|  |  |  |
| --- | --- | --- |
| NO | Script | Hasil Running |
| 1 | class Tumbuhan {  void tanam() {  System.out.println("tumbuhan ditanam");  }  void berkembang() {  System.out.println("tumbuhan tumbuh");  }  }  class Pohon extends Tumbuhan {  @Override  void tanam() {  System.out.println("pohon ditanam");  }  @Override  void berkembang() {  System.out.println("Pohon tumbuh");  }  }  class Bunga extends Tumbuhan {  @Override  void tanam() {  System.out.println("bunga ditanam");  }  @Override  void berkembang() {  System.out.println("Bunga tumbuh");  }  }  public class Flora {  public static void main(String[] args) {  Tumbuhan pohon = new Pohon();  Tumbuhan bunga = new Bunga();  pohon.tanam();  pohon.berkembang();    bunga.tanam();  bunga.berkembang();  }  } |  |
| 2 | import java.util.ArrayList;  class Tanaman {    }  class Melati extends Tanaman {  }  class Pohon extends Tanaman {  }  public class Gaga {  public static void prosesTanaman(Tanaman tanaman) {  if (tanaman instanceof Melati) {  System.out.println("Memproses Melati");  } else if (tanaman instanceof Pohon) {    System.out.println("Memproses Pohon");  } else {  // Metode lainnya  System.out.println("Memproses Tanaman Lainnya");  }  }  public static void main(String[] args) {  ArrayList<Tanaman> taman = new ArrayList<>();    taman.add(new Melati());  taman.add(new Pohon());  for (Tanaman tanaman : taman) {  prosesTanaman(tanaman);  }  }  } |  |
| 3 | class Mobil {  String merek;  Mobil(String merek) {  this.merek = merek;  }  void berkendara() {  System.out.println("Mobil " + merek + " sedang berkendara");  }  }  class Sedan extends Mobil {  Sedan(String merek) {  super(merek);  }  @Override  void berkendara() {  System.out.println("Sedan " + merek + " sedang berkendara");  }  }  class SUV extends Mobil {  SUV(String merek) {  super(merek);  }  @Override  void berkendara() {  System.out.println("SUV " + merek + " sedang berkendara");  }  }  public class Kendaraan {  public static void Proses(Mobil mobil) {  if (mobil instanceof Sedan) {  Sedan sedan = (Sedan) mobil;  sedan.berkendara();  } else if (mobil instanceof SUV) {  SUV suv = (SUV) mobil;  suv.berkendara();  } else {  System.out.println("Memproses Mobil Lainnya");  }  }  public static void main(String[] args) {  Mobil sedan = new Sedan("Toyota");  Mobil suv = new SUV("Honda");  Proses(sedan);  Proses(suv);  }  } |  |
| 4 | class Manusia {  String nama;  Manusia(String nama) {  this.nama = nama;  }  void bekerja() {  System.out.println(nama + " sedang bekerja");  }  }  class Pekerja extends Manusia {  Pekerja(String nama) {  super(nama);  }  void bekerjaKeras() {  System.out.println(nama + " sedang bekerja ");  }  }  class Pengantar extends Manusia {  Pengantar(String nama) {  super(nama);  }  void antarBarang() {  System.out.println(nama + " sedang mengantar barang");  }  }  public class Utama {  public static void Proses(Manusia manusia) {  if (manusia instanceof Pekerja) {  Pekerja pekerja = (Pekerja) manusia;  pekerja.bekerjaKeras();  } else if (manusia instanceof Pengantar) {  Pengantar pengantar = (Pengantar) manusia;  pengantar.antarBarang();  } else {  System.out.println(manusia.nama + " sedang bekerja");  }  }  public static void main(String[] args) {  Pekerja pekerja = new Pekerja("Fauzan");  Pengantar pengantar = new Pengantar("Fatah");  Proses(pekerja);  Proses(pengantar);  }  } |  |
