Praktikum 4

Inheritance dan Polymorphism

[**Macam-macam Inheritance**](#_fhr8ozvk6b6m) **2**

[Single](#_qwzotki5kra) 2

[Multilevel](#_8gnjzo8mg9vt) 2

[Hierarchical](#_5r2hyx7mqev0) 3

[Multiple](#_ppeigrlkjhqy) 4

[Hybrid](#_cswi76ic1no6) 4

[**Implementasi Macam-macam Inheritance**](#_lj5ycfsbfyb5) **6**

[Single](#_rrp1749h7di4) 6

[Multilevel](#_lgxk5m6rq9z7) 7

[Hierarchical](#_e9vr80j5ywt2) 8

[Multiple dan Hybrid](#_wakj14wy7hs7) 8

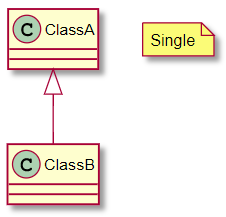
[**Implementasi Polymorphism**](#_nzocprx3q5sx) **9**

[Method Overloading/Static Polymorphism](#_77ly7vjz98tv) 9

[Method Overriding/Dynamic Polymorphism](#_r2x7be8fvg0n) 10

# Macam-macam Inheritance

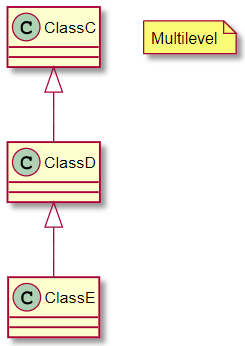
## Single



ClassA adalah Super Class/Parent dari ClassB

ClassB adalah Sub Class/Child dari ClassA

## Multilevel



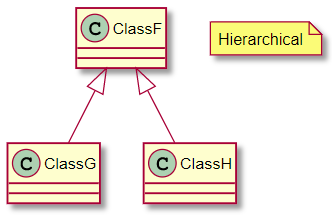
ClassC adalah Super Class/Parent dari ClassD

ClassD adalah Super Class/Parent dari ClassE

ClassE adalah Sub Class/Child dari ClassD

ClassD adalah Sub Class/Child dari ClassC

## Hierarchical

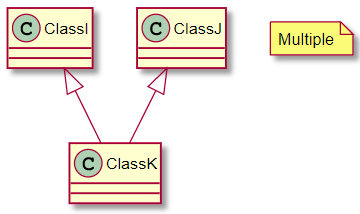


ClassF adalah Super Class/Parent dari ClassG dan ClassH

ClassG adalah Sub Class/Child dari ClassF

ClassH adalah Sub Class/Child dari ClassF

## Multiple

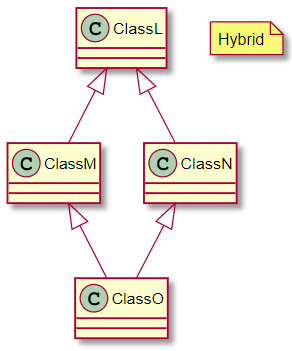


ClassI adalah Super Class/Parent dari ClassK

ClassJ adalah Super Class/Parent dari ClassK

ClassK adalah Sub Class/Child dari ClassI dan ClassK

## Hybrid



ClassL adalah Super Class/Parent ClassM dan ClassN

ClassM dan ClassN adalah Super Class/Parent ClassO

ClassM dan ClassN adalah Sub Class/Child ClassL

ClassO adalah Sub Class/Child Class M dan ClassN

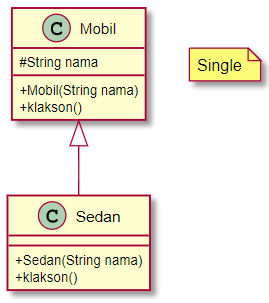
# 

Tugas Praktikum 04

# Implementasi Macam-macam Inheritance

Berikut adalah implementasi dari berbagai macam inheritance pada Java.

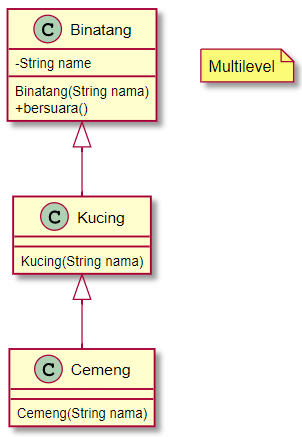
## Single



Buat Source Code dari Class Diagram di atas dengan Aplikasi:

1. System.out.println method klakson() di Mobil adalah “Teet”
2. System.out.println method klakson() di Sedan adalah “Tiin”
3. Buat objek m1 dengan nama “Hino” dari Class Mobil
4. Buat objek s1 dengan nama “Vios” dari Class Sedan
5. Mobil m1 melakukan klakson()
6. Sedan s1 melakukan klakson()
7. Print screen source code Class Mobil, Class Sedan, dan Implementasi dari nomor 1 hingga 6 di atas.

