

Object

Dalam paradigma OOP, obyek merupakan *instance* atau wujud nyata dari suatu kelas. Pemahaman mengenai obyek tidak kalah pentingnya pada konsep pemrograman berorientasi obyek. Salah satu kegunaan obyek adalah untuk mengakses berbagai fungsi dan metode pada kelas. Obyek bisa dibuat lebih dari satu.

Menciptakan Obyek

Karena obyek merupakan instansiasi dari kelas, untuk menciptakan suatu obyek kita perlu mendeklarasikan tiap obyek yang dibuat dengan mengikutsertakan kelasnya. Pada Java untuk membuat sebuah obyek terdapat 3 tahap :

- Declaration : mendeklarasikan nama sebuah obyek
- Instantiation : memerlukan sebuah perintah `new` untuk menciptakan obyek
- Initialization : inialisasi dari sebuah obyek setelah perintah `new`

Codelab Mencipatakan Obyek

Pada contoh berikut kita praktikkan cara menciptakan suatu obyek secara sederhana di dalam satu kelas.

1. Buatlah proyek baru dengan nama Obyek dengan nama package `com.dicoding.javafundamental.obyek` di dalamnya:



2. Buatlah sebuah kelas baru dengan nama `Hewan`, kemudian tambahkan kode berikut:

```
1. package com.dicoding.javafundamental.obyek;
2.
3. public class Hewan {
4.     public void cetakNama(String nama) {
5.         System.out.println("Nama hewan: " + nama);
6.     }
7. }
```

3. Setelah itu, buatlah kelas baru dengan nama `Main` dan masukkan kode berikut di dalamnya:

```
1. public class Main {
2.     public static void main(String[] args) {
3.         // menciptakan object dengan nama 'obyekHewan'
4.         Hewan obyekHewan = new Hewan();
5.         obyekHewan.cetakNama("Elang");
6.     }
7. }
```

4. Selanjutnya jalankanlah kode di atas pada IDE yang kalian gunakan. Bila sukses, seharusnya Console akan menampilkan output seperti ini.

```
Nama hewan: Elang
```

Nah, sampai di sini apakah Anda sudah paham ? Jika belum mari kita bahas kodenya secara jelas.

Bedah Code Menciptakan Obyek

Pada kode di atas kita mempunyai sebuah kelas bernama Hewan. Di dalam *class* tersebut kita mempunyai metode `cetakNama` yang mempunyai parameter nama. Pada metode `cetakNama` kita hanya melakukan perintah menampilkan berdasarkan parameter yang dikirim dengan menggunakan tipe data String.

```
1. public void cetakNama(String nama) {
2.     System.out.println("Nama hewan: " + nama);
3. }
```

Pada kode selanjutnya kita menciptakan suatu obyek dengan perintah `new`. Sedangkan `obyekHewan` merupakan nama dari obyek yang kita buat.

```
1. Hewan obyekHewan = new Hewan();
```

Untuk memanggil sebuah metode di dalam suatu kelas, kita dapat menggunakan `(.)` (titik).

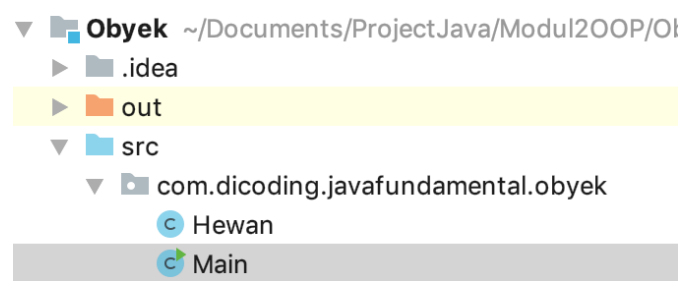
```
1. obyekNama.cetakNama("Elang");
```

Sampai di sini, kita telah mengerti bagaimana membuat obyek secara sederhana. Tahap selanjutnya kita akan mencoba membuat obyek pada sebuah kelas yang lebih kompleks.

Codelab Menggunakan Obyek

Selanjutnya kita akan coba menggunakan obyek yang ada di kelas `Hewan`.

1. Bukalah kembali proyek Obyek.



2. Masukkan kode berikut ke dalam kelas **Hewan** :

```
1. package com.dicoding.javafundamental.obyek;
2.
3. public class Hewan {
4.     String nama;
5.     int berat;
6.     int jumlahKaki;
7.
8.     // Ini merupakan konstruktor dari kelas hewan. konstruktor adalah method yang pertama kali dijalankan I
9.     public Hewan(String namaHewan) {
10.         nama = namaHewan;
11.     }
12.
13.     public void beratHewan(int beratHewan) {
14.         berat = beratHewan;
15.     }
```

Kelas **Hewan** memiliki beberapa atribut, seperti nama, berat dan jumlah kaki. Masing-masing dari atribut tersebut kita berikan parameter dengan membuat method.

3. Ubahlah kode yang ada di dalam kelas **Main** :

```
1. package com.dicoding.javafundamental.obyek;
2.
3. public class Main {
4.     public static void main(String[] args) {
5.         Hewan Elang = new Hewan("Elang");
6.         Hewan Kucing = new Hewan("Kucing");
7.
8.         Elang.beratHewan(5);
9.         Elang.jumlahKakiHewan(2);
10.        Elang.cetakHewan();
11.
12.        Kucing.beratHewan(2);
13.        Kucing.jumlahKakiHewan(4);
14.        Kucing.cetakHewan();
15.    }
```

4. Jalankan kode di atas maka hasilnya akan jadi seperti ini:

Nama hewan : Elang

Berat hewan : 5 kg

Jumlah kaki: 2

Nama hewan : Kucing

Berat hewan : 2 kg

Jumlah kaki: 4

Kini kita telah memahami cara membuat kelas dan menginstansiasinya pada Java. Ada yang perlu kita tekankan dalam bab ini . Dalam sebuah kelas kita bisa menciptakan beberapa obyek. Dengan menggunakan obyek kita dapat mengeksekusi baik metode maupun fungsi yang beraneka ragam, yang dipakai dalam sebuah kelas.

[< Sebelumnya](#)

[Selanjutnya >](#)



Dicoding Space

Jl. Batik Kumeli No.50, Sukaluyu,
Kec. Cibeunying Kaler, Kota Bandung
Jawa Barat 40123

Penghargaan

image
click bila
belum muncul

image
click bila
belum muncul

Decode Ideas
Discover Potential

[> Tentang Kami](#)

[Blog](#)

[Reward](#)

[Showcase](#)

[Hubungi Kami](#)

[FAQ](#)

© 2022 Dicoding | Dicoding adalah merek milik PT Presentologics, perusahaan induk dari PT Dicoding Akademi Indonesia.

[Terms](#) • [Privacy](#)