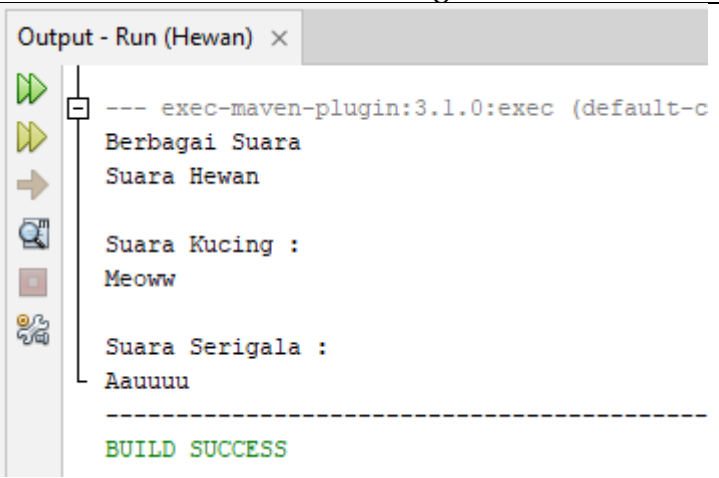


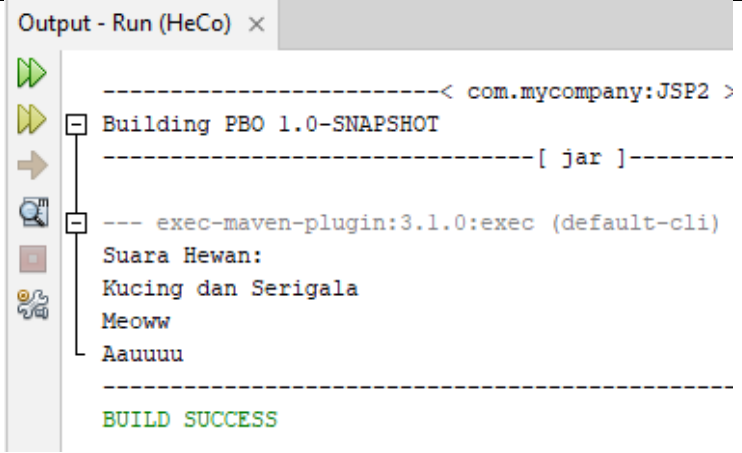
Nama : Intan Novita Sari

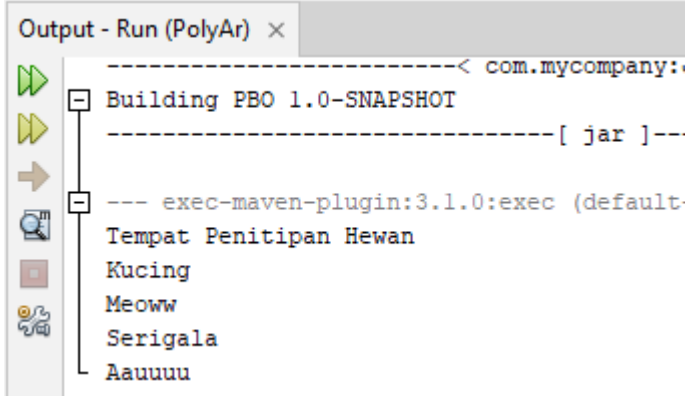
NIM : F1B020057

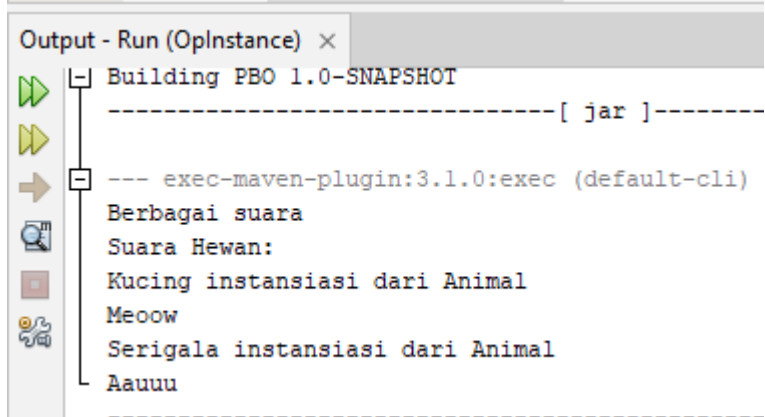
Klp : 4

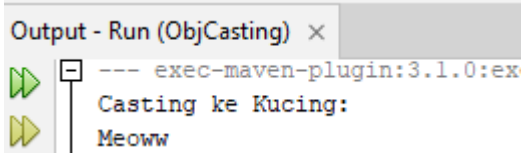
JOB SHEET P7 (*Polymorphisme*)

