

PBO JOBSHEET MINGGU 9

Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Objek

Dosen : Odhitya Desta Triswidrananta, S.Pd, M.Pd



Nama : Raden Dimas Erlangga

Kelas : D-III Manajemen Informatika 2E

Nim : 2031710121

PROGRAM STUDI D-III MANAJEMEN INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2021

Jobsheet 9

1. Percobaan 1

1.1 class karyawan

```
1 package percobaan1;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class Karyawan {
6     private String nama;
7     private String nip;
8     private String golongan;
9     private double gaji;
10
11     public void setNama(String nama) {
12         this.nama = nama;
13     }
14     public void setNip(String nip) {
15         this.nip = nip;
16     }
17     public void setGolongan(String golongan) {
18         this.golongan = golongan;
19
20         switch(golongan.charAt(0)) {
21             case '1': this.gaji = 5000000;
22                 break;
23             case '2': this.gaji = 3000000;
24                 break;
25             case '3': this.gaji = 2000000;
26                 break;
27             case '4': this.gaji = 1000000;
28                 break;
29             case '5': this.gaji = 750000;
30                 break;
31         }
32     }
33     public void setGaji(double gaji) {
34         this.gaji = gaji;
35     }
36     public String getNama() {
37         return nama;
38     }
39     public String getNip() {
40         return nip;
41     }
42     public String getGolongan() {
43         return golongan;
44     }
45     public double getGaji() {
46         return gaji;
47     }
48 }
49
```

1.2 class Manajer

```
1 package percobaan1;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class Manager extends Karyawan {
6     private double tunjangan;
7     private String bagian;
8     private Staff st[];
9
10    public void setTunjangan(double tunjangan){
11        this.tunjangan=tunjangan;
12    }
13    public double getTunjangan(){
14        return tunjangan;
15    }
16    public void setBagian(String bagian){
17        this.bagian=bagian;
18    }
19    public String getBagian(){
20        return bagian;
21    }
22    public void setStaff(Staff st[]){
23        this.st=st;
24    }
25    public void viewStaff(){
26        int i;
27        System.out.println("-----");
28        for(i=0; i<st.length; i++){
29            st[i].lihatInfo();
30        }
31        System.out.println("-----");
32    }
33    public void lihatInfo(){
34        System.out.println("Manager :"+this.getBagian());
35        System.out.println("NIP :"+this.getNip());
36        System.out.println("Nama :"+this.getNama());
37        System.out.println("Golongan :"+this.getGolongan());
38        System.out.println("Tunjangan :"+this.getTunjangan());
39        System.out.printf("Gaji :%.0f\n",this.getGaji());
40        System.out.printf("Bagian :",this.getBagian());
41        this.viewStaff();
42    }
43    public double getGaji(){
44        return super.getGaji()+tunjangan;
45    }
46 }
```

1.3 class Staff

```
1 package percobaan1;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class Staff extends Karyawan{
6     private int lembur;
7     private double gajiLembur;
8
9     public void setLembur(int lembur){
10         this.lembur=lembur;
11     }
12     public int getLembur(){
13         return lembur;
14     }
15     public void setGajiLembur(double gajiLembur){
16         this.gajiLembur=gajiLembur;
17     }
18     public double getGajiLembur(){
19         return gajiLembur;
20     }
21     //overloading
22     public double getGaji(int lembur, double gajiLembur){
23         return super.getGaji()+lembur*gajiLembur;
24     }
25     //overriding
26     public double getGaji(){
27         return super.getGaji()+lembur*gajiLembur;
28     }
29     public void lihatInfo(){
30         System.out.println("NIP :"+this.getNip());
31         System.out.println("Nama :"+this.getNama());
32         System.out.println("Golongan :"+this.getGolongan());
33         System.out.println("Jml Lembur :"+this.getGajiLembur());
34         System.out.printf("Gaji Lembur :%.0f\n",this.getGaji());
35         System.out.printf("Gaji :%.0f\n",this.getGaji());
36     }
37 }
```

