



unindra
universitas indraprasta PGRI

Kelas dan Objek

Class and Object

PERTEMUAN 2

Class A

State

Data member / variabel

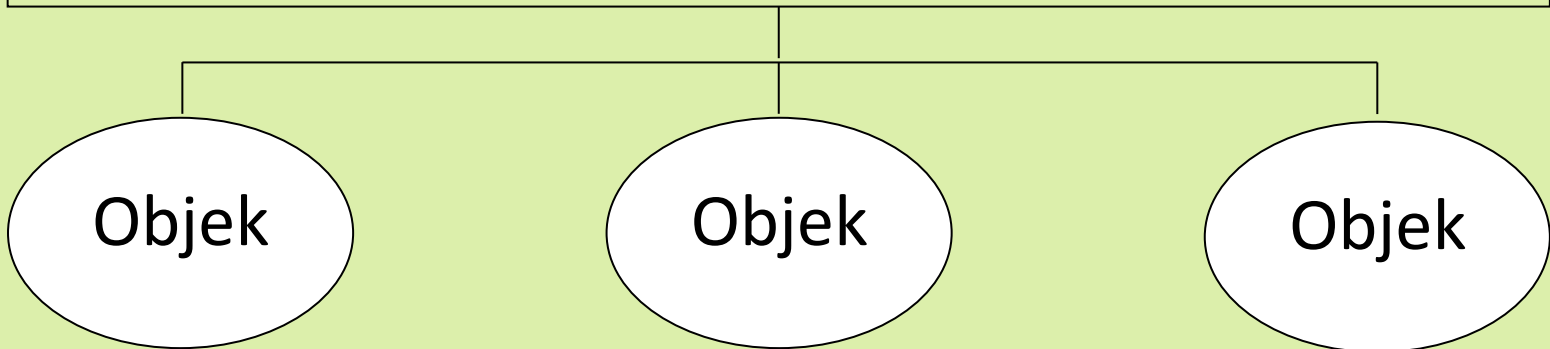
Behaviour

method1()
method2()

Objek

Objek

Objek



Jenis-Jenis Variable

Dalam bahasa pemrograman Java, terdapat beberapa jenis Variable diantaranya :

- 1. Variable Lokal**
- 2. Variable Kelas (Variable Static)**
- 3. Variable Instance (Non-Static)**

Variable Lokal

Variable Lokal memiliki ciri-ciri seperti berikut :

- Variabel lokal dideklarasikan didalam method, konstruktor atau blok.
- Variable lokal tidak dapat menggunakan Modifier seperti **public**, **protected**, **private**, dll.
- Variabel lokal hanya dapat digunakan didalam method, konstruktor, atau blok tempat pendeklarasiannya.
- Tidak ada nilai default untuk variabel lokal sehingga variabel lokal harus dideklarasikan dan diinisialisasikan sebelum digunakan.

Variable Kelas (Variable Static)

Variable Kelas (Variable Static) memiliki ciri-ciri seperti berikut:

- Variabel kelas/static dideklarasikan dengan katakunci static didalam kelas, tetapi diluar method, konstruktor, atau blok.
- Hanya ada satu salinan dari variabel kelas per kelas.
- Variabel static biasanya dideklarasikan sebagai variabel konstanta (nilainya tetap dan tidak berubah).
- Variabel static memiliki nilai default yang sama dengan variabel instance.
- Variabel static dapat diakses dengan memanggil nama kelas>NamaKelas>NamaVariabel.

Variable Instance (Non-Static)

Variable Instance (Variable Non-Static) memiliki ciri-ciri seperti berikut :

