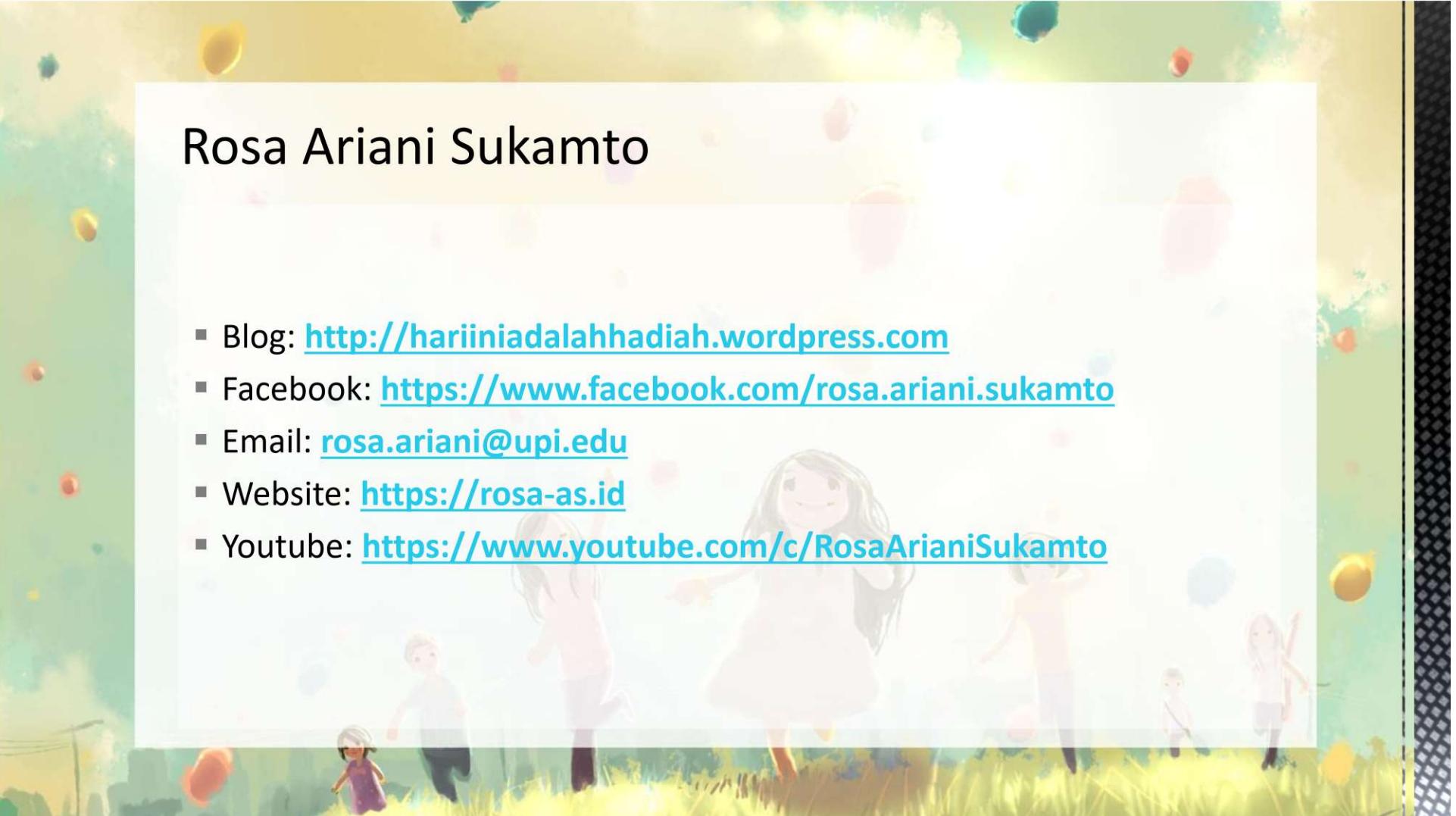




Desain dan Pemrograman Berorientasi Objek

Pewarisan (*Inheritance*)

Rosa A. S.



