**Jobsheet 10 – Abstract Class**

1. **Kompetensi**

Setelah menyelesaikan lembar kerja ini mahasiswa diharapkan mampu:

1. Menjelaskan maksud dan tujuan penggunaan Abstract Class;
2. Menerapkan Abstract Class di dalam pembuatan program.
3. **Pendahuluan**

**Abstract Class**

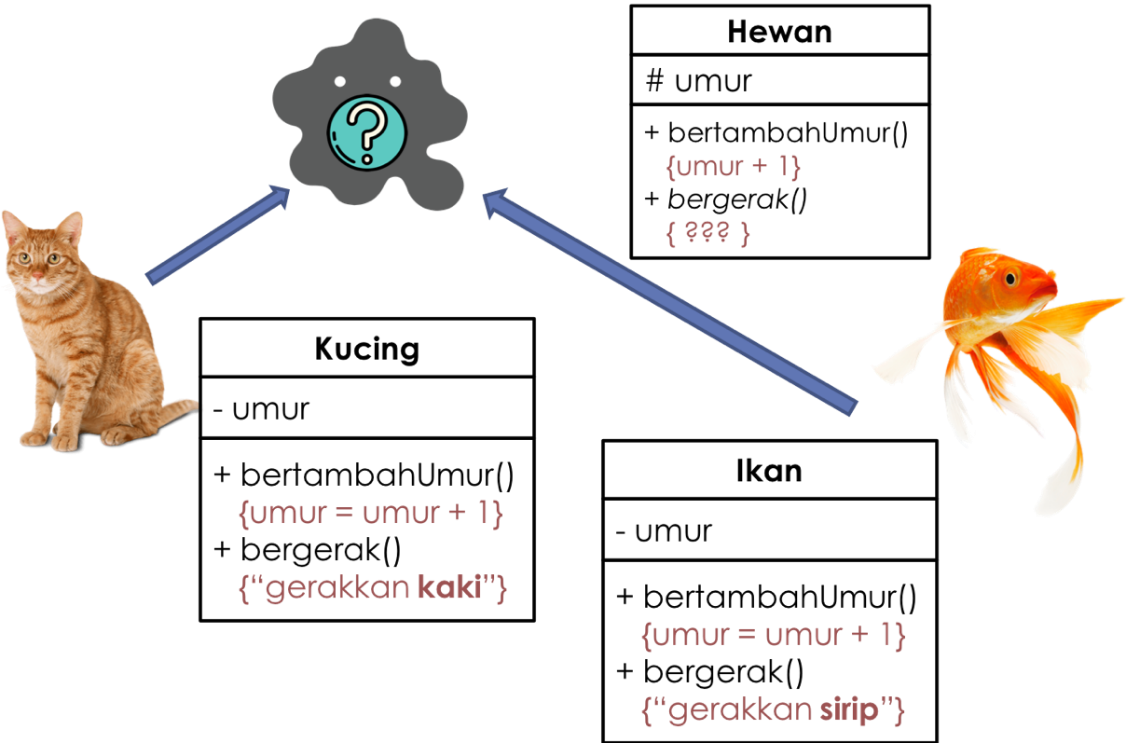
*Abstract Class* adalah class yang tidak dapat diinstansiasi namun dapat di-*extend*. *Abstract class baru* dapat dimanfaatkan ketika ia di-*extend*.

**Karakteristik:**

1. Dapat memiliki *properties* dan *methods* seperti class biasa.
2. Selalu memiliki *methods* yang tidak memiliki tubuh (hanya deklarasinya saja), disebut juga *abstract method*.
3. Selalu dideklarasikan dengan menggunakan kata kunci abstract class.

**Kegunaan:**

Menggambarkan sesuatu yang bersifat umum, yang hanya bisa berfungsi setelah ia dideskripsikan ke dalam bentuk yang lebih spesifik.

