

TUGAS PRAKTIKUM PBO

Nama: Mohammad Dwi Anshor Al Ghifari

Nim: 235150607111008

Kelas: PTI A

Asisten: Kak Devan F

Kak Rivaldo Sidique

Data dan Analisis hasil percobaan Pertanyaan

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

Jawab: Tidak terjadi kesalahan pada kode program berikut

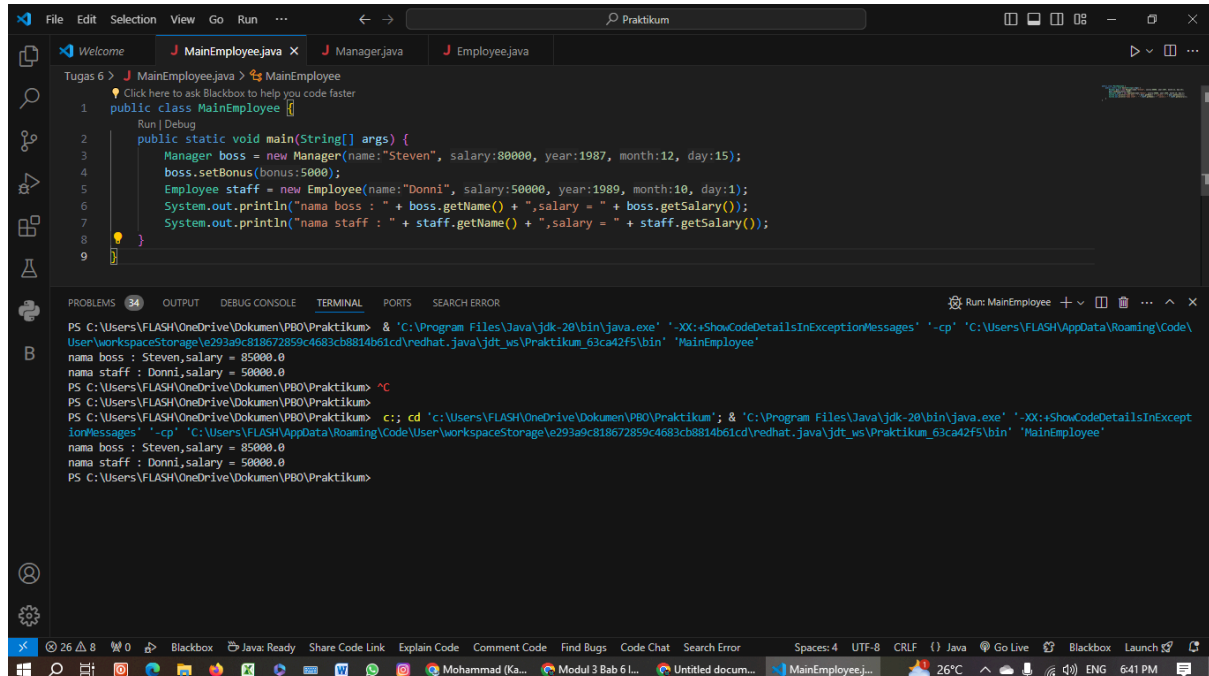
2. Bagaimana cara konstruktor pada subclass memanggil konstruktor di superclass nya?

Apakah hal itu perlu dilakukan? Sertakan alasan anda !

Jawab: konstruktor pada subclass dapat memanggil konstruktor di superclass-nya dengan menggunakan kata kunci `super()`. Hal ini harus dilakukan di baris pertama konstruktor subclass. Misalnya, di konstruktor Manajer, Anda dapat melihat bahwa subclass memanggil konstruktor Karyawan superkelas menggunakan `super(nama,gaji, tahun, bulan, hari)`. Hal ini diperlukan agar objek karyawan diinisialisasi dengan benar ketika objek manajer dibuat.

