

LAPORAN AKHIR PRAKTIKUM

Mata Praktikum : Rekayasa Perangkat Lunak 2
Kelas : 4IA11
Praktikum ke- 1
Tanggal : 18 Oktober 2024
Materi : Desain Perangkat Lunak Berbasis Komponen, Pettern, dan Skala
NPM : 50421786
Nama : Marsillo Dito Saputra
Ketua Asisten :
Paraf Asisten :
Nama Asisten : Dariwan
Jumlah Lembar : Lembar

LABORATORIUM TEKNIK INFORMATIKA

UNIVERSITAS GUNADARMA

2024

LISTING PROGRAM

(Jawab Soal Untuk Laporan Akhir)

LA

1. apa kekurangan dari desain pattern?
2. apa contoh software ERP?
3. sebutkan dan jelaskan macam macam design pattern dan sertakan gambar dari desain pattern itu!

Jawab :

1. Desain pattern sebenarnya punya banyak kelebihan, tapi ada juga beberapa kekurangan, seperti:
 - **Overengineering:** Kadang, kalau tidak digunakan dengan tepat, bisa bikin kode jadi lebih rumit daripada yang seharusnya. Misalnya, untuk masalah yang simpel, kita malah menggunakan pola desain yang terlalu kompleks.
 - **Steep Learning Curve:** Desain pattern mungkin sulit dipahami untuk pemula. Butuh waktu dan pengalaman untuk memahami kapan dan bagaimana menggunakan pola yang tepat.
 - **Tidak Fleksibel untuk Semua Situasi:** Meskipun pola desain menawarkan solusi umum untuk masalah umum, mereka tidak selalu cocok untuk semua situasi. Kadang, kita perlu mengubah atau mengadaptasi pola agar sesuai dengan kebutuhan.
2. Beberapa contoh software ERP yang terkenal adalah:
 - **SAP:** Salah satu ERP yang paling banyak digunakan di seluruh dunia, terutama di perusahaan besar.
 - **Oracle ERP:** Ini juga populer di perusahaan besar, terutama karena skalabilitas dan dukungan cloud-nya.

- **Microsoft Dynamics:** ERP dari Microsoft yang cocok untuk bisnis kecil hingga besar, dengan integrasi yang bagus ke produk Microsoft lainnya seperti Office 365.
- **Odoo:** ERP yang lebih fleksibel dan sering digunakan oleh usaha kecil hingga menengah karena sifatnya yang open-source.

3. Desain pattern dibagi menjadi tiga kategori utama:

a. Creational Patterns (Pola Kreasi)

Pola ini fokus pada cara menciptakan objek dengan cara yang lebih fleksibel dan reusable.

- **Singleton:** Pola ini memastikan bahwa hanya ada satu instance dari kelas tertentu yang bisa dibuat. Contohnya adalah saat kita ingin hanya ada satu koneksi ke database.
- **Factory Method:** Digunakan untuk membuat objek tanpa harus menyebutkan tipe objek secara eksplisit. Misalnya, jika kita punya berbagai jenis kendaraan (mobil, motor, dll.), kita bisa menggunakan Factory untuk membuat kendaraan tanpa menentukan jenisnya secara langsung.

b. Structural Patterns (Pola Struktural)

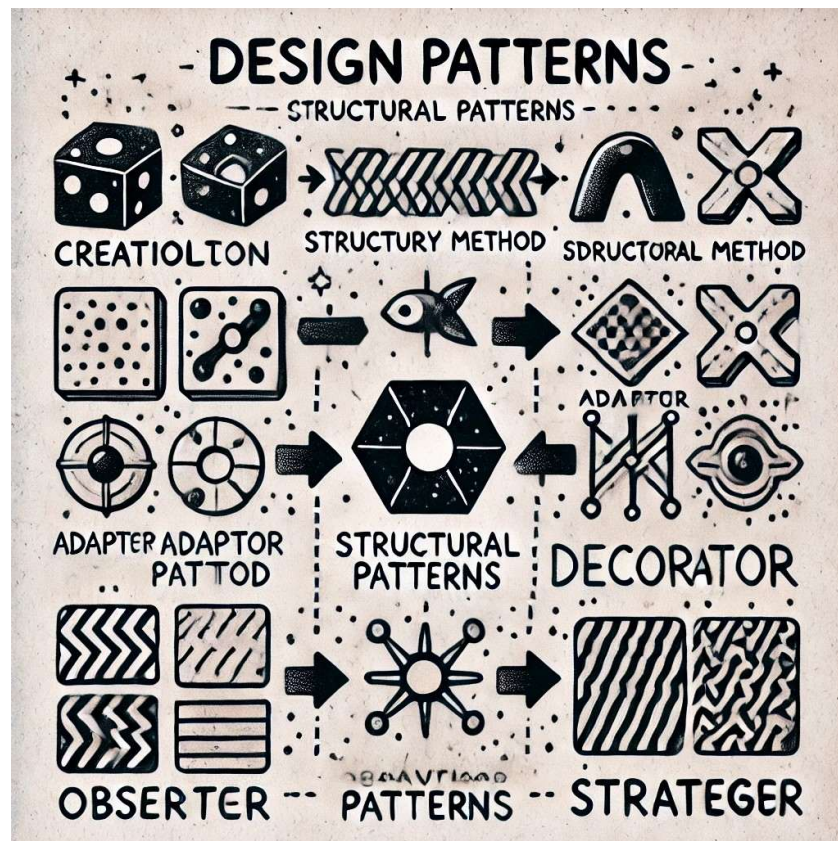
Pola ini membantu mengatur hubungan antar kelas dan objek agar bisa bekerja bersama.

- **Adapter:** Pola ini memungkinkan dua interface yang tidak kompatibel untuk bekerja sama. Misalnya, kita punya class lama dan ingin menghubungkannya dengan sistem baru, kita bisa menggunakan adapter.
- **Decorator:** Pola ini menambahkan fungsionalitas ke objek tanpa mengubah strukturnya. Misalnya, menambahkan fitur keamanan ke sistem yang sudah ada.

c. Behavioral Patterns (Pola Perilaku)

Pola ini berhubungan dengan cara objek berinteraksi dan bagaimana mereka bertukar pesan.

- **Observer:** Digunakan ketika satu objek harus memberi tahu objek lain tentang perubahan yang terjadi. Contoh sederhananya adalah ketika kamu berlangganan ke berita, dan setiap kali ada berita baru, kamu diberi tahu.
- **Strategy:** Pola ini memungkinkan kita untuk memilih algoritma yang akan digunakan pada saat runtime. Misalnya, kita punya berbagai cara untuk melakukan sorting, dan kita bisa memilih cara yang paling efisien di runtime.



Berikut adalah gambar diagram yang menunjukkan tiga kategori design pattern, yaitu: Creational Patterns, Structural Patterns, dan Behavioral Patterns, beserta contoh-contohnya. Diagram ini menunjukkan hubungan antar kategori dan pola yang ada di dalam

