

PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

JOBSHEET 6

INHERITANCE (PEWARISAN)



NAMA : SHEVA ANANDA ARIFIANSYAH

KELAS : SIB2B

NIM: 2241760114

PROGRAM STUDI D-IV SISTEM INFORMASI
BISNIS JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

2024

3.PERCOBAAN 1 (extends)

A. TAHAPAN PERCOBAAN

CLAASS A

```
1 public class ClassA {
2     public int x;
3     public int y;
4
5     public void getNilai (){
6         System.out.println("nilai x:" + x);
7         System.out.println("nilai y:" + y);
8     }
9 }
10
```

CLASS B

```
1 public class ClassB {
2     public int z;
3
4     public void getNilaiZ(){
5         System.out.println("nilai z:"+ z);
6     }
7     public void getJumlah(){
8         System.out.println("jumlah:" + (x+y+z));
9     }
10 }
11
```

PERCOBAAN 1

```
1 public class Percobaan1 {
2     public static void main(String[] args) {
3         ClassB hitung = new ClassB();
4         hitung.x=20;
5         hitung.y=30;
6         hitung.z=5;
7         hitung.getNilai();
8         hitung.getNilaiZ();
9         hitung.getJumlah();
10    }
11 }
12
```

B. PERTANYAAN

1. Pada percobaan 1 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!

```
1 public class ClassB extends ClassA {
2     public int z;
3
4     public void getNilaiZ(){
5         System.out.println("nilai z:"+ z);
6     }
7     public void getJumlah(){
8         System.out.println("jumlah:" + (x+y+z));
9     }
10 }
```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!

Penyebab error pada program di percobaan 1 adalah karena ClassB tidak memiliki akses langsung ke variabel x dan y, yang diperlukan dalam metode getJumlah(). Karena itu, ketika mencoba mengakses x dan y dalam getJumlah(), compiler menghasilkan error karena variabel tersebut tidak dideklarasikan dalam ruang lingkup ClassB. Dengan membuat ClassB menjadi subclass dari ClassA, kita membuat variabel x dan y menjadi bagian dari ClassB juga, sehingga dapat diakses di dalamnya.

4.PERCOBAAN 2 (Hak Akses)

A. TAHAPAN PERCOBAAN

Class A

```
3 public class ClassA {
4     private int x;
5     private int y;
6
7     public void setX(int x) {
8         this.x = x;
9     }
10    public void setY(int y){
11        this.y = y;
12    }
13    public void getNilai(){
14        System.out.println("nilai x:"+ x );
15        System.out.println("nilai y:"+ y);
16    }
17
18 }
```

Class B

```
3 public class ClassB {
4     private int z;
5
6     public void setZ(int z){
7         this.z = z;
8     }
9     public void getNilaiZ(){
10        System.out.println("nilai Z:"+ z);
11    }
12    public void getJumlah(){
13        System.out.println("jumlah:"+ (x+y+z));
14    }
15 }
16
```

Class percobaan 2

```
3 public class Percobaan2 {
4
5     public static void main(String[] args) {
6         ClassB hitung = new ClassB();
7         hitung.setX(20);
8         hitung.setY(30);
9         hitung.setZ(z:5);
10        hitung.getNilai();
11        hitung.getNilaiZ();
12        hitung.getJumlah();
13    }
14 }
15
```

Hasil

```
nilai Z:5
jumlah:55
```

B. PERTANYAAN

1. Pada percobaan 2 diatas program yang dijalankan terjadi error, kemudian perbaiki sehingga program tersebut bisa dijalankan dan tidak error!

```
2
3 public class ClassB {
4     private int z;
5     private int x;
6     private int y;
7
8     public void setZ(int z){
9         this.z = z;
10    }
11    public void setX(int x){
12        this.x = x;
13    }
14
15    public void setY(int y){
16        this.y = y;
17    }
18
19    public void getNilaiZ(){
20        System.out.println("nilai Z:"+ z);
21    }
22    public void getJumlah(){
23        System.out.println("jumlah:"+ (x+y+z));
24    }
25 }
```

2. Jelaskan apa penyebab program pada percobaan 1 ketika dijalankan terdapat error!

Penyebab error pada program di percobaan 2 adalah karena variabel x dan y memiliki hak akses private, yang berarti tidak dapat diakses langsung dari luar kelas. Oleh karena itu, ketika mencoba mengakses x dan y dalam getJumlah() di ClassB, compiler menghasilkan error karena variabel tersebut tidak dideklarasikan dalam ruang lingkup ClassB. Dengan menambahkan variabel x dan y serta metode setX() dan setY() di ClassB, kita dapat mengatasi masalah tersebut dengan menggunakan metode setter untuk mengatur nilai variabel x dan y dari luar kelas.