Rosa Ariani Sukamto

- Blog: <http://hariiniadalhhadiah.wordpress.com>
- Facebook: <https://www.facebook.com/rosa.ariani.sukamto>
- Email: rosa.ariani@upi.edu
- Website: <https://rosa-as.id>
- Youtube: <https://www.youtube.com/c/RosaArianiSukamto>

Pewarisan atau *Inheritance* (1)

- suatu hubungan dua buah kelas atau lebih dimana ada kelas yang memiliki atribut dan metode yang sama dengan kelas lainnya beserta atribut dan metode tambahan yang merupakan sifat khusus kelas yang menjadi turunannya
- Kelas yang merupakan kelas turunan biasa disebut kelas anak (*subclass*) dan kelas yang menjadi dasar penurunan adalah kelas orang tua (*superclass*)
- Kelas Anak berjenis sama dengan kelas Orang Tua/Bapak hanya saja kelas Anak memiliki hal yang lebih detail dari kelas Orang Tua/ Bapak

Pewarisan atau *Inheritance* (2)

Kelas Manusia

nama
alamat
no_ktp
no_telepon

Kelas Karyawan

nama
alamat
no_ktp
no_telepon
nomor_pegawai
jabatan
departemen

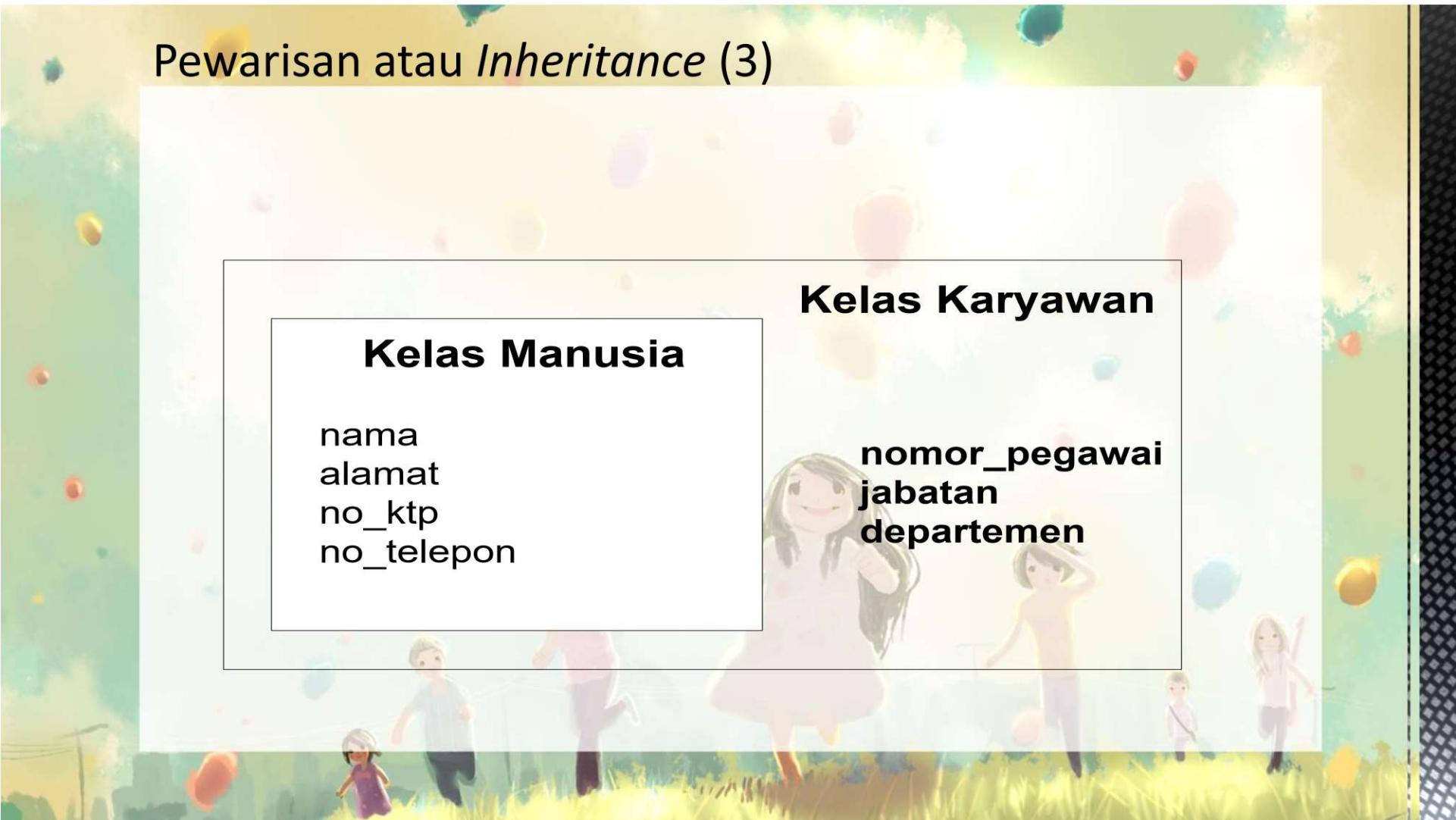
Pewarisan atau *Inheritance* (3)

