Nama : Kelvianto Pratama Harum

NIM : 200210500016

MK : Pemrograman Lanjut

Dosen : Muhammad Fajar B, S.Pd., M.Cs.

PERTEMUAN IV

INHERITANCE

A. Contoh 4.1

Source code:

KelasSatu.java

```java
package com.example.Pertemuan4;
//Bagian dari Contoh 4.1 - App.java
public class KelasSatu {
    public KelasSatu(){
        System.out.println("Konstruktor Kelas Satu");
    }
}
```

KelasDua.java

```java
package com.example.Pertemuan4;
//Bagian dari Contoh 4.1 - App.java
public class KelasDua extends KelasSatu{
    public KelasDua(){
        System.out.println("Kelas Dua");
    }
}
```

Contoh41.java

```java
package com.example.Pertemuan4;

public class Contoh41 {
    public static void main(String[] args) {
        KelasDua kd = new KelasDua();
    }
}
```

Output:

```
> Task :app:run
Konstruktor Kelas Satu
Kelas Dua

BUILD SUCCESSFUL in 2s
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

KelasSatu.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **KelasSatu.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **KelasSatu**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **7**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **KelasSatu**.

Pada baris **4**, dideklarasikan konstruktor untuk class **KelasSatu**, pemanggilan konstruktor ini tidak menggunakan parameter. Scope constructor ini dari baris **4** sampai baris **6**.

Pada baris **5**, digunakan untuk string **"Konstruktor Kelas Satu"**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

KelasDua.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **KelasDua.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **KelasDua** yang diturunkan dari class **KelasSatu**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **7**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **KelasDua**.

Pada baris **4**, dideklarasikan konstruktor untuk class **KelasDua**, pemanggilan konstruktor ini tidak menggunakan parameter. Scope constructor ini dari baris **4** sampai baris **6**.

Pada baris **5**, digunakan untuk string **"Kelas Dua"**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Contoh41.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh41.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Contoh41**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **7**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh41**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **6**.

Pada baris **5**, dibuat sebuah object dari class **mahasiswa4** dengan nama **mhs** tanpa argumen konstruktor.

B. Contoh 4.2

Source code:

superKlas.java

```
1   package com.example.Pertemuan4;
2   //Bagian dari Contoh 4.2 - Super dan Sub Class
    public class superKlas {
        private int nilaiSuper;
5       public superKlas(int nilaiSuper){
6           this.nilaiSuper = nilaiSuper;
7       }
8       public int getNilaiSuper(){
9           return nilaiSuper;
10      }
        private void methodPrivate(){
12          System.out.println("Ini method private");
13      }
14      protected void methodProtected(){
15          System.out.println("Ini method protected");
16      }
17  }
```

subKelas.java

```java
package com.example.Pertemuan4;
//Bagian dari Contoh 4.2 - Super dan Sub Class
public class subKelas extends superKlas{
    private double nilaiSub;
    public subKelas(int nilaiSuper, double nilaiSub){
        super(nilaiSuper);
        this.nilaiSub = nilaiSub;
    }
    public void methodSub(){
        super.methodProtected();
        System.out.println("Nilai Super: "+super.getNilaiSuper());
        System.out.println("Nilai sub: "+this.nilaiSub);
    }
}
```

Contoh42.java

```java
package com.example.Pertemuan4;

public class Contoh42 {
    public static void main(String args[]){
        //bentuk objek superclass
        System.out.println("Objek superclass");
        superKlas sup = new superKlas(5);
        System.out.println("Nilai super: "+sup.getNilaiSuper());
        sup.methodProtected();

        //bentuk objek subclass
        System.out.println("\nObjek Subclass");
        subKelas sub = new subKelas(10, 9.5);
        System.out.println("Pemanggilan method superclass dari objek subclass");
        System.out.println("Nilai super: "+sub.getNilaiSuper());
        sub.methodProtected();
        System.out.println("Pemanggilan method superclass dari subclass");
        sub.methodSub();
    }
}
```

Output:

```
> Task :app:run
Objek superclass
Nilai super: 5
Ini method protected

Objek Subclass
Pemanggilan method superclass dari objek subclass
Nilai super: 10
Ini method protected
Pemanggilan method superclass dari subclass
Ini method protected
Nilai Super: 10
Nilai sub: 9.5

BUILD SUCCESSFUL in 659ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

superKlas.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **superKlas.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **superKlas**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **17**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **superKlas**.