1.4 class Utama

```
1  package percobaan1;
2  /**
3   * Raden Dimas Erlangga
4   */
5  public class Utama {
6
7      public static void main(String[] args) {
8          System.out.println("Program Testing Class Manager & Staff");
9          System.out.println();
10         Manager man[]=new Manager[2];
11         Staff staff1[]=new Staff[2];
12         Staff staff2[]=new Staff[3];
13
14         //Pembuatan Manager
15
16         man[0]=new Manager();
17         man[0].setNama("Tedjo");
18         man[0].setNip("101");
19         man[0].setGolongan("1");
20         man[0].setTunjangan(5000000);
21         man[0].setBagian("Administrasi");
22
23         man[1]=new Manager();
24         man[1].setNama("Atika");
25         man[1].setNip("102");
26         man[1].setGolongan("1");
27         man[1].setTunjangan(2500000);
28         man[1].setBagian("Pemasaran");
29
```

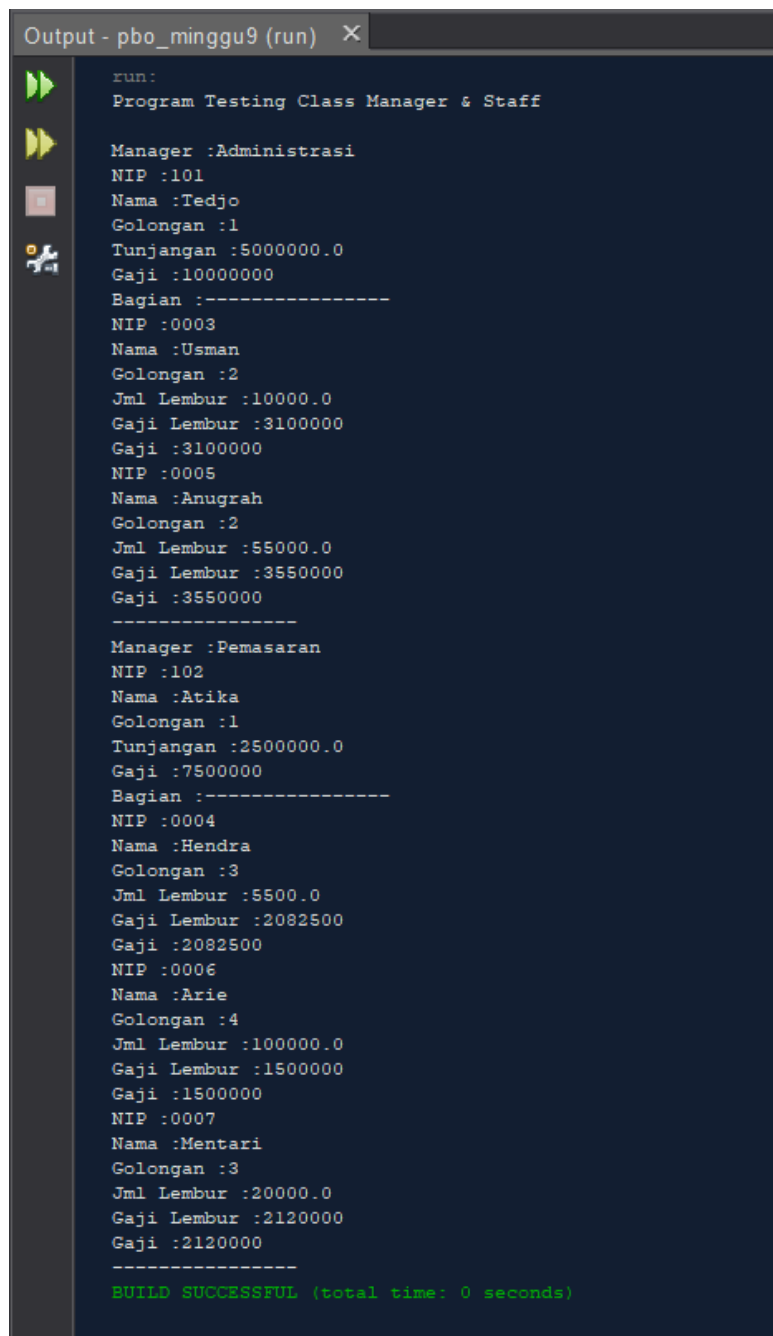
```
30         staff1[0]=new Staff();
31         staff1[0].setNama("Usman");
32         staff1[0].setNip("0003");
33         staff1[0].setGolongan("2");
34         staff1[0].setLembur(10);
35         staff1[0].setGajiLembur(10000);
36
37         staff1[1]=new Staff();
38         staff1[1].setNama("Anugrah");
39         staff1[1].setNip("0005");
40         staff1[1].setGolongan("2");
41         staff1[1].setLembur(10);
42         staff1[1].setGajiLembur(55000);
43         man[0].setStaff(staff1);
44
45         staff2[0]=new Staff();
46         staff2[0].setNama("Hendra");
47         staff2[0].setNip("0004");
48         staff2[0].setGolongan("3");
49         staff2[0].setLembur(15);
50         staff2[0].setGajiLembur(5500);
51
52         staff2[1]=new Staff();
53         staff2[1].setNama("Arie");
54         staff2[1].setNip("0006");
55         staff2[1].setGolongan("4");
56         staff2[1].setLembur(5);
57         staff2[1].setGajiLembur(100000);
58
```

```

59     staff2[2]=new Staff();
60     staff2[2].setNama("Mentari");
61     staff2[2].setNip("0007");
62     staff2[2].setGolongan("3");
63     staff2[2].setLembur(6);
64     staff2[2].setGajiLembur(20000);
65     man[1].setStaff(staff2);
66
67     //cetak informasi dari manager + staffnya
68     man[0].lihatInfo();
69     man[1].lihatInfo();
70 }
71 }