5.PERCOBAAN 3 (Super)

A. TAHAPAN PERCOBAAN

Class Bangun

```
2
3 public class Bangun {
4     protected double phi;
5     protected int r;
6 }
```

Class Tabung

```
3 public class Tabung extends Bangun{
4     protected int t;
5     public void setSuperPhi(double phi){
6         super.phi = phi;
7     }
8     public void setSuperR(int r){
9         super.r = r;
10    }
11    public void setT(int t){ // Corrected method name
12        this.t = t;
13    }
14    public void volume(){
15        System.out.println("Volume Tabung adalah: " + (super.phi * super.r * super.r * this.t));
16    }
17 }
18
```

Class Percobaan 3

```
3 public class Percobaan3 {
4     Run | Debug
5     public static void main(String[] args){
6         Tabung tabung = new Tabung();
7         tabung.setSuperPhi(phi:3.14);
8         tabung.setSuperR(r:10);
9         tabung.setT(t:3);
10        tabung.volume();
11    }
12 }
```

Hasil

```
Volume Tabung adalah: 942.8
PS C:\Users\USER\Documents\Semester 4\Prak PBO\Jobsheet 6>
```

B. PERTANYAAN

1. Jelaskan fungsi “super” pada potongan program berikut di class Tabung!

```
super.phi = phi;
super.r = r;
```

Dalam potongan program di class Tabung, kata kunci super digunakan untuk merujuk ke anggota-anggota dari superclass, yaitu Bangun. Dalam konteks ini, super.phi mengacu pada atribut phi dari kelas Bangun, dan super.r mengacu pada atribut r dari kelas Bangun. Dengan menggunakan super, kita dapat mengakses anggota-anggota yang diwarisi dari superclass, bahkan jika ada nama yang sama di kelas turunan.

2. Jelaskan fungsi “super” dan “this” pada potongan program berikut di class Tabung!

```
(super.phi * super.r * super.r * this.t));
```

Pada potongan program (super.phi * super.r * super.r * this.t), super dan this digunakan untuk merujuk ke konteks tertentu saat mengakses atribut.

3. Jelaskan mengapa pada class Tabung tidak dideklarasikan atribut “phi” dan “r” tetapi class tersebut dapat mengakses atribut tersebut!

Pada potongan program (super.phi * super.r * super.r * this.t), super dan this digunakan untuk merujuk ke konteks tertentu saat mengakses atribut.

6.PERCOBAAN 4 (super constructor)

A. TAHAPAN PERCOBAAN

Class A

```
3 public class ClassA {
4     ClassA() {
5         System.out.println(x:"konstruktor A dijalankan");
6     }
7 }
```

Class B

```
3 public class ClassB extends ClassA {
4     ClassB() {
5         super();
6         System.out.println(x:"konstruktor B dijalankan");
7     }
8 }
9
```

Class C

```
3 public class ClassC extends ClassB {
4     ClassC() {
5         super();
6         System.out.println(x:"konstruktor C dijalankan");
7     }
8 }
```

Percobaan 4

```
3 public class Percobaan4 {
4     public static void main(String[] args) {
5         ClassC test= new ClassC();
6     }
7 }
```

Hasil

```
konstruktor A dijalankan
konstruktor B dijalankan
konstruktor C dijalankan
PS C:\Users\USER\Documents\Semester 4\Prak PBO\Jobsheet 6>
```

B. PERTANYAAN

1. Pada percobaan 4 sebutkan mana class yang termasuk superclass dan subclass, kemudian jelaskan alasannya!

Superclass:

- ClassA adalah superclass karena tidak memiliki superclass lain yang diperluas. Ini adalah kelas dasar dalam hierarki kelas dan tidak mewarisi sifat atau perilaku dari kelas lain.