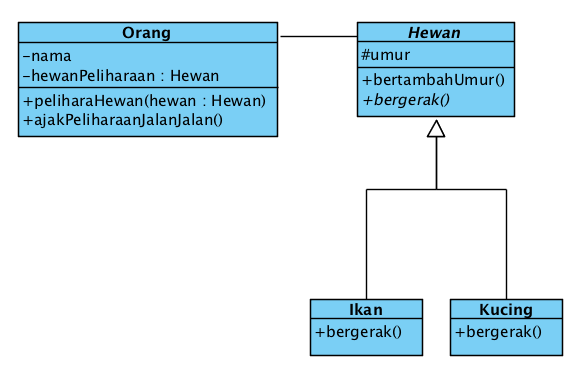


1. **Praktikum**

**Percobaan 1: Abstract Class**

Di dunia ini terdapat banyak jenis hewan. Semua hewan memiliki beberapa karakteristik yang sama, seperti contohnya semua hewan memiliki umur, hewan apapun itu, umurnya akan bertambah sama jumlahnya setiap tahun.

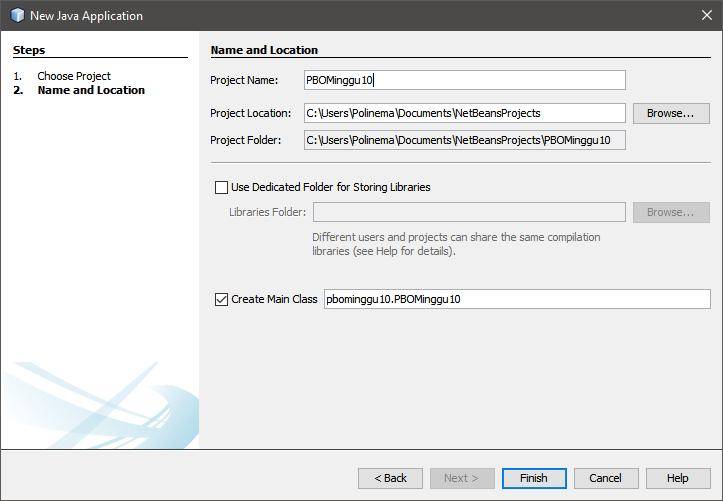
Selain karakteristik yang sama, masing-masing hewan juga memiliki karakteristik yang berbeda satu dengan yang lainnya. Contohnya dalam hal **bergerak**. Cara kucing bergerak berbeda dengan cara ikan bergerak. Kucing bergerak dengan cara melangkahkan kaki-kakinya sedangkan ikan bergerak dengan cara menggerakkan siripnya.