Kelas Manusia

nama
alamat
no_ktp
no_telepon

Kelas Karyawan

nomor_pegawai
jabatan
departemen



Pewarisan atau *Inheritance* (4)



Implementasi pewarisan (*inheritance*) - C++ (1)

```
class Manusia{

private:
    string nama;
    string alamat;
    string no_ktp;
    string no_telepon;

public:

    Manusia() {
    }

    void setNama(string nama){
        this->nama = nama;
    }

};

string getNama() {
    return this->nama;
}

.....
//buat set dan get atribut lainnya

~Manusia() {
}

};
```

Implementasi pewarisan (*inheritance*) - C++ (2)

```
class Karyawan : public Manusia{  
  
private:  
  
    string nomor_pegawai;  
    string jabatan;  
    string departemen;  
  
public:  
  
    Karyawan() {  
    }  
  
    void setNomorPegawai(string nomor) {  
        this->nomor_pegawai = nomor;  
    }  
  
    string getNomorPegawai() {  
        return this->nomor_pegawai;  
    }  
  
    .....  
    /* set dan get dari atribut yang  
       hanya milik karyawan */  
  
    .....  
  
    ~Karyawan() {  
    }  
};
```

Implementasi pewarisan (*inheritance*) - C++ (3)

```
#include <string>
#include <iostream>
using namespace std;
#include "Manusia.cpp"
#include "Karyawan.cpp"
int main(){

    Manusia kmanusia;
    Karyawan kkaryawan;

    kmanusia.setNama("Gina");
    kmanusia.setAlamat("Bandung");
    kmanusia.setNoKTP("320.120.777");
    kmanusia.setNoTelepon("08188899977" );

    cout << kmanusia.getNama() << endl;
    cout << kmanusia.getAlamat() << endl;
    cout << kmanusia.getNoKTP() << endl;
    cout << kmanusia.getNoTelepon() << endl;
```

```
    kkaryawan.setNama("Gina");
    kkaryawan.setAlamat("Bandung");
    kkaryawan.setNoKTP("320.120.777");
    kkaryawan.setNoTelepon("08188899977");
    kkaryawan.setNomorPegawai("13202020");
    kkaryawan.setJabatan("-");
    kkaryawan.setDepartemen("-");
    cout << kkaryawan.getNama() << endl;
    cout << kkaryawan.getAlamat() << endl;
    cout << kkaryawan.getNoKTP() << endl;
    cout << kkaryawan.getNoTelepon() << endl;
    cout << kkaryawan.getNomorPegawai() << endl;
    cout << kkaryawan.getJabatan() << endl;
    cout << kkaryawan.getDepartemen() << endl;
    return 0;
}
```