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

Jawab: Tidak ada perubahan yang terjadi ketika kode dijalankan

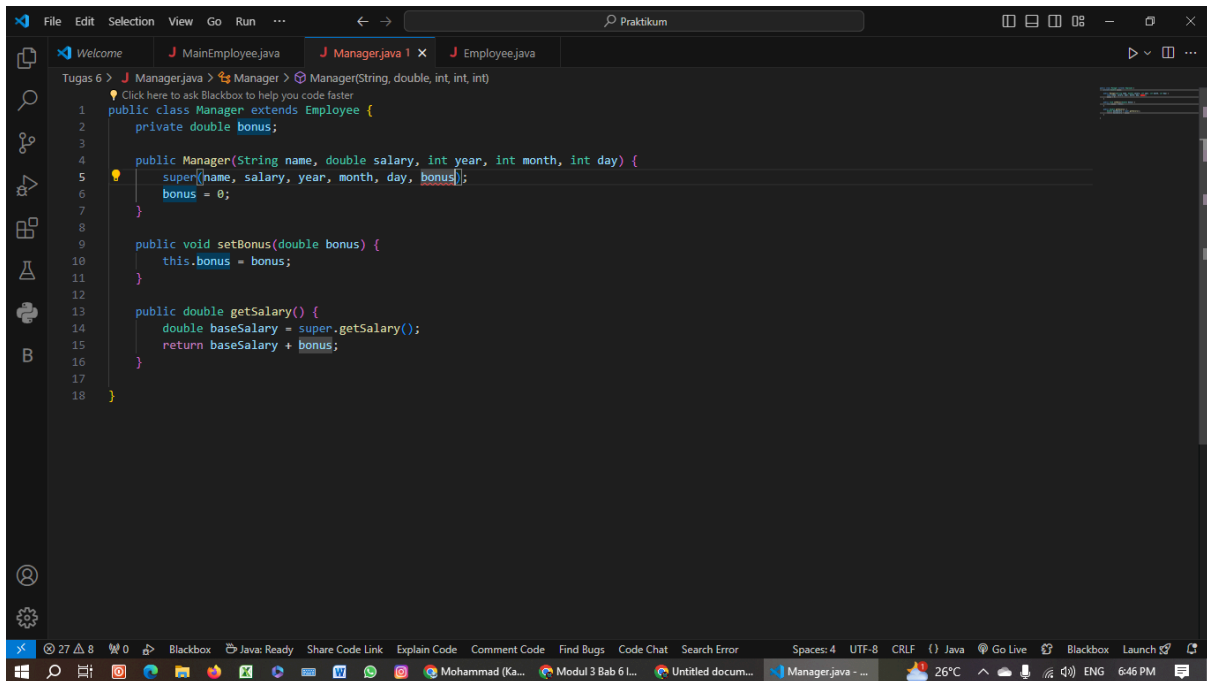


The screenshot shows an IDE with a Java file named `MainEmployee.java`. The code defines a `MainEmployee` class with a `main` method. Inside `main`, it creates a `Manager` object named `boss` and an `Employee` object named `staff`. It then prints their details. The output window shows the following text:

```
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum> & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat_java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 85000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum> cd 'C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat_java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 85000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum>
```

4. Pada Class Manager baris ke 5, setelah variable day tambahkan variable bonus! Amati apa yang terjadi dan mengapa demikian?

