1. Coding
2. Code smell merupakan gejala dalam source code pada sebuah program yang mungkin mengindikasikan problem yang lebih besar akan terjadi. Contoh code smell pada Object Oriented yang sering terjadi adalah *Feature envy. Feature Envy* yaitu jika sebuah class menggunakan method class lain secara berlebihan. Hal tersebut menimbulkan high coupling dan low cohesion dalam istilah Grasp design pattern, tentu saja hal tersebut kurang bagus dalam design class pada software engineering. Jika high coupling terjadi maka ketergantungan class satu dengan class yang lain tinggi, serta jika low cohesion maka class tersebut tidak fokus dengan apa yang seharusnya dikerjakan. Untuk menanggulangi problem-problem seperti itu bisa menggunakan aturan-aturan dalam GRASP design pattern dan mengikuti aturan Law of demeter untuk mendesign ulang class design tersebut.
3. Pertama kali mendengar istilah **Dependency Injection** pada saat saya belajar Angular.js <https://docs.angularjs.org/guide/di>. Konsep DI yang saya tangkap waktu itu adalah menginject / memasukkan suatu service atau module ke dalam controller kita jika kita membutuhkannya. Sebagai contoh ketika saya ingin melakukan **AJAX call** menggunakan $http service, kita tinggal inject $http service ke dalam controller kita. . Setelah saya baca lebih general lagi mengenai DI, inilah definisinya : sebuah prinsip design dalam software engineering dimana code membuat objek baru dengan cara mengimport / menginject object / class lain diman objek baru tersebut tergantung dengan proses konstruksinya. Tujuan menggunakan DI disini adalah supaya suatu module / class tidak bergantung satu sama lain dengan module / class lain (loose coupled)

2. REST

Hal yang seharusnya dilakukan dan tidak boleh dilakukan ketika menghandle request dengan method:

|  |  |  |
| --- | --- | --- |
| Method | Yang seharusnya dilakukan | Tidak dilakukan |
| GET | * Retrieve sebuah resource dari system yang parameternya bersifat tidak privacy. * Mengambil parameter query dari request query pada url. | * Melakukan request untuk melakukan perubahan atau store sebuah resource ke dalam system atau meretrieve resource yang mengandung informasi yang sensitif seperti password karena parameter tersebut akan dimunculkan dalam url. * Melakukan retrieve request body. |
| POST | * Parameter yang masuk harus divalidasi terlebuh dahulu, karena bisa jadi mengandung hal-hal yang tidak diinginkan seperti sql injection dll. * Meretrieve parameter request dari request body bukan dari query url * Menggunakan csrf token untuk menambah proteksi | * Mengambil parameter request dengan tidak menggunakan request body tetapi menggunakan query pada url (seperti pada method GET) |

NOTE: akun github

* https://github.com/rizqyfaishal