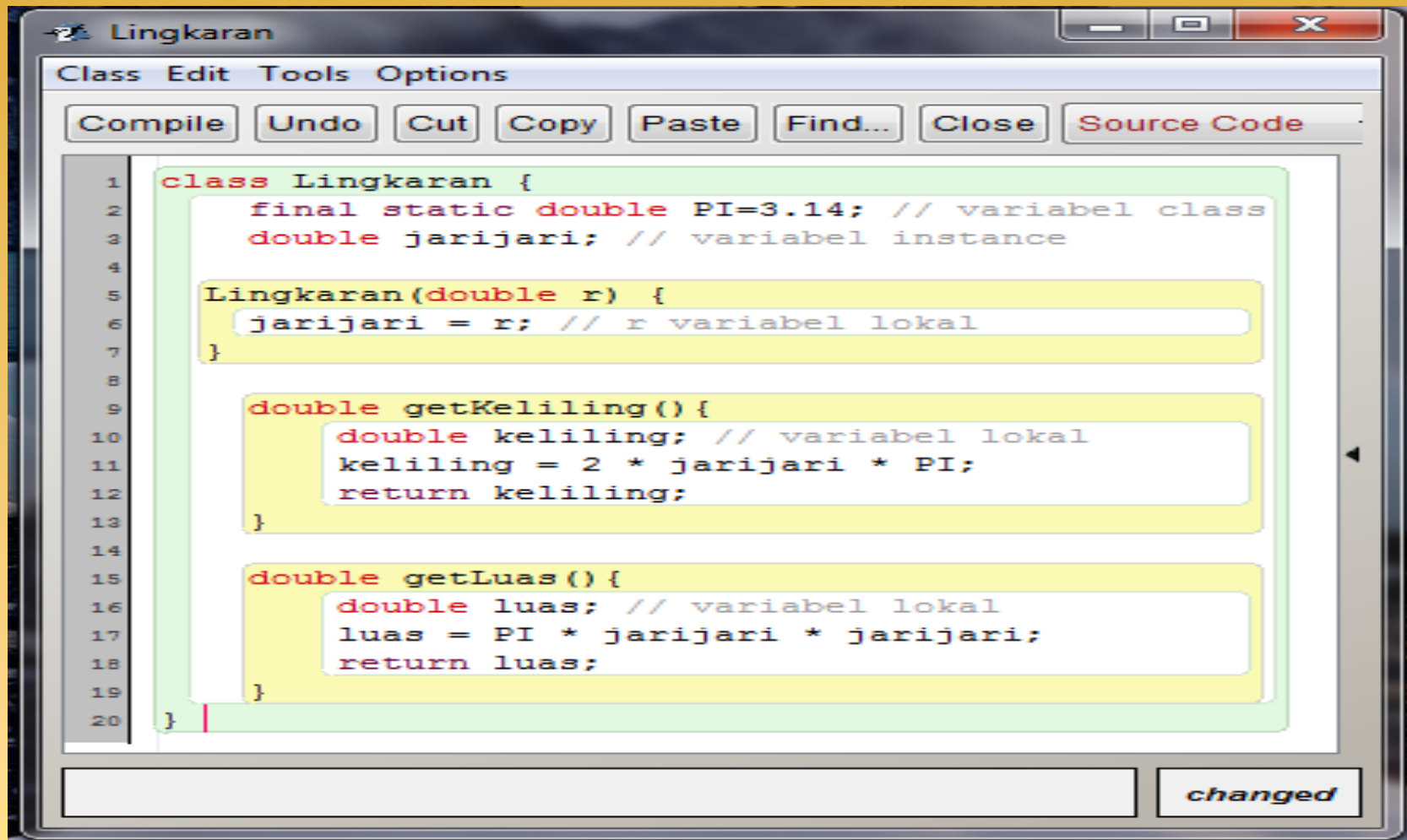
Variabel instance dideklarasikan didalam kelas, tetapi di luar method, constructor, atau blok.

Variabel instance dibuat saat sebuah objek dibuat dengan kata kunci 'new' dan dihapus saat objek tersebut dihapus.

Variabel instance dapat dideklarasikan dalam kelas sebelum atau sesudah penggunaan.

Modifier akses dapat digunakan oleh semua method, konstruktor, dan blok di dalam kelas.

Contoh Variable Lokal, Kelas dan Instance



```
1  class Lingkaran {
2      final static double PI=3.14; // variabel class
3      double jarijari; // variabel instance
4
5      Lingkaran(double r) {
6          jarijari = r; // r variabel lokal
7      }
8
9      double getKeliling(){
10         double keliling; // variabel lokal
11         keliling = 2 * jarijari * PI;
12         return keliling;
13     }
14
15     double getLuas(){
16         double luas; // variabel lokal
17         luas = PI * jarijari * jarijari;
18         return luas;
19     }
20 }
```

The screenshot shows a Java IDE window titled "Lingkaran". The window has a menu bar with "Class", "Edit", "Tools", and "Options". Below the menu bar is a toolbar with buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", "Close", and "Source Code". The main area displays the source code for the "Lingkaran" class. The code defines a class with a final static variable "PI", an instance variable "jarijari", and three methods: a constructor "Lingkaran(double r)", "getKeliling()", and "getLuas()". Each method body is highlighted in yellow. The status bar at the bottom right indicates "changed".