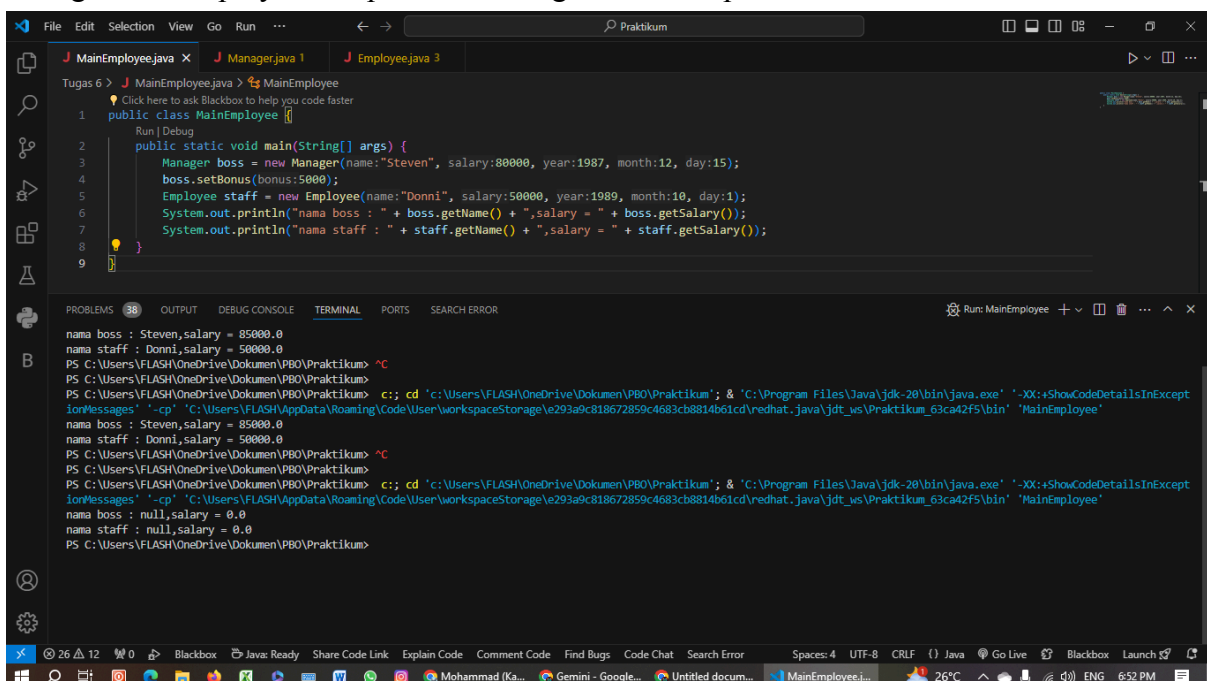
Jawab: Yang terjadi adalah error, karena variabel bonus tidak ada pada superclass Employee sehingga menghasilkan error.



```
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, bonus);
6         bonus = 0;
7     }
8
9     public void setBonus(double bonus) {
10         this.bonus = bonus;
11     }
12
13     public double getSalary() {
14         double baseSalary = super.getSalary();
15         return baseSalary + bonus;
16     }
17 }
18 }
```

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

Jawab: This pada java digunakan untuk mengacu pada contoh objek saat ini, memanggil konstruktor lain dari dalam konstruktor, melewati instance objek saat ini sebagai argumen, mengakses anggota kelas terlampir dari kelas, kelas anonim. Jika keyword this pada class manager dan Employee dihapus maka menghasilkan output 0.



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

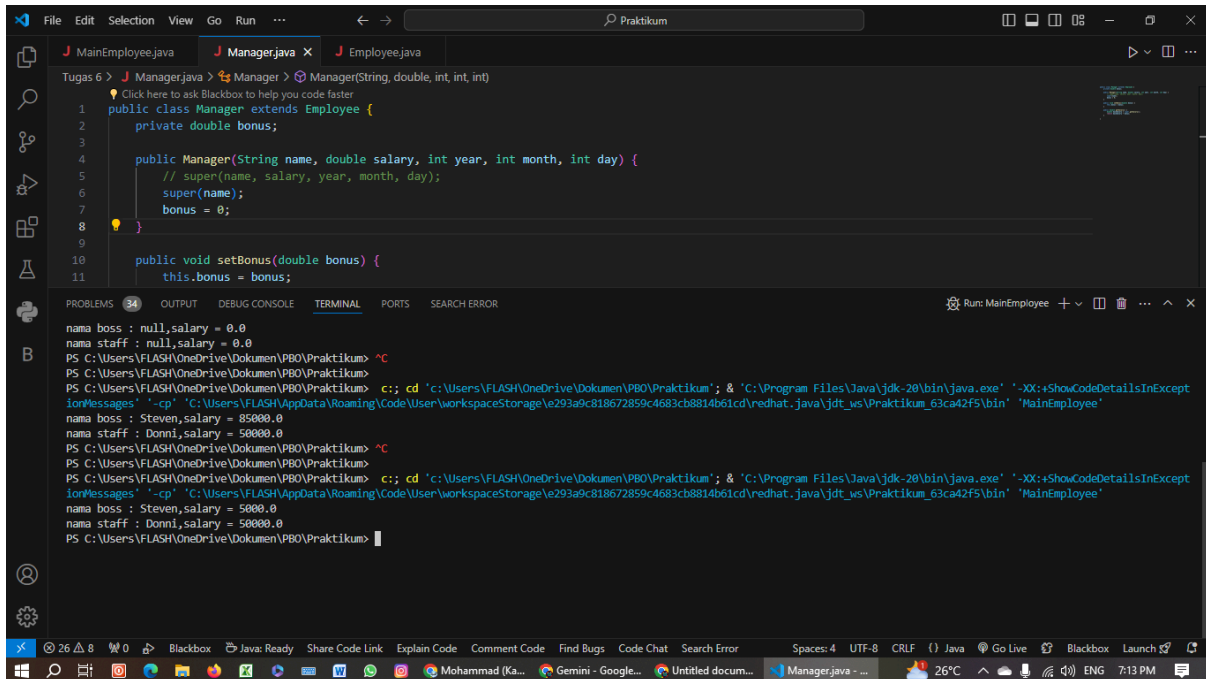
nama boss : Steven,salary = 85000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum> cd 'C:\Users\FLASH\OneDrive\Dokumen\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\293a9c818672859c4683cb8814b61cd\redhat.java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'

