

LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

BAB : INHERITANCE
NAMA : Dani Adrian
NIM : 225150201111009
ASISTEN : Tengku Muhammad Rafi Rahardiansyah
Muhammad Bin Djafar Almasyhur
TGL PRAKTIKUM : 29 Maret 2023

Data dan Hasil Percobaan**Pertanyaan**

1. Jalankan code program diatas dan benahi jika menemukan kesalahan!

Source Code

1	Tulis source code di sini pake courier new 12
---	---

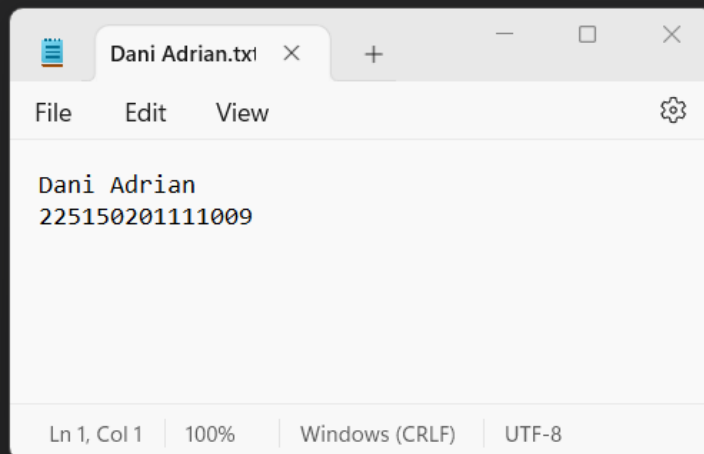
Output

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam> & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenam_d5fa685a\bin' 'MainEmployee'

nama boss : Steven, salary = 85000.0
nama staff : Donni, salary = 50000.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam> 
```



Penjelasan

Tidak ada kesalahan pada program di atas

2. Bagaimana cara konstruktor pada subclass memanggil konstruktor di superclass nya? Apakah hal itu perlu dilakukan? Sertakan alasan anda !

Source Code

1	Tulis source code di sini pake courier new 12
---	---

Ouput**Penjelasan**

Syntax `super()` digunakan untuk memanggil konstruktor dari superclass pada konstruktor dari subclass. Konstruktor yang dipanggil pada superclass akan cocok dengan isi parameter pada syntax `super()` pada konstruktor dari subclass. Pemanggilan konstruktor superclass secara manual diperlukan jika Anda ingin memanggil konstruktor superclass yang memerlukan parameter. Selain itu, konstruktor superclass default (tanpa parameter) akan dipanggil secara otomatis pada baris pertama konstruktor subclass.

3. Tambahkan constructor pada class Employee dengan parameter String name! amati perubahan apa yang terjadi, jelaskan jawaban anda!

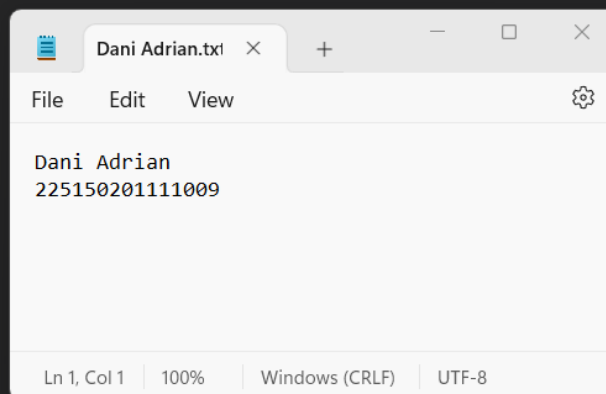
Source code

1.	<code>import java.util.*;</code>
2.	
3.	<code>public class Employee {</code>
4.	<code> private String name;</code>
5.	<code> private double salary;</code>
6.	<code> private Date hireday;</code>
7.	
8.	<code> public Employee(String name, double salary, int year, int month, int day) {</code>
9.	<code> this.name = name;</code>
10.	<code> this.salary = salary;</code>
11.	<code> GregorianCalendar calendar = new GregorianCalendar(year, month - 1, day);</code>
12.	<code> this.hireday = calendar.getTime();</code>
13.	<code> }</code>
14.	<code> public Employee(String name) {</code>
15.	<code> this.name = name;</code>

16.	}
17.	public Date getHireDay() {
18.	return hireday;
19.	}
20.	
21.	public String getName() {
22.	return name;
23.	}
24.	
25.	public double getSalary() {
26.	return salary;
27.	}
28.	
29.	public void raiseSalary(double byPercent) {
30.	double raise = salary * byPercent / 100;
31.	salary += raise;
32.	}
33.	}

Output

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam> c++; cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam';
& 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspac
eStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenam_d5fa685a\bin' 'MainEmployee'
nama boss : Steven, salary = 85000.0
nama staff : Donni, salary = 50000.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam>
```



