

Catatan: Karena Jurusan yang saya ambil bukan programming maka kebanyakan dari jawaban akan berdasarkan pengalaman dan pembelajaran mandiri.

ESSAY

Jelaskan apa itu Inheritance, Polymorphism, dan Abstraction dalam Object Oriented Programming? Buatlah Rekaman Video Maksimal 3 Menit yang menjelaskan hal tersebut disertai contoh kodingan! (Live Coding adalah Opsional, Direkomendasikan untuk menggunakan Screen Recorder)

Inheritance adalah ketika sebuah **Class B** berasal dari **Class A**, ini berarti dia akan "inherit" setiap method dan property dari **Class A** tersebut.

Polymorphism adalah penggunaan inheritance dalam pengimplementasian method yang sama untuk beberapa class yang berasal dari satu class. Contohnya **Class Bentuk** dapat berupa dalam berbagai class seperti **Class Lingkaran**, **Class Segitiga**, **Class Kotak** yang semua berasal dari **Class Bentuk**.

Abstraction adalah pembuatan sebuah class tanpa menginginkan adanya instance yang terbuat, dalam arti lain class tersebut hanya sebuah fondasi atas class lain yang mengimplementasikan class tersebut. Dalam contoh tadi, **Class Bentuk** dapat menjadi abstrak sehingga kita tidak bisa membuat suatu bentuk baru, tetapi kita dapat membuat sebuah lingkaran dan mengetahui bahwa ia sebuah bentuk.

Jelaskan apa itu SOLID Principle?

SOLID principle adalah sebuah prinsip yang digunakan agar kode lebih mudah dipahami dan digunakan oleh kode lain (dalam OOP):

S - Single-responsibility Principle

Berarti sebuah kode hanya melakukan satu hal.

O - Open-closed Principle

Berarti sebuah kode dapat diperluas dengan Inheritance, Polymorphism dan Abstraction tetapi sebuah kode yang sudah ditetapkan sebaiknya tidak perlu disentuh kembali untuk menambahkan sesuatu.

L - Liskov Substitution Principle

Berarti jika ada **Class A** dan **Class B : Class A**, maka segala hal yang dapat dilakukan pada **Class A** harus dapat dilakukan pada **Class B** juga.

I - Interface Segregation Principle

Berarti jika kita memiliki **Class A** dan **Class B : Class A**, dan **Class B** tidak perlu mengimplementasikan sesuatu yang tidak dibutuhkan, jika tidak berarti **Class A** dapat dipisah menjadi class lain.

D - Dependency Inversion Principle

Berarti sebisa mungkin sebuah kode akan menggunakan abstract class atau inheritance dan polymorphism untuk mengimplementasikan penggunaan ulang kode sehingga fungsi setiap kode dapat dijamin dan mudah dipahami karena melakukan hal yang sama.

Jelaskan apa itu Singleton, Object Pooling, dan Observer dalam Design Pattern? Sertakan Masing-masing Contoh Kasus Terbaik dalam bentuk Kodingan dengan menggunakan Unity Game Engine!

Singleton

adalah sebuah design pattern di mana suatu class hanya memiliki satu instance. Ini berarti kita tidak ingin ada dua hal yang sama. Sebuah singleton biasanya akan menjadi instance yang mengatur kejadian apa saja yang terjadi dan dalam urutan apa dengan memanggil method dari instance lain. Hal ini dapat dilihat pada KASUS [InstancePool.cs](#) dan juga [GameManager.cs](#)

Object Pooling

adalah sebuah design pattern untuk menggunakan reference yang sudah ada, lebih biasanya dalam Unity untuk instanciate sebuah benda. Karena setiap pembuatan instance MonoBehaviour object pada Unity cukup berat, kita tidak akan menghancurkan sebuah benda saat ia meninggalkan Scene melainkan kita menyembunyikannya dan memperlihatkannya kembali Ketika dibutuhkan. Hal ini dapat dilihat pada KASUS [InstancePool.cs](#)

Observer

adalah sebuah design pattern di mana kode lain akan memberikan methodnya kepada sebuah state yang akan berubah. Pada perubahan state tersebut method itu akan dipanggil sehingga kode tersebut tidak perlu memeriksa setiap frame atas state tersebut. Hal ini dapat dilihat pada [Damageable.cs](#)