6. Tambahkan constructor pada class Employee dengan parameter Bertipe data string bernama name yang nantinya bila constructor ini akan dipanggil akan menginisialisasi

variable name! Amati perubahannya pada class anak dan jelaskan! Benahi bila terjadi kesalahan!

Jawab:

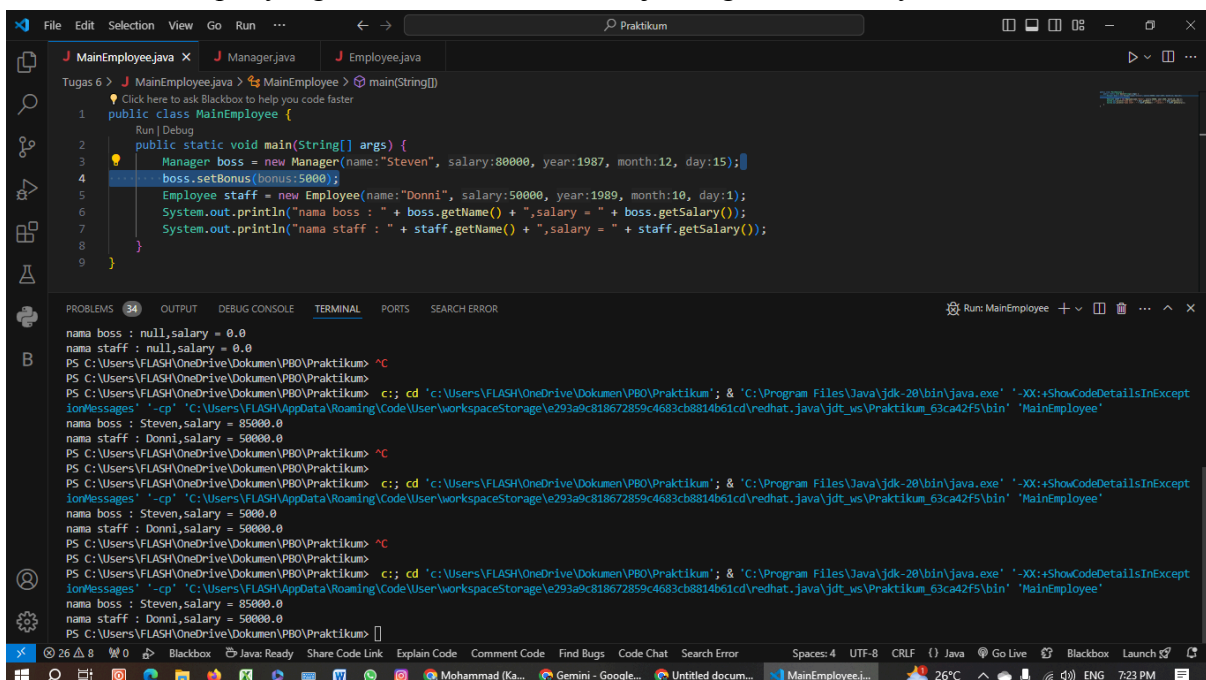
1. Jika tidak membuat parameter baru tetapi membuat comment pada baris ke-5 dan kemudian menambahkan parameter `super(name)`. maka constructor name yang dipanggil saja maka constructor lain seperti `salary`, `year`, `month`, `day` tidak akan menampilkan hasil pada output tersebut.



```
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         super(name);
7         bonus = 0;
8     }
9
10    public void setBonus(double bonus) {
11        this.bonus = bonus;
12    }
13 }
```

```
nama boss : null,salary = 0.0
nama staff : null,salary = 0.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> c:: cd 'c:\Users\FLASH\OneDrive\Documents\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' ^-XX:+ShowCodeDetailsInExcept
ionMessages' ^-cp 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat.java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 85000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> c:: cd 'c:\Users\FLASH\OneDrive\Documents\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' ^-XX:+ShowCodeDetailsInExcept
ionMessages' ^-cp 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat.java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 50000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
```

