

PERAWATAN PERANGKAT LUNAK

Resume code Refactor



Disusun oleh :

RIFQI ROSYID ERDIANSYAH

E41192306

TEKNIK INFORMATIKA 2019

GOLONGAN D

PROGRAM STUDI TEKNIK INFORMATIKA

KAMPUS JEMBER

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2021

Pengertian

Refactoring merupakan salah satu kegiatan untuk memperbaiki atau membangun kembali struktur kode, dengan cara menghapus baris kode yang tidak terpakai, serta mencegah baris code yang berulang-ulang (boilerplate), agar perancangan code yang telah dibangun dapat lebih jelas dipahami. Refactoring menurut Martin Fowler dan Kent Beck adalah perubahan yang dilakukan pada struktur internal dari sebuah software agar lebih mudah dipahami dan lebih murah untuk dimodifikasi tanpa mengubah tingkah laku kode tersebut. Ini adalah cara disiplin untuk membersihkan kode, serta meminimalkan kemungkinan terjadinya bug pada program.

Dalam software development, seringkali kita membuat production code susah dibaca, tidak rapi bahkan ada duplikasi function atau memiliki kompleksitas yang tinggi sehingga menyebabkan code sulit untuk di-maintain saat sudah menjadi aplikasi yang besar. Untuk itulah dibutuhkan refactoring. Jadi sederhananya, Refactoring adalah sebuah teknik untuk meningkatkan desain code kita sehingga membuat code kita menjadi mudah dibaca dan di-maintain. Salah satu cara refactoring code adalah menerapkan design pattern.

Kode yang sebaiknya di-refactor :

1. Long Method

Method yang baik adalah method yang pendek dan fokus pada suatu job tertentu. Long method tidak bagus secara desain karena semakin panjang method tersebut maka semakin sulit untuk dimengerti. Jika ada method yang memiliki banyak parameter dan temporary variables, hal tersebut bisa diselesaikan dengan Extract Method.

2. Duplicated Code

Duplicated Code merupakan jenis code yang di-refactor karena membuat tidak good design. Jenis duplicated code yang paling sederhana adalah adanya dua ekspresi yang sama pada dua method di class yang sama.

Kita bisa melakukan Extract Method dan melakukan invoke pada code tersebut. Jika, ada duplicated code di class yang berbeda, kita bisa melakukan Extract Class pada salah satu class dan menggunakan component baru di class yang lain

3. Large Class

Suatu class yang berusaha didesain untuk melakukan banyak job maka artinya akan ada banyak instance variable. Jika class memiliki banyak instance variable, artinya kita mendorong class kita memiliki duplicated

code. Hal paling sederhana yang bisa dilakukan adalah mengeliminasi code yang redundant pada class tersebut.

Fungsi dan tujuan code refactor

Refactor dapat dilakukan ketika kita ingin menambahkan fitur tertentu di software kita. Biasanya alasan utama untuk refactor adalah untuk membantu kita mengerti beberapa code yang harus kita modifikasi. Code tersebut bisa jadi dibuat oleh orang lain atau bahkan diri kita sendiri. Jadi ketika kita akan melakukan refactor pastikan kita mengerti apa yang dilakukan oleh code tersebut. Jika tidak mengerti, maka bisa membuat code design kita menjadi buruk. Lalu ketika menemukan bahwa code design tidak membantu dalam menambahkan fitur, maka kita harus coba memperbaiki code design dan otomatis kita sudah melakukan refactoring. Setelah code design dirasa sudah baik untuk menambahkan fitur, barulah kita melakukan penambahan fitur

Refactor juga biasa dilakukan jika sebuah software mengalami kasus seperti :

1. Ketika memperbaiki bug
Ketika memperbaiki bug, secara tidak langsung kita juga berusaha mengerti apa yang suatu code lakukan. Ketika sedang memahami code tersebut, sebaiknya kita juga melakukan refactor untuk menambah rasa percaya diri kita bahwa kita memahami code tersebut.
2. Ketika melakukan Code Review
Pada saat kerja di Industri, biasanya para developer akan melakukan code review ketika suatu fitur selesai dikerjakan. Code review membantu berbagi pengetahuan ke seluruh anggota tim. Dengan code review, developer yang lebih berpengalaman dapat berbagi ke yang kurang berpengalaman. Developer yang berpengalaman ini membantu tim untuk mengerti lebih dalam terkait aspek large software system. Code review juga menjadi ajang untuk tiap orang memberikan ide yang lebih bermanfaat terkait suatu fitur

Teknik atau cara melakukan kode refactor

Teknik atau cara melakukan kode refactor diantaranya adalah dengan :

1. mendeteksi masalah
Langkah awal dalam melakukan kegiatan refactoring adalah dengan menganalisa apakah sebuah kode perlu untuk dilakukan refactoring
2. Mendeskripsikan masalah

Mendeskripsikan sebuah masalah yang terjadi seperti apakah harus dilakukan perubahan pada code

3. Modifikasi code

Langkah terakhir adalah dengan cara melakukan transformasi struktur code di dalamnya, tanpa harus merubah fungsi dari code tersebut.

Contoh implementasi code refectory

1. Sebelum melakukan tahap code refectory

```
90 <!-- about -->
91 <div id="abt" class="abt">
92   <div class="container">
93     <div class="row display_boxflex">
94       <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
95         <div class="abtbox">
96           <h2>Membersihkan Halaman Depan</h2>
97           <p>Salah satu layanan kami adalah dapat membersihkan halaman depan anda dan anda dapat memilih
98             layanan ini saat anda membutuhkannya</p>
99           <!-- <a href="Javascript:void(0)">Read More</a> -->
100         </div>
101       </div>
102       <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
103         <div class="abtbox">
104           <figure></figure>
105         </div>
106       </div>
107     </div>
108   </div>
109 </div>
110 <!-- end abouts -->
```

2. Sesudah melakukan tahap code

```
91 <!-- about -->
92 <div id="about" class="about">
93   <div class="container">
94     <div class="row display_boxflex">
95       <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
96         <div class="about-box">
97           <h2>Membersihkan Halaman Depan</h2>
98           <p>Salah satu layanan kami adalah dapat membersihkan halaman depan anda dan anda dapat memilih
99             layanan ini saat anda membutuhkannya</p>
100           <!-- <a href="Javascript:void(0)">Read More</a> -->
101         </div>
102       </div>
103       <div class="col-xl-6 col-lg-6 col-md-6 col-sm-12">
104         <div class="about-box">
105           <figure></figure>
106         </div>
107       </div>
108     </div>
109   </div>
110 </div>
111 <!-- end abouts -->
```

Refrensi

<https://medium.com/ppl-a1-iebs/refactor-your-code-10bf2238047>

<https://www.okedroid.com/2017/10/belajar-mengenal-apa-itu-refactoring.html>

<https://id.quora.com/Apa-yang-dimaksud-dengan-code-refactoring-dalam-pemrograman>