**LAPORAN PEMROGRAMAN BERBASIS OBJECT**

**“Pholimorpysm”**



**DOSEN PENGAMPU:**

**Lukie Perdanasari, S.Kom., M.T.**

**DISUSUN OLEH:**

Aisyah Currents

E31241326

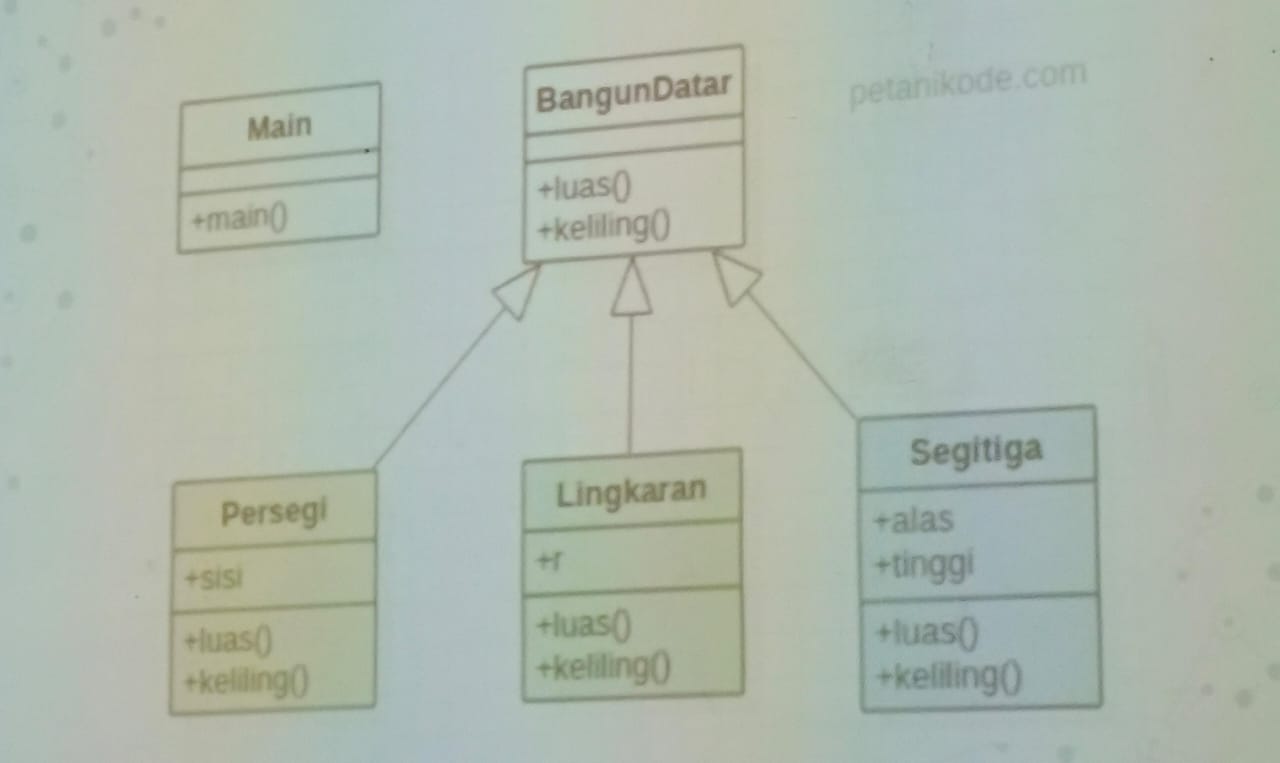
Golongan C

**PROGRAM STUDI MANAJEMEN INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**2025**

1. Ubah kedalam Kode



Source Code:

1. package pholimorpysm;
2. public abstract class PholimorpysmBangunDatar {
3. public abstract double luas();
4. public abstract double keliling();
5. }
6. class Persegi extends PholimorpysmBangunDatar {
7. private double sisi;
8. public Persegi(double sisi) {
9. this.sisi = sisi;
10. }
11. @Override
12. public double luas() {
13. return sisi \* sisi;
14. }
15. @Override
16. public double keliling() {
17. return 4 \* sisi;
18. }
19. }
20. class Lingkaran extends PholimorpysmBangunDatar {
21. private double r;
22. public Lingkaran(double r) {
23. this.r = r;
24. }
25. @Override
26. public double luas() {
27. return Math.PI \* r \* r;
28. }
29. @Override
30. public double keliling() {
31. return 2 \* Math.PI \* r;
32. }
33. }
34. class Segitiga extends PholimorpysmBangunDatar {
35. private double alas;
36. private double tinggi;
37. public Segitiga(double alas, double tinggi) {
38. this.alas = alas;
39. this.tinggi = tinggi;
40. }
41. @Override
42. public double luas() {
43. return (alas \* tinggi) / 2;
44. }
45. @Override
46. public double keliling() {
47. double sisiMiring = Math.sqrt((alas \* alas) + (tinggi \* tinggi));
48. return alas + tinggi + sisiMiring;
49. }
50. }
51. package pholimorpysm;
52. public class Main {
53. public static void main(String[] args) {
54. PholimorpysmBangunDatar[] daftarBangunDatar = {
55. new Persegi(8),
56. new Lingkaran(14),
57. new Segitiga(7, 8)
58. };
59. for (PholimorpysmBangunDatar bangun : daftarBangunDatar) {
60. System.out.println("Luas: " + bangun.luas());
61. System.out.println("Keliling: " + bangun.keliling());
62. System.out.println("-------------------");
63. }
64. }
65. }

Analisa:

1. Konsep OOP yang Digunakan

Source code ini menerapkan **Object-Oriented Programming (OOP)** dengan beberapa prinsip utama:

**Inheritance (Pewarisan):**

* PholiomorpysmHewan adalah kelas abstrak yang menjadi **superclass**.
* Kelas Kucing, Elang, dan LumbaLumba adalah **subclass** yang mewarisi properti dan metode dari superclass.

**Polymorphism (Polimorfisme):**

* Metode Bersuara() dan Bergerak() di-**override** di masing-masing subclass dengan implementasi spesifik sesuai jenis hewan.

**Encapsulation (Enkapsulasi):**

* Atribut nama pada kelas PholiomorpysmHewan dibuat **private**, sehingga hanya bisa diakses melalui metode getter (getNama()).

**2. Struktur Kelas**

**a) PholiomorpysmHewan (Superclass / Abstract Class)**

* Memiliki atribut private String nama untuk menyimpan nama hewan.
* Metode getNama() untuk mengembalikan nama hewan.
* Metode abstrak Bersuara() dan Bergerak() yang wajib diimplementasikan oleh setiap subclass.

**b) Kucing (Subclass)**

* Atribut tinggiLompat menyimpan ketinggian lompat dalam cm.

Bersuara(): Menghasilkan suara "Meow!".

Bergerak(): Menampilkan aksi melompat sesuai tinggiLompat.

**c) Elang (Subclass)**

* Atribut lebarSayap menyimpan rentang sayap dalam meter.

Bersuara(): Menghasilkan suara "Cit!".

Bergerak(): Menampilkan aksi meluncur dengan lebar sayap tertentu.

**d) LumbaLumba (Subclass)**

* Atribut kedalamanRenang menyimpan kedalaman renang dalam meter.
* Bersuara(): Menghasilkan suara "Eee-ee-ee!".
* Bergerak(): Menampilkan aksi menyelam hingga kedalaman tertentu.