2. Jika kita membuat parameter yang baru tanpa membuat comment pada baris ke-5 maka output yang dihasilkan akan sama saja dengan sebelumnya.



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

```
nama boss : null,salary = 0.0
nama staff : null,salary = 0.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> c:: cd 'c:\Users\FLASH\OneDrive\Documents\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' ^-XX:+ShowCodeDetailsInExcept
ionMessages' ^-cp 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat.java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 85000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> c:: cd 'c:\Users\FLASH\OneDrive\Documents\PBO\Praktikum'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' ^-XX:+ShowCodeDetailsInExcept
ionMessages' ^-cp 'C:\Users\FLASH\AppData\Roaming\Code\User\workspaceStorage\e293a9c818672859c4683cb8814b61cd\redhat.java\jdt_ws\Praktikum_63ca42f5\bin' 'MainEmployee'
nama boss : Steven,salary = 50000.0
nama staff : Donni,salary = 50000.0
PS C:\Users\FLASH\OneDrive\Documents\PBO\Praktikum> ^C
```

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

Jawab: Karena penggunaan modifier protected pada superclass akan memungkinkan kontrol akses yang akan terkendali, dan juga mendukung inheritance dan polymorphism, menjaga enkapsulasi modular dan juga meningkatkan keamanan kode. Sehingga menjadikan protected sebagai alat yang penting dalam desain kelas yang efektif dan juga berorientasi objek. Jika superclass modifier menggunakan public maka method akan dapat diakses dari berbagai class bukan hanya dari subclassnya saja, tetapi jika superclass menggunakan modifier private maka method tidak dapat diakses oleh semua class maka method tidak dapat mewarisi kepada subclassnya.

8. Ubahlah acces modifier method pada kelas employee menjadi : a. Private b. Protected Amati perubahan apa yang terjadi? Jelaskan jawaban anda dengan detail!

Jawab:

a. Private

Jika kita mengubah semua method menjadi private maka akan terjadi ppada main dan subclassnya seperti penjelasan saya di nomor 7

b. Protected

Jika kita menggunakan modifier protected maka codingan tersebut tidak ada error seperti kita menggunakan modifier public pada sebelumnya dan hasil outputnya pun akan tetap sama seperti 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)

- o Tunjangan untuk yang telah menikah adalah apabila laki-laki akan mendapat \$25 sedangkan perempuan mendapat \$20.
- o Tunjangan untuk yang belum menikah adalah \$15 .
- o 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)

- o Beasiswa untuk ipk 3.0 – 3.5 mendapat \$50 dan untuk 3.5 – 4 mendapat \$75
- o Status untuk mendapatkan angkatan dan prodi (menurut kaidah FILKOM UB) 1651506XXXXXXX

- o Digit ke 1-2 adalah angkatan
- o 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

- o 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)

- o Bonus didapatkan oleh pegawai sesuai lama bekerja :
 - o Jika lama bekerja 0 – 5 tahun, maka bonus sebesar 5% dari gaji
 - o Jika lama bekerja 5 – 10 tahun, maka bonus sebesar 10% dari gaji
 - o Jika lebih dari 10 tahun, maka bonus sebesar 15% dari gaji
- o Tunjangan ditambah apabila memiliki anak, yaitu \$20 per anak.
- o 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)

- o Tunjangan ditambah sebesar 10% dari gaji.
- o 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. Mahasiswa Filkom (sesuai status Anda)
 - a. $\text{IpK} < 3$
 - b. $\text{IpK} 3 - 3.5$
 - c. $\text{IpK} 3.5 - 4$
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