Software Design Pattern

2 Tiga Pola Desain

Pola Desain

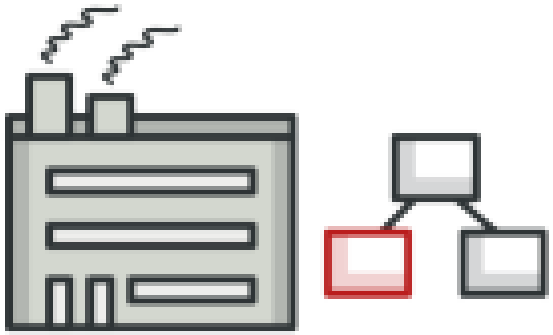
Terdapat tiga pola utama yang dapat digunakan

- Creational Pattern (Pola Pembuatan)
- Behavioral Pattern (Pola Perilaku)
- Structural Patternr (Pola Struktural)

Creational Pattern

- Creational Pattern menyediakan berbagai mekanisme pembuatan objek, sehingga meningkatkan fleksibilitas dan penggunaan Kembali kode yang ada
- Creational pattern dapat dibagi menjadi lima yaitu : Factory Method, Abstract Factory,Bulder, Prototype dan Singleton,

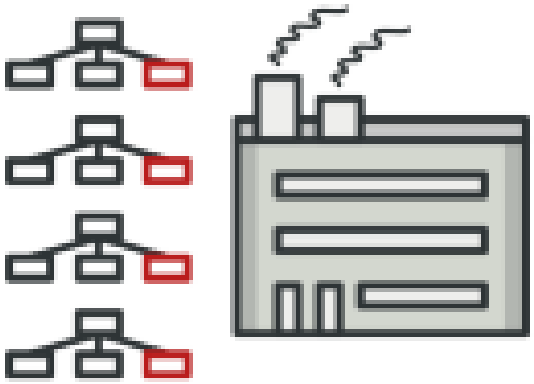
Creational Pattern



Factory Method

Menyediakan antarmuka untuk membuat objek di super class, tetapi memungkinkan subclass untuk mengubah jenis objek yang akan dibuat

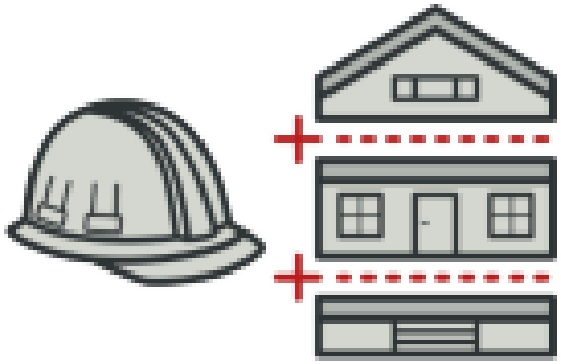
Creational Pattern



Abstract Factory

Pola ini hanya dapat mendefinisikan interface atau abstract class untuk membuat suatu objek tanpa harus menentukan sub-classnya.

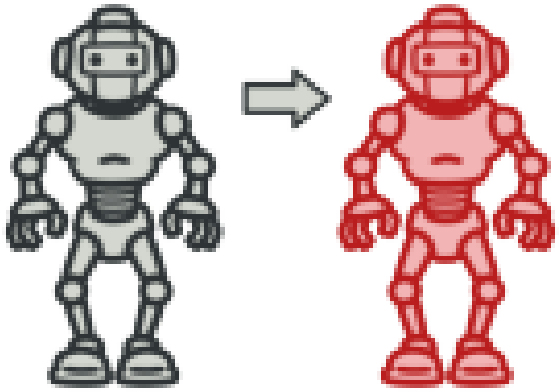
Creational Pattern



Builder

Builder adalah creational pattern yang membuat sebuah objek yang kompleks dari objek yang sederhana menggunakan pendekatan yang dilakukan secara bertahap (step-by-step)

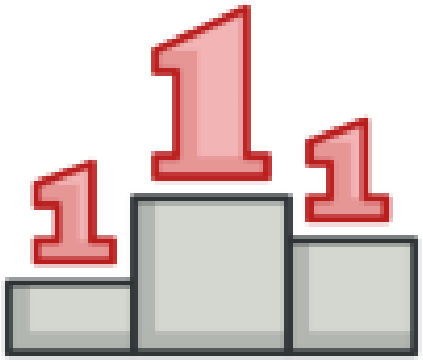
Creational Pattern



Prototype

Pola ini memungkinkan kamu untuk menyalin objek menggunakan instance prototype yang ada dengan membuat objek yang baru dan dapat dikustomisasi sesuai dengan kebutuhan tanpa bergantung dengan objek aslinya.

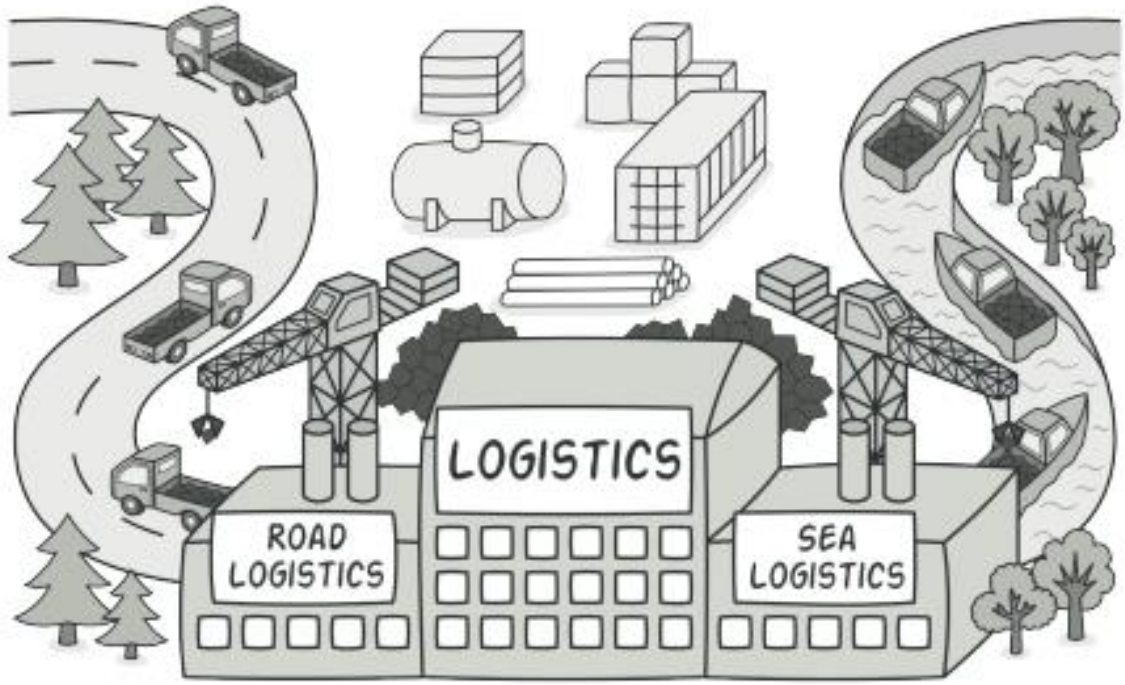
Creational Pattern



Singleton

Singleton ini memastikan suatu class hanya memiliki satu instance dan menyediakan global access point ke instance tersebut.

Creational Pattern



FACTORY METHOD

Factory Method

- Factory Method adalah pola desain kreasi yang menyediakan antarmuka untuk membuat objek dalam super class, tetapi memungkinkan subclass untuk mengubah jenis objek yang akan dibuat

Permasalahan

- Programmer membuat aplikasi manajemen logistic
- Versi pertama aplikasi hanya dapat menangani transportasi dengan truk
- Sebagian kode berada di dalam class truk.

