program sederhana dengan tema bebas yang melibatkan pembuatan class dan object! Syarat dan ketentuan:

- Minimal 2 file, satu untuk class object dan satu untuk class main.
- Sertakan source code dan screenshot output saat running program di laporan praktikum.
- Jangan lupa beri penjelasan tentang alur kerja program di laporan.
- Nilai plus jika program memiliki menu.
- Nilai plus plus jika source code rapi, enak dilihat, dan mudah dipahami saat pertama kali dibaca.

Jawab:

Pada program aplikasi kali ini saya membuat program aplikasi matematika sederhana untuk menghitung keliling atau luas bangun datar. Bangun datar tersebut meliputi persegi, persegi panjang, lingkaran dan segitiga.

Pertama, buat sebuah class bernama lingkaran, dengan source code seperti dibawah:

```
public class Lingkaran {
    int r;
    double keliling,a, luas;

    public Lingkaran(int r) {
        this.r = r;
    }

    public Lingkaran() {
    }

    public void setR(int r) {
        this.r = r;
    }

    double kelilingLingkaran(){
        keliling = 2 * 3.14 * r;
        a = (keliling*100)/100;
        System.out.println("Keliling lingkaran adalah: " + a + "cm");
        return keliling;
    }

    double luasLingkaran(){
        luas = 3.14*r*r;
        System.out.println("Luas lingkaran adalah: " + luas + "cm^2");
        return luas;
    }
}
```

Pada class lingkaran, terdapat tipe data untuk menyimpan variabel jari jari, luas dan juga keliling. Dengan metodh **kelilingLingkaran**, **luasLingkaran**.

Lalu buat class baru lagi yaitu persegi, dan copy kode tersebut seperti dibawah

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.baim.pertemuan3.unguided;
|
/**
 *
 * @author ASUS
 */
public class Persegi {
    int sisi, keliling, luas;

    public Persegi(int sisi) {
        this.sisi = sisi;
    }

    public void setSisi(int sisi) {
        this.sisi = sisi;
    }

    int keliling(){
        keliling = sisi*4;
        System.out.println("Keliling persegi adalah: " + keliling + "cm");
        return keliling;
    }

    int luas(){
        luas = sisi * sisi;
        System.out.println("Luas persegi adalah: " + luas + "cm^2");
        return luas;
    }
}
```

Disitu dideklarasikan variabel sisi, keliling, luas dengan tipe data integer. Dengan methode keliling dan luas.

Lalu buat lagi class baru bernama persegi panjang, dengan source code seperti dibawah ini

```

public class PersegiPanjang {
    int luas, keliling, panjang, lebar;

    public PersegiPanjang(int panjang, int lebar) {
        this.panjang = panjang;
        this.lebar = lebar;
    }

    public void setPanjang(int panjang) {
        this.panjang = panjang;
    }

    public void setLebar(int lebar) {
        this.lebar = lebar;
    }

    int kelilingPersegiPanjang(){
        keliling = 2*(panjang+lebar);
        System.out.println("Keliling Persegi Panjang Adalah: " + keliling + "cm");
        return keliling;
    }

    int luasPersegiPanjang(){
        luas = panjang * lebar;
        System.out.println("Luas Persegi Panjang Adalah: " + luas + "cm^2");
        return luas;
    }
}

```

Pada class diatas terdapat variable luas, keliling, panjang, lebar dengan tipe data integer. Lalu methodenya adalah kelilingPersegiPanjang dan juga luasPersegiPanjang.

Lalu buat lagi class bernama Segitiga dengan source code seperti dibawah

