

PPL 2017

PANDUAN PENGEMBANGAN

30 JANUARY 2017

PENGANTAR

Paparan pada dokumen panduan ini hanya merupakan arahan/ *pointer* singkat saja, peserta kuliah diminta untuk membaca dan memahami sumber acuan yang diberikan. Syarat minimal yang dituliskan pada panduan ini direfleksikan dengan nilai poin B pada komponen kompetensi/topik terkait. Memahaminya dengan mendalam akan memberikan manfaat yang sangat besar dikemudian hari pada karir di bidang software engineering. Nilai poin A akan diberikan pada peserta yang melakukan lebih baik/banyak ataupun lebih *advanced* daripada panduan ini.

Panduan Pengembangan ini ditulis khusus untuk acuan penulisan code program dalam pengembangan perangkat lunak pada kuliah PPL 2017. Panduan ini diambil dari beberapa sumber baik literature maupun pendapat dan kebutuhan industri serta ditetapkan untuk kemudahan proses perkuliahan. Pada PPL 2017, digunakan pendekatan *Test-Driven Development*. Pada penulisan code-nya peserta diharapkan mengikuti arahan dari buku *Clean Code* oleh Robert C. Martin.

SUMBER ACUAN:

- 1) Test-Driven Development With Python, Harry J.W. Percival, O'Reilly 2014. Edisi 2 tersedia online di URL: <http://www.obeythetestinggoat.com>
- 2) Clean Code: A Handbook of Agile Software Craftmanship, Robert C. Martin.

Pada buku acuan, dijelaskan rinci prinsip prinsip dan *best practices* yang bermanfaat. Peserta sangat disarankan untuk membaca kedua buku acuan tersebut untuk dapat memahami dan menerapkan panduan ini lebih jelas dan tepat. Selamat berlatih untuk menjadi programmer yang berkualitas dan menghasilkan code yang bermanfaat.

TEST DRIVEN DEVELOPMENT (TDD)

DISPLIN DALAM METODOLOGI PENGEMBANGAN DEMI TEAM WORK YANG BERKUALITAS

Metodologi Test Driven Development dipilih untuk kuliah PPL 2017 demi membantu peserta untuk lebih mengenali perkembangan dan kualitas software engineering. Metode ini menuntut programmer untuk berpikir tentang kualitas dan API terlebih dahulu sebelum membuat program, sehingga tolak ukur kualitas dan integrasi dengan bagian lain bisa lebih awal dipikirkan. Pada PPL 2017, berikut ini 7 (tujuh) poin-poin tentang *Test Driven Development* yang harus diterapkan:

- 1) **Test First.** Pastikan memulai pengembangan dengan membuat *unit test* terlebih dahulu. Sesuai metode TDD, Terapkan prinsip: **red/green/refactor**. Pada PPL 2017, melakukan *refactoring* tidak wajib. Bila melakukan, catat dengan pesan *commit* yang sesuai di repository: Commit Milestone [Refactor]. Akan jadi nilai tambah bila peserta sudah bisa menerapkan dengan baik. Aspek ini bisa dipelajari/ diterapkan lebih dalam pada kuliah lain seperti Advanced Programming, atau PMPL)
- 2) **Definition of Done.** Semua test yang dibuat mengacu pada *Definition of Done* (kriteria penyelesaian) yang wajib ditulis rinci pada dokumen kerja *user story*.
- 3) **Online Repository.** Setiap pengembangan code harus tercatat di repository bersama yang dapat di-akses oleh asisten dan dosen serta rekan tim lainnya.
- 4) **Commit Milestone [Red].** Setiap langkah, dalam arti penyusunan unit test dan functional test (*red*, hanya test nya saja belum ada implementasi) harus tercatat di repository. Berikan pesan *commit* yang sesuai.
- 5) **Clean test.** Terap kan aturan F.I.R.S.T pada penyusunan test nya.
- 6) **Commit Milestone [Green].** Setiap pengerjaan mengimplementasikan sebuah metode, bila sudah lolos test (*green*), wajib perlu di-catat pada pesan *commit*.
- 7) **Code coverage.** *Unit testing* yang dibuat harus cukup komprehensif sehingga menguji seluruh bagian code yang diimplementasikan. Hal ini dievaluasi dengan nilai *code coverage* dalam satuan persen. Gunakan tools yang sesuai untuk mengetahui nilai *code coverage*. Code yang sudah *green* belum bisa diterima bila code coveragennya belum terpenuhi. Pada kuliah PPL 2017, **Minimal code coverage adalah 80%**. Lebih baik lagi bila bisa mencapai **95%**.

CLEAN CODE

PENULISAN CODE MUDAH DIPAHAMI, EFISIEN DAN EFEKTIF

Pada pelaksanaan PPL 2017, peserta tidak hanya diharapkan menghasilkan produk perangkat lunak yang bermanfaat namun juga dibangun dari *source code* yang berkualitas sehingga peserta lebih terbiasa untuk menghasilkan source code yang bisa lebih mudah dibaca dan di-*maintain*. Pada PPL 2017, berikut ini 6 (enam) poin-poin minimal tentang Clean Code yang harus diterapkan:

- 1) **API Documentation dan Komentar yang efisien.** Komentar biasanya dibutuhkan untuk menjelaskan kode yang kurang bagus. Dari pada menulis komentar untuk menjelaskan kode tersebut, peserta diminta memperbaiki kodenya agar mudah dipahami. Pastikan komentar hanya ditulis untuk bagian-bagian yang membutuhkan penjelasan ekstra dan bukan karena kode program yang ditulis tidak jelas atau tidak bagus. API Documentation wajib ditulis dengan efektif dan efisien sesuai standard yang dipilih atau mengikuti tools tertentu.
- 2) **Penamaan yang baik.** Pastikan penamaan, variable, fungsi mengikuti convention umum yang ditetapkan untuk bahasa pemrograman atau *framework* yang ditetapkan. Contoh: bila menggunakan python, perlu mengikuti PEP-8.
- 3) **Penulisan fungsi (*method*) yang simple dan efektif.** Program perlu disusun dari fungsi-fungsi yang cukup kecil dan modular. **Setiap satu fungsi/method hanya melakukan satu hal.** Hindari adanya *side-effect* dalam pembuatan fungsi yang dapat membuat pengembang selanjutnya keliru dalam mengantisipasi dan memahami *behaviour* dari fungsi tersebut.
- 4) **Exception Handling.** Perlu antisipasi setiap kemungkinan *exception* dengan *handling* yang sesuai. Susun **error code** yang baik dan disepakati bersama minimal oleh anggota tim lainnya.
- 5) **Don't Repeat yourself.** Jangan ada duplikasi kode dan buatlah program yang modular dan terstruktur rapih.
- 6) **Formatting.** Terapkan aturan formating code yang sesuai dengan standard dari bahasa pemrograman dan framework yang digunakan.

_____ Akhir Panduan Pengembangan PPL 2017 _____