**3. Kelas Zoo (Main Class)**

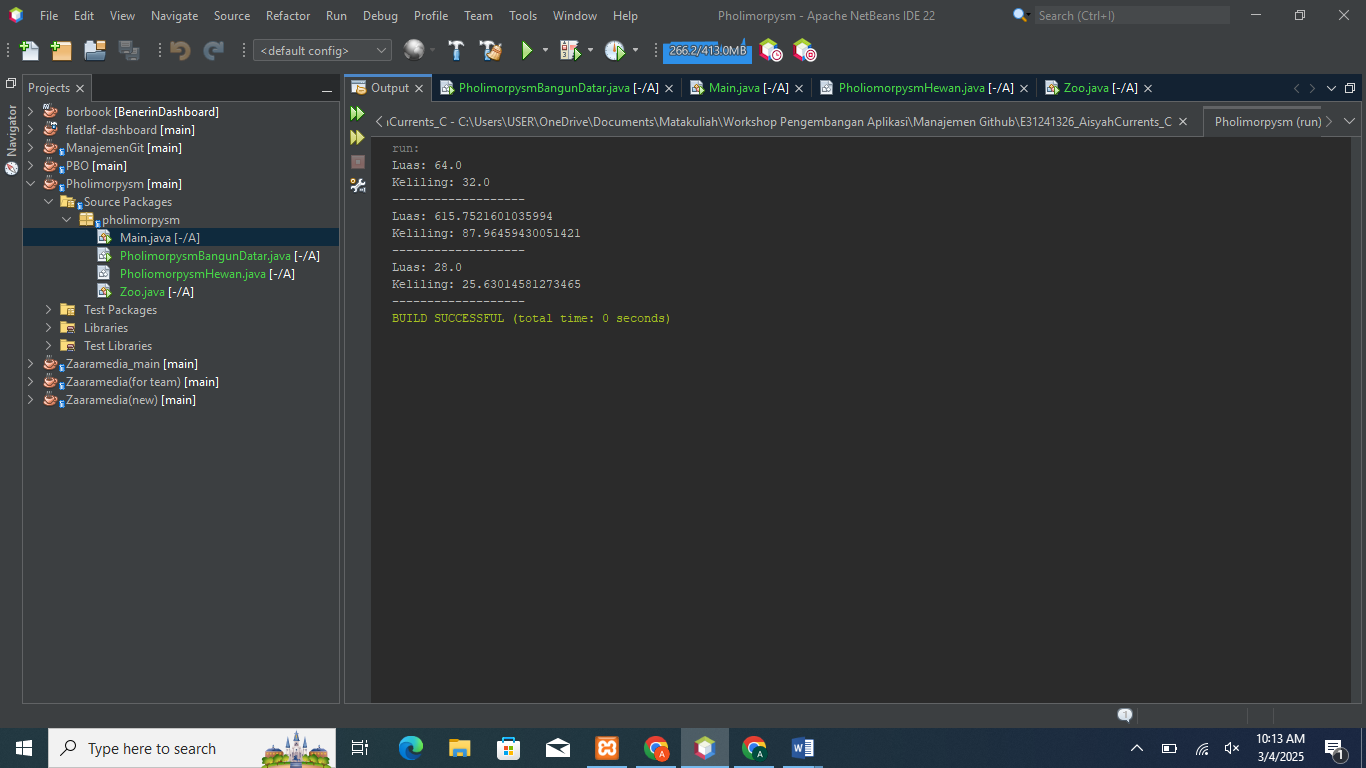
* Membuat **array** PholiomorpysmHewan[] zoo yang menyimpan objek dari berbagai subclass.
* Menggunakan **loop for-each** untuk menampilkan informasi setiap hewan dengan:

Nama hewan

Suara hewan (dari Bersuara())

Cara bergerak (dari Bergerak()).