Permasalahan

- Kemudian aplikasi yang dibuat programmer tersebut menjadi popular
- Setiap hari programmer tersebut menerima puluhan permintaan dari perusahaan transportasi laut untuk memasukan logistic ke dalam aplikasi

Permasalahan

Permasalahan

- Apakah sebuah Berita bagus?
- Bagus…bagai mana dengan kodenya digabungkan ke kelas truk?
- Menambah kapal ke aplikasi akan membutuhkan perubahan pada kode

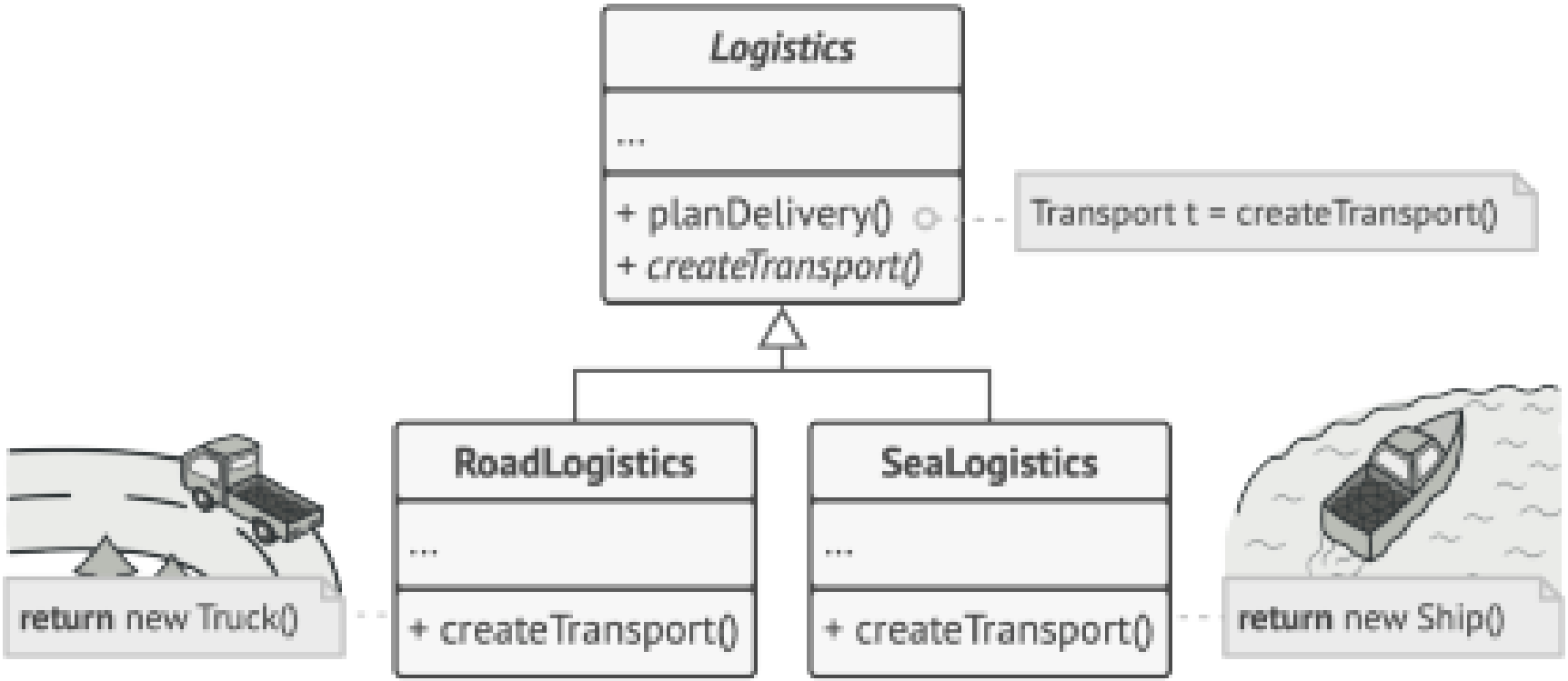
Permasalahan

- Jika memutuskan menambahkan jenis transportasi lain ke aplikasi, programmer perlu membuat banyak perubahan
- Akibatnya programmer akan berakhir dengan kode yang tidak bagus, penuh dengan kondisional yang mengubah perilaku aplikasi tergantung pada kelas objek tranportasi

