



Design Pattern

Suatu pengantar

Apa itu Design Pattern

Design Pattern adalah penggambaran bagaimana objek dan class saling berkomunikasi yang disesuaikan untuk memecahkan problem design secara umum dalam konteks tertentu

Design Pattern memberi gambaran, mengabstraksi dan mengidentifikasi aspek2 penting dalam design sehingga dapat tercipta reusable OO design

Not code reuse

- Instead, solution/strategy reuse
- Sometimes, interface reuse

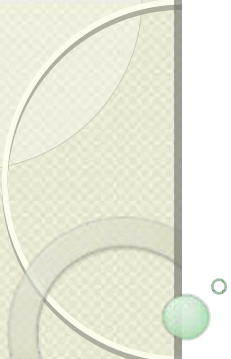
Apa itu Design Pattern

Empat elemen utama dalam pattern:

1. Nama pattern: penggambaran problem, solusi dan konsekwensinya
2. Problem: waktu yang tepat menggunakan suatu pattern
3. Solusi: Elemen yang menghasilkan design, hubungan, responsibilities dan kolaborasi
4. Konsekwensi: keuntungan dan kerugian menerapkan suatu pattern; space, time, fleksibilitas, extensibilitas dan portabilitas

Klasifikasi Design Pattern

- Berdasarkan tujuan class/objek, design pattern diklasifikasikan ke dalam:
 - Creational
 - Structural
 - Behavioral
- Berdasarkan scopenya, menentukan apakah pattern diaplikasikan untuk class atau objek



		Purpose		
		Creational (process of object creation)	Structural (composition of classes or objects)	Behavioral (how classes interacts and distribute responsibility)
Scope	Class (relationship between classes & subclasses, fixed at compile time)	Factory Method	Adapter	Interpreter Template Method
	Object (can be changed at run time and more dynamic)	Abstract Factory Builder Prototype Singleton	Adapter Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor



Bagaimana Memilih Design Pattern

- Pikirkan bagaimana Design Pattern memecahkan Design Problem
- Pelajari tujuan design pattern
- Pelajari hubungan antar pattern
- Pelajari patterns dengan tujuan yang serupa
- Telaah penyebab redesign
- Pikirkan apa yang menjadi variable dari suatu design



Creational Design Pattern

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton



Builder Design Pattern

- Tujuan:
memisahkan konstruksi dari objek kompleks dari representasi kelasnya sehingga proses konstruksi yang sama dapat digunakan untuk berbagai macam jenis representasi
- Konteks:
 - Ketika algoritma untuk menkonstruksi objek kompleks harus independen terhadap bagian-bagian apa saja yang menyusun dan bagaimana bagian-bagian tersebut tersusun
 - Proses konstruksi harus memungkinkan adanya berbagai jenis representasi objek yang berbeda-beda