Pada baris **4**, dideklarasikan field dengan nama **nilaiSuper** dengan tipe data **integer** dan modifier **private**, modifier ini berarti **field/method** hanya dapat diakses **didalam class** itu saja, tidak pada object maupun class turunannya.

Pada baris **5**, dideklarasikan konstruktor untuk class **superKlas**, pemanggilan menggunakan parameter variable **nilaiSuper** dengan tipe data **integer**. Scope constructor ini dari baris **5** sampai baris **7**.

Pada baris **6**, didefinisikan nilai **field** nilaiSuper dengan nilai parameter variabel **nilaiSuper**.

Pada baris **8**, dideklarasikan method **getNilaiSuper** dengan tipe data nilai balik **integer** tanpa parameter pemanggilan dan modifier **public**. Scope method ini dari baris **8** hingga baris **10**.

Pada baris **9**, digunakan nilai **field nilaiSuper** sebagai nilai balik untuk method **getNilaiSuper**.

Pada baris **11**, dideklarasikan method tanpa nilai balik (**void**) **methodPrivate** tanpa parameter pemanggilan dengan modifier **private**. Scope method ini dari baris **11** hingga baris **13**.

Pada baris **12**, digunakan untuk menampilkan string **“Ini method private”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **14**, dideklarasikan method tanpa nilai balik (**void**) **methodProtected** tanpa parameter pemanggilan dengan modifier **protected**. Modifier ini berarti **field/method** ini dapat diakses dari objek maupun class turunannya, tapi tidak pada class lain atau package lain.

Pada baris **15**, digunakan untuk menampilkan string **“Ini method protected”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

subKelas.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **subKelas.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **subKelas** yang diturunkan dari class **superKlas**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **14**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **subKelas**.

Pada baris **4**, dideklarasikan field dengan nama **nilaiSub** dengan tipe data **double** dan modifier **private**, modifier ini berarti **field/method** hanya dapat diakses **didalam class** itu saja, tidak pada object maupun class turunannya.

Pada baris **5**, dideklarasikan konstruktor untuk class **subKelas**, pemanggilan konstruktor ini menggunakan parameter variabel **nilaiSuper** dengan tipe data **integer**, dan variabel **nilaiSub** dengan tipe data **double**. Scope constructor ini dari baris **5** sampai baris **8**.

Pada baris **6**, dipanggil konstruktor **super** class dari **subKelas** yaitu **superKlas** dengan argumen pemanggilan nilai parameter **nilaiSuper**.

Pada baris **7**, didefinisikan nilai field **nilaiSub** dengan nilai parameter variabel **nilaiSub**.

Pada baris **9**, dideklarasikan method tanpa nilai balik (**void**) **methodSub** tanpa parameter pemanggilan dengan modifier **public**. Scope method ini dari baris **9** hingga baris **13**.

Pada baris **10**, dipanggil method **methodProtected** dari **super** class (**superKlas**).

Pada baris **11**, digunakan untuk menampilkan string **“Nilai super: ”** diikuti dengan nilai balik pemanggilan method **getNilaiSuper** dari **super** class (**superKlas**). Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **12**, digunakan untuk menampilkan string **"Nilai sub: "** diikuti dengan nilai **field nilaiSub**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Contoh42.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Contoh42.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Contoh42**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **20**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Contoh42**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **19**.

Pada baris **6,** digunakan untuk menampilkan string **"Objek superclass".** Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **7**, dibuat object **sup** dari class **superKlas** dengan argumen konstruktor nilai **5.**

Pada baris **8,** digunakan untuk menampilkan string **"Nilai super: "** diikuti dengan nilai balik pemanggilan method **getNilaiSuper** pada object **sup**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **9**, dipanggil method **methodProtected** dari object **sup**.

Pada baris **12**, digunakan untuk menampilkan string **"Objek Subclass"** diawali dengan baris baru **(\n)**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **13**, dibuat object **sub** dari class **subKelas** dengan argumen konstruktor nilai **10** dan nilai **9.5**.