| 5 | class Kendaraan {  void info() {  System.out.println("Ini adalah kendaraan");  }  }  class Mobil extends Kendaraan {  void infoMobil() {  System.out.println("Ini adalah mobil");  }  }  class SepedaMotor extends Kendaraan {  void infoSepedaMotor() {  System.out.println("Ini adalah sepeda motor");  }  }  public class Gaga {  public static void main(String[] args) {  Kendaraan kendaraan1 = new Mobil();  Kendaraan kendaraan2 = new SepedaMotor();  if (kendaraan1 instanceof Mobil) {  Mobil mobil = (Mobil) kendaraan1;  mobil.infoMobil();  } else if (kendaraan1 instanceof SepedaMotor) {  SepedaMotor sepedaMotor = (SepedaMotor) kendaraan1;  sepedaMotor.infoSepedaMotor();  }  if (kendaraan2 instanceof Mobil) {  Mobil mobil = (Mobil) kendaraan2;  mobil.infoMobil();  } else if (kendaraan2 instanceof SepedaMotor) {  SepedaMotor sepedaMotor = (SepedaMotor) kendaraan2;  sepedaMotor.infoSepedaMotor();  }  }  } |  |
| 6 | class Binatang {  void bersuara() {  System.out.println("Binatang sedang bersuara");  }  }  class Kucing extends Binatang {  void meong() {  System.out.println("Kucing sedang meong");  }  }  class Anjing extends Binatang {  void gonggong() {  System.out.println("Anjing sedang gonggong");  }  }  public class Gaga {  public static void main(String[] args) {  Kucing binatang1 = new Kucing();  Anjing binatang2 = new Anjing();  binatang1.meong();  binatang2.gonggong();  }  } |  |
| 7 | class Kendaraan {  void info() {  System.out.println("Ini adalah kendaraan");  }  }  class Mobil extends Kendaraan {  void infoMobil() {  System.out.println("Ini mobil");  }  }  class SepedaMotor extends Kendaraan {  void infoSepedaMotor() {  System.out.println("Ini sepeda motor");  }  }  public class Gaga {  public static void main(String[] args) {  Kendaraan kendaraan = new Mobil();  if (kendaraan instanceof Mobil) {  Mobil mobil = (Mobil) kendaraan;  mobil.infoMobil();  }  Kendaraan kendaraan2 = new SepedaMotor();  if (kendaraan2 instanceof SepedaMotor) {  SepedaMotor sepedaMotor = (SepedaMotor) kendaraan2;  sepedaMotor.infoSepedaMotor();  }  }  } |  |
| 8 | public class Gaga {  static int bandingkan(int nilai1, int nilai2) {  System.out.println("Membandingkan dua nilai integer:");  if (nilai1 > nilai2) {  return 1;  } else if (nilai1 < nilai2) {  return -1;  } else {  return 0;  }  }  static double bandingkan(double nilai1, double nilai2) {  System.out.println("Membandingkan dua nilai double:");  if (nilai1 > nilai2) {  return 1.0;  } else if (nilai1 < nilai2) {  return -1.0;  } else {  return 0.0;  }  }  public static void main(String[] args) {  int hasilInt = bandingkan(5, 10);  System.out.println("Hasil perbandingan integer: " + hasilInt);  double hasilDouble = bandingkan(3.5, 2.0);  System.out.println("Hasil perbandingan double: " + hasilDouble);  }  } |  |
| 9 | class Kendaraan {  void info() {  System.out.println("Diatas termasuk kendaraan");  }  }  class Mobil extends Kendaraan {  @Override  void info() {  System.out.println("Ini adalah mobil");  }  }  class SepedaMotor extends Kendaraan {  @Override  void info() {  System.out.println("Ini adalah sepeda motor");  }  }  public class Gaga {  public static void tampilkanInfo(Kendaraan kendaraan) {  kendaraan.info();  }  public static void main(String[] args) {  Kendaraan kendaraan1 = new Mobil();  Kendaraan kendaraan2 = new SepedaMotor();  Kendaraan kendaraan3 = new Kendaraan();  tampilkanInfo(kendaraan1);  tampilkanInfo(kendaraan2);  tampilkanInfo(kendaraan3);  }  } |  |