Implementasi pewarisan (*inheritance*) - PHP (1)

```
<?php

class Manusia{

    private $nama;
    private $alamat;
    private $no_ktp;
    private $no_telepon;

    function __construct(){
    }

    function setNama($nama){
        $this->nama = $nama;
    }

    function getNama(){
        return $this->nama;
    }

    //set dan get
    .....
}

function __destruct(){
}

?>
```

Implementasi Pewarisan (*Inheritance*) - PHP (2)

```
<?php

class Karyawan extends Manusia{

    private $nomor_pegawai;
    private $jabatan;
    private $departemen;

    function __construct() {
    }

    function setNomorPegawai($nomor) {
        $this->nomor_pegawai = $nomor;
    }

    function getNomorPegawai() {
        return $this->nomor_pegawai;
    }

    //set dan get
    .....
}

function __destruct() {
}

?>
```

Implementasi pewarisan (*Inheritance*) - PHP (3)

```
<?php
include "Manusia.php";
include "Karyawan.php";

$kmanusia = new Manusia();
$kmanusia->setNama("Gina");
$kmanusia->setAlamat("Bandung");
$kmanusia->setNoKTP("320.120.777");
$kmanusia->setNoTelepon("08188899977");

echo $kmanusia->getNama()."<br/>";
echo $kmanusia->getAlamat()."<br/>";
echo $kmanusia->getNoKTP()."<br/>";
echo $kmanusia->getNoTelepon()."<br/>";

$kkaryawan = new Karyawan();
```

```
$kkaryawan->setNama("Gina");
$kkaryawan->setAlamat("Bandung");
$kkaryawan->setNoKTP("320.120.777");
$kkaryawan->setNoTelepon("08188899977");
$kkaryawan->setNomorPegawai("13202020");
$kkaryawan->setJabatan("-");
$kkaryawan->setDepartemen("-");

echo $kkaryawan->getNama()."<br/>";
echo $kkaryawan->getAlamat()."<br/>";
echo $kkaryawan->getNoKTP()."<br/>";
echo $kkaryawan->getNoTelepon()."<br/>";
echo $kkaryawan->getNomorPegawai()."<br/>";
echo $kkaryawan->getJabatan()."<br/>";
echo $kkaryawan->getDepartemen()."<br/>";

?>
```

Implementasi pewarisan (*Inheritance*) - Java (1)

```
class Manusia{

    private String nama;
    private String alamat;
    private String no_ktp;
    private String no_telepon;

    Manusia(){
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return this.nama;
    }

    //set dan get
    .....
}
```

Implementasi pewarisan (*Inheritance*) - Java (2)

```
class Karyawan extends Manusia{

    private String nomor_pegawai;
    private String jabatan;
    private String departemen;

    Karyawan() {
    }

    public void setNomorPegawai(String nomor) {
        this.nomor_pegawai = nomor;
    }

    public String getNomorPegawai(){
        return this.nomor_pegawai;
    }

    //set dan get
    .....
}
```

Implementasi pewarisan (*Inheritance*) - Java (3)

```
class Main{
public static void main(String[] args){
    Manusia kmanusia;
    Karyawan kkaryawan;

    kmanusia = new Manusia();
    kmanusia.setNama("Gina");
    kmanusia.setAlamat("Bandung");
    kmanusia.setNoKTP("320.120.777");
    kmanusia.setNoTelepon("08188899977");

    System.out.println(kmanusia.getNama());
    System.out.println(kmanusia.getAlamat());
    System.out.println(kmanusia.getNoKTP());
    System.out.println(
        kmanusia.getNoTelepon());

    kkaryawan = new Karyawan();
```

```
kkaryawan.setNama("Gina");
kkaryawan.setAlamat("Bandung");
kkaryawan.setNoKTP("320.120.777");
kkaryawan.setNoTelepon("08188899977");
kkaryawan.setNomorPegawai("13202020");
kkaryawan.setJabatan("-");
kkaryawan.setDepartemen("-");
System.out.println(kkaryawan.getNama());
System.out.println(kkaryawan.getAlamat());
System.out.println(kkaryawan.getNoKTP());
System.out.println(kkaryawan.getNoTelepon());
System.out.println(kkaryawan.getNomorPegawai());
System.out.println(kkaryawan.getJabatan());
System.out.println(kkaryawan.getDepartemen());
}
```