Pada baris **14,** digunakan untuk menampilkan string **"Pemanggilan method superclass dari objek subclass".** Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **15,** digunakan untuk menampilkan string **"Nilai super: "** diikuti dengan nilai balik pemanggilan method **getNilaiSuper** pada object **sub**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **16**, dipanggil method **methodProtected** dari object **sub**.

Pada baris **17,** digunakan untuk menampilkan string **"Pemanggilan method superclass dari objek subclass".** Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **18**, dipanggil method **methodSub** dari object **sub**.

C. Praktikum 4.1

Source code:

Manusia.java

```java
package com.example.Pertemuan4;
//Bagian dari Praktikum 4.1
public class Manusia {
    protected String nama;
    protected int umur;
    public String getNama(){
        return this.nama;
    }
    public void setNama(String nama){
        this.nama = nama;
    }
    public int getUmur(){
        return this.umur;
    }
    public void setUmur(int umur){
        this.umur = umur;
    }
    public void siapaKamu(){
        System.out.println("Saya manusia");
    }
}
```

Dosen.java

```java
package com.example.Pertemuan4;
//Bagian dari Praktikum 4.1
public class Dosen extends Manusia {
    private String nip;
    private String matakuliah;
    public String getNIP(){
        return this.nip;
    }
    public void setNIP(String nip){
        this.nip = nip;
    }
    public String getMatakuliah(){
        return this.matakuliah;
    }
    public void setMatakuliah(String matakuliah){
        this.matakuliah = matakuliah;
    }
    public void mengajarApa(){
        System.out.println("Saya " + nama +
                           " umur " + umur +
                           " mengajar " + matakuliah);
    }
}
```

Mahasiswa.java

```java
package com.example.Pertemuan4;
//Bagian dari Praktikum 4.1
public class Mahasiswa extends Manusia{
    private String nim;
    private String kelas;
    public String getNIM(){
        return this.nim;
    }
    public void setNIM(String nim){
        this.nim = nim;
    }
    public String getKelas(){
        return this.kelas;
    }
    public void setKelas(String kelas){
        this.kelas = kelas;
    }
    public void kelasApa(){
        System.out.println("Saya " + nama +
                           " umur " + umur +
                           " mahasiswa di kelas " + kelas);
    }
}
```

Praktikum41.java

```java
package com.example.Pertemuan4;

public class Praktikum41 {
    public static void main(String args[]){
        Dosen dosen = new Dosen();
        dosen.setNama("Fhatiah Adiba");
        dosen.setUmur(30);
        dosen.setNIP("0228079202");
        dosen.setMatakuliah("Pemrograman Lanjut");
        System.out.println("NIP dosen: "+ dosen.getNIP());
        dosen.mengajarApa();
        System.out.println();

        Mahasiswa mahasiswa = new Mahasiswa();
        mahasiswa.setNama("Kelvin Pratama");
        mahasiswa.setUmur(21);
        mahasiswa.setNIM("20021050016");
        mahasiswa.setKelas("Network and Security");
        System.out.println("NIM Mahasiswa: "+ mahasiswa.getNIM());
        mahasiswa.kelasApa();
    }
}
```