```

public class Segitiga {
    int sisiA, sisiB, sisiC, alas, tinggi,
        a, b, c, kelilingSegitiga;
    double luas;

    public Segitiga(int sisiA, int sisiB, int sisiC) {
        this.sisiA = sisiA;
        this.sisiB = sisiB;
        this.sisiC = sisiC;
    }

    public Segitiga(int alas, int tinggi) {
        this.alas = alas;
        this.tinggi = tinggi;
    }

    public void setAlas(int alas) {
        this.alas = alas;
    }

    public void setTinggi(int tinggi) {
        this.tinggi = tinggi;
    }

    public void setSisiA(int sisiA) {
        this.sisiA = sisiA;
    }

    public void setSisiB(int sisiB) {
        this.sisiB = sisiB;
    }
}

```



```

public void setSisiC(int sisiC) {
    this.sisiC = sisiC;
}

int keliling(){
    kelilingSegitiga = sisiA+sisiB+sisiC;
    System.out.println("Keliling segitiga adalah: " + kelilingSegitiga + "cm");
    return kelilingSegitiga;
}

double luas(){
    this.alas = alas;
    this.tinggi = tinggi;

    luas = 0.5 * alas * tinggi;
    System.out.println("Luas segitiga adalah: " + luas + "cm^2");
    return luas;
}

```

Pada class diatas mempunyai methodode yaitu keliling dan juga luas, yang berfungsi untuk menghitung kelililing dan juga luas.

Lalu buat main class, dan ketik kode seperti dibawah:

```

package com.baim.pertemuan3.unguided;

import java.util.Scanner;

/**
 *
 * @author ASUS
 */
public class main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        boolean keeplooping = true;

        do{
            Scanner input = new Scanner(System.in);
            System.out.println("=====");
            System.out.println("Selamat Datang di Aplikasi Matematika Sederhana");
            System.out.println("=====");
            System.out.println("");
            System.out.println("1. Segitiga");
            System.out.println("2. Persegi");
            System.out.println("3. Lingkaran");
            System.out.println("4. Persegi Panjang");
            System.out.println("5. Exit");
            System.out.println(">");
            int pilihan = input.nextInt();

            if (pilihan == 1){
                System.out.println("1. Keliling Segitiga");
            }

```

```

        System.out.println("2. Luas Segitiga");
        System.out.println(">");
        int masukan = input.nextInt();
        if (masukkan == 1) {
            System.out.println("Masukkan sisi A");
            int sisiA = input.nextInt();
            System.out.println("Masukkan sisi B");
            int sisiB = input.nextInt();
            System.out.println("Masukkan sisi C");
            int sisiC = input.nextInt();
            Segitiga segitiga = new Segitiga(sisiA, sisiB, sisiC);
            segitiga.keliling();
        } else if (masukkan == 2) {
            System.out.println("Masukkan Alas");
            int alas = input.nextInt();
            System.out.println("Masukkan Tinggi");
            int tinggi = input.nextInt();
            Segitiga segitiga = new Segitiga(alas, tinggi);
            segitiga.luas();
        }

    } else if (pilihan == 2) {
        System.out.println("1. Keliling Persegi");
        System.out.println("2. Luas Persegi");
        System.out.println(">");
        int masukan = input.nextInt();
        if (masukkan == 1) {
            System.out.println("Masukkan Sisi Persegi");
            int sisi = input.nextInt();
            Persegi persegi = new Persegi(sisi);
            persegi.keliling();
        } else if (masukkan == 2) {
            System.out.println("Masukkan Sisi Persegi");

            int sisi = input.nextInt();
            Persegi persegi = new Persegi(sisi);
            persegi.keliling();
        } else if (masukkan == 2) {
            System.out.println("Masukkan Sisi Persegi");

            int sisi = input.nextInt();
            Persegi persegi = new Persegi(sisi);
            persegi.luas();
        }
    }

} else if (pilihan == 3) {
    System.out.println("1. Keliling Lingkaran");
    System.out.println("2. Luas Lingkaran");
    int masukan = input.nextInt();
    if (masukkan == 1) {
        System.out.println("Masukkan jari jari lingkaran: ");
        int jari = input.nextInt();
        Lingkaran lingkaran = new Lingkaran(jari);
        lingkaran.kelilingLingkaran();
    } else if (masukkan == 2) {
        System.out.println("Masukkan jari jari lingkaran: ");
        int jari = input.nextInt();
        Lingkaran lingkaran = new Lingkaran(jari);
        lingkaran.luasLingkaran();
    }
}