MULTIPLE INHERITANCE - C++ (1)



MULTIPLE INHERITANCE - C++ (2)

```
class Aset{
    private:

        int kodeAset;
        int jenisAset;
        double nilaiAset;

    public:
        Aset() {
        }

    void setKodeAset(int kodeAset) {
        this->kodeAset = kodeAset;
    }

    int getKodeAset () {
        return this->kodeAset;
    }
}
```

//set dan get

.....

~Aset() {

}

} ;

MULTIPLE INHERITANCE - C++ (3)

```
class Bangunan{  
    private:  
  
        int kodeBangunan;  
        string PemilikBangunan;  
  
    public:  
        Bangunan() {  
        }  
  
        void setKodeBangunan(int  
kodeBangunan) {  
            this->kodeBangunan =  
kodeBangunan;  
        }  
  
        int getKodeBangunan() {  
            return this->kodeBangunan;  
        }  
}
```

```
//set dan get
```

```
.....
```

```
~Bangunan () {  
}  
};
```

MULTIPLE INHERITANCE - C++ (4)

```
class Gudang : public Bangunan, public Aset{  
  
private:  
  
    int kodeGudang;  
    string namaGudang;  
    string alamatGudang;  
  
public:  
    Gudang() {  
    }  
  
    void setKodeGudang(  
        int kodeGudang){  
        this->kodeGudang = kodeGudang;  
    }  
  
    int getKodeGudang(){  
        return this->kodeGudang;  
    }  
}
```

//set dan get

.....

~Gudang () {

}

};

MULTIPLE INHERITANCE - C++ (5)

```
#include <string>
#include <iostream>
using namespace std;
#include "Aset.cpp"
#include "Bangunan.cpp"
#include "Gudang.cpp"

int main(){
    Aset oaset;
    Bangunan obangunan;
    Gudang ogudang;

    oaset.setKodeAset(1);
    oaset.setJenisAset(1);
    oaset.setNilaiAset(300000000);

    cout << oaset.getKodeAset() << endl;
    cout << oaset.getJenisAset() << endl;
    cout << oaset.getNilaiAset() << endl;
```

```
obangunan.setKodeBangunan(1);
obangunan.setPemilikBangunan("BDG Aamiin");

cout << obangunan.getKodeBangunan() << endl;
cout << obangunan.getPemilikBangunan() << endl;

ogudang.setKodeAset(3);
ogudang.setJenisAset(3);
ogudang.setNilaiAset(300000000);
ogudang.setKodeBangunan(3);
ogudang.setPemilikBangunan("BDG Aamiin");
ogudang.setKodeGudang(1);
ogudang.setNamaGudang("BDG Aamiin I");
ogudang.setAlamatGudang("Bandung");
```

MULTIPLE INHERITANCE - C++ (6)

```
cout << ogudang.getKodeAset() << endl;
cout << ogudang.getJenisAset() << endl;
cout << ogudang.getNilaiAset() << endl;
cout << ogudang.getKodeBangunan() << endl;
cout << ogudang.getPemilikBangunan() << endl;
cout << ogudang.getKodeGudang() << endl;
cout << ogudang.getNamaGudang() << endl;
cout << ogudang.getAlamatGudang() << endl;

return 0;
}
```

Konsep

- Inner Class – Java The One and Only
 - Jika dalam objek sesungguhnya ada objek di dalam objek lain, jenis berbeda
- Inheritance
 - Jika objek sejenis dan semua atribut kelas Orang Tua/Bapak dapat dipakai kelas Anak, kelas anak memiliki atribut lebih banyak dari kelas Orang Tua/Bapak
- Saling lepas
 - Jika berbeda jenis dan hubungannya adalah saling memakai atau dipakai