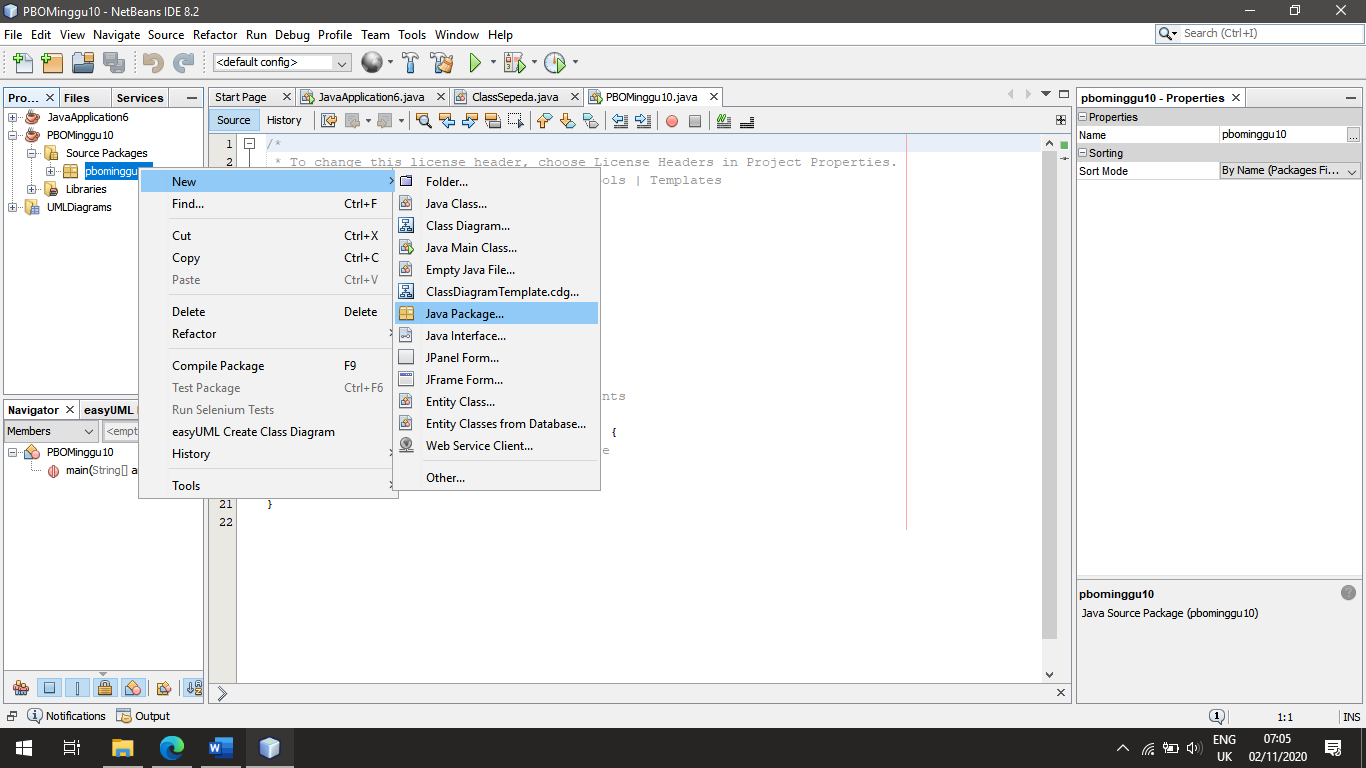
Setiap orang yang memelihara hewan dapat mengajak hewan peliharaannya berjalan (membuat agar hewan peliharaannya bergerak). Namun orang yang memelihara hewan yang berbeda, akan berbeda pula cara hewan peliharaannya dalam bergerak.

Pada percobaan pertama ini kita akan membuat sebuah program yang menggambarkan skenario di atas dengan memanfaatkan **abstract class**.

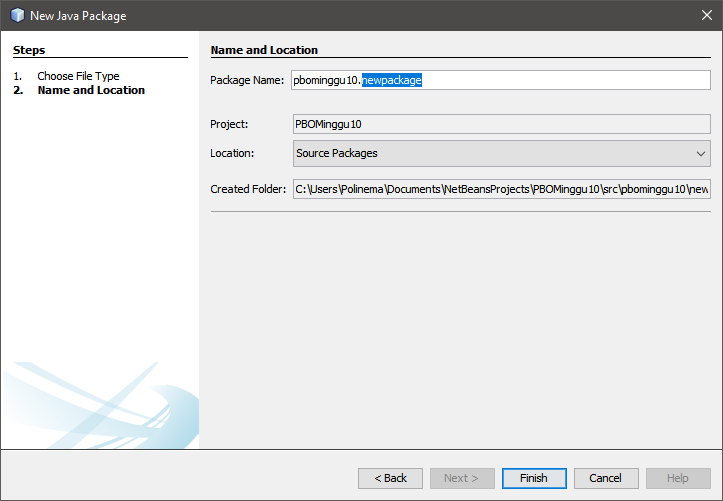
1. Buatlah sebuah project baru di NetBeans dengan nama **PBOMinggu10**