Output:

```
> Task :app:run
NIP dosen: 0228079202
Saya Fhatiah Adiba umur 30 mengajar Pemrograman Lanjut

NIM Mahasiswa: 20021050016
Saya Kelvin Pratama umur 21 mahasiswa di kelas Network and Security

BUILD SUCCESSFUL in 1s
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Manusia.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Manusia.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Manusia**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **21**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Manusia**.

Pada baris **4**, dideklarasikan **field** dengan nama **nama** dan tipe data **String** dengan modifier **protected**.

Pada baris **5**, dideklarasikan **field** dengan nama **umur** dan tipe data **integer** dengan modifier **protected**.

Pada baris **6**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getNama**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **6** hingga baris **8**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **7**, nilai **field nama** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **9**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setNama**, dalam method ini parameter yang digunakan adalah variabel **nama** dengan tipe data **String**. Scope method ini mulai dari baris **9** hingga baris **11**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **10**, diberikan nilai pada field **nama** dengan nilai dari argumen variabel **nama** pada saat pemanggilan method **setNama**.

Pada baris **12**, dideklarasikan method dengan modifier **public**, tipe data balik **int**, nama **getUmur**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **12** hingga baris **14**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **13**, nilai **field umur** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **15**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setUmur**, dalam method ini parameter yang digunakan adalah variabel **umur** dengan tipe data **integer**. Scope method ini mulai dari baris **15** hingga baris **17**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **16**, diberikan nilai pada field **umur** dengan nilai dari argumen variabel **umur** pada saat pemanggilan method **setUmur**.

Pada baris **18**, dideklarasikan method tanpa nilai balik (**void**) dengan nama **siapaKamu** tanpa parameter pemanggilan dan modifier **public**.

Pada baris **19**, digunakan untuk menampilkan string **“Saya manusia”**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Dosen.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Dosen.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Dosen**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **23**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Dosen**.

Pada baris **4**, dideklarasikan **field** dengan nama **nip** dan tipe data **String** dengan modifier **private**.

Pada baris **5**, dideklarasikan **field** dengan nama **matakuliah** dan tipe data **String** dengan modifier **private**.

Pada baris **6**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getNIP**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **6** hingga baris **8**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **7**, nilai **field nip** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **9**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setNIP**, dalam method ini parameter yang digunakan adalah variabel **nip** dengan tipe data **String**. Scope method ini mulai dari baris **9** hingga baris **11**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **10**, diberikan nilai pada field **nip** dengan nilai dari argumen variabel **nip** pada saat pemanggilan method **setNIP**.

Pada baris **12**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getMatakuliah**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **12** hingga baris **14**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **13**, nilai **field matakuliah** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **15**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setMatakuliah**, dalam method ini parameter yang digunakan adalah variabel **matakuliah** dengan tipe data **String**. Scope method ini mulai dari baris **15** hingga baris **17**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **16**, diberikan nilai pada field **matakuliah** dengan nilai dari argumen variabel **matakuliah** pada saat pemanggilan method **setMatakuliah**.

Pada baris **18**, dideklarasikan method tanpa nilai balik (**void**) dengan nama **mengajarApa** tanpa parameter pemanggilan dan modifier **public**.

Pada baris **19**, digunakan untuk menampilkan string **“Saya ”** diikuti nilai field **nama** lalu string **“ umur “** diikuti nilai field **umur** lalu string **“ mengajar “** diikuti nilai field **matakuliah**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Mahasiswa.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Mahasiswa.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Mahasiswa**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **23**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Mahasiswa**.

Pada baris **4**, dideklarasikan **field** dengan nama **nim** dan tipe data **String** dengan modifier **private**.

Pada baris **5**, dideklarasikan **field** dengan nama **kelas** dan tipe data **String** dengan modifier **private**.

Pada baris **6**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getNIM**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **6** hingga baris **8**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **7**, nilai **field nim** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **9**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setNIM**, dalam method ini parameter yang digunakan adalah variabel **nim** dengan tipe data **String**. Scope method ini mulai dari baris **9** hingga baris **11**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **10**, diberikan nilai pada field **nim** dengan nilai dari argumen variabel **nim** pada saat pemanggilan method **setNIM**.

Pada baris **12**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getKelas**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **12** hingga baris **14**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **13**, nilai **field kelas** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **15**, dideklarasikan method tanpa nilai balik **(void)** dengan nama **setKelas**, dalam method ini parameter yang digunakan adalah variabel **kelas** dengan tipe data **String**. Scope method ini mulai dari baris **15** hingga baris **17**. Style coding ini biasa disebut **setter** pada style **setter-getter**.

Pada baris **16**, diberikan nilai pada field **kelas** dengan nilai dari argumen variabel **kelas** pada saat pemanggilan method **setKelas**.

Pada baris **18**, dideklarasikan method tanpa nilai balik (**void**) dengan nama **kelasApa** tanpa parameter pemanggilan dan modifier **public**.

Pada baris **19**, digunakan untuk menampilkan string **"Saya "** diikuti nilai field **nama** lalu string **" umur "** diikuti nilai field **umur** lalu string **" mahasiswa di kelas "** diikuti nilai field **kelas**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Praktikum41.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Praktikum41.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Praktikum41**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **22**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Praktikum41**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **21**.

Pada baris **5**, dibuat object **dosen** dari class **Dosen** tanpa argumen konstruktor.

Pada baris **6**, dipanggil method **setNama** dari object **dosen** dengan argumen pemanggilan **"Fhatiah Adiba"**.

Pada baris **7**, dipanggil method **setUmur** dari object **dosen** dengan argumen pemanggilan **30**.

Pada baris **8**, dipanggil method **setNIP** dari object **dosen** dengan argumen pemanggilan **“0228079202”**.

Pada baris **9**, dipanggil method **setMatakuliah** dari object **dosen** dengan argumen pemanggilan **“Pemrograman Lanjut”**.

Pada baris **10**, digunakan untuk menampilkan string **“NIP dosen: ”** diikuti nilai balik pemanggilan method **getNIP** dari object **dosen**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **11**, dipanggil method **mengajarApa** dari object **dosen**.

Pada baris **12**, dibuat output baris kosong ke console.

Pada baris **14**, dibuat object **mahasiswa** dari class **Mahasiswa** tanpa argumen konstruktor.

Pada baris **15**, dipanggil method **setNama** dari object **mahasiswa** dengan argumen pemanggilan **“Kelvin Pratama”**.

Pada baris **16**, dipanggil method **setUmur** dari object **mahasiswa** dengan argumen pemanggilan **21**.

Pada baris **17**, dipanggil method **setNIM** dari object **mahasiswa** dengan argumen pemanggilan **“200210500016”**.

Pada baris **18**, dipanggil method **setKelas** dari object **mahasiswa** dengan argumen pemanggilan **“Network and Security”**.

Pada baris **19**, digunakan untuk menampilkan string **“NIM mahasiswa: ”** diikuti nilai balik pemanggilan method **getNIM** dari object **mahasiswa**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **11**, dipanggil method **kelasApa** dari object **mahasiswa**.

D. Tantangan

Buatlah sebuah program implementasi inheritance yang berkaitan dengan kehidupan sehari-hari.

Source code:

Kendaraan.java