```

1.5 Hasil Running :



```

Output - pbo_minggu9 (run) X
run:
Program Testing Class Manager & Staff

Manager :Administrasi
NIP :101
Nama :Tedjo
Golongan :1
Tunjangan :5000000.0
Gaji :10000000
Bagian :-----
NIP :0003
Nama :Usman
Golongan :2
Jml Lembur :10000.0
Gaji Lembur :3100000
Gaji :3100000
NIP :0005
Nama :Anugrah
Golongan :2
Jml Lembur :55000.0
Gaji Lembur :3550000
Gaji :3550000
-----
Manager :Pemasaran
NIP :102
Nama :Atika
Golongan :1
Tunjangan :2500000.0
Gaji :7500000
Bagian :-----
NIP :0004
Nama :Hendra
Golongan :3
Jml Lembur :5500.0
Gaji Lembur :2082500
Gaji :2082500
NIP :0006
Nama :Arie
Golongan :4
Jml Lembur :100000.0
Gaji Lembur :1500000
Gaji :1500000
NIP :0007
Nama :Mentari
Golongan :3
Jml Lembur :20000.0
Gaji Lembur :2120000
Gaji :2120000
-----
BUILD SUCCESSFUL (total time: 0 seconds)

```

2. Latihan

```
1  package latihan;
2  /**
3   * Raden Dimas Erlangga
4   */
5  public class PerkalianKu {
6      void perkalian(int a, int b){
7          System.out.println(a * b);
8      }
9
10     void perkalian(int a, int b, int c){
11         System.out.println(a*b*c);
12     }
13
14     public static void main(String[] args) {
15         PerkalianKu objek = new PerkalianKu();
16
17         objek.perkalian(25, 43);
18         objek.perkalian(34, 23, 56);
19     }
20 }
21
```

2.1 Dari source coding diatas terletak dimanakah overloading?

Jawaban : overloading terletak pada method perkalian

```
void perkalian(int a, int b){
    System.out.println(a * b);
}

void perkalian(int a, int b, int c){
    System.out.println(a*b*c);
}
```

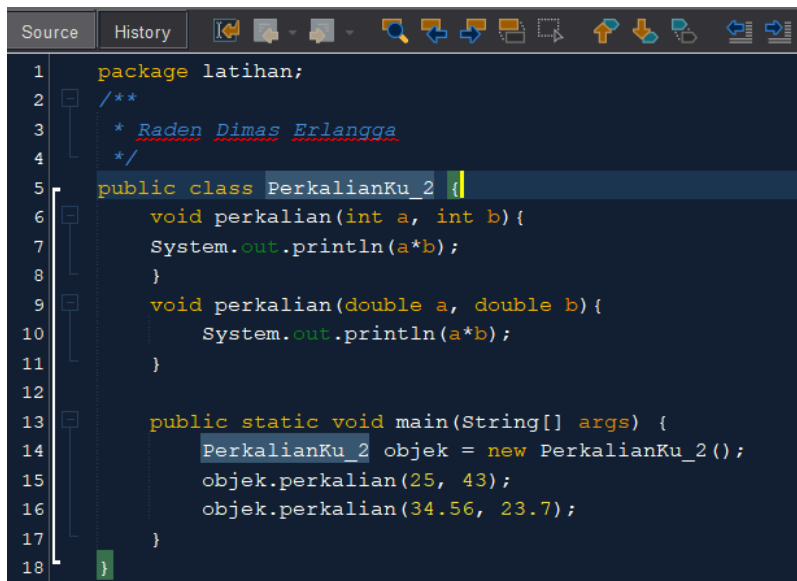
2.2 Jika terdapat overloading ada berapa jumlah parameter yang berbeda?