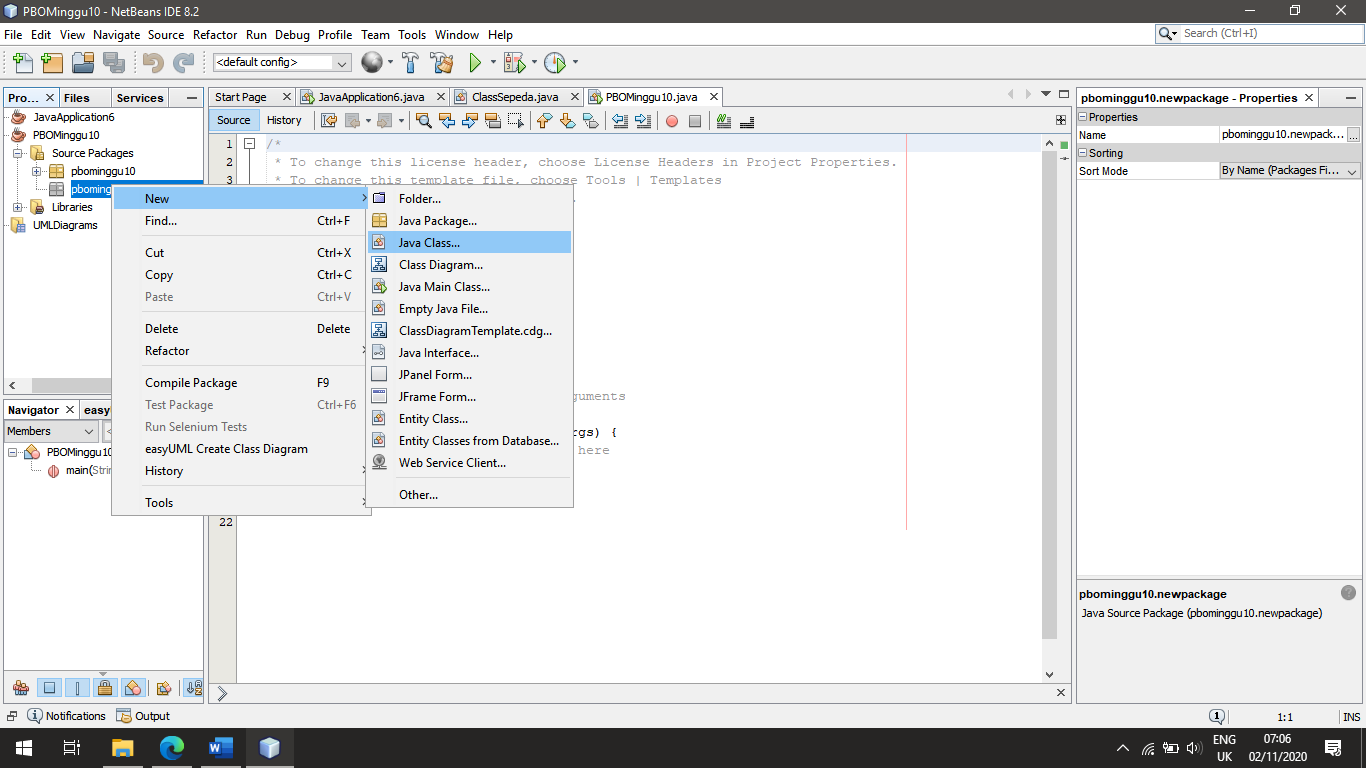
1. Pada package **pbominggu10**, tambahkan package baru dengan cara klik kanan nama package 🡪 New 🡪 Java Package…



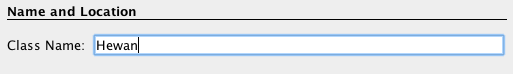
1. Beri nama package tersebut dengan nama **abstractclass**. Semua class yang dibuat pada percobaan 1 ini **diletakkan pada package yang sama**, yaitu *package abstractclass* ini,



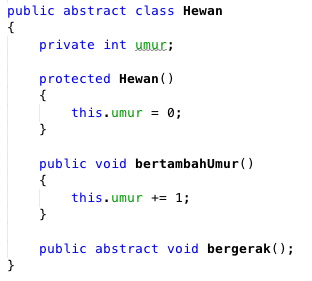
1. Pada package baru tersebut tambahkan *class* baru.



1. Beri nama *class* baru tersebut, yaitu class **Hewan**.



1. Pada *class* Hewan tersebut, ketikkan kode berikut ini.

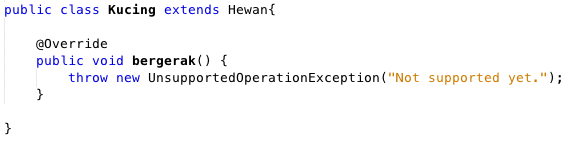


Class Hewan tersebut adalah class abstract berisi property dan method biasa, ditambah sebuah *method abstract* bernama **bergerak()**. *Method* tersebut didepannya terdapat kata kunci **abstract** dan tidak memiliki badan fungsi. Method ini nantinya akan di-*override* oleh *class* mana saja yang menjadi *class* turunan dari class Hewan tersebut.