Pengertian Kelas/Class

- Kelas merupakan blok pembangun terpenting di pendekatan berorientasi objek yang bersifat abstrak.
- Kelas mendeskripsikan kelompok objek dengan properti-properti dan atribut-atribut serupa di dalam java.
- Kelas merupakan kumpulan dari definisi data dan fungsi – fungsi dalam suatu unit untuk suatu tujuan yang di dalamnya terdapat kumpulan atribut dan method.

Class dapat di umpamakan seperti spesifikasi atau blueprint.

Contoh :

1. Kendaraan
2. Binatang
3. Pohon
4. Universitas

Kegunaan Kelas/Class

Berikut ini adalah kegunaan kelas :

- Mengelompokkan object-object menjadi kelas-kelas berarti mengabstraksikan persoalan yang dihadapi.
- Definisi-definisi umum (seperti nama kelas dan atribut) cukup disimpan sekali per kelas bukan sekali per instan kelas (object). Operasi-operasi dapat ditulis sekali per kelas. Semua object dalam kelas mendapat manfaat penggunaan kembali (reuse) kode itu.

Struktur Kelas/Class di Dalam Java

Secara umum, komponen kelas/class dalam JAVA adalah :

```
class NamaClass
{
    variabel/field
    (statement)...
    konstruktor
    (statement)...
    method/behavior/aksi/fungsi
    (statement)...
}
```

Isi dalam Kelas/Class

variabel/field	Konstruktor	Method/behavior/aksi/fungsi
<p>Berisikan tipe data dalam java yang disertai nama variabel itu sendiri .</p>	<p>Cara khusus untuk membangun atau memanggil objek dari definisi class dan biasanya sama dengan nama class .</p>	<p>Komponen dalam class yang memberikan nilai atau keluaran jika menerima parameter sebagai input datanya.</p>
<p>Contoh: contoh : int nilai; , char huruf; , String kalimat dll ;</p>	<p>Contoh:</p> <pre>class <u>hitung</u> { public void <u>hitung</u> () { } }</pre> <p>(<u>Method sama dengan nama class</u>)</p>	<p>Contoh: pada object mangga : terdapat method ambilRasa , kupasKulit dan lain-lain.</p>

Pengertian Objek

Object adalah instance dari class. Proses pembuatan objek sering disebut sebagai **instansiasi class**.

Jika class secara umum merepresentasikan (template) sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri. Objek merupakan turunan atau hasil dari suatu class.

Pengertian Objek

Pada dasarnya ada dua karakteristik yang utama pada sebuah objek , yaitu : state dan behaviour

State merupakan atribut /variable

Contoh : Atribut dari class Kendaraan (jenis, merk, warna)

Behaviour merupakan tingkah laku/method

Contoh : Behaviour dari class Kendaraan (jalan, mundur, berhenti)

Pada dasarnya semua benda yang ada di dunia nyata dapat dianggap sebagai sebuah objek.

Contoh :

Dari **class Kendaraan** kita dapat membuat **object** roda_dua, roda_tiga, roda_empat dan lain-lain.

Dari **class Binatang** kita dapat membuat **object** mamalia, reptil, amfibi dan lain-lain.