```java
package com.example.Pertemuan4;
//Bagian dari Tantangan
public class Kendaraan {
    private String merek;
    private int tahunProduksi;
    public Kendaraan(String merek, int tahunProduksi){
        this.merek = merek;
        this.tahunProduksi = tahunProduksi;
    }
    public String getMerek(){
        return this.merek;
    }
    public int getTahunProduksi(){
        return this.tahunProduksi;
    }
    public void Informasi(){
        System.out.println("Merek: " + merek + ", Tahun Produksi: " + tahunProduksi);
    }
}
```

Mobil.java

```java
package com.example.Pertemuan4;
//Bagian dari Tantangan
public class Mobil extends Kendaraan{
    private String kategori;
    private String model;
    public Mobil(String merek, int tahunProduksi, String kategori, String model){
        super(merek, tahunProduksi);
        this.kategori = kategori;
        this.model = model;
    }
    public String getKategori(){
        return this.kategori;
    }
    public String getModel(){
        return this.model;
    }
    public void tampilkanInformasi(){
        super.Informasi();
        System.out.println("Model: "+model);
        System.out.println("Kategori: "+kategori);
    }
}
```

Motor.java

```java
package com.example.Pertemuan4;
//Bagian dari Tantangan
public class Motor extends Kendaraan{
    private int ukuranSilinder;
    private String model;
    public Motor(String merek, int tahunProduksi, int ukuranSilinder, String model){
        super(merek, tahunProduksi);
        this.ukuranSilinder = ukuranSilinder;
        this.model = model;
    }
    public int getUkuranSilinder(){
        return this.ukuranSilinder;
    }
    public String getModel(){
        return this.model;
    }
    public void tampilkanInformasi(){
        super.Informasi();
        System.out.println("Model: "+model);
        System.out.println("Ukuran silinder: " + ukuranSilinder + " cc");
    }
}
```

Tantangan.java