1. Dengan cara yang sama, buatlah class dengan nama **Kucing** yang meng-*extend* class Hewan. Di dalam class Kucing tersebut, setelah Anda menuliskan kode seperti di bawah, maka akan muncul ikon lampu peringatan. Klik lampu tersebut dan kemudian pilih **implement all abstract methods**.



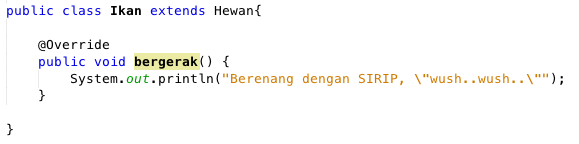
1. Maka akan secara otomatis dibuatkan fungsi yang meng-*override* fungsi *abstract* **bergerak()** yang ada pada class hewan.



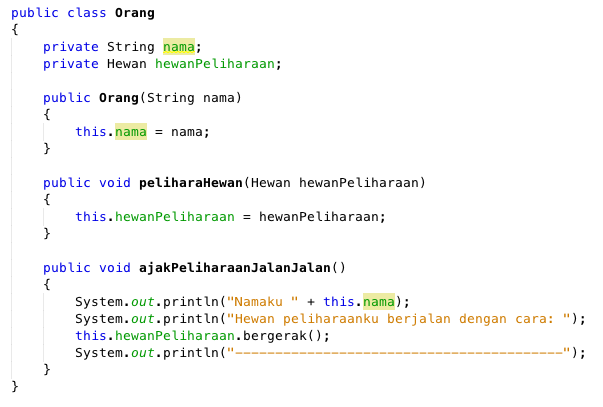
1. Ubahlah badan fungsi tersebut dengan mengganti kode didalamnya menjadi seperti berikut.



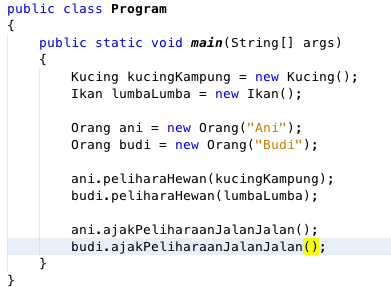
1. Dengan cara yang sama seperti ketika Anda membuat class Kucing, buatlah class Hewan baru bernama **Ikan** dan buatlah kodenya seperti pada gambar dibawah.



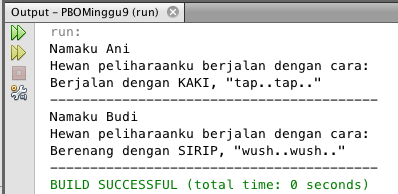
1. Selanjutnya, buatlah class biasa baru yang bernama class **Orang**. Class ini adalah class yang menjadi pengguna dari *class abstract* Hewan yang sudah dibuat sebelumnya. Ketikkan pada class Orang tersebut, baris-baris kode seperti di bawah.



1. Terakhir, buatlah sebuah *Main Class* baru di dalam *package* yang sama. Beri nama class baru tersebut dengan nama class **Program**. Ketikkan didalamnya seperti kode di bawah ini.



1. Jalankan class tersebut dengan cara klik kanan pada class Program kemudian pilih **Run File** (Shift + F6).
2. Perhatikan dan amati hasilnya!



1. Pertanyaan diskusi:

Bolehkah apabila sebuah class yang meng-*extend* suatu *abstract class* tidak mengimplementasikan *method abstract* yang ada di class induknya? Buktikan!

**Pertanyaan**

1. Berikan penjelasan terkait tentang jalannya program diatas
2. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika *method* **bergerak()** diubah menjadi *method abstract*!
3. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika tidak dilakukan *overriding* terhadap *method* **bergerak()**
4. Tunjukkan hasil kompilasi program dan berikan penjelasan singkat jika *abstract method* **bergerak()**yang dideklarasikan dalam Class Ikan