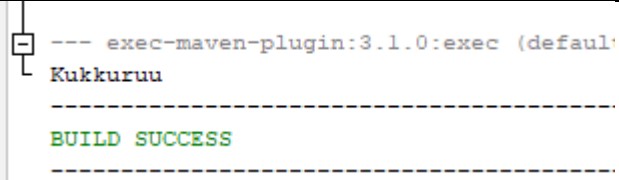
NO	Kegiatan	Script	Hasil Running
1	Buatlah program bebas menggunakan virtual method invocation	<pre>package JSP7; class Animal { public void suara() { System.out.println("Suara Hewan"); } } class Kucing extends Animal { @Override public void suara() { System.out.println("Meoww"); } } class Serigala extends Animal { @Override public void suara() { System.out.println("Aauuuu"); } } public class Hewan { public static void main(String[] args) { Animal animal = new Animal(); Animal Kucing = new Kucing(); Animal Serigala = new Serigala(); } }</pre>	 <p>Output - Run (Hewan) ×</p> <pre>--- exec-maven-plugin:3.1.0:exec (default-c Berbagai Suara Suara Hewan Suara Kucing : Meoww Suara Serigala : Aauuuu ----- BUILD SUCCESS</pre>

		<pre> System.out.println("Berbagai Suara"); animal.suara(); System.out.println("\nSuara Kucing :"); Kucing.suara(); System.out.println("\nSuara Serigala :"); Serigala.suara(); } } </pre>	
2	Buatlah program bebas atau modifikasi program disamping menggunakan Heterogeneous Collection	<pre> package JSP7; import java.util.ArrayList; class Animal { public void suara() { System.out.println("Kucing dan Serigala"); } } class Kucing extends Animal { public void suara() { System.out.println("Meoww"); } } class Serigala extends Animal { public void suara() { System.out.println("Aauuuu"); } } public class HeCo { public static void main(String[] args) { </pre>	 <pre> Output - Run (HeCo) x -----< com.mycompany:JSP2 > Building PBO 1.0-SNAPSHOT -----[jar]----- --- exec-maven-plugin:3.1.0:exec (default-cli) Suara Hewan: Kucing dan Serigala Meoww Aauuuu ----- BUILD SUCCESS </pre>

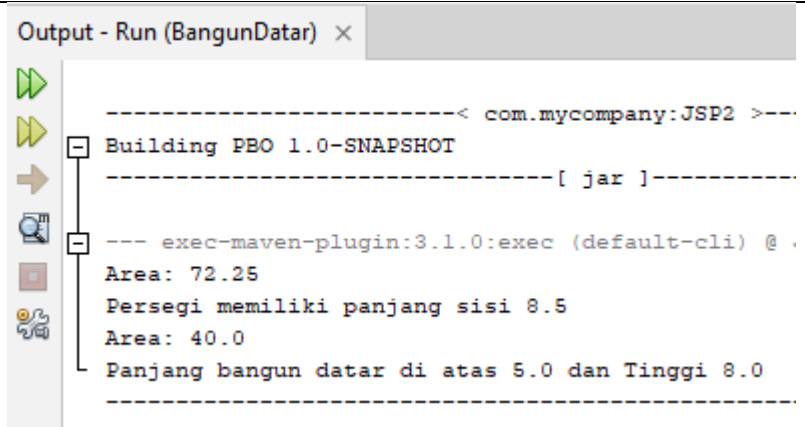
		<pre> ArrayList<Object> animals = new ArrayList<>(); animals.add(new Animal()); animals.add(new Kucing()); animals.add(new Serigala()); System.out.println("Suara Hewan:"); for (Object animal : animals) { if (animal instanceof Animal) { Animal hewan = (Animal) animal; hewan.suara(); } } } } </pre>	
3	Buatlah program bebas atau modifikasi program disamping menggunakan polymorphic argument	<pre> package JSP7; class Animal { public void suara() { System.out.println("Tempat Penitipan Hewan"); } } class Kucing extends Animal { public void suara() { System.out.println("Meoww"); } } class Serigala extends Animal { public void suara() { System.out.println("Aauuuu"); } } </pre>	

		<pre> } class AnimalShelter { public void introduce(Animal animal) { animal.suara(); } } public class PolyAr { public static void main(String[] args) { AnimalShelter Hwn = new AnimalShelter(); Animal hewan = new Animal(); Kucing a = new Kucing(); Serigala b = new Serigala(); Hwn.introduce(hewan); System.out.println("Kucing"); Hwn.introduce(a); System.out.println("Serigala"); Hwn.introduce(b); } } </pre>	
4	Buatlah program bebas atau modifikasi program disamping menggunakan operator instanceof	<pre> package JSP7; class Animal { public void suara() { System.out.println("Suara Hewan:"); } } class Cat extends Animal { public void suara() { System.out.println("Meoow"); } } </pre>	