| 10 | import java.util.Scanner;  class Employee {  private String name;  private double salary;  private static double salary\_rise\_percent = 0.2;  // Konstruktor  public Employee(String nm, double sly) {  this.setName(nm);  this.setSalary(sly);  }  // Setter dan Getter  public void setName(String nm) {  name = nm;  }  public void setSalary(double sly) {  salary = sly;  }  public String getName() {  return name;  }  public double getSalary() {  return salary;  }  public static double getPresentase() {  return salary\_rise\_percent;  }  // Metode untuk menaikkan gaji  public void salaryUp() {  salary += (salary \* salary\_rise\_percent);  }  // Setter untuk presentase kenaikan gaji  public static void setPresentase(double percent) {  salary\_rise\_percent = percent;  }  }  class Manager extends Employee {  private static double bonus = 500;  // Konstruktor  public Manager(String nm, double sly) {  super(nm, sly);  }  // Getter dan Setter untuk bonus  public double getBonus() {  return bonus;  }  public void setBonus(double bns) {  bonus = bns;  }  // Override metode getSalary untuk menghitung gaji total  @Override  public double getSalary() {  double salaryBase = super.getSalary();  return (salaryBase + bonus);  }  }  public class Gaga {  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  System.out.print("Masukkan nama manager: ");  String namaManager = scanner.nextLine();  System.out.print("Masukkan gaji manager: ");  double gajiManager = scanner.nextDouble();  Manager manager = new Manager(namaManager, gajiManager);  System.out.println("Nama: " + manager.getName());  System.out.println("Bonus: " + manager.getBonus());  System.out.println("Gaji Total: " + manager.getSalary());  }  } |  |
| 11 | class Person {  String nama = "Fauzan";  int usia = 21;  }  class Dosen extends Person {  float gaji = 200000f;  void tampilkanInfo() {  System.out.println("Nama: " + super.nama);  System.out.println("Usia: " + super.usia);  System.out.println("Gaji: Rp" + gaji);  }  }  class Mahasiswa extends Person {  String jurusan;  Mahasiswa(String nama, int usia, String jurusan) {  this.nama = nama;  this.usia = usia;  this.jurusan = jurusan;  }  void tampilkanInfo() {  System.out.println("Nama: " + this.nama);  System.out.println("Usia: " + this.usia);  System.out.println("Jurusan: " + this.jurusan);  }  }  public class Gaga {  public static void main(String[] args) {  Dosen rismon = new Dosen();  rismon.tampilkanInfo();  System.out.println("\nInformasi Mahasiswa:");  Mahasiswa mahasiswa = new Mahasiswa("Fauzan Al-Farisi", 21, "Elektro");  mahasiswa.tampilkanInfo();  }  } |  |
| 12 | package package9;  import java.util.Date;  class Employee {  private static final double BASE\_SALARY = 15000.00;  private String name;  private double salary;  private Date birthDate;  public Employee(String name, double salary, Date DoB) {  this.name = name;  this.salary = salary;  this.birthDate = DoB;  }  public Employee(String name, double salary) {  this(name, salary, null);  }  public Employee(String name, Date DoB) {  this(name, BASE\_SALARY, DoB);  }  public Employee(String name) {  this(name, BASE\_SALARY);  }  public String getName() {  return name;  }  }  class Manager extends Employee {  private String department;  public Manager(String name, double salary, String dept) {  super(name, salary);  department = dept;  }  public Manager(String name, String dept) {  super(name);  department = dept;  }  public String getDepartment() {  return department;  }  }  public class Gaga {  public static void main(String[] args) {  Employee man = new Manager("Fauzan", 16000.00, "Electrical");  if (man instanceof Manager) {  Manager manager = (Manager) man;  System.out.println("Name: " + man.getName());  System.out.println("Department: " + manager.getDepartment());  }  }  } |  |