Contoh soal *Inheritance*

Pengusaha

- nomorUsaha
- nama
- alamat
- namaUsaha

PengerajinKayu

- noKTP
- nama
- alamat
- namaUsaha
- omsetPerBulan
- deskripsi

Kayu

- kodeKayu
- jenisPohon
- berat
- Volume

HasilKerajinan

- kodeKerajinan
- bahan
- namaKerajinan

KomoditasEksport

- kodeKomoditas
- negaraTujuan
- jenisKomoditas

KerajinanKayu

- kodeKerajinan
- namaKerajinan
- fungsiKerajinan
- harga

semuanya memiliki metode get set. Buatlah main dengan beberapa kelas di atas yang mampu menerima masukan berupa data pengrajin kayu dan kerajinan kayu untuk kemudian ditampilkan masukan itu dalam bentuk tabel. Gunakan bahasa C++ dan Java.

Cara menganalisis hubungan antar kelas (1)

- Pengusaha dengan Pengrajin Kayu
- secara objek adalah objek yang sama yaitu manusia



Cara menganalisis hubungan antar kelas (2)

- Pengusaha dengan Kayu
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (3)

- Pengusaha dengan HasilKerajinan
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (4)

- Pengusaha dengan Komoditas Eksport
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (5)

- Pengusaha dengan Kerajinan Kayu
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (6)

- Pengrajin Kayu dengan Kayu
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (7)

- PengrajinKayu dengan HasilKerajinan
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (8)

- Pengrajin Kayu dengan Komoditas Eksport
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (9)

- PengrajinKayu dengan KerajinanKayu
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (10)

- Kayu dengan HasilKerajinan
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (11)

- Kayu dengan KomoditasEksport
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (12)

- Kayu dengan Kerajinan Kayu
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (13)

- HasilKerajinan dengan KomoditasEksport
- secara objek bukan merupakan jenis objek yang sama



Cara menganalisis hubungan antar kelas (14)

- HasilKerajinan dengan KerajinanKayu
- secara objek adalah objek yang sama yaitu kerajinan?



Cara menganalisis hubungan antar kelas (15)

- KerajinanKayu dengan KomoditasEksport
- secara objek bukan merupakan jenis objek yang sama



Contoh soal *Inheritance*

Pengusaha

- nomorUsaha
- nama
- alamat
- namaUsaha

PengerajinKayu

- noKTP
- nama
- alamat
- namaUsaha
- omsetPerBulan
- deskripsi

Kayu

- kodeKayu
- jenisPohon
- berat
- Volume

HasilKerajinan

- kodeKerajinan
- bahan
- namaKerajinan

KomoditasEksport

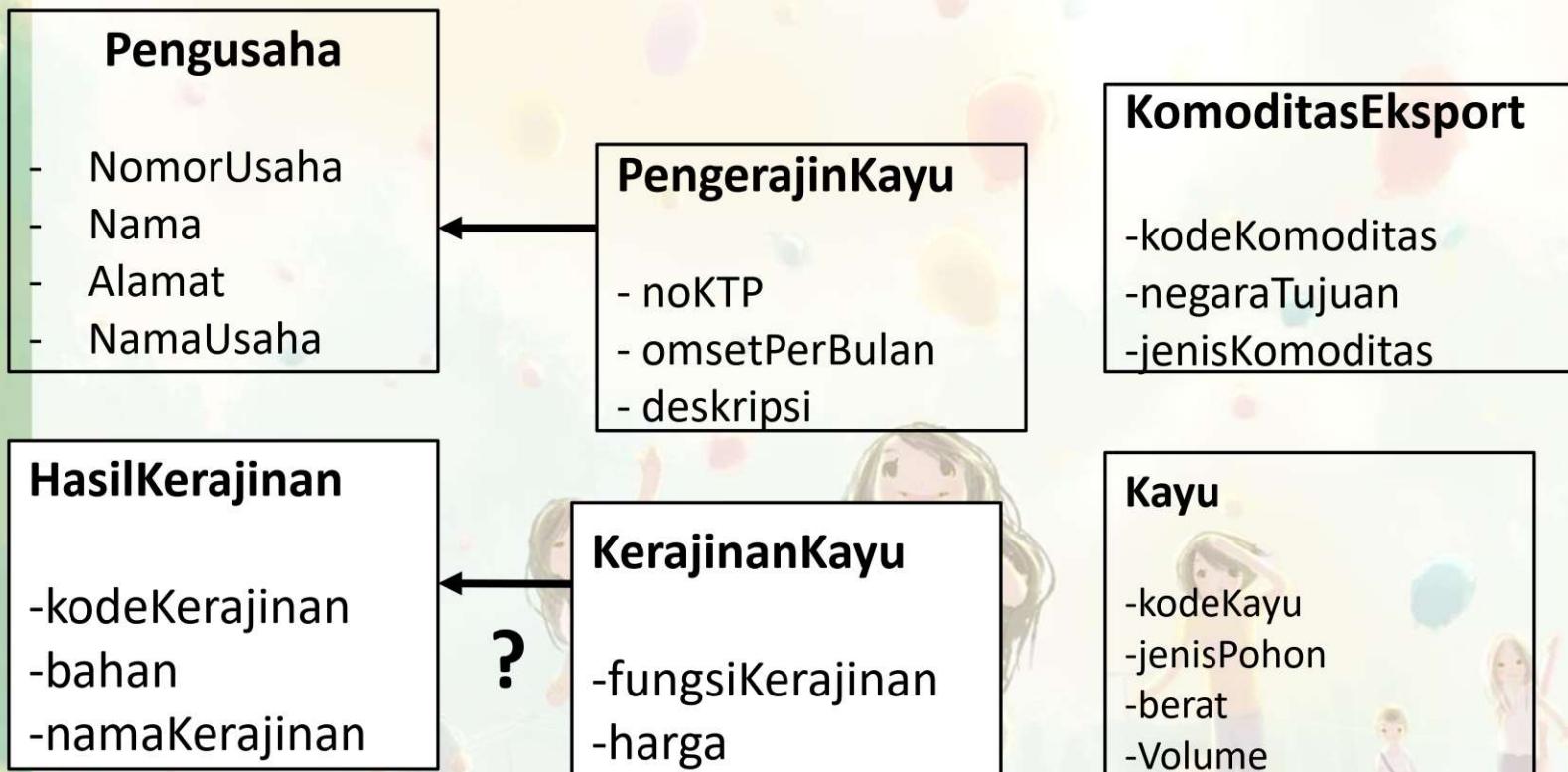
- kodeKomoditas
- negaraTujuan
- jenisKomoditas

KerajinanKayu

- kodeKerajinan
- namaKerajinan
- fungsiKerajinan
- harga

semuanya memiliki metode get set. Buatlah main dengan beberapa kelas di atas yang mampu menerima masukan berupa data pengrajin kayu dan kerajinan kayu untuk kemudian ditampilkan masukan itu dalam bentuk tabel. Gunakan bahasa C++ dan Java.

Desain



```
class Pengusaha{  
    private String nomorUsaha;  
    private String nama;  
    private String alamat;  
    private String namaUsaha;  
  
    Pengusaha() {  
    }  
  
    //get set  
    .....  
}
```

```
class PengerajinKayu extends Pengusaha{  
    private String noKTP;  
    private Double omsetPerBulan;  
    private String deskripsi;  
  
    PengerajinKayu() {  
    }  
  
    //get set  
    .....  
}
```

Restoran

- Kode
- Jenis
- PangsaPasar

IndustriKuliner

- NoIjin
- NamaPemilik
- Kota
- Deskripsi

MasakanIndonesia

- Kode
- Nama
- Bahan
- Harga

BahanMasakan

- Kode
- Nama
- Deskripsi

BahanIkan

- Kode
- Nama
- JenisIkan
- Deskripsi

MasakanAsing

- Kode
- Nama
- Bahan
- Harga
- Negara

PengusahaRestoran

- Kode
- Nama
- Kota
- KodeRestoran

20 % desain
10 % kompilasi
30 % eksekusi
40 % kode program

How About This?



Daftar Pustaka