```java
package com.example.Pertemuan4;

public class Tantangan {
    public static void main(String args[]){
        Mobil mobil = new Mobil("Toyota", 2007, "Sedan", "Accord 7th Gen");
        Motor motor = new Motor("Honda", 2015, 110, "Beat Sporty");

        System.out.println("Informasi mobil: ");
        mobil.tampilkanInformasi();
        System.out.println();

        System.out.println("Informasi motor: ");
        motor.tampilkanInformasi();
    }
}
```

Output:

```
> Task :app:run
Informasi mobil:
Merek: Toyota, Tahun Produksi: 2007
Model: Accord 7th Gen
Kategori: Sedan

Informasi motor:
Merek: Honda, Tahun Produksi: 2015
Model: Beat Sporty
Ukuran silinder: 110 cc

BUILD SUCCESSFUL in 985ms
2 actionable tasks: 1 executed, 1 up-to-date
```

Penjelasan:

Kendaraan.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Kendaraan.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Kendaraan**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **19**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Kendaraan**.

Pada baris **4**, dideklarasikan **field** dengan nama **merek** dan tipe data **String** dengan modifier **private**.

Pada baris **5**, dideklarasikan **field** dengan nama **tahunProduksi** dan tipe data **integer** dengan modifier **private**.

Pada baris **6**, dideklarasikan konstruktor untuk class **Kendaraan**, pemanggilan konstruktor ini menggunakan parameter variabel **merek** dengan tipe data **String**, dan variabel **tahunProduksi** dengan tipe data **integer**. Scope constructor ini dari baris **6** sampai baris **9**.

Pada baris **7**, didefinisikan nilai **field merek** dengan nilai parameter pemanggilan **merek**.

Pada baris **8**, didefinisikan nilai **field tahunProduksi** dengan nilai parameter pemanggilan **tahunProduksi**.

Pada baris **10**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getMerek**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **10** hingga baris **12**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **11**, nilai **field merek** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **13**, dideklarasikan method dengan modifier **public**, tipe data balik **integer**, nama **getTahunProduksi**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **13** hingga baris **15**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **14**, nilai **field tahunProduksi** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **16**, dideklarasikan method **Informasi** tanpa nilai balik (**void**) dengan modifier **public** dan tidak menggunakan parameter pemanggilan.

Pada baris **17**, digunakan untuk menampilkan string **"Merek: "** diikuti nilai field **merek** lalu string **", Tahun Produksi: "** diikuti nilai **field tahunProduksi**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Mobil.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Mobil.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Mobil** yang diturunkan dari class **Kendaraan**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **22**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Mobil**.

Pada baris **4**, dideklarasikan **field** dengan nama **kategori** dan tipe data **String** dengan modifier **private**.

Pada baris **5**, dideklarasikan **field** dengan nama **model** dan tipe data **String** dengan modifier **private**.

Pada baris **6**, dideklarasikan konstruktor untuk class **Mobil**, pemanggilan konstruktor ini menggunakan parameter variabel **merek** dengan tipe data **String**, variabel **tahunProduksi** dengan tipe data **integer**, variabel **kategori** dengan tipe data **String** dan variabel **model** dengan tipe data **String**. Scope constructor ini dari baris **6** sampai baris **10**.

Pada baris **7**, dipanggil konstruktor dari **super** class dengan argumen nilai variabel **merek** dan **tahunProduksi**.

Pada baris **8**, didefinisikan nilai **field kategori** dengan nilai parameter pemanggilan **kategori**.

Pada baris **9**, didefinisikan nilai **field model** dengan nilai parameter pemanggilan **model**.

Pada baris **11**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getKategori**, dan tidak menggunakan parameter. Scope method ini mulai

dari baris **11** hingga baris **13**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **12**, nilai **field kategori** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **14**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getModel**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **14** hingga baris **16**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **15**, nilai **field model** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **17**, dideklarasikan method **tampilkanInformasi** tanpa nilai balik (**void**) dengan modifier **public** dan tidak menggunakan parameter pemanggilan.

