

Pola Desain Perangkat Lunak (PDPL)



[Minggu 1] - 17 Februari 2020
00 Pengantar PDPL

Prepared by Tifanny Nabarian (nabarian@gmail.com)
Tribute to Hilmi A.T. for magnificent previous lecture slides

Profil Dosen



Pendidikan

S1 – Teknik Informatika UNSRI

S2 – Magister Teknologi Informasi UI

085267393710

nabarian@gmail.com

Pengalaman Kerja



PT. BANK NEGARA INDONESIA (Persero) Tbk.
5 thn 4 bln

- **Channel System & SOA Analyst**
Nov 2015 – Feb 2018 · 2 thn 4 bln
IT Solution & Security Division, DKI Jakarta - Indonesia

- **Switching System & Middleware Programmer**
Mar 2015 – Okt 2015 · 8 bln
IT Division, DKI Jakarta - Indonesia

- **Channel & Connectivity Programmer**
Nov 2013 – Feb 2015 · 1 thn 4 bln
IT Division, DKI Jakarta - Indonesia

- **ODP Trainee Batch 108, Specific Banking**
Nov 2012 – Okt 2013 · 1 thn
BNI Corporate University, DKI Jakarta - Indonesia

Satuan Acara Perkuliahan



Mata Kuliah : Pola Desain Perangkat Lunak

Kode : TI32012

Jumlah SKS : 3 SKS

Rencana Perkuliahan

Tatap Muka

Pemaparan Materi

Tanya Jawab

Diskusi

Tugas

Presentasi

Evaluasi:

UTS : 25%

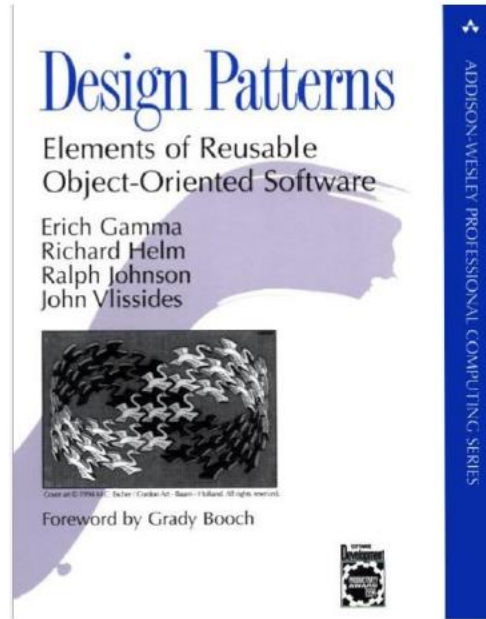
UAS : 30%

Quiz : 10%

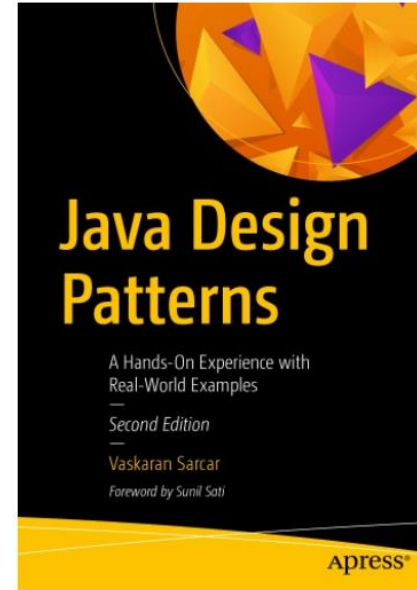
Tugas : 30%

Kehadiran/Keaktifan : 5%

References



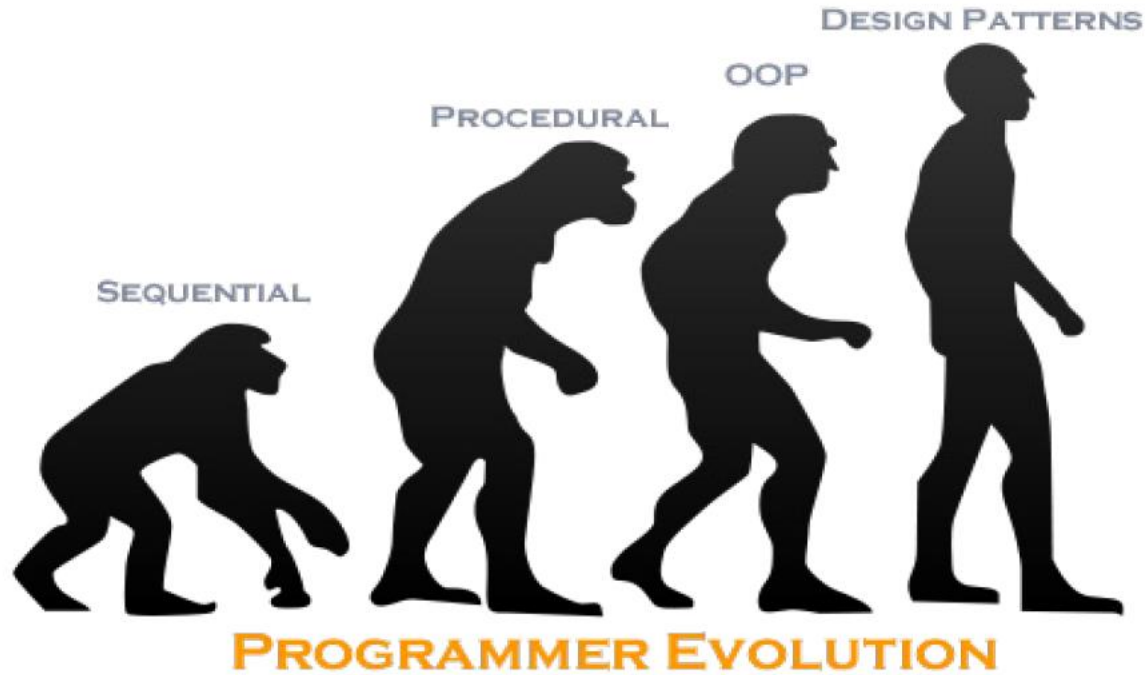
Erich Gamma, Design Pattern Elemen of Reusable Object Oriented Software, Addison-Wesley, 1994