Solusi

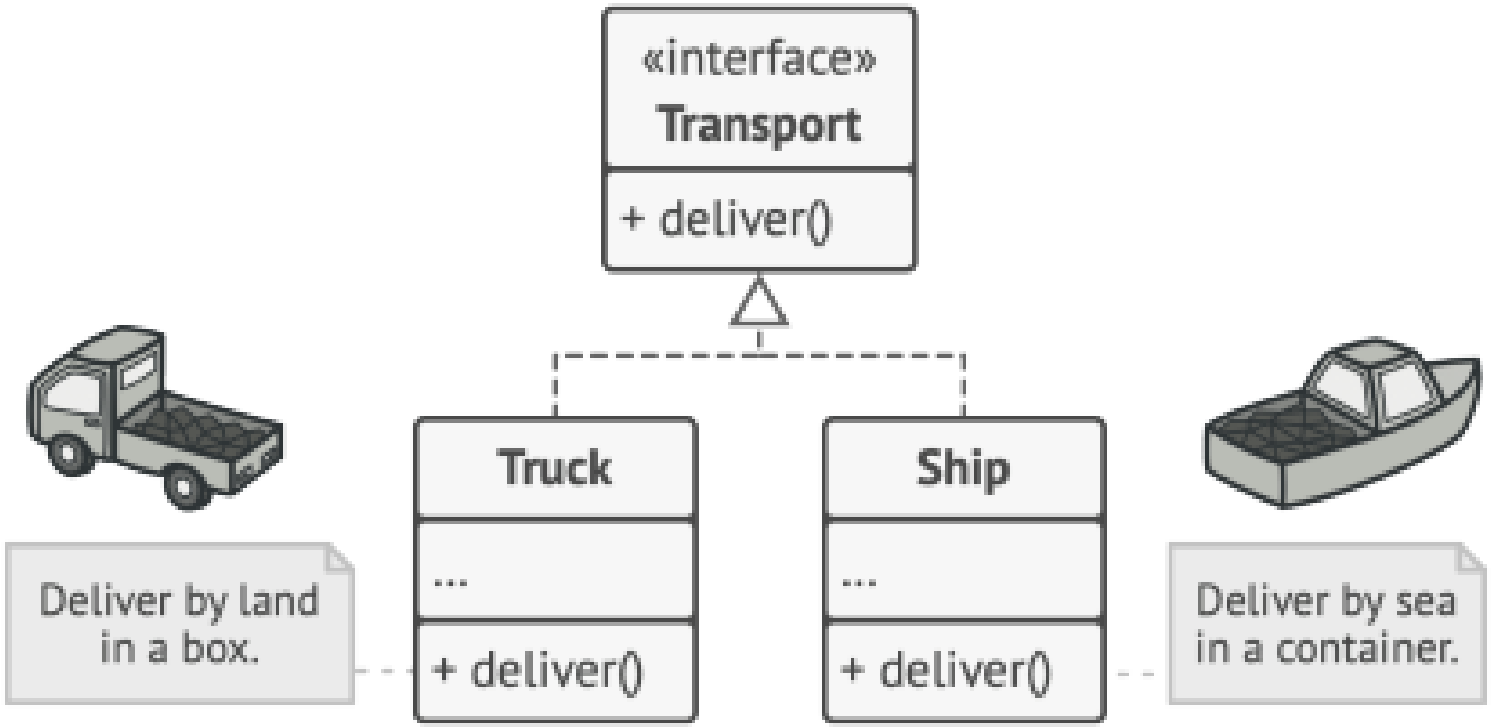
- Pola factory method programmer mengganti langsung panggilan kontruksi objek (menggunakan operator new)
- Dengan panggilan ke factory method khusus
- Objek dibuat melalui operator new, tetapi dipanggil dari dalam factory method.
- Objek yang dikembalikan dengan factory method disebut dengan produk