Subclass:

- ClassB adalah subclass dari ClassA karena memperluas atau mewarisi dari ClassA. ClassB mewarisi semua sifat dan perilaku dari ClassA dan dapat menambahkan perilaku tambahan atau mengubah perilaku yang ada.
- ClassC adalah subclass dari ClassB karena memperluas atau mewarisi dari ClassB. Seperti halnya ClassB, ClassC juga mewarisi semua sifat dan perilaku dari ClassB dan dapat menambahkan perilaku tambahan atau mengubah perilaku yang ada.

2. Ubahlah isi konstruktor default ClassC seperti berikut:

Tambahkan kata super() di baris Pertama dalam konstruktor defaultnya. Coba jalankan kembali class Percobaan4 dan terlihat tidak ada perbedaan dari hasil outputnya!

Menambahkan kata super() di baris pertama konstruktor default ClassC tidak berpengaruh karena kelas induknya (Class) tidak memiliki konstruktor default. Output program tetap sama dengan atau tanpa kata super() pada baris pertama konstruktor default ClassC.

3. Ublah isi konstruktor default ClassC seperti berikut: Ketika mengubah posisi super() dibaris kedua dalam konstruktor defaultnya dan terlihat ada error. Kemudian kembalikan super() kebaris pertama seperti

sebelumnya, maka errornya akan hilang. Perhatikan hasil keluaran ketika class Percobaan4 dijalankan. Kenapa bisa tampil output seperti berikut pada saat instansiasi objek test dari class ClassC Jelaskan bagaimana urutan proses jalannya konstruktor saat objek test dibuat!

Urutan proses jalannya konstruktor:

Ketika objek `test` dibuat dari kelas `ClassC`, urutan proses jalannya konstruktor adalah sebagai berikut:

- 1. **Konstruktor kelas `A` dijalankan terlebih dahulu.**
 - 2. **Kemudian, konstruktor kelas `B` dijalankan.**
 - 3. **Terakhir, konstruktor kelas `C` dijalankan.**
4. Apakah fungsi `super()` pada potongan program dibawah ini di `ClassC`!

Fungsi `super()` pada potongan program di atas digunakan untuk memanggil konstruktor kelas dasar (`ClassB`). Hal ini penting karena konstruktor kelas dasar perlu dijalankan terlebih dahulu sebelum konstruktor kelas turunan (`ClassC`).

7.PERCOBAAN 5

A. TAHAPAN PERCOBAAN

Class Karyawan

```
3 public class Karyawan {
4     public String nama, alamat, jk;
5     public int umur, gaji;
6
7     public Karyawan(){
8
9     }
10    public Karyawan(String nama, String alamat, String jk, int umur, int gaji){
11        this.nama=nama;
12        this.alamat=alamat;
13        this.jk=jk;
14        this.umur=umur;
15        this.gaji=gaji;
16
17    public void tampilDataKaryawan(){
18        System.out.println("Nama          =" +nama);
19        System.out.println("Alamat       =" +alamat);
20        System.out.println("Jenis Kelamin =" +jk);
21        System.out.println("Umur        =" +umur);
22        System.out.println("Gaji        =" +gaji);
23    }
24 }
25
26
27
```

Class Manager

```
3 public class Manager extends Karyawan {
4     public int tunjangan;
5
6     public Manager(){
7
8     }
9     public void tampilDataManager(){
10        super.tampilDataKaryawan();
11        System.out.println("Tunjangan      =" +tunjangan);
12        System.out.println("Total Gaji     =" +(super.gaji+tunjangan));
13    }
14 }
15
```

Class Staff

```

3 public class Staff extends Karyawan{
4     public int lembur, potongan;
5
6     public Staff(){
7
8     }
9     public Staff(String nama, String alamat, String jk, int umur, int gaji, int lembur, int potongan){
10         super(nama,alamat,jk,umur,gaji);
11         this.lembur=lembur;
12         this.potongan=potongan;
13     }
14     public void tampilDataStaff(){
15         super.tampilDataKaryawan();
16         System.out.println("lembur          ="+lembur);
17         System.out.println("Potongan        ="+potongan);
18         System.out.println("Total Gaji       ="+(gaji+lembur-potongan));
19     }
20 }
21