Jawaban ada 2 jumlah parameter yang berbeda 2, parameter method perkalian pertama menggunakan dua parameter, lalu method perkalian kedua menggunakan 3 parameter.

```
void perkalian(int a, int b){
    System.out.println(a * b);
}

void perkalian(int a, int b, int c){
    System.out.println(a*b*c);
}
```

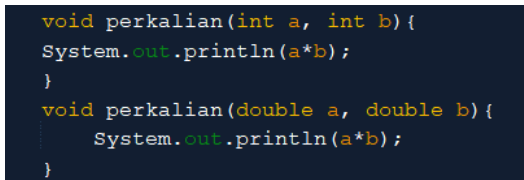
Soal (2.3 & 2.4) :



```
1 package latihan;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class PerkalianKu_2 {
6     void perkalian(int a, int b) {
7         System.out.println(a*b);
8     }
9     void perkalian(double a, double b) {
10         System.out.println(a*b);
11     }
12
13     public static void main(String[] args) {
14         PerkalianKu_2 objek = new PerkalianKu_2();
15         objek.perkalian(25, 43);
16         objek.perkalian(34.56, 23.7);
17     }
18 }
```

2.3 Dari source coding diatas terletak dimanakah overloading?

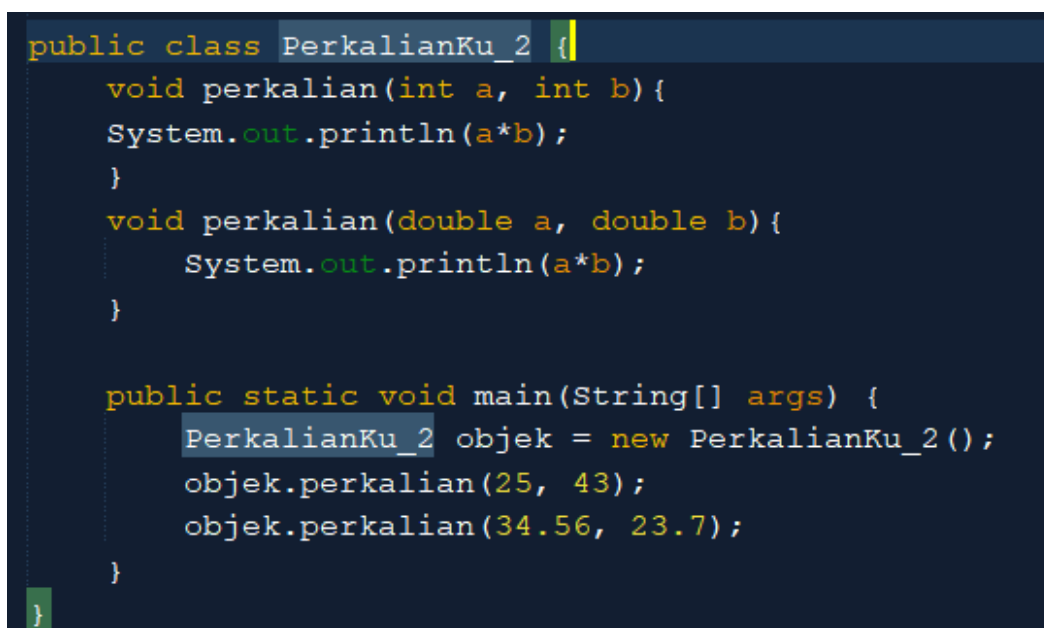
Jawaban : masih sama dengan soal sebelum nya, yaitu terletak pada method perkalian



```
void perkalian(int a, int b) {
    System.out.println(a*b);
}
void perkalian(double a, double b) {
    System.out.println(a*b);
}
```

2.4 Jika terdapat overloading ada berapa tipe parameter yang berbeda?

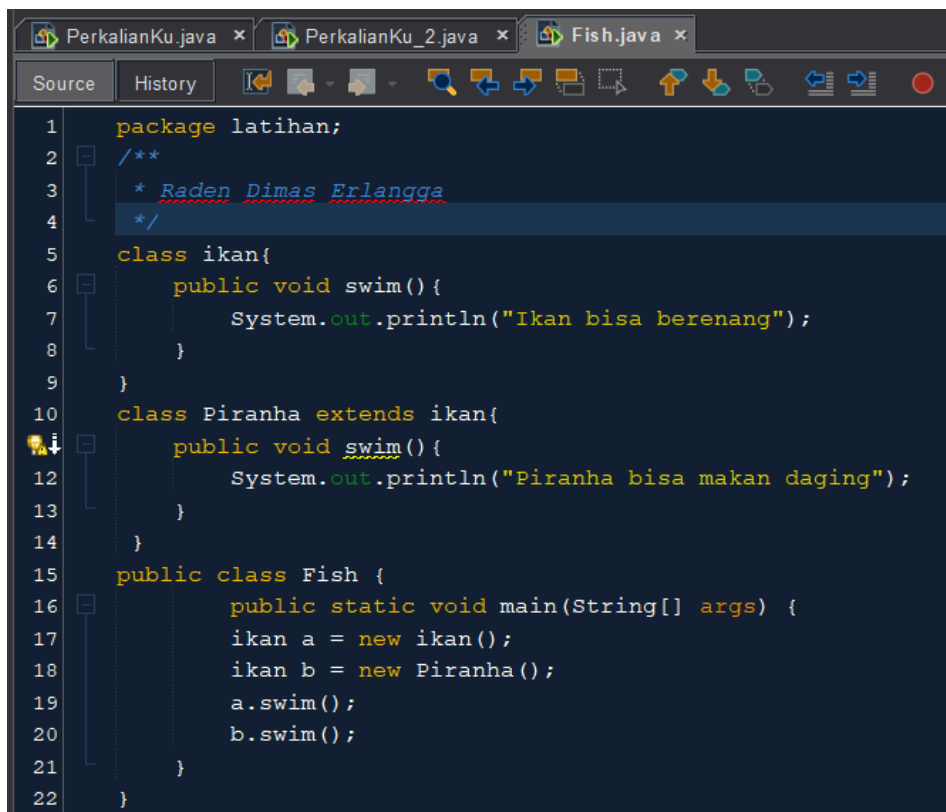
Jawaban : ada dua tipe parameter yang berbeda, parameter method pertama menggunakan int dan parameter method kedua menggunakan double.



```
public class PerkalianKu_2 {
    void perkalian(int a, int b) {
        System.out.println(a*b);
    }
    void perkalian(double a, double b) {
        System.out.println(a*b);
    }

    public static void main(String[] args) {
        PerkalianKu_2 objek = new PerkalianKu_2();
        objek.perkalian(25, 43);
        objek.perkalian(34.56, 23.7);
    }
}
```

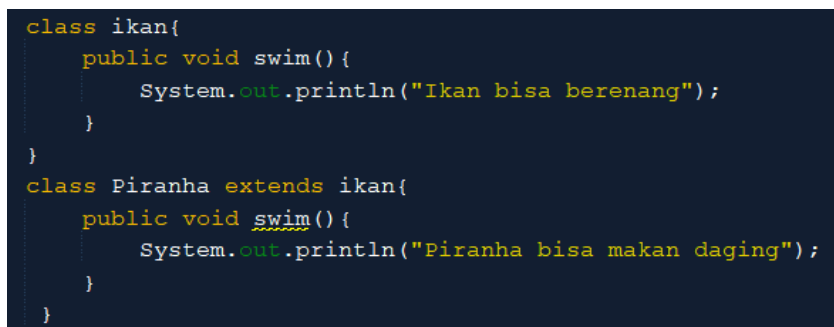

Soal (2.5 & 2.6) :



```
1 package latihan;
2 /**
3  * Raden Dimas Erlangga
4  */
5 class ikan{
6     public void swim(){
7         System.out.println("Ikan bisa berenang");
8     }
9 }
10 class Piranha extends ikan{
11     public void swim(){
12         System.out.println("Piranha bisa makan daging");
13     }
14 }
15 public class Fish {
16     public static void main(String[] args) {
17         ikan a = new ikan();
18         ikan b = new Piranha();
19         a.swim();
20         b.swim();
21     }
22 }
```

2.5 Dari source coding diatas terletak dimanakah overriding?

Jawaban : overriding terletak pada method swim



```
class ikan{
    public void swim(){
        System.out.println("Ikan bisa berenang");
    }
}
class Piranha extends ikan{
    public void swim(){
        System.out.println("Piranha bisa makan daging");
    }
}
```

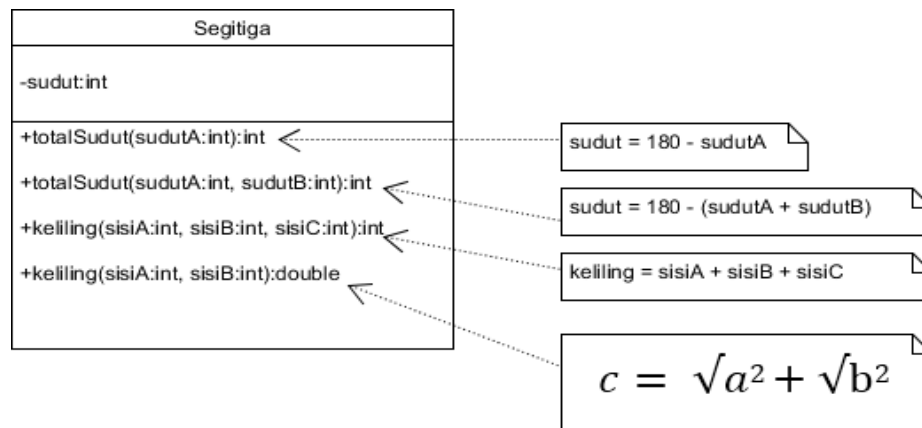
2.6 Jelaskanlah apabila sourcoding diatas jika terdapat overriding?

Jawaban : alasan method swim adalah overriding, dikarenakan method tersebut memiliki nama yang sama namun output yang dihasilkan berbeda, pada method swim yang pertama ditujukan pada Ikan, dan method yang kedua ditujukan pada piranha.

3. Tugas

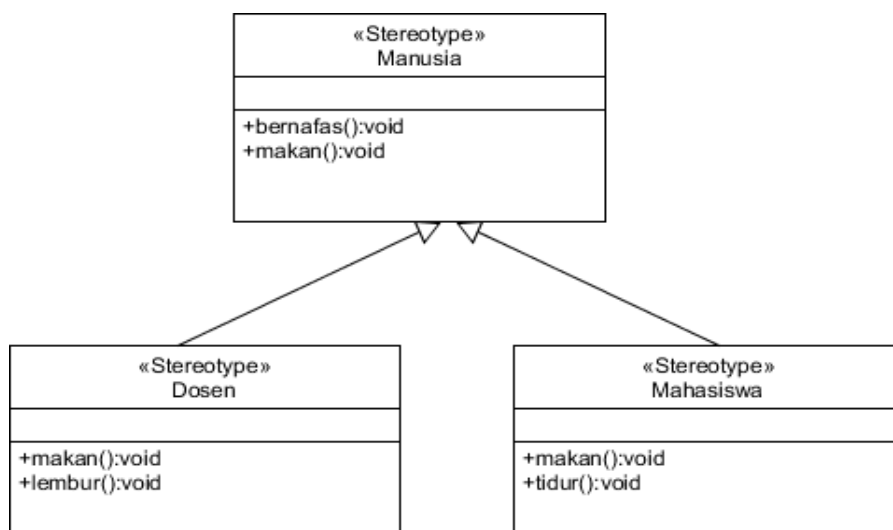
3.1 Overloading

Implementasikan konsep overloading pada class diagram dibawah ini :



3.2 Overriding

Implementasikan class diagram dibawah ini dengan menggunakan teknik dynamic method dispatch :



Jawaban :

3.1 class Segitiga

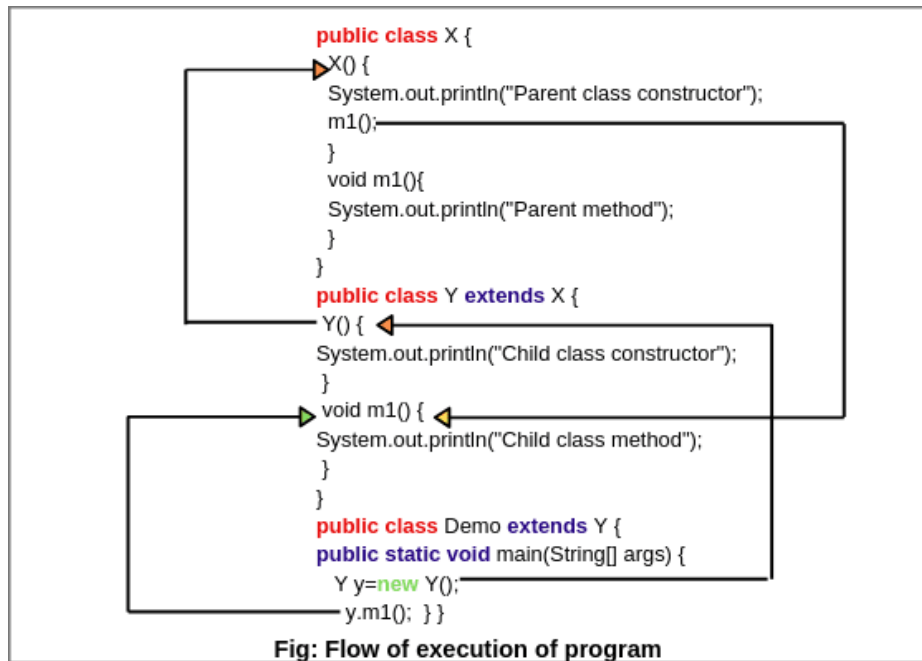
```
1 package Tugas;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class Segitiga {
6
7     private int sudut;
8
9     public int totalSudut(int suduta) {
10         return sudut = 180 - suduta;
11     }
12     public int totalSudut(int suduta, int sudutB) {
13         return sudut = 180 - (suduta - sudutB);
14     }
15     public int keliling(int sisiA, int sisiB, int sisiC) {
16         int keliling = sisiA + sisiB + sisiC;
17         return keliling;
18     }
19
20     public double keliling(int sisiA, int sisiB) {
21         //math.sqrt adalah fungsi yang digunakan
22         //untuk mencari hasil dari akar kuadrat pada sebuah nilai
23         double keliling = Math.sqrt(Math.pow(sisiA, 2) + Math.pow(sisiB, 2));
24         return keliling;
25     }
26
27     //main class Segitiga
28     public static void main(String[] args) {
29         Segitiga s = new Segitiga();
30         System.out.println("total Sudut pertama: " + s.totalSudut(60));
31         System.out.println("total Sudut kedua: " + s.totalSudut(60, 60));
32         System.out.println("Keliling pertama: " + s.keliling(3, 5, 7));
33         System.out.printf("Keliling kedua: %.2f\n", s.keliling(3, 9));
34     }
35 }
```

Hasil Running :

```
Output - pbo_minggu9 (run) X
run:
total Sudut pertama: 120
total Sudut kedua: 180
Keliling pertama: 15
Keliling kedua: 9.49
BUILD SUCCESSFUL (total time: 0 seconds)
```

3.2 class Manusia

Konsep Dynamic Method Dispatch :



Penerapan :

```
1 package Tugas;
2 /**
3  * Raden Dimas Erlangga
4  */
5 public class Manusia {
6
7     public void bernafas() {
8         System.out.println("Manusia perlu bernafas");
9     }
10    public void makan() {
11        System.out.println("Manusia perlu makan");
12    }
13 }
14 class Dosen {
15     public void makan() {
16         System.out.println("Dosen juga manusia, maka Dosen perlu makan");
17     }
18     public void lembur() {
19         System.out.println("Sebagai manusia harus bekerja, dosen juga bekerja lembur");
20     }
21 }
22 class Mahasiswa {
23     public void makan() {
24         System.out.println("Mahasiswa juga manusia, maka Mahasiswa perlu makan");
25     }
26     public void tidur() {
27         System.out.println("Mahasiswa juga manusia, sebagai manusia pasti nya perlu tidur yang cukup");
28     }
29 }
```