Sarcar, Verkaran. Java Design Patterns: A Hands-On Experience with Real-World Examples, Apress, 2019



Latar Belakang



Latar Belakang (1)



What's wrong with sequential & procedural programming?

```
<?php
```

```
$firstNumber=20;  
$secondNumber=40;  
$total= $firstNumber + $secondNumber;  
echo $total;
```

```
?>
```

```
<?php
```

```
function addEmUp($first,$second)  
{  
    $total=$first + $second;  
    echo $total;  
}  
addEmUp(20,40);
```

```
?>
```

Latar Belakang (2)

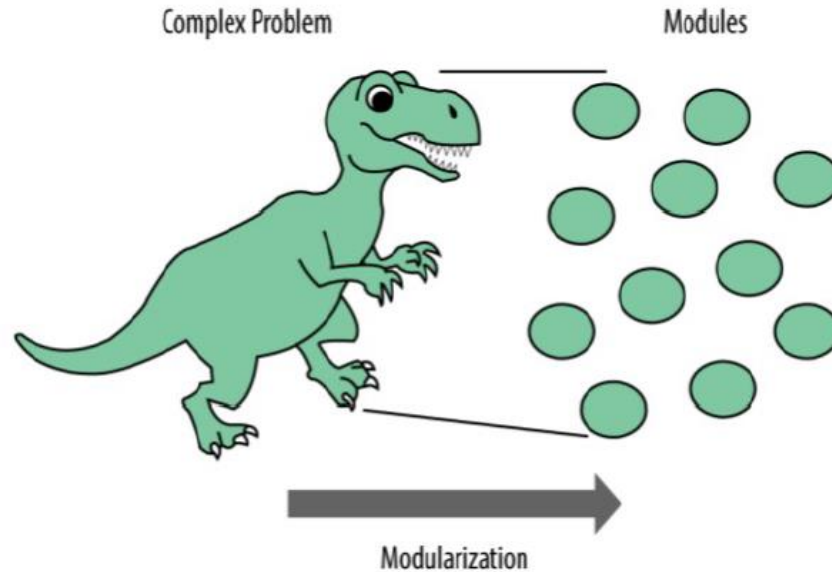


What's wrong with sequential & procedural programming?

- As programmers began to write longer and longer programs with more complex tasks, the sequences began to be entangled into what was called spaghetti code.
- Like OOP, procedural programming uses modularity and reuse. However, procedural programming does not provide for classes where programming tasks can be bundled into objects. Class objects (instances of classes) can operate on their own data structures, and that cannot be done by functions alone.

Latar Belakang (3)

Why Object Oriented Programming?



Latar Belakang (4)



- Designing object-oriented software is **hard**
- Designing reusable object-oriented software is even **harder**
- Your design should be specific to the problem at hand but also general enough to address future problems and requirements





Tujuan Mata Kuliah

“Mata kuliah ini dimaksudkan untuk memberi kemampuan pada mahasiswa untuk dapat memahami **pola desain** serta **arsitektur** dari **struktur kode program** pada pengembangan ***software***.”



Design Thinking



- Memecahkan masalah menggunakan pendekatan desain.
- Melihat masalah secara komprehensif (keseluruhan)
- Mengambil keputusan yang sesuai untuk masalah yang dialami



What is Design Patterns?

“descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.”

Why Design Patterns?



Why Design Patterns?

- Experienced designers reuse solutions that have worked in the past
- Well-structured object-oriented systems have recurring patterns of classes and objects
- Knowledge of the patterns that have worked in the past allows a designer to be more productive and the resulting designs to be more flexible and reusable

Manfaat



Thank You!

*Subhaanakallohumma wa bihamdika,
asy-hadu alla ilaha illa anta, as-tagh-
firuka wa atuubu ilaik*