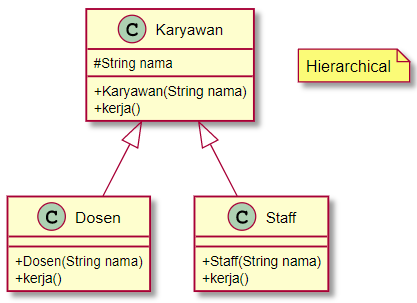
## Multilevel



Buat Source Code dari Class Diagram di atas dengan Aplikasi:

1. System.out.println method bersuara() di class Binatang adalah “Mbeek”
2. System.out.println method bersuara() di class Kucing adalah “Ngeoong”
3. System.out.println method bersuara() di class Cemeng adalah “Meoong”
4. Buat objek kambing dengan nama “Kambing” dari class Binatang
5. Buat objek mudreg dengan nama “Mudreg” dari class Kucing
6. Buat objek mocha dengan nama “Mocha” dari class Cemeng
7. Kambing bersuara()
8. Mudreg bersuara()
9. Mocha bersuara()
10. Print screen source code Class Binatang, Class Kucing, Class Cemeng, dan Implementasi dari nomor 1 hingga 9 di atas.

## Hierarchical



Buat Source Code dari Class Diagram di atas dengan Aplikasi:

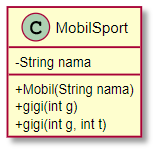
1. System.out.println method kerja() di Class Karyawan adalah “kerja”
2. System.out.println method kerja() di Class Dosen adalah “mengajar”
3. System.out.println method kerja() di Class Staff adalah “bantu dosen”
4. Buat objek wahyu dengan nama “Wahyu” dari class Karyawan
5. Buat objek amir dengan nama “Amir” dari class Dosen
6. Buat objek melan dengan nama “Melan” dari class Staff
7. Wahyu kerja()
8. Amir kerja()
9. Melan kerja()
10. Print screen source code Class Karyawan, Class Dosen, Class Staff, dan Implementasi dari nomor 1 hingga 9 di atas.

## Multiple dan Hybrid

Di Java, Inheritance tipe multiple dan hybrid tidak didukung. Misalkan ada nama method yang sama di Super Class dan method tersebut dipanggil di Sub Class, maka akan terjadi ambigu, method dari class mana yang akan digunakan. Pada Java, hal ini akan mengakibatkan Compile Time Error.

# Implementasi Polymorphism

## Method Overloading/Static Polymorphism



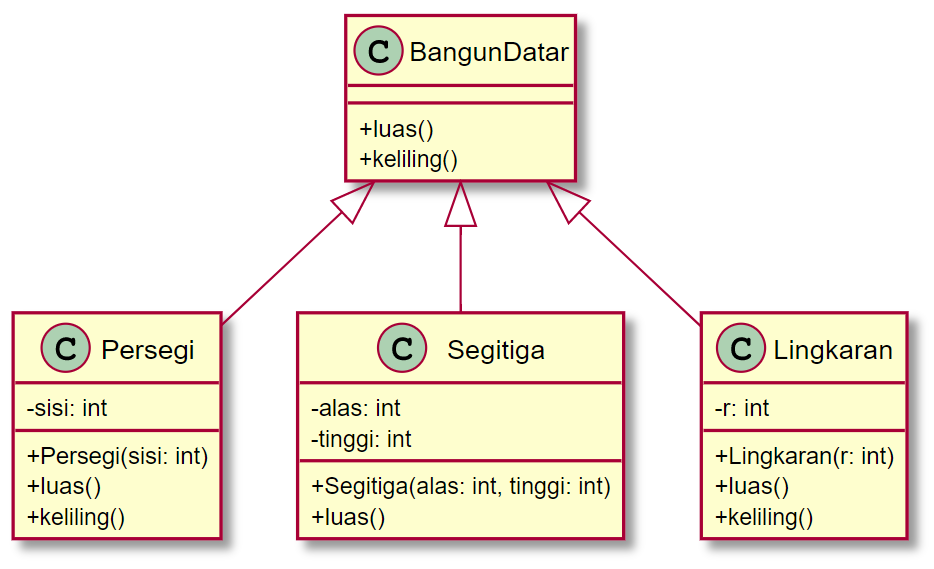
Buat Source Code dari Class Diagram di atas dengan Aplikasi:

1. System.out.println method gigi(int g) di Class MobilSport adalah “nama mobil masuk gigi ke-g”
2. System.out.println method gigi(int g, int t) di Class MobilSport adalah “nama mobil masuk gigi ke-g dengan turbo t”
3. Buat objek jaguar dengan nama “Jaguar” dari class MobilSport
4. Buat app dengan output sebagai berikut:

| Jaguar masuk gigi ke-1 Jaguar masuk gigi ke-2 Jaguar masuk gigi ke-3 Jaguar masuk gigi ke-4 Jaguar masuk gigi ke-5 Jaguar masuk gigi ke-5 dengan turbo 1 Jaguar masuk gigi ke-5 dengan turbo 2 Jaguar masuk gigi ke-5 dengan turbo 3 Jaguar masuk gigi ke-5 dengan turbo 4 Jaguar masuk gigi ke-5 dengan turbo 5 |
| --- |

1. Print screen source code Class MobilSport dan Implementasi dari nomor 1 hingga 4 di atas.

## Method Overriding/Dynamic Polymorphism



Source code <https://github.com/adiwp/pbo20192020II/>. Buat Source Code dari Class Diagram di atas dengan Aplikasi:

1. Buat objek p dengan sisi=3 dari class Persegi
2. Buat objek s dengan alas=4, tinggi=4 dari class Segitiga
3. Buat objek el dengan r=7 dari class Lingkaran
4. Hitung luas() dan keliling() objek p
5. Hitung luas() objek s
6. Hitung luas() dan keliling() objek el
7. Print screen output dari nomor 1 hingga 7 di atas.