Contoh Pembuatan Class dan Object pada Java (tanpa Method)

The image shows two windows from an IDE. The left window, titled 'Kendaraan', displays the source code for a class named 'Kendaraan'. The code defines three String attributes: 'jenis', 'merk', and 'warna'. The right window, titled 'Kendaraan_Aksi', displays the source code for a class named 'Kendaraan_Aksi'. It contains a 'main' method that creates an instance of the 'Kendaraan' class named 'rodadua' and initializes its attributes with the values 'Gunung', 'Polygon', and 'Merah'. Below the code windows, a terminal window titled 'BlueJ: Terminal Window - PB...' shows the output of the program: 'Jenis : Gunung', 'Merk : Polygon', and 'Warna : Merah'.

```
public class Kendaraan {
    String jenis ;
    String merk ;
    String warna ;
}

public class Kendaraan_Aksi {
    public static void main (String [] args) {
        Kendaraan rodadua = new Kendaraan () ;
        rodadua.jenis = "Gunung" ;
        rodadua.merk = "Polygon" ;
        rodadua.warna = "Merah" ;
        System.out.println ("Jenis : " +rodadua.jenis);
        System.out.println ("Merk : " +rodadua.merk);
        System.out.println ("Warna : " +rodadua.warna);
    }
}
```

Jenis : Gunung
Merk : Polygon
Warna : Merah

Nama Class : Kendaraan dan Kendaraan_Aksi

Nama Atribut : jenis, merk, warna dengan bertipe data String

Nama Objek : rodadua (instance class) (tidak boleh ada objek yang sama)

Inisialisasi dengan menggunakan kata *new* di ikuti dengan konstruktor (sama dengan nama Class yg di panggil (Kendaraan))

Contoh Pembuatan Class dan Object pada Java (dengan Method)

```
1 public class Kendaraan_1 {
2     String jenis ;
3     String merk ;
4     String warna ;
5     String keterangan ;
6     String kendaraan ;
7
8     public void rusak () {
9         System.out.println ("Kendaraan : " +kendaraan);
10        System.out.println ("Jenis : " +jenis);
11        System.out.println ("Merk : " +merk);
12        System.out.println ("Warna : " +warna);
13        System.out.println ("Keterangan : " +keterangan);
14        System.out.println ( );
15    }
16
17    public void jalan () {
18        System.out.println ("Kendaraan : " +kendaraan);
19        System.out.println ("Jenis : " +jenis);
20        System.out.println ("Merk : " +merk);
21        System.out.println ("Warna : " +warna);
22        System.out.println ("Keterangan : " +keterangan);
23        System.out.println ( );
24    }
25 }
```

```
1 public class Kendaraan_Aksi_1 {
2     public static void main (String [] args) {
3         Kendaraan_1 rodadua = new Kendaraan_1 ();
4         rodadua.kendaraan = "Kendaraan Roda Dua" ;
5         rodadua.jenis = "Gunung" ;
6         rodadua.merk = "Polygon" ;
7         rodadua.warna = "Merah" ;
8         rodadua.keterangan = "2 Rusak" ;
9         rodadua.rusak ();
10
11        Kendaraan_1 rodaempat = new Kendaraan_1 ();
12        rodaempat.kendaraan = "Kendaraan Roda Empat" ;
13        rodaempat.jenis = "Sedan" ;
14        rodaempat.merk = "Toyota" ;
15        rodaempat.warna = "Hitam" ;
16        rodaempat.keterangan = "4 Jalan";
17        rodaempat.jalan ();
18    }
19 }
20
21 }
```

Options

```
Kendaraan : Kendaraan Roda Dua
Jenis : Gunung
Merk : Polygon
Warna : Merah
Keterangan : 2 Rusak

Kendaraan : Kendaraan Roda Empat
Jenis : Sedan
Merk : Toyota
Warna : Hitam
Keterangan : 4 Jalan
```

Nama Class : Kendaraan_1 dan Kendaraan_Aksi_1

Nama Atribut : jenis, merk, warna, keterangan dan kendaraan dengan bertype data String

Nama Objek : rodadua dan rodaempat (instance class) (tidak boleh ada objek yang sama)

Inisialisasi dengan menggunakan kata *new* di ikuti dengan konstruktor (sama dengan nama Class yg di panggil (Kendaraan_1))

Method : rusak dan jalan (tidak boleh ada method yang sama)

Latihan

Class dan Object

1. Berikan simpulan menurut penjelasan dari masing – masing mahasiswa/i apa itu class dan object !
2. Berikan contoh Program Sederhana class dan object, dengan ketentuan :
 - a. NPM Ganjil dengan Tema Binatang.
 - b. NPM Genap dengan Tema Tumbuhan.
 - c. Tiap class terdiri minimal 3 atribut, 2 method dan 2 object