Hasil Output:



1. Berikan contoh lain untuk penerapan pholimorpysm

Source Code:

1. package pholimorpysm;
2. public abstract class PholiomorpysmHewan {
3. private String nama;
5. public PholiomorpysmHewan(String nama) {
6. this.nama = nama;
7. }
9. public String getNama() {
10. return nama;
11. }
13. public abstract void Bersuara();
14. public abstract void Bergerak();
15. }
16. class Kucing extends PholiomorpysmHewan {
17. private int tinggiLompat;
18. public Kucing(String nama, int tinggiLompat) {
19. super(nama);
20. this.tinggiLompat = tinggiLompat;
21. }
22. @Override
23. public void Bersuara() {
24. System.out.println(getNama() + " Bersuara: Meow!");
25. }
26. @Override
27. public void Bergerak() {
28. System.out.println(getNama() + " melompat setinggi" + tinggiLompat + " cm");
29. }
30. }
31. class Elang extends PholiomorpysmHewan {
32. private double lebarSayap;
33. public Elang(String nama, double lebarSayap) {
34. super(nama);
35. this.lebarSayap = lebarSayap;
36. }
37. @Override
38. public void Bersuara() {
39. System.out.println(getNama() + " bersuara: Cit!");
40. }
41. @Override
42. public void Bergerak() {
43. System.out.printf("%s meluncur dengan lebar sayap %.1f meter\n", getNama(), lebarSayap);
44. }
45. }
46. class LumbaLumba extends PholiomorpysmHewan {
47. private int kedalamanRenang;
48. public LumbaLumba(String nama, int kedalamanRenang) {
49. super(nama);
50. this.kedalamanRenang = kedalamanRenang;
51. }
52. @Override
53. public void Bersuara() {
54. System.out.println(getNama() + " berkata: Eee-ee-ee!");
55. }
56. @Override
57. public void Bergerak() {
58. System.out.println(getNama() + " menyelam hingga " + kedalamanRenang + " meter");
59. }
60. }
61. package pholimorpysm;
62. public class Zoo {
63. public static void main(String[] args) {
64. PholiomorpysmHewan[] zoo= {
65. new Kucing("Yoyo", 50),
66. new Elang("Sky King", 2.3),
67. new LumbaLumba("Bubbles", 70)
68. };
70. for(PholiomorpysmHewan hewan : zoo) {
71. System.out.println("=== " + hewan.getNama() + " ===");
72. hewan.Bersuara();
73. hewan.Bergerak();
74. System.out.println();
75. }
76. }
77. }

Analisa:

**1. Konsep OOP yang Digunakan**

Kode ini menggunakan **Object-Oriented Programming (OOP)** dengan konsep:  
 **Inheritance (Pewarisan):**

* PholiomorpysmHewan adalah **kelas abstrak** yang menjadi parent.
* Kucing, Elang, dan LumbaLumba adalah subclass yang mewarisi dari PholiomorpysmHewan.

**Polymorphism (Polimorfisme):**

* Method Bersuara() dan Bergerak() diimplementasikan secara berbeda di masing-masing subclass.

**Encapsulation (Enkapsulasi):**

* Variabel nama, tinggiLompat, lebarSayap, dan kedalamanRenang bersifat private untuk menjaga data tetap aman.

**2. Struktur Kelas**

**Kelas PholiomorpysmHewan (Abstract Class)**

* Memiliki atribut nama dan method **abstract** Bersuara() & Bergerak().
* Tidak bisa dibuat objek langsung, harus diwarisi oleh subclass.

**Subclass Kucing**

* Memiliki atribut tambahan tinggiLompat.
* Implementasi Bersuara() → "Meow!"
* Implementasi Bergerak() → Melompat sesuai tinggi lompatnya.

**Subclass Elang**

* Memiliki atribut tambahan lebarSayap.
* Implementasi Bersuara() → "Cit!"
* Implementasi Bergerak() → Meluncur dengan lebar sayapnya.

**Subclass LumbaLumba**

* Memiliki atribut tambahan kedalamanRenang.
* Implementasi Bersuara() → "Eee-ee-ee!"
* Implementasi Bergerak() → Menyelam hingga kedalaman tertentu.

**Kelas Zoo (Main Class)**

* Membuat array PholiomorpysmHewan[] zoo untuk menyimpan objek Kucing, Elang, dan LumbaLumba.
* Menggunakan **perulangan for-each** untuk memanggil method Bersuara() dan Bergerak() dari setiap objek.

Hasil Output:

