

[DISKUSIKAN MATERI](#)[LAPORKAN MATERI](#)

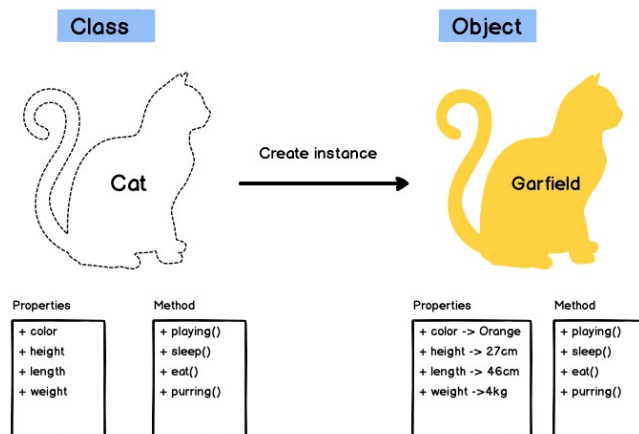
Daftar Modul

[Persetujuan Hak Cipta](#) ✓[Modul 1: Introduction Course \(Gratis\)](#) ✓[Prasyarat Kelas \(Gratis\)](#) ✓[Apa yang Akan Kita Pelajari? \(Gratis\)](#) ✓[Tools Requirement](#) ✓[Modul 2: ECMAScript 6 \(Gratis\)](#) ✓[Running Snippet Code \(Gratis\)](#) ✓[Persiapan Project Latihan \(Gratis\)](#) ✓

Class

Potongan kode pada materi ini: <https://repl.it/@dicodingacademy/163-02-class?lite=true>

Dalam paradigma *Object-Oriented Programming* (OOP), class merupakan sebuah blueprint yang dapat dikembangkan untuk membuat sebuah objek. *Blueprint* ini merupakan sebuah template yang di dalamnya menjelaskan seperti apa perilaku dari objek itu (berupa properti ataupun method).



Paradigma OOP selalu digambarkan dengan kehidupan nyata. Visualisasi di atas mencontohkan gambaran umum OOP di mana terdapat sebuah blueprint kucing, nilai yang dimiliki kucing, dan kemampuan yang dapat dilakukan olehnya. Dalam OOP blueprint tersebut dikenal dengan class (kelas), nilai yang dimiliki olehnya dikenal dengan properti, kemampuan yang dimilikinya dikenal sebagai *behaviour/method* dan realisasi dari sebuah blueprint tersebut disebut *instance*.

Dalam dunia pemrograman khususnya dalam paradigma OOP, kita sering membuat banyak objek dengan jenis yang sama. Dengan membuat blueprint ini kita dapat mengurangi duplikasi kode dalam membuat objek yang serupa.

[← KEMBALI KE MATERI SEBELUMNYA](#)[LANJUTKAN KE MATERI BERIKUTNYA →](#)[Image](#)
click here
to open modal[Video](#)
click here
to open modal

PERUSAHAAN

[Tentang Kami](#)[Blog](#)[Berita Terbaru](#)

PROGRAM

[Academy](#)[Challenge](#)[Event](#)[Job](#)[Rewards](#)

SUPPORT

[Bantuan](#)[FAQ](#)[Hubungi Kami](#)