```

Class inheritance1

```

5
6 public class Inheritance1 {
7     Run | Debug
8     public static void main(String[] args) {
9         Manager M = new Manager();
10         M.nama="Vanda";
11         M.alamat="Jl. Ngingit";
12         M.umur=20;
13         M.jk="Laki-Laki";
14         M.gaji=3000000;
15         M.tunjangan=1000000;
16         M.tampilDataManager();
17
18         Staff S = new Staff();
19         S.nama="Lestari";
20         S.alamat="Jl. Vanilla";
21         S.umur=20;
22         S.jk="Perempuan";
23         S.gaji=2000000;
24         S.lembur=500000;
25         S.potongan=250000;
26         S.tampilDataStaff();
27     }
28 }

```

Hasil

```

Nama          =Vanda
Alamat        =Jl. Ngingit
Jenis Kelamin =Laki-Laki
Umur          =20
Gaji          =3000000
Tunjangan     =1000000
Total Gaji    =4000000
Nama          =Lestari
Alamat        =Jl. Vanilla
Jenis Kelamin =Perempuan
Umur          =20
Gaji          =2000000
lembur        =500000
Potongan      =250000
Total Gaji    =2250000
PS C:\Users\USER\Documents\Semester 4\Prak PBO\Jobsheet 6>

```

B. PERTANYAAN

1. Sebutkan class mana yang termasuk super class dan sub class dari percobaan 1 diatas!

- Super class: **Karyawan**
- Sub class: **Manager** dan **Staff**

2. Kata kunci apakah yang digunakan untuk menurunkan suatu class ke class yang lain?

extends

3. Perhatikan kode program pada class Manager, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Karyawan!

- tunjangan (atribut yang dimiliki oleh class Manager sendiri)
- nama, alamat, jk, umur, gaji (atribut yang diwarisi dari class Karyawan)

4. Jelaskan kata kunci super pada potongan program dibawah ini yang terdapat pada class Manager!

Kata kunci super digunakan untuk merujuk ke superclass dari suatu class. Pada potongan program yang Anda tunjukkan, super.gaji mengacu pada atribut gaji yang ada di superclass Karyawan. Dengan menggunakan super, kita bisa mengakses atribut atau method dari superclass saat kita berada di dalam subclass.

5. Program pada percobaan 1 diatas termasuk dalam jenis inheritance apa? Jelaskan alasannya!

Program pada percobaan 1 di atas termasuk dalam jenis inheritance kelas tunggal (single inheritance). Hal ini karena setiap subclass (Manager dan Staff) hanya mewarisi sifat dari satu superclass (Karyawan). Dalam Java, hanya mendukung inheritance tunggal, yang berarti sebuah class hanya bisa memiliki satu superclass.

8.PERCobaan 6

A. TAHAPAN PERCobaan

Class Staff Tetap

```
3 public class StaffTetap extends Staff {
4     public String golongan;
5     public int asuransi;
6
7     public StaffTetap(){
8
9     }
10    public StaffTetap(String nama, String alamat, String jk, int umur, int gaji, int lembur, int potongan, String golongan, int asuransi){
11        super(nama,alamat,jk,umur,gaji,lembur,potongan);
12        this.golongan=golongan;
13        this.asuransi=asuransi;
14    }
15    public void tampilDataStaffTetap(){
16        System.out.println("=====Data Staff Tetap=====");
17        super.tampilDataStaff();
18        System.out.println("Golongan"      +": "+golongan);
19        System.out.println("Jumlah Asuransi"  +": "+asuransi);
20        System.out.println("Gaji Bersih"      +": "+(gaji+lembur-potongan-asuransi));
21    }
22
23 }
24
```

Class Staff Harian

```
3 public class StaffHarian extends Staff {
4     public int jellamKerja;
5
6     public StaffHarian(){
7
8     }
9     public StaffHarian(String nama, String alamat, String jk, int umur, int gaji, int lembur, int potongan,int jellamKerja){
10        super(nama,alamat,jk,umur,gaji,lembur,potongan);
11        this.jellamKerja=jellamKerja;
12    }
13    public void tampilDataStaffHarian(){
14        System.out.println("=====Data Staff Harian=====");
15        super.tampilDataStaff();
16        System.out.println("Jumlah Jam Kerja"      +": "+jellamKerja);
17        System.out.println("Gaji Bersih"          +": "+(gaji*jellamKerja+lembur-potongan));
18    }
19 }
20
```

