

Hari-18-Clean Code

"Clean Code" sebagai salah satu aspek penting dalam *software development* tentu ada bukan tanpa sebab. *Clean code* pada pengembangan proyek khususnya perangkat lunak merupakan hal yang sebaiknya dilakukan oleh para developer. Clean code juga diterapkan pada Agile Methodology (berdasarkan buku **Clean Code** karangan Robert C. Martin). Sistem dapat berubah dengan cepat, maka pemanfaatan *Clean code* juga penting untuk menyeimbangi kecepatan development.

Kode yang memanfaatkan *Clean code* diperlukan dengan maksud untuk membuat kode rapi dan baik. Setiap potongan kode memiliki arti dan fungsi sesuai tugasnya masing-masing. Terlebih lagi ketika melakukan kolaborasi pekerjaan, akan sangat memudahkan apabila program yang dibuat mengimplementasikan *Clean code* di dalamnya sehingga programmer/developer lain dapat melanjutkan pekerjaan tersebut.

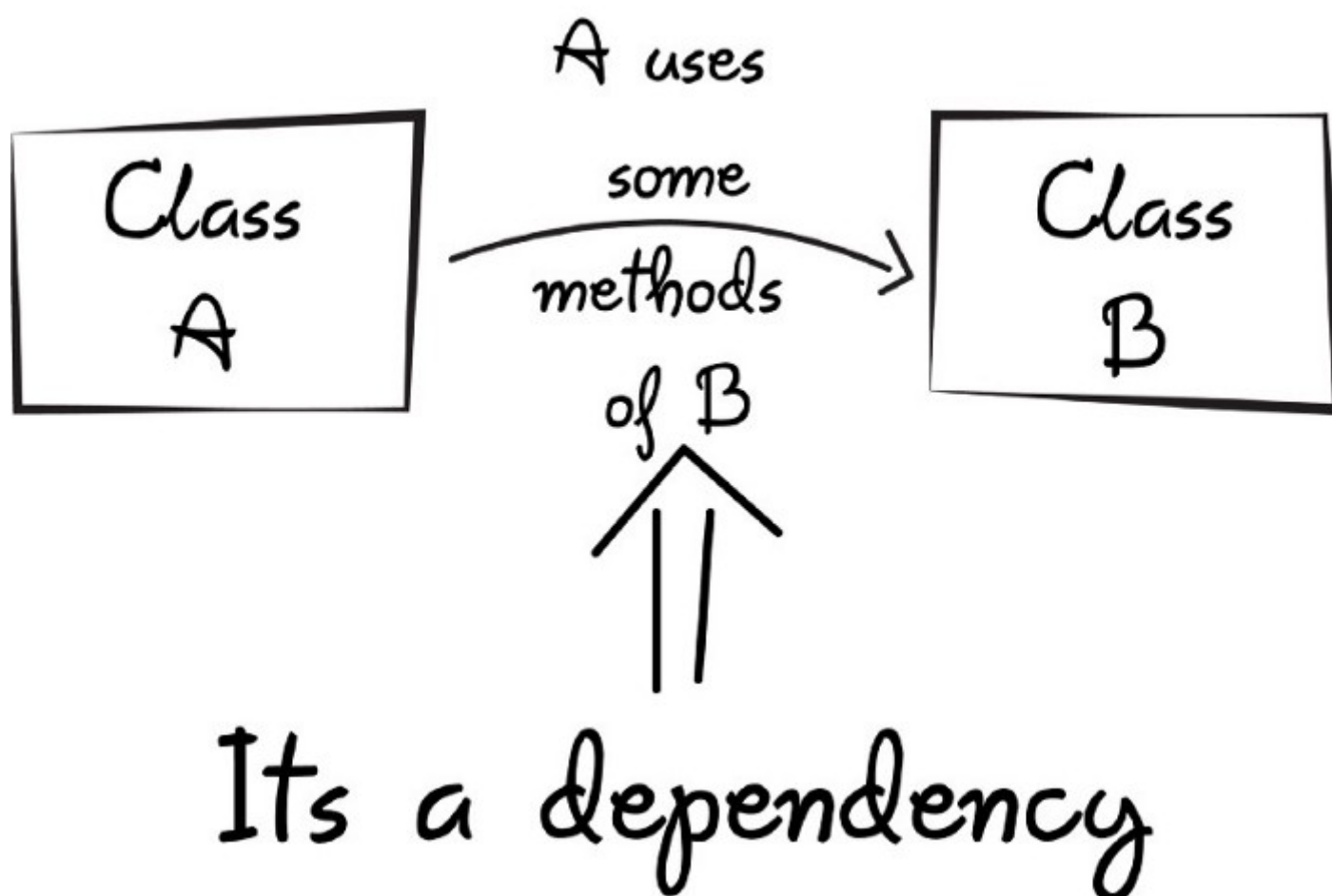
Adapun langkah-langkah untuk menerapkan *Clean code* (*ini bukan hal yang pasti, tergantung konvensi dan keinginan*).

Aturan umum

- Mengikuti standar umum dalam bahasa pemrograman, contoh : Python (PEP 8)
- **KISS (Keep It Simple Stupid)**, diusahakan untuk selalu seminimal mungkin dalam mengimplementasikan kode.
- Cari akar permasalahan, jangan hanya bagian-bagian sekitar yang terkena dampak kode tersebut.

Aturan Desain Arsitektur Kode

- Penerapan **SOLID Principles**. <https://medium.com/@dorela/s-o-l-i-d-principles-explained-in-python-with-examples-3332520b90ff>
- *Dependency Injection*



- Mencegah konfigurasi berlebih, seperti tidak memaksakan semua best practice diterapkan pada kode, meskipun kode tersebut memang sudah dipastikan kebenarannya.
- Konsistensi perlu diterapkan dalam arsitektur.
-

Aturan penamaan

Tidak mungkin dalam suatu kode kita tidak membuat suatu variabel. Variabel adalah hal penting dalam membuat suatu kode yang nantinya akan menjadi sebuah fitur. Pada penerapan *clean code* berikut adalah beberapa hal yang dapat diterapkan ketika akan memulai menulis sebuah kode.

1. Nama variabel bersifat deskriptif
2. Nama variable yang mudah dicari
3. Pemberian nama constant untuk *magic number*, seperti gravitasi.

Aturan Fungsi

Sebuah fungsi pada kode sangat penting keberadaanya, ketika fungsi tersebut tidak dibuat secara baik, maka ketika akan dilakukan pembuatan implementasi, test, dan refactoring akan semakin sulit. Jika kita mengabaikan hal ini, maka tentu kita bisa saja terlewat oleh para developer lain. Berikut adalah beberapa cara untuk menerapkan *clean code* pada fungsi.

- 1. Fungsi hanya memiliki satu tujuan dan seminimal mungkin.
- 2. Nama dan isi fungsi harus sesuai dengan apa yang dilakukan.
- 3. Penamaan fungsi yang deskriptif dan mudah.
- 4. Argument tidak terlalu banyak pada suatu fungsi.
- 5. Tidak ada *side-effect* pada kode.
- 6. Penerapan *exception* untuk handling kode

Aturan *comments*

- Komentar pada kode harus deskriptif dengan penjelasan singkat.
- Jika tidak perlu komentar, maka tidak perlu dituliskan.

Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

Submit