Penjelasan

Output akan tetap sama karena di kelas Manager, konstruktor superclass yang dipanggil adalah konstruktor yang memiliki parameter lengkap, bukan hanya parameter nama saja. Karena

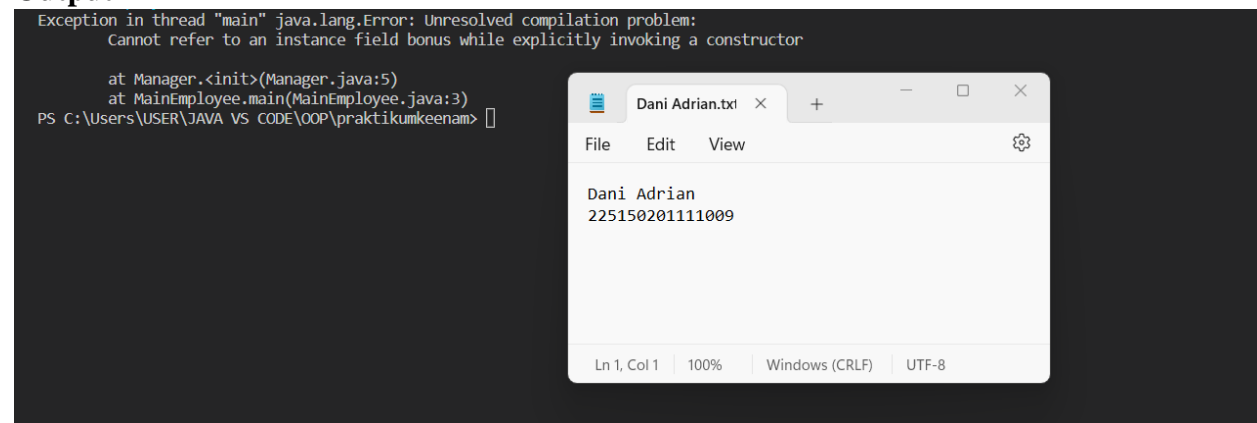
parameter yang digunakan sama seperti pada kode sebelumnya, maka output yang dihasilkan juga akan tetap sama.

4. Pada class manager baris ke 5. Setelah variable day tambahkan variable bonus! Amati apa yang terjadi dan mengapa demikian?

Source Code

1.	<code>public class Manager extends Employee {</code>
2.	<code> private double bonus;</code>
3.	
4.	<code> public Manager(String name, double salary, int year,</code> <code>int month, int day) {</code>
5.	<code> super(name, salary, year, month, day, bonus);</code>
6.	<code> bonus = 0;</code>
7.	<code> }</code>
8.	
9.	<code> public void setBonus(double bonus) {</code>
10.	<code> this.bonus = bonus;</code>
11.	<code> }</code>
12.	
13.	<code> public double getSalary() {</code>
14.	<code> double baseSalary = super.getSalary();</code>
15.	<code> return baseSalary + bonus;</code>
16.	<code> }</code>
17.	
18.	<code>}</code>

Output



Penjelasan

Ketika menggunakan `super()` dalam subclass, konstruktor superclass akan dipanggil dengan parameter yang sesuai dengan parameter yang diberikan pada `super()`. Jika tidak ada

konstruktor pada superclass yang memiliki parameter yang sesuai dengan parameter yang dipanggil menggunakan `super()`, maka akan terjadi error pada saat kompilasi. Dalam kasus ini, terjadi error karena tidak ada konstruktor pada superclass yang memiliki parameter yang sesuai dengan parameter yang dipanggil pada subclass menggunakan `super()`.

5. Untuk apa digunakan keyword `this` pada class `manager` dan `employee`? Hapus keyword `this` dan amati apa yang terjadi?

Source Code

1.	<code>import java.util.*;</code>
2.	
3.	<code>public class Employee {</code>
4.	<code> private String name;</code>
5.	<code> private double salary;</code>
6.	<code> private Date hireday;</code>
7.	
8.	<code> public Employee(String name, double salary, int year, int month, int day) {</code>
9.	<code> name = name;</code>
10.	<code> salary = salary;</code>
11.	<code> GregorianCalendar calendar = new GregorianCalendar(year, month - 1, day);</code>
12.	<code> hireday = calendar.getTime();</code>
13.	<code> }</code>
14.	<code> public Employee(String name) {</code>
15.	<code> name = name;</code>
16.	<code> }</code>
17.	<code> public Date getHireDay() {</code>
18.	<code> return hireday;</code>
19.	<code> }</code>
20.	
21.	<code> public String getName() {</code>
22.	<code> return name;</code>
23.	<code> }</code>
24.	
25.	<code> public double getSalary() {</code>
26.	<code> return salary;</code>
27.	<code> }</code>
28.	
29.	<code> public void raiseSalary(double byPercent) {</code>
30.	<code> double raise = salary * byPercent / 100;</code>
31.	<code> salary += raise;</code>

