

Nama	Ronald Arsyad
Kelas	GT-A 10
NRP	5221600028

## Laporan Praktikum 08

Soal :

Coba dan pelajari contoh program pada Materi Teori 7: Command Pattern

Buat summary dari apa yg kalian pahami tentang command pattern dan penjelasan dari contoh program tersebut

Jawab :

Command Pattern adalah pola desain yang memungkinkan pemisahan antara pemanggil perintah (Invoker) dan penerima perintah (Receiver) dengan menggunakan objek perintah (Command) sebagai perantara. Ini memungkinkan fleksibilitas, skalabilitas, dan keterbacaan kode yang lebih baik dengan mengkapsulasi permintaan sebagai objek.

Dalam contoh program yang diberikan:

- **Command (Perintah):** Interface Command adalah kerangka yang didefinisikan oleh metode `Execute()`, yang semua perintah harus mengimplementasikannya. Ini memungkinkan setiap perintah, baik sederhana maupun kompleks, untuk dijalankan dengan cara yang seragam.
- **Concrete Command (Perintah Khusus):** Kelas `SimpleCommand` dan `ComplexCommand` adalah implementasi konkret dari antarmuka Command. `SimpleCommand` melakukan tindakan sederhana seperti mencetak pesan, sementara `ComplexCommand` melakukan tindakan lebih kompleks dengan menggunakan objek Receiver.
- **Receiver (Penerima):** Kelas Receiver berisi logika bisnis yang sebenarnya. Dalam contoh ini, itu memiliki metode `DoSomething()` dan `DoSomethingElse()` yang akan dipanggil oleh `ComplexCommand`.
- **Invoker (Pengundang):** Kelas Invoker bertanggung jawab untuk mengirimkan permintaan ke objek Command. Dalam contoh ini, Invoker dapat mengatur perintah sebelum dan sesudah menjalankan tindakan pentingnya.

Dengan Command Pattern, setiap elemen memiliki tanggung jawab yang jelas: Invoker bertindak sebagai pengatur tingkat tinggi yang tidak perlu tahu detail implementasi perintah, Command sebagai perantara yang memungkinkan pemanggilan tanpa harus bergantung pada penerima, dan Receiver sebagai pemegang logika bisnis sebenarnya. Ini memungkinkan fleksibilitas dalam menambah, menghapus, atau mengganti perintah tanpa mempengaruhi kode klien. Semua ini membuat aplikasi lebih modular, mudah dipelihara, dan mudah diperluas.