**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 5**

**“POLYMORPHISM”**

Icon

Description automatically generated

**Disusun oleh**

Bunga Laelatul Muna

NIM : 21102010

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2023**

**BAB 1 – TUJUAN PRAKTIKUM**

Tujuan dari praktikum instalasi Java (IntelliJ IDEA) adalah untuk memberikan pengetahuan dan keterampilan kepada mahasiswa tentang Polymorphism

* Mahasiswa dapat memahami tentang konsep utama pbo yaitu polymorphism
* Mahasiswa dapat membuat polimorphism dalam program.

**BAB II- DASAR TEORI**

Polimorfisme merupakan konsep pembuatan method berbeda dengan nama yang

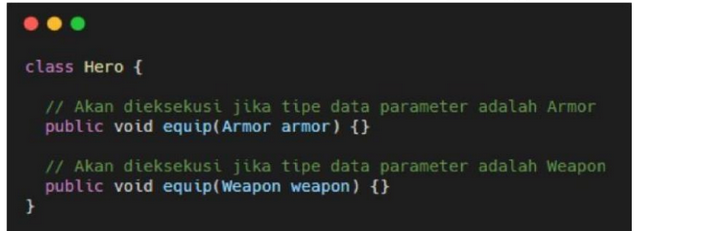
sama. Polimorfisme dibagi menjadi dua, yaitu:

1. **Overloading**

Overloading adalah ketika di sebuah class ada 2 atau lebih method dengan

nama sama namun masing-masing memiliki parameter berbeda. Walaupun

namanya sama, program tetap mengerti harus menjalankan method yang mana berdasarkan parameter yang dimasukan saat pemanggilan method.

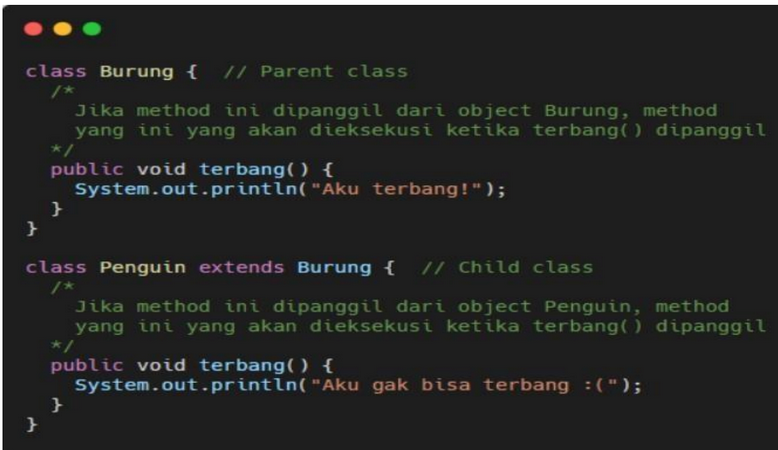


1. **Overiding**

Overriding adalah ketika sebuah Child class memiliki method dengan nama

yang sama dengan Parent class-nya, namun dengan isi yang berbeda (parameter

boleh sama). Nantinya program akan memprioritaskan method milik Child class ini jika ternyata ketahuan Child class memiliki method yang di-Override dari Parent class-nya.



**BAB III – GUIDED**

**Guided 1**

* **Source Code (Buah.Java)**

|  |
| --- |
| //Bunga Laelatul Muna // 21102010  package com.Bunga\_Laelatul\_Muna.PBO.Pertemuan5.LatihanKelas;  public class Buah {  // Atribut  String nama;  int jumlah;   // Constructor  public Buah(){}  public Buah(String nama, int jumlah){  this.nama = nama;  this.jumlah = jumlah;  }   // Method  public void showInfo(){  System.*out*.println("Nama buah : " + nama);  System.*out*.println("Jumlah : " + jumlah);   extraInfo();  }   public void extraInfo(){} } |

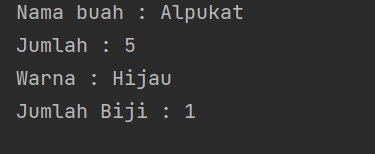
* **Source Code (Monokotil.java)**

|  |
| --- |
| package com.Bunga\_Laelatul\_Muna.PBO.Pertemuan5.LatihanKelas;   public class Monokotil extends Buah{  // Atributr  String warna;  int jumlahBiji;   // Constructor  public Monokotil(){}  // Overloading  public Monokotil(String warna, int jumlahBiji){  this.warna = warna;  this.jumlahBiji = jumlahBiji;  }  public Monokotil(String nama, int jumlah, String warna, int jumlahBiji){  super(nama, jumlah);  this.warna = warna;  this.jumlahBiji = jumlahBiji;  }   // Method  // Overriding  public void extraInfo(){  System.*out*.println("Warna : " + warna);  System.*out*.println("Jumlah Biji : " + jumlahBiji);  } } |

* **Source Code (Main.Java)**

|  |
| --- |
| package com.Bunga\_Laelatul\_Muna.PBO.Pertemuan5.LatihanKelas; public class Main {  public static void main(String[] args){  // Object  Monokotil mono = new Monokotil();  mono.nama = "Alpukat";  mono.jumlah = 5;  mono.warna = "Hijau";  mono.jumlahBiji = 1;   mono.showInfo();  } } } |

* **SS Output**

****

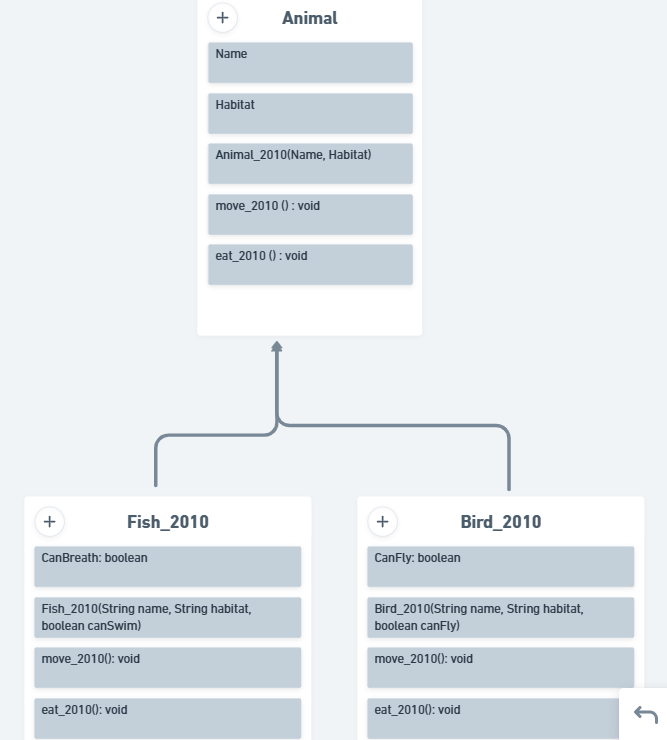
* **Penjelasan**

Ini programa Polymoprhism yang memiliki 1 parent dan 1 child. Buah sebagai parent memiliki constructor dengan 2 parameter yaitu nama dan jumlah yang kemudian di tampilkan di method showInfo. Lalu pada monokotil sebagai child. Kelas ini memiliki dua atribut yaitu warna dan jumlahBiji, dan tiga konstruktor yaitu konstruktor tanpa parameter, konstruktor dengan parameter warna dan jumlahBiji, dan konstruktor dengan parameter nama, jumlah, warna, dan jumlahBiji. Kelas Monokotil juga memiliki metode extraInfo() yang meng-overriding metode yang sama pada kelas Buah dan menambahkan informasi mengenai warna dan jumlah biji.

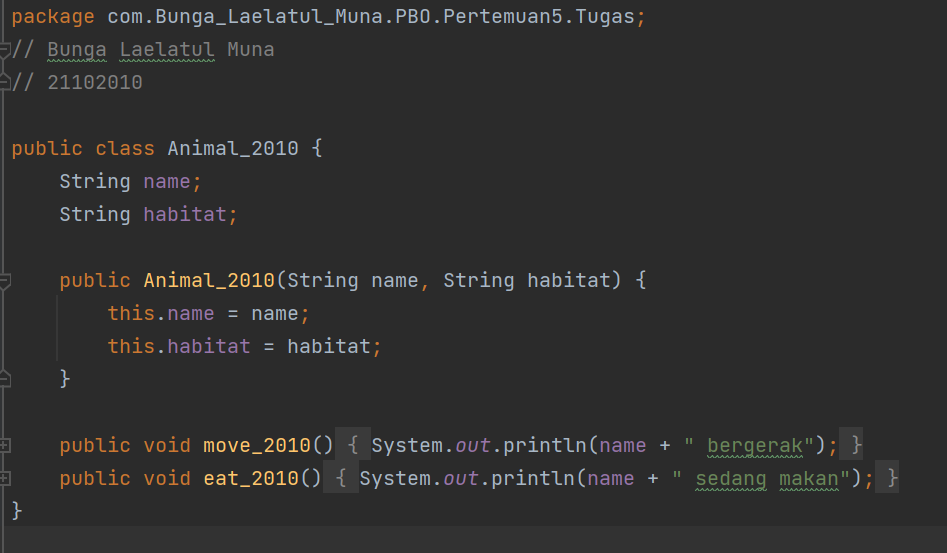
**BAB IV – UNGUIDED**

**UnGuided 1**

* **Diagram**

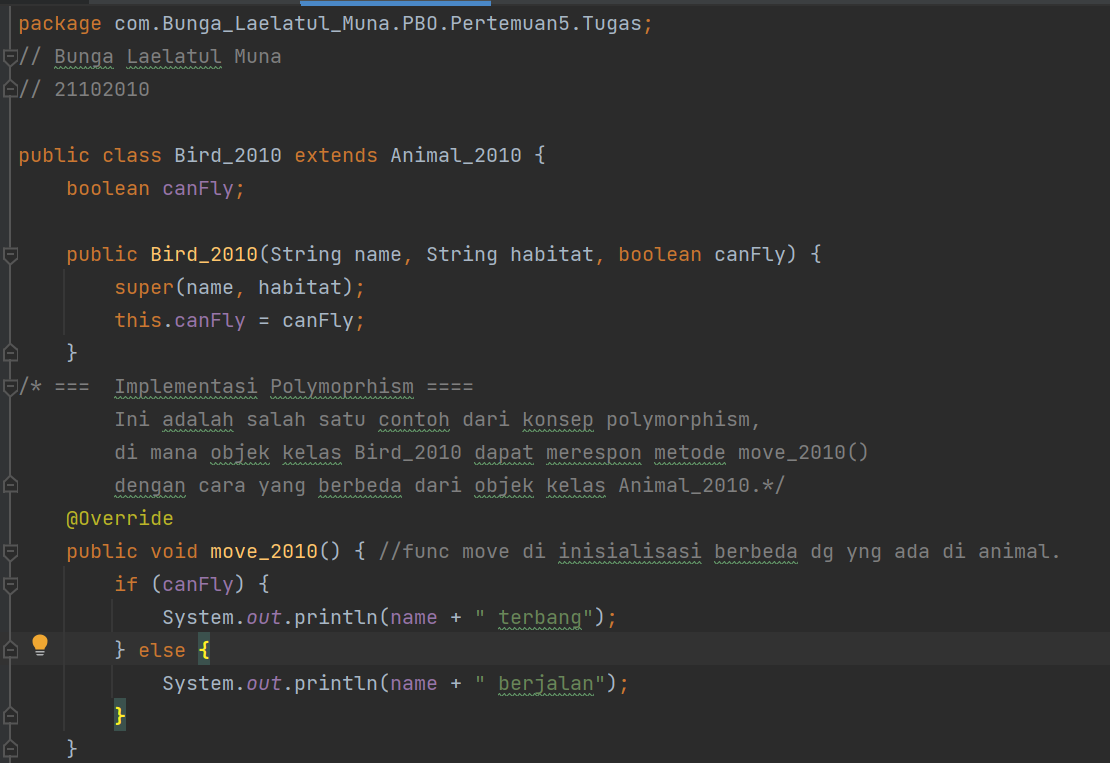
****

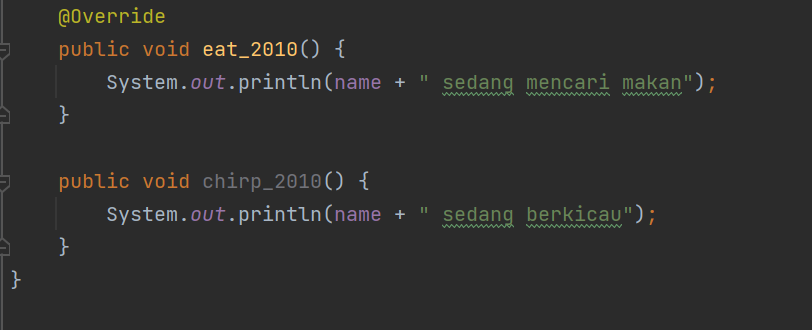
* **Source Code(Animal\_2010.Java)**

****

|  |
| --- |
| package com.Bunga\_Laelatul\_Muna.PBO.Pertemuan5.Tugas; // Bunga Laelatul Muna // 21102010  public class Animal\_2010 {  String name;  String habitat;   public Animal\_2010(String name, String habitat) {  this.name = name;  this.habitat = habitat;  }    public void move\_2010() {  System.*out*.println(name + " bergerak");  }  public void eat\_2010() {  System.*out*.println(name + " sedang makan");  } } |

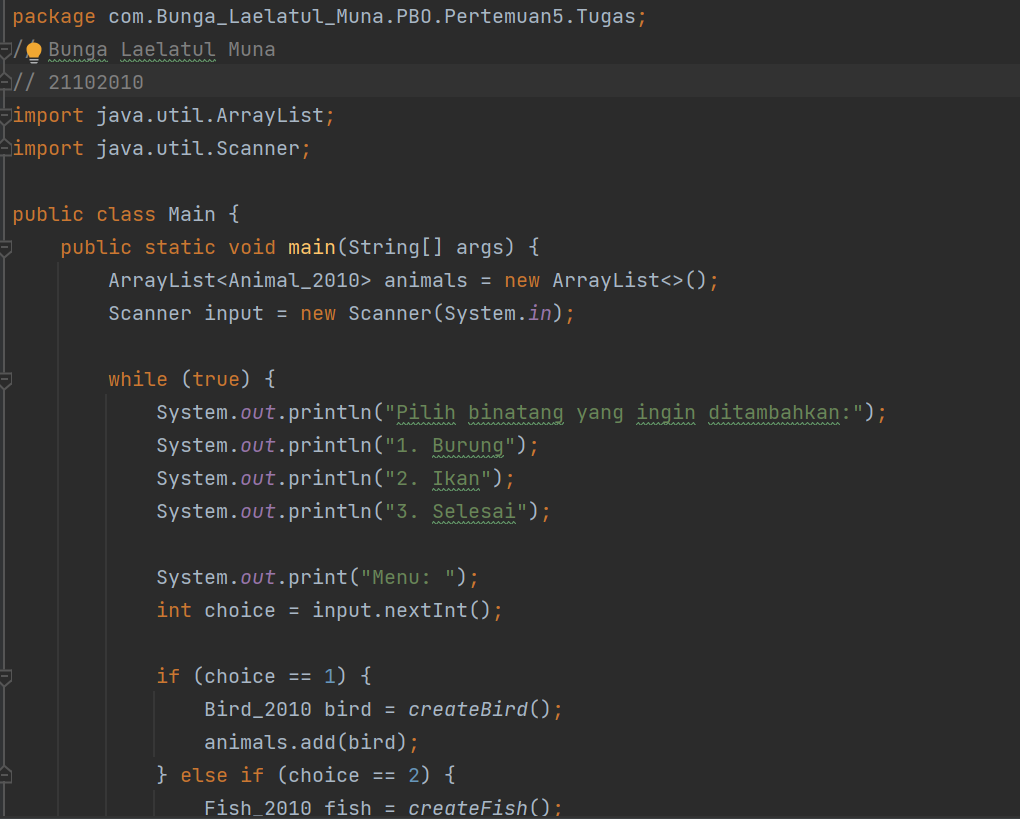
* **Source Code (Bird\_2010.java)**

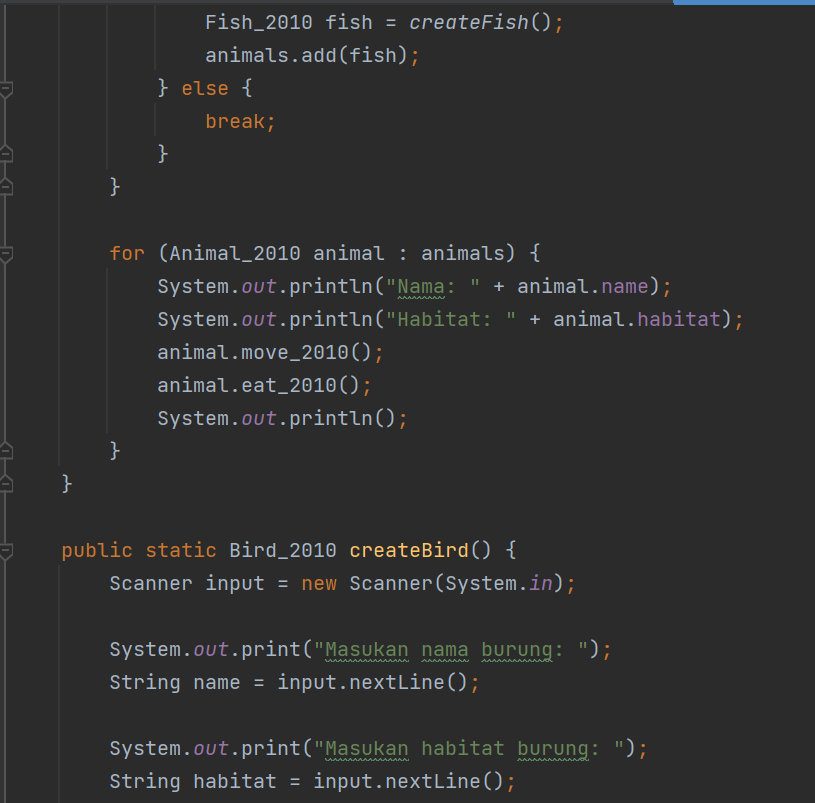
****

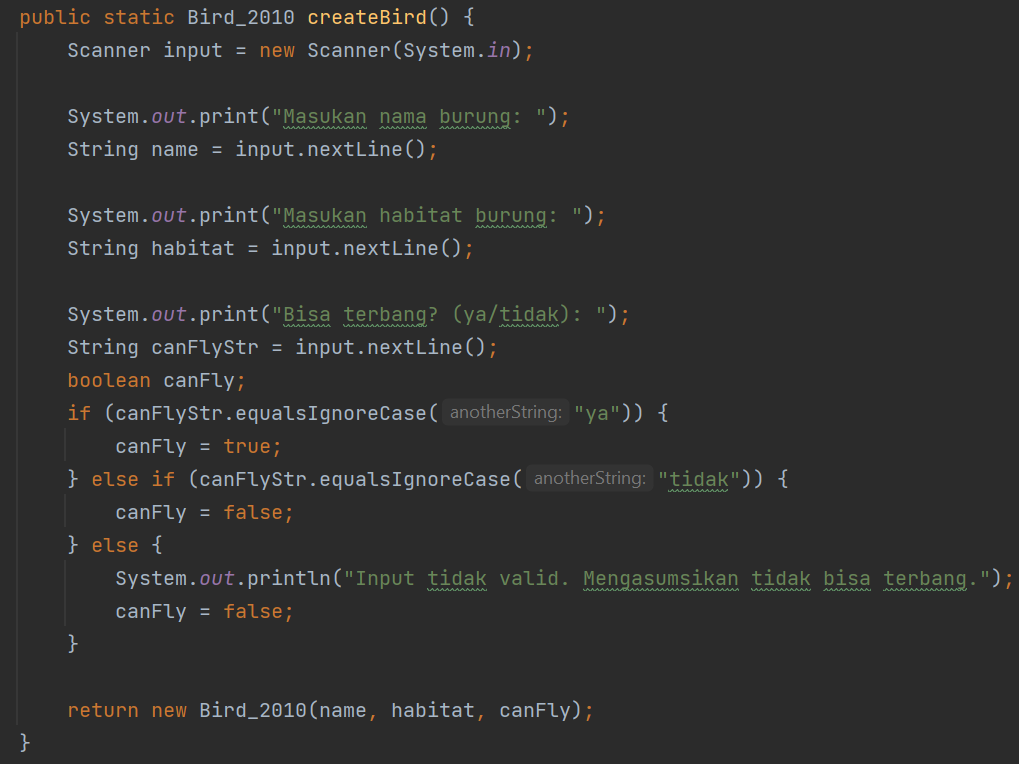
****

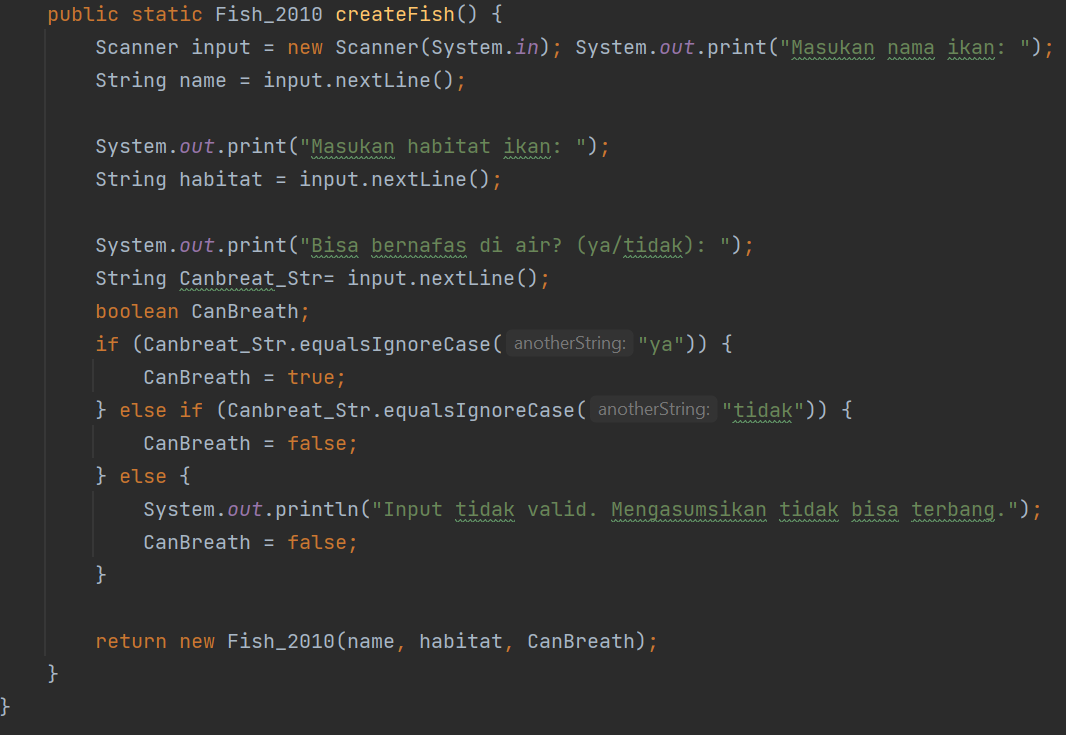
|  |
| --- |
| package com.Bunga\_Laelatul\_Muna.PBO.Pertemuan5.Tugas; // Bunga Laelatul Muna // 21102010  public class Bird\_2010 extends Animal\_2010 {  oolean canFly;   public Bird\_2010(String name, String habitat, oolean canFly) {  super(name, habitat);  this.canFly = canFly;  }  /\* === Implementasi Polymoprhism ====  Ini adalah salah satu contoh dari konsep polymorphism,  di mana objek kelas Bird\_2010 dapat merespon metode move\_2010()  dengan cara yang berbeda dari objek kelas Animal\_2010.\*/  @Override  public void move\_2010() { //func move di inisialisasi berbeda dg yng ada di animal.  if (canFly) {  System.*out*.println(name + “ terbang”);  } else {  System.*out*.println(name + “ berjalan”);  }  }   @Override  public void eat\_2010() {  System.*out*.println(name + “ sedang mencari makan”);  }   public void chirp\_2010() {  System.*out*.println(name + “ sedang berkicau”);  } } |

* **Source Code (Main.Java)**

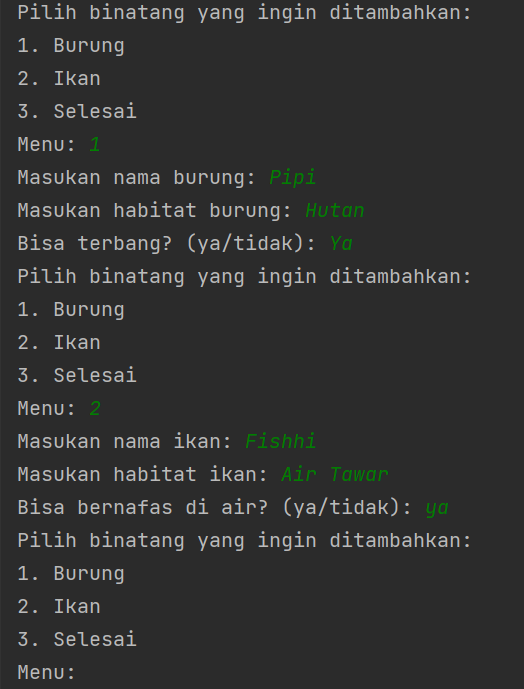
****

****

****

****

* **SS Output**

****

* **Penjelasan**

Ini program program Polymorpishm dengan menu yang mengimplementasi jenis overriding. Dimana di kelas Animal dia memiliki method move\_2010 yang menampilkan output ‘bergerak’ namun di kelas fish dan bird method tersebut memiliki output yang berbeda dengan yang ada di animal.

Lalu di main kelas – kelas tsb di panggil dengan menu yang menggunakan perulangan for. Di program main juga ada menggunakan Boolean untuk menjawab pertanyaan dr methode move.

* **Alur Program**

Ketika Program di run, yang akan muncul pertama ada pilihan menu untuk option 1 Bird dan Option 2 Fish. Ketika user memilih 1 maka akan langsung ke program di kelas Bird. Di kelas Bird akan muncul pertanyaan yang ada di kelas animal yaitu nama dan habitat kemudian disusul pertanyaan dari kelas burung yaitu method Move\_2010(terbang/tidak). Jika user menjawab iya maka akan muncul “Bisa terbang” dan sebaliknya jika tidak “Tidak Terbang”. Iya/Tidak ini adalah bentuk Boolean yang di konversi ke string menggunakan if else.

Ketika user memilih angka 2 yang muncul adalah pertanyaan dr kelas animal dan juga kelas fish yaitu methode move(Bisa bernafas dalam air atau tidak)

1. **REFERENSI**
2. PetaniKode
3. Java Documentation
4. Modul 5