32.	}
33.	}

1.	public class Manager extends Employee {
2.	private double bonus;
3.	
4.	public Manager(String name, double salary, int year, int month, int day) {
5.	super(name, salary, year, month, day);
6.	bonus = 0;
7.	}
8.	
9.	public void setBonus(double bonus) {
10.	bonus = bonus;
11.	}
12.	
13.	public double getSalary() {
14.	double baseSalary = super.getSalary();
15.	return baseSalary + bonus;
16.	}
17.	
18.	}

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam> c::; cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe'
'-cp' 'c:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenam_d5fa685a\bin' 'MainEmployee
nama boss : null, salary = 0.0
nama staff : null, salary = 0.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam>

```

Dani Adrian
225150201111009

Penjelasan

Fungsi "this" adalah untuk mengakses variable instance dalam suatu class dan bukan variable lokal, terutama ketika keduanya memiliki nama yang sama. Tujuan penggunaan "this" adalah untuk membedakan antara kedua jenis variable tersebut. Dengan menggunakan "this," kita dapat memastikan bahwa yang kita akses adalah variable instance dalam class tersebut, bukan variable lokal.

6. Tambahkan constructor pada class Employee dengan parameter bertipe data string bernama name yang nantinya bila constructor ini akan dipanggil akan menginisiasi variable name! Amati perubahannya pada class anak dan jelaskan! Benahi bila terjadi kesalahan!

Source Code

1.	public class MainEmployee {
2.	public static void main(String[] args) {
3.	Manager boss = new Manager("Steven", 80000,
4.	1987, 12, 15);
5.	boss.setBonus(5000);
6.	Employee staff = new Employee("Donni", 50000,
7.	1989, 10, 1);
8.	System.out.println("nama boss : " +
9.	boss.getName() + ", salary = " + boss.getSalary());
10.	System.out.println("nama staff : " +
11.	staff.getName() + ", salary = " + staff.getSalary());
12.	}

Output

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenam_d5fa685a\bin' 'MainEmployee'
nama boss : Steven, salary = 85000.0
nama staff : Donni, salary = 50000.0
nama staff2 : Rahmat, salary = 0.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenam>
  
```

Penjelasan

Dengan menambahkan konstruktor baru yang memiliki parameter string bernama "name," kita dapat membuat instance hanya dengan satu parameter string saja yang akan diisi ke dalam instance variable "name." Namun, nilai dari sisa instance variable masih akan kosong karena belum diisi.

7. Pada bab sebelumnya anda telah belajar mengenai konsep encapsulation, jelaskan mengapa pada super class menggunakan midfier protected? Apa yang terjadi jika modifier anda ubah menjadi private atau public? Jelaskan!

Source Code

1	Tulis source code di sini pake courier new 12
---	---

Output

Penjelasan

Dengan menggunakan kata kunci "protected", kita diperbolehkan untuk mengakses variabel atau metode yang berasal dari superclass pada subclass, meskipun subclass berada dalam paket yang berbeda. Jika kita menggunakan kata kunci "private", maka tidak dapat mengakses variabel atau metode tersebut pada subclass. Namun, jika kita menggunakan kata kunci "public", class yang tidak inherit dari superclass juga dapat mengakses variabel atau metode tersebut, bahkan jika mereka berada di paket yang berbeda. Prinsip encapsulation yang bertujuan untuk menjaga akses dari variabel agar tidak sembarangan akan terganggu.

8. Ubahlah acces modifier method pada kelas employee menjadi :
- Private
 - Protected

Amati perubahan apa yang terjadi? Jelaskan jawaban anda dengan detail!

Source Code

a. Private

1.	import java.util.*;
2.	public class Employee {
3.	private String name;
4.	private double salary;
5.	private Date hireday;
6.	
7.	public Employee(String name, double salary, int year, int month, int day) {
8.	this.name = name;
9.	this.salary = salary;
10.	GregorianCalendar calendar = new GregorianCalendar(year, month - 1, day);
11.	hireday = calendar.getTime();
12.	}