Class Inheritance

```
3 public class Inheritance1 {
4     Run | Debug
5     public static void main(String[] args) {
6         StaffTetap ST = new StaffTetap(nama:"Budi", alamat:"Malang", jk:"Lakilaki", umur:20, gaji:2000000, lembur:250000, potongan:200000, go-"2A", 10000);
7         ST.tampilDataStaffTetap();
8
9         StaffHarian SH = new StaffHarian(nama:"Indah", alamat:"Malang", jk:"Perempuan", umur:27, gaji:10000, lembur:100000, potongan:50000, jm_100);
10        SH.tampilDataStaffHarian();
11    }
12 }
```

Hasil

```
=====Data Staff Tetap=====
Nama          =Budi
Alamat        =Malang
Jenis Kelamin  =Lakilaki
Umur          =20
Gaji          =2000000
lembur        =250000
Potongan      =200000
Total Gaji    =1950000
Golongan      =2A
Jumlah Asuransi =10000
Gaji Bersih   =1940000
=====Data Staff Harian=====
Nama          =Indah
Alamat        =Malang
Jenis Kelamin  =Perempuan
Umur          =27
Gaji          =10000
lembur        =100000
Potongan      =50000
Total Gaji    =60000
Jumlah Jam Kerja =100
Gaji Bersih   =1050000
PS C:\Users\USER\Documents\Semester 4\Prak PB0\Jobsheet 6> |
```

B. PERTANYAAN

1. Berdasarkan class diatas manakah yang termasuk single inheritance dan mana yang termasuk multilevel inheritance?
- Single inheritance: Manager, Staff, StaffTetap, dan StaffHarian masing-masing merupakan contoh dari single inheritance karena masing-masing hanya memiliki satu superclass.
 - Multilevel inheritance: Tidak ada contoh multilevel inheritance dalam program yang diberikan.

2. Perhatikan kode program class StaffTetap dan StaffHarian, atribut apa saja yang dimiliki oleh class tersebut? Sebutkan atribut mana saja yang diwarisi dari class Staff!

- nama, alamat, jk, umur, gaji, lembur, potongan (atribut yang diwarisi dari class Staff)
- golongan (atribut tambahan yang dimiliki oleh class StaffTetap)
- asuransi dan jmlJamKerja (atribut tambahan yang dimiliki oleh class StaffHarian)

3. Apakah fungsi potongan program berikut pada class StaffHarian

```
super(nama, alamat, jk, umur, gaji, lembur, potongan);
```

Potongan program `super(nama, alamat, jk, umur, gaji, lembur, potongan);` pada class StaffHarian adalah untuk memanggil konstruktor dari superclass (Staff). Dengan cara ini, kita dapat menggunakan parameter yang diterima oleh konstruktor superclass untuk menginisialisasi objek StaffHarian

4. Apakah fungsi potongan program berikut pada class StaffHarian

```
super.tampilDataStaff();
```

Potongan program `super.tampilDataStaff();` pada class StaffHarian adalah untuk memanggil method `tampilDataStaff()` dari superclass (Staff). Dengan cara ini, kita dapat menggunakan method yang telah didefinisikan di superclass untuk menampilkan data staff, sehingga menghindari duplikasi kode.

5. Perhatikan kode program dibawah ini yang terdapat pada class StaffTetap Terlihat dipotongan program diatas atribut gaji, lembur dan potongan dapat diakses langsung. Kenapa hal ini bisa terjadi dan bagaimana class StaffTetap memiliki atribut gaji, lembur, dan potongan padahal dalam class tersebut tidak dideklarasikan atribut gaji, lembur, dan potongan?

```
System.out.println("Gaji Bersih      =" + (gaji + lembur - potongan - asuransi));
```

Atribut gaji, lembur, dan potongan dapat diakses langsung karena class StaffTetap mewarisi atribut-atribut tersebut dari superclass-nya, yaitu class Staff. Meskipun tidak ada deklarasi langsung dalam class StaffTetap, namun karena class tersebut adalah subclass dari Staff, maka class StaffTetap memiliki akses ke atribut-atribut yang telah didefinisikan dalam Staff.