} else if (pilihan == 4) {
    System.out.println("1. Keliling Persegi Panjang");
    System.out.println("2. Luas Persegi Panjang");
    System.out.println(">");
    int masukan = input.nextInt();
    if (masukkan == 1) {
        System.out.println("Masukkan Alas Persegi Panjang: ");

        int alas = input.nextInt();
        System.out.println("Masukkan Tinggi Persegi Panjang");
        int tinggi = input.nextInt();
        PersegiPanjang persegiPjg = new PersegiPanjang(alas, tinggi);
        persegiPjg.kelilingPersegiPanjang();
    } else if (masukkan == 2) {
        System.out.println("Masukkan Alas Persegi Panjang: ");
        int alas = input.nextInt();
        System.out.println("Masukkan Tinggi Persegi Panjang");
        int tinggi = input.nextInt();
        PersegiPanjang persegiPjg = new PersegiPanjang(alas, tinggi);
        persegiPjg.luasPersegiPanjang();
    }
}

} else if (pilihan == 5) {
    System.out.println("Terimakasih sudah menggunakan aplikasi ini");
    keeplooping = false;
}

} while (keeplooping);

}
}

```

Jika main codenya dirun maka akan menghasilkan output seperti berikut:

1. Menghitung keliling segitiga

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
1
1. Keliling Segitiga
2. Luas Segitiga
>
1
Masukkan sisi A
10
Masukkan sisi B
19
Masukkan sisi C
20
Keliling segitiga adalah: 49cm
```

2. Menghitung luas segitiga

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
1
1. Keliling Segitiga
2. Luas Segitiga
>
2
Masukkan Alas
10
Masukkan Tinggi
19
Luas segitiga adalah: 95.0cm^2
```

3. Menghitung keliling persegi

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
2
1. Keliling Persegi
2. Luas Persegi
>
1
Masukkan Sisi Persegi
10
Keliling persegi adalah: 40cm
```

4. Menghitung luas persegi

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
2
1. Keliling Persegi
2. Luas Persegi
>
2
Masukkan Sisi Persegi
10
Luas persegi adalah: 100cm^2
```

5. Menghitung keliling lingkaran

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
3
1. Keliling Lingkaran
2. Luas Lingkaran
1
Masukkan jari jari lingkaran:
10
Keliling lingkaran adalah: 62.8cm
```

6. Menghitung Luas Lingkaran

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
3
1. Keliling Lingkaran
2. Luas Lingkaran
2
Masukkan jari jari lingkaran:
15
Luas lingkaran adalah: 706.5cm^2
```

7. Menghitung Keliling Persegi Panjang

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
4
1. Keliling Persegi Panjang
2. Luas Persegi Panjang
>
1
Masukkan Alas Persegi Panjang:
10
Masukkan Tinggi Persegi Panjang
10
Keliling Persegi Panjang Adalah: 40cm
```

8. Menghitung Luas Persegi Panjang

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
4
1. Keliling Persegi Panjang
2. Luas Persegi Panjang
>
2
Masukkan Alas Persegi Panjang:
10
Masukkan Tinggi Persegi Panjang
2
Luas Persegi Panjang Adalah: 20cm^2
```

9. Exit

```
=====
Selamat Datang di Aplikasi Matematika Sederhana
=====

1. Segitiga
2. Persegi
3. Lingkaran
4. Persegi Panjang
5. Exit
>
5
Terimakasih sudah menggunakan aplikasi ini
```

Kesimpulan

- Abstraksi berarti menggunakan hal-hal sederhana untuk mewakili hal-hal rumit.
- Enkapsulasi berarti menyembunyikan data-data kompleks di balik modifier Private dan menyediakan sedikit method Public sebagai “jalan” untuk mengakses data-data Private tersebut.
- Inheritance (arti: Pewarisan) merupakan konsep menurunkan atribut dan method milik sebuah class ke class lain.
- Polimorfisme merupakan konsep pembuatan method berbeda dengan nama yang sama.