13.	public Employee(String name) {
14.	this.name = name;
15.	}
16.	private Date getHireday() {
17.	return hireday;
18.	}
19.	private String getName() {
20.	return name;
21.	}
22.	private double getSalary() {
23.	return salary;
24.	}
25.	private void raiseSalary(double byPercent) {
26.	double raise = salary * byPercent / 100;
27.	salary += raise;
28.	}
29.	}

b. Protected

1.	import java.util.*;
2.	public class Employee {
3.	private String name;
4.	private double salary;
5.	private Date hireday;
6.	
7.	public Employee(String name, double salary, int year, int month, int day) {
8.	this.name = name;
9.	this.salary = salary;
10.	GregorianCalendar calendar = new GregorianCalendar(year, month - 1, day);
11.	hireday = calendar.getTime();
12.	}
13.	public Employee(String name) {
14.	this.name = name;
15.	}
16.	protected Date getHireday() {
17.	return hireday;
18.	}
19.	protected String getName() {
20.	return name;
21.	}

22.	<code>protected double getSalary() {</code>
23.	<code> return salary;</code>
24.	<code>}</code>
25.	<code>protected void raiseSalary(double byPercent) {</code>
26.	<code> double raise = salary * byPercent / 100;</code>
27.	<code> salary += raise;</code>
28.	<code>}</code>
29.	<code>}</code>

Output

a. Private

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\jav
a.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenan_d5fa685a\bin' '
MainEmployee'
Exception in thread "main" java.lang.Error: Unresolved compilation problems:
    The method getName() from the type Employee is not visible
    The method getName() from the type Employee is not visible
    The method getSalary() from the type Employee is not visible
    The method getName() from the type Employee is not visible
    The method getSalary() from the type Employee is not visible
    at MainEmployee.main(MainEmployee.java:6)
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan>

```

b. Protected