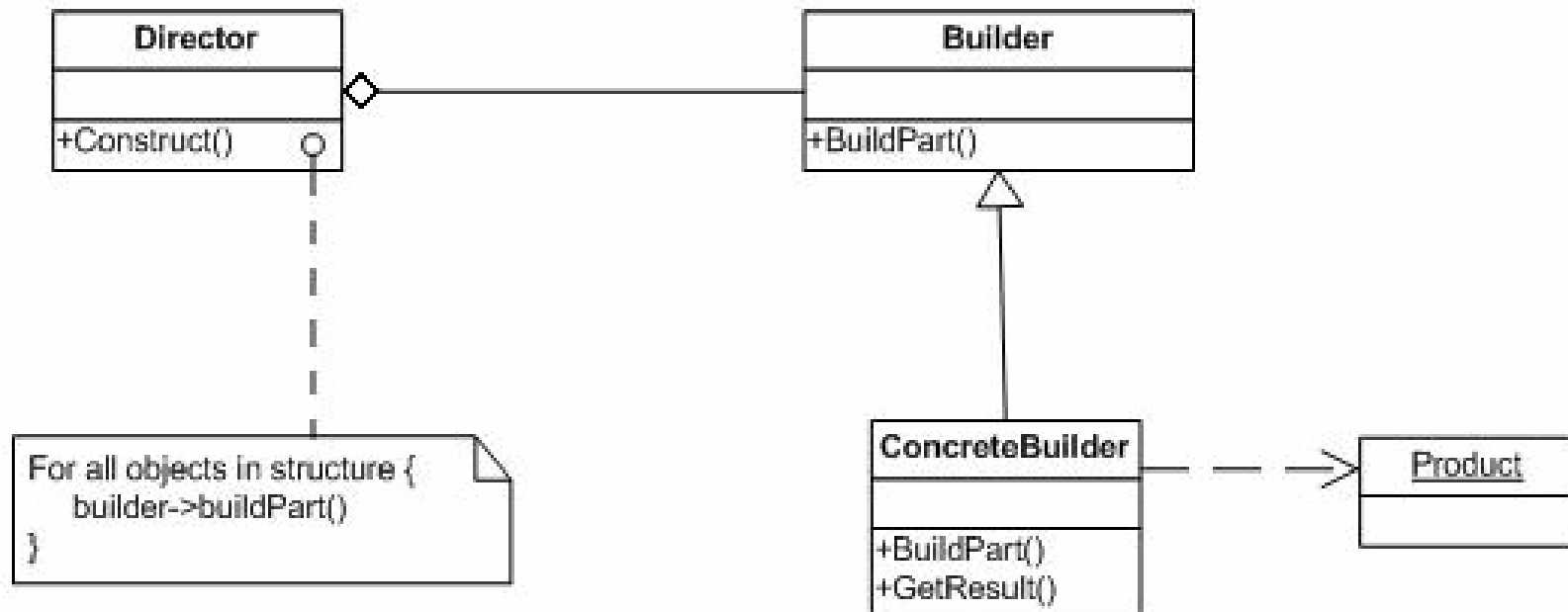


Builder Design Pattern

- Actors:
 - Builder
 - ConcreteBuilder
 - Director
 - Product

Builder Design Pattern

- Structure





Builder Design Pattern

- Implementation:
 - See Code !



Factory Method DP

- Tujuan:

Menyediakan pembuatan interface untuk objek, tetapi mendelegasikan instantiasi kelas kepada subclassnya.

- Konteks:

- Ketika sebuah kelas perlu membuat objek yang lain, tetapi tidak dapat mengetahui kelas dari objek yang akan dibuatnya
- Sebuah kelas ingin agar subclass-nya yang menentukan objek yang akan dibuat
- Sebuah kelas ingin mendelegasikan

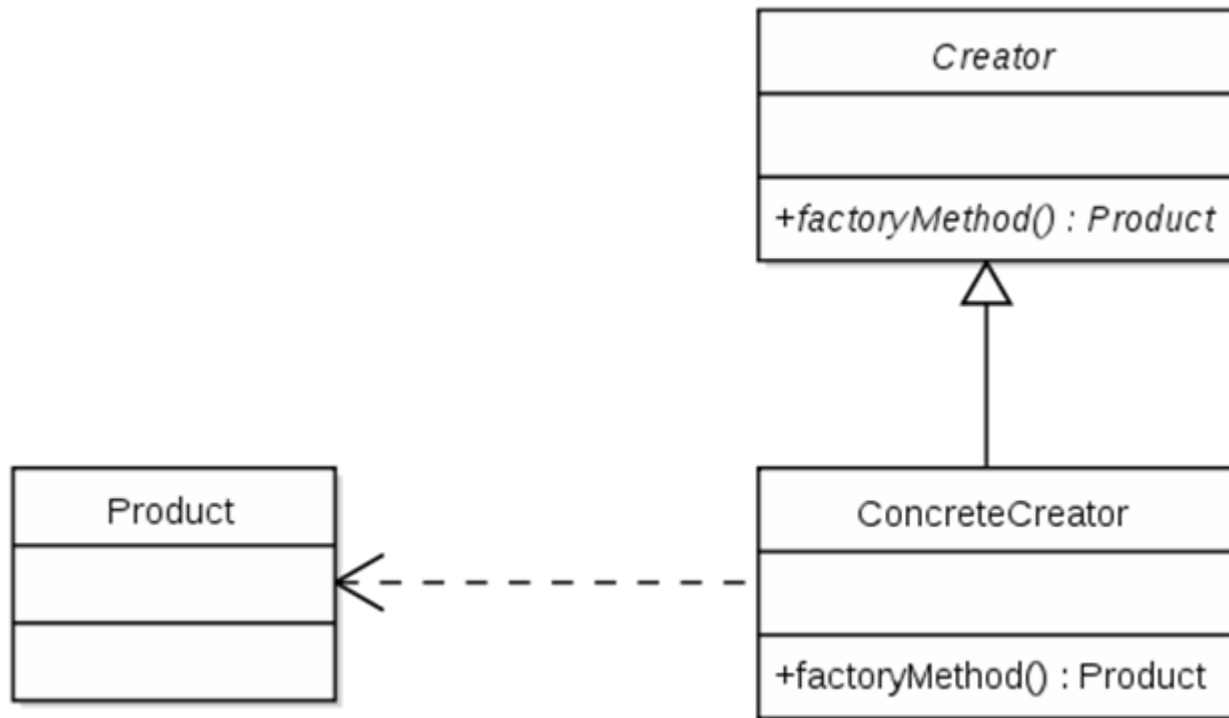


Factory Method DP

- Actors:
 - Product
 - ConcreteProduct
 - Creator
 - ConcreteCreator

Factory Method DP

- Structure





Factory Method DP

- Implementation:
 - See Code !



Creational Design Pattern

- Masih ada beberapa Creational DP yang lain:
 - Abstract Factory
 - Prototype
 - Singleton
 - Lazy Initialization
 - Object pool
 - dll...



Tugas

- Buat review singkat mencakup:
 - Tujuan/motivasi
 - Konteks
 - Struktur
 - Contoh Implementasi
- dari masing-masing DP Creational yang belum disebutkan!