

RPL A

## B. Object Oriented Programming (OOP)

1. Jelaskan kelebihan dan kekurangan pemrograman berorientasi objek?

## ▷ Kelebihan:

a. Paralell Development.

Pengembang dalam tim dapat membagi tugas dalam membangun class secara individu, untuk kemudian disatukan diakhir. Hal ini mendukung penghematan waktu.

b. Lebih mudah dalam mengelola sistem besaran kompleks.

c. Reusable

Class dapat digunakan kembali di project lainnya.

d. Mendukung enkapsulasi, abstraksi, pewarisan dan polimorfisme.

e. Kemudahan pengelolaan coding

## ▷ Kekurangan:

a. Kompleksitas dan kemungkinan penggunaan lebih memakan waktu untuk pengembang baru.

b. Overhead memori

c. Tidak selalu cocok untuk proyek kecil / skrip sederhana.



## 2. Sejarah OOP.

- Th. 1960an, mengembangkan small talk dengan konsep OOP yang dasar konsepnya seperti penyembunyian informasi dan pewarisan.
- 1970, Konsep awal muncul dari bahasa simula, dan berkembang melalui Smalltalk, program terdiri dari objek yang saling berinteraksi.
- 1980, Bahasa C++ dibangun dari hasil penambahan fitur OOP ke C. Bahasa seperti C++ mulai populer, membawa fitur OOP ke dunia nyata.
- 1990, Java lahir dengan filosofi OOP murni dan dukungan platform-independen. Peluncuran para diikuti Python & Perl.
- 2000an-hingga sekarang; OOP menjadi standar dominan di berbagai bahasa modern seperti C#, Python, Swift, & Kotlin.

## 3. Ilustrasi dalam dunia nyata.

- Kelas: cetakan atau blueprint. ex: "Mobil".
- Objek: Instansiasi dari kelas. ex: "Mobil Toyota Avanza milik Budi".
- Method: Fungsi yang dimiliki objek. ex: nyalakanMesin()
- Atribut: Properti objek, ex: warna, merk, atau kapasitas mesin.
- Interface: Kontrak perilaku, ex: "kendaraan" bisa memiliki bergerak() - yang wajib diimplementasikan oleh kelas seperti Mobil, Motor.

- Package, folder-logis yang mengelompokkan kelas-kelas.  
ex: kendaraan.mobil atau kendaraan.motor

4. Mengapa metodologi berorientasi objek berkembang lebih pesat?

= karena OOP:

- lebih cocok untuk sistem besar dan kompleks.
- mendukung prinsip desain perangkat lunak yang baik.
- meningkatkan skalabilitas dan pemeliharaan kode.
- lebih mudah dimodelkan dengan dunia nyata, sehingga intuitif untuk developer
- didukung oleh banyak tools modern, framework dan komunitas yang besar