```

PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan> c:: cd 'c:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan'; & 'C:\Program Files\Java\jdk1.8.0_111\bin\jav
a.exe' '-cp' 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\046e179603cdd7319eb81daa567a0ea5\redhat.java\jdt_ws\praktikumkeenan_d5fa685a\bin' '
MainEmployee'
nama boss : Steven, salary = 85000.0
nama staff : Donni, salary = 50000.0
nama staff2 : Rahmat, salary = 0.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikumkeenan>

```

Penjelasan

- Private : akan terjadi error karena tidak dapat diakses oleh subclassnya
- Protected : tetap bekerja seperti akses modifier public

Tugas Praktikum

Susunlah program sesuai studi kasus di bawah ini :

1. Manusia.java

Kelas manusia merupakan kelas induk dengan definisi sebagai berikut:

- nama : String
- jenisKelamin : Boolean (true : laki-laki, false : perempuan)
- nik : String
- menikah : boolean
- + setter, getter
- + getTunjangan() : double
- + getPendapatan() : double
- + toString() : String

(Keterangan)

- Tunjangan untuk yang telah menikah adalah apabila laki-laki akan mendapat \$25 sedangkan perempuan mendapat \$20
- Tunjangan untuk yang belum menikah adalah \$15
- toString() menampilkan nama, nik, jenis kelamin dan jumlah pendapatan.

2. MahasiswaFilkom.java

Kelas mahasiswa merupakan kelas turunan dari Manusia dengan definisi sebagai berikut:

- nim : String
- ipk : double
- + setter, getter
- + getStatus() : String
- + getBeasiswa() : double
- + toString() : String

(Keterangan)

- Beasiswa untuk ipk 3.0 – 3.5 mendapat \$50 dan untuk 3.5 – 4 mendapat \$75
- Status untuk mendapatkan angkatan dan prodi (menurut kaidah FILKOM UB)

1651506XXXXXXX

- Digit ke 1-2 adalah Angkatan
- Digit ke 7 adalah prodi

- 2 Teknik Informatika
- 3 Teknik Komputer
- 4 Sistem Informasi
- 6 Pendidikan Teknologi Informasi
- 7 Teknologi Informasi

Dengan pengembalian dengan format : prodi Angkatan, contoh : Sistem Informasi, 2017

- toString() menampilkan atribut induk + nim, ipk, dan status

3. Pekerja.java

Kelas Pekerja merupakan kelas turunan dari Manusia dengan definisi sebagai berikut:

- gaji : double
- tahunMasuk : LocalDate
- jumlahAnak : int
- + setter, getter
- + getBonus() : double
- + getGaji() : double
- + toString() : String

(Keterangan)

- Bonus didapatkan oleh pegawai sesuai lama bekerja :
 - Jika lama bekerja 0 – 5 tahun, maka bonus sebesar 5 % dari gaji
 - Jika lama bekerja 5 – 10 tahun, maka bonus sebesar 10% dari gaji
 - Jika lebih dari 10 tahun, maka bonus sebesar 15% dari gaji

Tunjangan ditambah apabila memiliki anak, yaitu \$20 per anak.

- toString() menampilkan atribut induk + tahun masuk, jumlah anak, dan gaji

4. Manager.java

Kelas Manager merupakan kelas turunan dari Pekerja dengan definisi sebagai berikut:

- departemen : String
- + setter, getter

(Keterangan)

- Tunjangan ditambah sebesar 10% dari gaji.
- toString() menampilkan atribut induk + departemen

Dari semua kelas yang telah dibuat, buatlah testcase (mencetak objek / toString()) untuk kasus berikut :

1. Manusia
 - a. Laki-Laki telah menikah.
 - b. Perempuan telah menikah.
 - c. Belum menikah.
2. MahasiswaFilkom (sesuai status Anda)
 - a. Ipk < 3
 - b. Ipk 3 – 3.5
 - c. Ipk 3.5 – 4
3. Pekerja
 - a. Lama bekerja 2 tahun, anak 2
 - b. Lama bekerja 9 tahun
 - c. Lama bekerja 20 tahun, anak 10
4. Manager, lama bekerja 15 tahun dengan gaji \$7500

Source Code

Manusia.java	
1.	public class Manusia {
2.	private String nama;
3.	private boolean jenisKelamin;
4.	private String nik;
5.	private boolean menikah;
6.	
7.	protected Manusia(String nama, boolean jenisKelamin, String nik, boolean menikah)
8.	{
9.	this.nama = nama;
10.	this.jenisKelamin = jenisKelamin;
11.	this.nik = nik;
12.	this.menikah = menikah;
13.	}