		<pre> class Wolf extends Animal { public void suara() { System.out.println("Aauuu"); } } public class OpInstance { public static void main(String[] args) { Animal animal = new Animal(); Cat a = new Cat(); Wolf b = new Wolf(); if (animal instanceof Animal) { System.out.println("Berbagai suara"); animal.suara(); } if (a instanceof Animal) { System.out.println("Kucing instansiasi dari Animal"); a.suara(); } if (b instanceof Animal) { System.out.println("Serigala instansiasi dari Animal"); b.suara(); } } } </pre>	
5	Buatlah program bebas atau modifikasi program disamping	<pre> package JSP7; class Animal { public void suara() { System.out.println("Suara Hewan"); } } </pre>	

	menggunakan Object Casting	<pre> } class Kucing extends Animal { public void suara() { System.out.println("Meooow"); } public void playandrun() { System.out.println("Kucing merupakan hewan imut"); } } public class ObjCasting { public static void main(String[] args) { Kucing a = new Kucing(); Animal animal = a; if (animal instanceof Kucing) { Kucing b = (Kucing) animal; System.out.println("Casting ke Kucing:"); b.suara(); b.playandrun(); } else { System.out.println("Gagal untuk casting ke Kucing"); } } } </pre>	
6	Buatlah program bebas dengan Up Casting	<pre> package JSP7; class Animal { void suara() { System.out.println("Hewan Berkaki Dua"); } } </pre>	 <pre> --- exec-maven-plugin:3.1.0:exec (default: Kukkuruu ----- BUILD SUCCESS ----- </pre>

		<pre> class Ayam extends Animal { void suara() { System.out.println("Kukkuruu"); } } public class UpCasting { public static void main(String[] args) { Ayam a = new Ayam(); Animal animal = a; animal.suara(); } } </pre>	
7	Buatlah program bebas dengan Down Casting	<pre> package JSP7; class Animal { public void suara() { System.out.println("Hewan Buas"); } } class Serigala extends Animal { public void suara() { System.out.println("Serigala"); } public void playandrun() { System.out.println("Serigala Putih"); } } public class DownCast { public static void main(String[] args) { Serigala s = new Serigala(); </pre>	 <pre> Output - Run (DownCast) x --- exec-maven-plugin:3.1.0:exec (default-c Casting ke Serigala: Auuuuu </pre>

		<pre> Animal animal = s; System.out.println("Casting ke Serigala:"); s.suara(); s.playandrun(); } } </pre>	
8	Buatlah program untuk membandingkan kedua nilai menggunakan polimorfis statis		
9	Buatlah program bebas menggunakan polimorfis dinamis dengan jumlah class : Akhiran NIM ganjil : 3 class Akhiran NIM genap : 4 class	<pre> package JSP7; class datarBangun { public double getArea() { return 0.0; } } class Persegi extends datarBangun { private double s; public Persegi(double s) { this.s = s; } public double getArea() { return s * s; } public void printInfo() { System.out.println("Persegi memiliki panjang sisi " + s); } } </pre>	 <pre> Output - Run (BangunDatar) x -----< com.mycompany:JSP2 >----- Building PBO 1.0-SNAPSHOT -----[jar]----- --- exec-maven-plugin:3.1.0:exec (default-cli) @ . Area: 72.25 Persegi memiliki panjang sisi 8.5 Area: 40.0 Panjang bangun datar di atas 5.0 dan Tinggi 8.0 ----- </pre>

		<pre>class PersegiPanjang extends datarBangun { private double p; private double t; public PersegiPanjang(double p, double t) { this.p = p; this.t = t; } public double getArea() { return p * t; } public void printInfo() { System.out.println("Persegi Panjang bangun datar di atas " + p + " dan Tinggi " + t); } } public class BangunDatar { public static void main(String[] args) { datarBangun a = new Persegi(8.5); datarBangun b = new PersegiPanjang(5, 8); displayBangunDatarInfo(a); displayBangunDatarInfo(b); } public static void displayBangunDatarInfo(datarBangun shape) { System.out.println("Area: " + shape.getArea()); if (shape instanceof Persegi) {</pre>	
--	--	--	--

		<pre> ((Persegi) shape).printInfo(); } else if (shape instanceof PersegiPanjang) { ((PersegiPanjang) shape).printInfo(); } } </pre>	
10	Modifikasi program disamping menggunakan inputan dinamis		
11	Buatlah program tambahan mengikuti contoh disamping lalu mengganti keyword super menjadi this	<pre> package JSP7; class Orang { String nama = "Intan"; int age = 21; } class Pelatih extends Orang { float gaji = 100000; String nama = "Akbar"; int umur = 25; public void tampilkanInfo() { System.out.println("Nama : " + this.nama); System.out.println("Umur : " + this.umur); System.out.println("Gaji: RP." + gaji); } } public class Key { public static void main(String[] args) { Pelatih Akbar = new Pelatih(); Akbar.tampilkanInfo(); } } </pre>	<pre> --- exec-maven-plugin:3.1.0:exec (defau Nama : Akbar Umur : 25 Gaji: RP.100000.0 ----- BUILD SUCCESS </pre>

		}	
--	--	---	--