Solusi



Subclass dapat mengubah kelas objek yang dikembalikan oleh factory method

Solusi

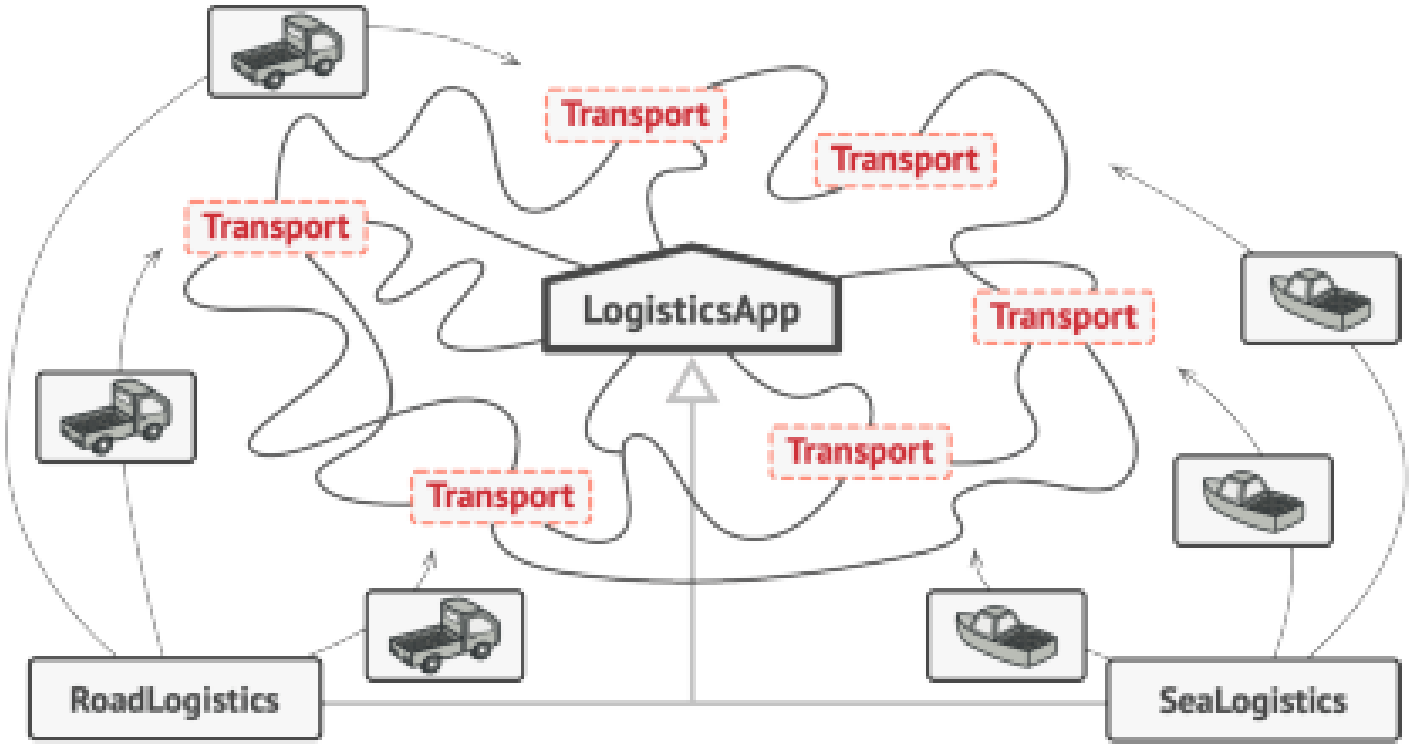


Semua produk harus mengikuti antarmuka yang sama.

Solusi

- Mis:kelas Truk dan Kapal harus mengimplementasikan tranport antarmuka yang menyatakan suatu metod deliver
- Setiap class menerapkan metode secara berbeda
- Truk mengirimkann kargo melalui darat
- Kapal mengirimkan kargo dari laut
- Factory method di class RoadLogistics mengembalikan truk objek
- Sedangkan factory method di Sealogistics clss mengembalikan kapal

Solusi



Selama semua kelas produk menerapkan antarmuka umum, programmer dapat meneruskan objek mereka ke kode klien tanpa merusaknya

TO …… BE ….. CONTINUE