14.	
15.	public String getNama() {
16.	return nama;
17.	}
18.	public void setNama(String nama) {
19.	this.nama = nama;
20.	}
21.	public boolean isJenisKelamin() {

22.	return jenisKelamin;
23.	}
24.	public void setJenisKelamin(boolean jenisKelamin) {
25.	this.jenisKelamin = jenisKelamin;
26.	}
27.	public String getNik() {
28.	return nik;
29.	}
30.	public void setNik(String nik) {
31.	this.nik = nik;
32.	}
33.	public boolean isMenikah() {
34.	return menikah;
35.	}
36.	public void setMenikah(boolean menikah) {
37.	this.menikah = menikah;
38.	}
39.	
40.	public double getTunjangan()
41.	{
42.	if (menikah)
43.	{
44.	return jenisKelamin ? 25 : 20;
45.	} else
46.	{
47.	return 15;
48.	}
49.	}
50.	
51.	public double getPendapatan()
52.	{
53.	return getTunjangan();
54.	}
55.	
56.	public String toString()
57.	{
58.	String jk = isJenisKelamin()? "Laki-laki" : "Perempuan";
59.	String output =
60.	>Nama : " + getName() + "\n" +
61.	"NIK : " + getNik() + "\n" +
62.	"Jenis Kelamin : " + jk + "\n" +
63.	"Pendapatan : " + getPendapatan() + "\n";

64.	
65.	return output;
66.	}
67.	}

MahasiswaFilkom.java	
1.	public class MahasiswaFilkom extends Manusia{
2.	private String nim;
3.	private double ipk;
4.	
5.	public MahasiswaFilkom(String nama, boolean jenisKelamin,
6.	String nik, boolean menikah, String nim, double ipk) {
7.	super(nama, jenisKelamin, nik, menikah);
8.	this.nim = nim;
9.	this.ipk = ipk;
10.	}
11.	
12.	public String getNim() {
13.	return nim;
14.	}
15.	public void setNim(String nim) {
16.	this.nim = nim;
17.	}
18.	public double getIpk() {
19.	return ipk;
20.	}
21.	public void setIpk(double ipk) {
22.	this.ipk = ipk;
23.	}
24.	
25.	public String getStatus()
26.	{
27.	String angaktan = getNim().substring(0, 2);
28.	String prodi = "";
29.	switch(getNim().charAt(6))
30.	{
31.	case '2' :
32.	prodi = "Teknik Informatika"; break;
33.	case '3' :
34.	prodi = "Teknik Komputer"; break;

35.	case '4' :
36.	prodi = "Sistem Informasi"; break;
37.	case '6' :
38.	prodi = "Pendidikan Teknologi Informasi"; break;
39.	case '7' :
40.	prodi = "Teknologi Informasi"; break;
41.	}
42.	
43.	return prodi + ", " + angaktan;
44.	}
45.	
46.	public double getBeasiswa()
47.	{
48.	return (ipk >= 3 && ipk < 3.5) ? 50 : (ipk >= 3.5 && ipk <= 4) ? 75 : 0;
49.	}
50.	
51.	public String toString()
52.	{
53.	String jk = isJenisKelamin()? "Laki-laki" : "Perempuan";
54.	String output =
55.	"Nama : " + getName() + "\n" +
56.	"NIK : " + getNik() + "\n" +
57.	"Jenis Kelamin : " + jk + "\n" +
58.	"Pendapatan : " + (getTunjangan()+getBeasiswa()) + "\n" +
59.	"NIM : " + getNim() + "\n" +
60.	"IPK : " + getIpk() + "\n" +
61.	"Status : " + getStatus() + "\n";
62.	
63.	return output;
64.	}
65.	
66.	}

Manager.java	
1.	public class Manager extends Pekerja{
2.	private String departemen;
3.	
4.	public Manager(String nama, boolean jenisKelamin,

	String nik, boolean menikah, double gaji, int tahun, int bulan,
5.	int hari, int jumlahAnak, String departemen)
6.	{
7.	super(nama, jenisKelamin, nik, menikah, gaji, tahun, bulan, hari, jumlahAnak);
8.	this.departemen = departemen;
9.	}
10.	public String getDepartemen() {
11.	return departemen;
12.	}
13.	
14.	public void setDepartemen(String departemen) {
15.	this.departemen = departemen;
16.	}
17.	
18.	public double getTunjangan()
19.	{
20.	return (.1*super.getGaji()+super.getTunjangan());
21.	}
22.	
23.	
24.	public String toString()
25.	{
26.	String output =
27.	super.toString() +
28.	"Departemen : " + getDepartemen() + "\n";
29.	
30.	return output;
31.	}
32.	}

Pekerja.java	
1.	import java.time.LocalDate;
2.	
3.	public class Pekerja extends Manusia{
4.	private double gaji;
5.	private LocalDate tahunMasuk;
6.	private int jumlahAnak;
7.	

8.	public Pekerja(String nama, boolean jenisKelamin,
9.	String nik,