Pada baris **18**, dipanggilan method **Informasi** dari **super** class.

Pada baris **19**, digunakan untuk menampilkan string **"Model: "** diikuti nilai field **model**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **20**, digunakan untuk menampilkan string **"Kategori: "** diikuti nilai field **kategori**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Motor.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Motor.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Motor** yang diturunkan dari class **Kendaraan**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **22**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Motor**.

Pada baris **4**, dideklarasikan **field** dengan nama **ukuranSilinder** dan tipe data **int** dengan modifier **private**.

Pada baris **5**, dideklarasikan **field** dengan nama **model** dan tipe data **String** dengan modifier **private**.

Pada baris **6**, dideklarasikan konstruktor untuk class **Motor**, pemanggilan konstruktor ini menggunakan parameter variabel **merek** dengan tipe data **String**, variabel **tahunProduksi** dengan tipe data **integer**, variabel **ukuranSilinder** dengan tipe data **int** dan variabel **model** dengan tipe data **String**. Scope constructor ini dari baris **6** sampai baris **10**.

Pada baris **7**, dipanggil konstruktor dari **super** class dengan argumen nilai variabel **merek** dan **tahunProduksi**.

Pada baris **8**, didefinisikan nilai **field ukuranSilinder** dengan nilai parameter pemanggilan **ukuranSilinder**.

Pada baris **9**, didefinisikan nilai **field model** dengan nilai parameter pemanggilan **model**.

Pada baris **11**, dideklarasikan method dengan modifier **public**, tipe data balik **int**, nama **getUkuranSilinder**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **11** hingga baris **13**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **12**, nilai **field ukuranSilinder** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **14**, dideklarasikan method dengan modifier **public**, tipe data balik **String**, nama **getModel**, dan tidak menggunakan parameter. Scope method ini mulai dari baris **14** hingga baris **16**. Style coding ini biasa disebut **getter** pada style **setter-getter**.

Pada baris **15**, nilai **field model** dimuat dan dikembalikan kepada pemanggil method.

Pada baris **17**, dideklarasikan method **tampilkanInformasi** tanpa nilai balik (**void**) dengan modifier **public** dan tidak menggunakan parameter pemanggilan.

Pada baris **18**, dipanggilan method **Informasi** dari **super** class.

Pada baris **19**, digunakan untuk menampilkan string **“Model: ”** diikuti nilai field **model**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **20**, digunakan untuk menampilkan string **“Ukuran silinder: ”** diikuti nilai field **ukuranSilinder**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini

pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Tantangan.java

Pada baris **1**, dideklarasikan package dimana file source ini bergabung pada contoh ini file **Tantangan.java** tergabung pada package **com.example.Pertemuan4**

Pada baris **3**, dideklarasikan Class bernama **Tantangan**, class ini dideklarasikan dengan modifier **public** yang berarti class ini dapat diakses dari file maupun package manapun. Class ini mencakup semua kode yang ada pada baris **3** hingga baris **15**, sehingga semua kode baik method dan identifiers dalam blok ini adalah scope dari class **Tantangan**.

Pada baris **4**, dideklarasikan method **main** dengan parameter String args[], method ini adalah method yang akan pertama kali dieksekusi pada saat runtime program berjalan. Scope method ini adalah baris **4** hingga baris **14**.

Pada baris **5**, dibuat sebuah object **mobil** dari class **Mobil** dengan argumen konstruktor **“Toyota”**, **2007**, **“Sedan”**, dan **“Accord 7th Gen”**.

Pada baris **6**, dibuat sebuah object **motor** dari class **Motor** dengan argumen konstruktor **“Honda”**, **2015**, **110**, dan **“Beat Sporty”**.

Pada baris **8**, digunakan untuk menampilkan string **“Informasi mobil: “**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **9**, dipanggil method **tampilkanInformasi** dari object **mobil**.

Pada baris **10**, ditampilkan baris kosong pada console.

Pada baris **11**, digunakan untuk menampilkan string **“Informasi motor: “**. Output ini dihasilkan menggunakan method bawaan dari Java yang berada pada class **System**, member **out**, method **println()**. Method ini pada dasarnya akan menampilkan apapun yang diberikan pada calling argument ke runtime console.

Pada baris **12**, dipanggil method **tampilkanInformasi** dari object **motor**.