

Nama : Haikal Fadhilah Mufid

NIM : 2311104027

MENJELASKAN DESIGN PATTERN SINGLETON

A. Berikan salah dua contoh kondisi dimana design pattern “Singleton” dapat digunakan

1. Manajemen Koneksi ke Database

Dalam sebuah aplikasi, sering kali hanya diperlukan satu instance objek koneksi ke database agar resource efisien dan konsisten. Singleton memastikan hanya satu objek koneksi yang digunakan selama aplikasi berjalan, menghindari overhead karena pembuatan koneksi baru berulang kali.

2. Pengaturan Aplikasi (Configuration Settings)

Saat aplikasi membutuhkan akses ke konfigurasi global (seperti setting dari file .env atau .json), Singleton bisa digunakan untuk memuat dan menyimpan konfigurasi tersebut satu kali saja, dan digunakan ulang di seluruh bagian aplikasi.

B. Berikan penjelasan singkat mengenai langkah-langkah dalam mengimplementasikan design pattern “Singleton”.

Langkah-langkah singkat mengimplementasikan design pattern **Singleton**:

- 1. Buat konstruktor sebagai private** agar objek tidak bisa dibuat langsung dari luar kelas.
- 2. Buat static instance** di dalam kelas untuk menyimpan satu-satunya objek.
- 3. Sediakan method static (biasanya getInstance)** untuk mengakses instance tersebut. Jika instance belum ada, buat dulu; jika sudah, kembalikan instance yang sama.

C. Berikan tiga kelebihan dan kekurangan dari design pattern “Singleton”.

Kelebihan:

- 1. Menghemat resource** karena hanya ada satu instance yang digunakan bersama.
- 2. Kontrol akses global** ke instance yang sama di seluruh program.
- 3. Mudah untuk logging dan konfigurasi bersama**, karena semua bagian sistem mengakses objek yang sama.

Kekurangan:

1. **Sulit untuk diuji** karena ketergantungan terhadap instance global membuat pengujian unit menjadi lebih rumit.
2. **Melanggar prinsip Single Responsibility**, karena kelas mengelola logika dan kontrol instance sekaligus.
3. **Potensi masalah concurrency**, terutama di lingkungan multi-thread jika tidak diimplementasikan dengan benar.

IMPLEMENTASI DAN PEMAHAMAN DESIGN PATTERN SINGLETON

```
const Singleton = (function () {  
  let instance;  
  function create() {  
    return { name: "Only One" };  
  }  
  return {  
    getInstance: function () {  
      if (!instance) instance = create();  
      return instance;  
    }  
  };  
})();  
  
const instance1 = Singleton.getInstance();  
console.log(instance1);
```

Outputnya :

```
PS D:\HAIKAL DISK\Kuliah'  
{ name: 'Only One' }  
PS D:\HAIKAL DISK\Kuliah'
```