10.	boolean menikah, double gaji, int tahun, int bulan,
11.	int hari, int jumlahAnak) {
12.	super(nama, jenisKelamin, nik, menikah);
13.	this.gaji = gaji;
14.	this.tahunMasuk = LocalDate.of(tahun, bulan,
15.	hari);
16.	this.jumlahAnak = jumlahAnak;
17.	}
18.	
19.	public double getGaji() {
20.	return gaji;
21.	}
22.	public void setGaji(double gaji) {
23.	this.gaji = gaji;
24.	}
25.	public LocalDate getTahunMasuk() {
26.	return tahunMasuk;
27.	}
28.	public void setTahunMasuk(LocalDate tahunMasuk) {
29.	this.tahunMasuk = tahunMasuk;
30.	}
31.	public int getJumlahAnak() {
32.	return jumlahAnak;
33.	}
34.	
35.	public void setJumlahAnak(int jumlahAnak) {
36.	this.jumlahAnak = jumlahAnak;
37.	}
38.	
39.	public double getBonus()
40.	{
41.	return (2023-tahunMasuk.getYear() <= 5) ?
42.	getGaji()/20 :
43.	(2023-tahunMasuk.getYear() <= 10) ? getGaji()/10
44.	: (3*getGaji())/20;
45.	}
	public double getTunjangan()
	{
	return super.getTunjangan()+(20*jumlahAnak);
	}

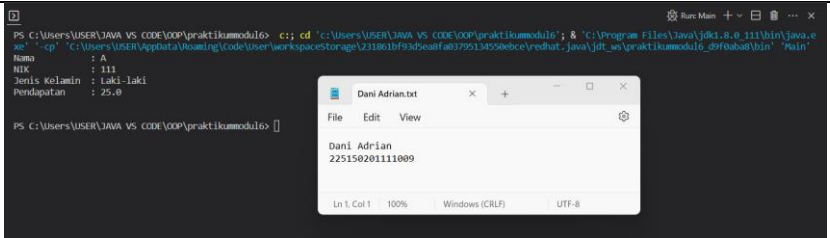
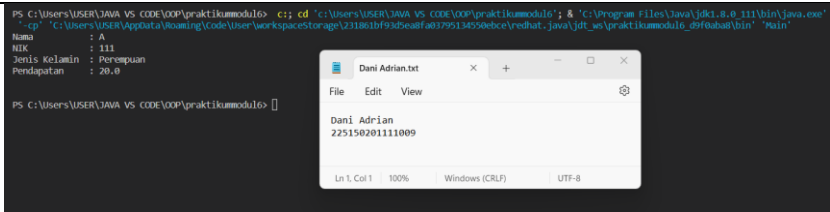
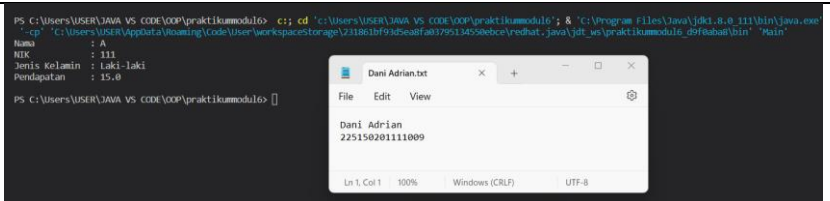
46.	public double getPendapatan()
47.	{
48.	return (getBonus()+getGaji()+getTunjangan());
49.	}
50.	
51.	public String toString()
52.	{
53.	String output =
54.	super.toString() +
55.	"Tahun Masuk : " + getTahunMasuk() + "\n" +
56.	"Jumlah Anak : " + getJumlahAnak() + "\n" +
57.	"Gaji : " + getGaji() + "\n";
58.	
59.	return output;
60.	}
61.	
62.	
63.	}

Main.java	
1.	public class Main {
2.	public static void main(String[] args) {
3.	Manusia a = new Manusia("A", true, "111", true);
4.	System.out.println(a);
5.	System.out.println();
6.	Manusia b = new Manusia("A", false, "111", true);
7.	System.out.println(b);
8.	System.out.println();
9.	Manusia c = new Manusia("A", true, "111", false);
10.	System.out.println(c);
11.	MahasiswaFilkom a = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 2.9);
12.	System.out.println(a);
13.	System.out.println();
14.	MahasiswaFilkom b = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 3.4);
15.	System.out.println(b);
16.	System.out.println();
17.	MahasiswaFilkom c = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 3.8);
18.	System.out.println(c);
19.	Pekerja a = new Pekerja("C", true,

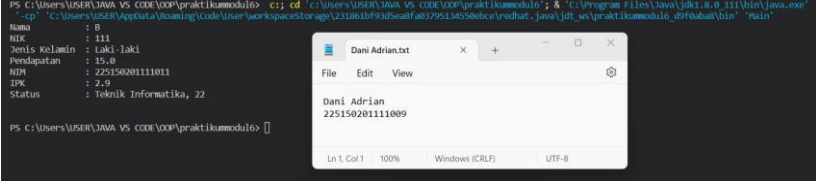
	"225150200111013", false, 100, 2021, 1, 1, 2);
20.	System.out.println(a);
21.	System.out.println();
22.	Pekerja b = new Pekerja("C", true, "225150200111013", false, 100, 2014, 1, 1, 0);
23.	System.out.println(b);
24.	System.out.println();
25.	Pekerja c = new Pekerja("C", true, "225150200111013", false, 100, 2003, 1, 1, 10);
26.	System.out.println(c);
27.	System.out.println();
28.	Manager a = new Manager("D", true, "225150200111013", false, 7500, 2008, 1, 1, 0, "Manager");
29.	System.out.println(a);
30.	
31.	}
32.	}

Output

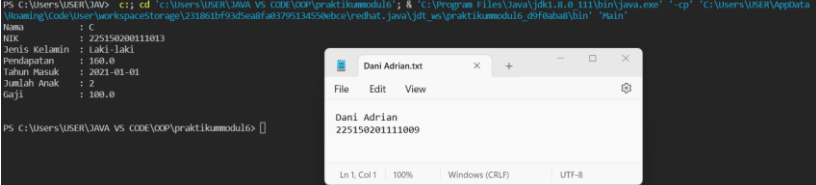
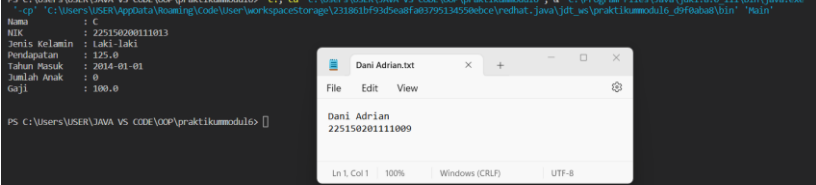
1. Manusia

Manusia a = new Manusia("A", true, "111", true); System.out.println(a);	
Manusia b = new Manusia("A", false, "111", true); System.out.println(b);	
Manusia c = new Manusia("A", true, "111", false); System.out.println(c);	

2. MahasiswaFilkom

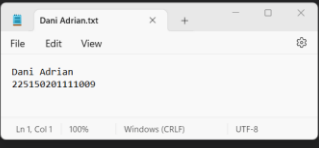
<pre> MahasiswaFilkom a = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 2.9); System.out.println(a); </pre>	
<pre> MahasiswaFilkom b = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 3.4); System.out.println(b); </pre>	
<pre> MahasiswaFilkom c = new MahasiswaFilkom("B", true, "111", false, "225150201111011", 3.8); System.out.println(c); </pre>	

3. Pekerja

<pre> Pekerja a = new Pekerja("C", true, "225150200111013", false, 100, 2021, 1, 1, 2); System.out.println(a); </pre>	
<pre> Pekerja b = new Pekerja("C", true, "225150200111013", false, 100, 2014, 1, 1, 0); System.out.println(b); </pre>	

```
Pekerja c = new
Pekerja("C", true,
"225150200111013",
false, 100, 2003, 1,
1, 10);
System.out.println(c
);
```

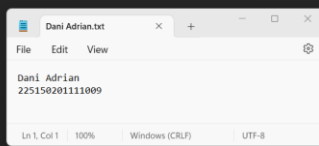
```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6> cd 'C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6' & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' -cp 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\231861bf93d50a7a0379513055ebce\redhat_java\jdk_vs\praktikummodul6_09f8abab\bin' "Main"
Nama      : C
NIK       : 225150200111013
Jenis Kelamin : Laki-laki
Pendapatan : 330.0
Tahun Masuk : 2003-01-01
Jumlah Anak : 10
Gaji      : 100.0
PS C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6>
```



4. Manager

```
Manager a = new
Manager("D", true,
"225150200111013",
false, 7500, 2008,
1, 1, 0, "Manager");
System.out.println(a
);
```

```
PS C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6> cd 'C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6' & 'C:\Program Files\Java\jdk1.8.0_111\bin\java.exe' -cp 'C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\231861bf93d50a7a0379513055ebce\redhat_java\jdk_vs\praktikummodul6_09f8abab\bin' "Main"
Nama      : D
NIK       : 225150200111013
Jenis Kelamin : Laki-laki
Pendapatan : 9300.0
Tahun Masuk : 2008-01-01
Jumlah Anak : 0
Gaji       : 7500.0
Departemen : Manager
PS C:\Users\USER\JAVA VS CODE\OOP\praktikummodul6>
```



Penjelasan

Kelas "Manusia" memiliki beberapa atribut yang digunakan untuk menyimpan informasi tentang objek manusia. Atribut-atribut tersebut adalah nama, jenis kelamin, NIK, dan status pernikahan. Atribut nama merupakan tipe data string, sedangkan atribut jenis kelamin dan status pernikahan merupakan tipe data boolean. Atribut NIK juga merupakan tipe data string. Kelas "Manusia" memiliki constructor dengan beberapa parameter untuk mengatur nilai awal dari atribut-atribut tersebut saat objek dibuat. Kelas "Manusia" juga memiliki getter dan setter untuk masing-masing atribut, yang digunakan untuk mengakses dan mengubah nilai atribut pada objek.

Selain itu, kelas "Manusia" memiliki beberapa method lain, yaitu `getTunjangan()`, `getPendapatan()`, dan `toString()`. Method `getTunjangan()` digunakan untuk menghitung tunjangan berdasarkan status pernikahan dan jenis kelamin dari objek "Manusia". Method `getPendapatan()` menghitung pendapatan objek "Manusia" berdasarkan tunjangan yang telah dihitung menggunakan method `getTunjangan()`. Method `toString()` digunakan untuk menghasilkan representasi string dari objek "Manusia" yang berisi informasi nama, NIK, jenis kelamin, dan jumlah pendapatan.

Kelas "MahasiswaFILKOM" merupakan turunan dari kelas "Manusia". Kelas ini memiliki atribut khusus yaitu `nim` dan `IPK`, dengan tipe data string dan double. Kelas "MahasiswaFILKOM" memiliki constructor dengan beberapa parameter untuk menginisialisasi atribut-atribut pada kelas ini dan juga memanggil constructor kelas induk "Manusia" menggunakan `super()`. Kelas "MahasiswaFILKOM" memiliki method `getStatus()` dan `getBeasiswa()` untuk menghitung status mahasiswa dan jumlah beasiswa yang akan diterima oleh mahasiswa. Kelas ini juga memiliki method `toString()` yang menghasilkan representasi string dari objek "MahasiswaFILKOM" yang berisi informasi seperti nama, NIK, jenis kelamin, pendapatan (tunjangan dan beasiswa), `nim`, `IPK`, dan status mahasiswa.

Kelas "Pekerja" juga merupakan turunan dari kelas "Manusia". Kelas ini memiliki atribut khusus yaitu gaji dan tahunMasuk, dengan tipe data double dan LocalDate. Kelas "Pekerja" memiliki constructor dengan beberapa parameter untuk menginisialisasi atribut-atribut pada kelas ini dan juga memanggil constructor kelas induk "Manusia" menggunakan super(). Kelas "Pekerja" memiliki method getLamaKerja() dan getBonus() untuk menghitung lama kerja pekerja dan bonus yang akan diterima. Kelas ini juga memiliki method toString() yang menghasilkan representasi string dari objek "Pekerja" yang berisi informasi seperti nama, NIK, jenis kelamin, pendapatan (gaji dan bonus), dan lama kerja.