

MAKALAH
“MODEL PROSES AGILE - TEST DRIVEN DEVELOPMENT”



Disusun Oleh :

5200411332 - Dwi Muhammad Faza

5200411409 - Pramudea Yohano Firmansyah

5200411416 - Muhammad Ilham Triwibowo

5200411417 - Rusli Pramono

PROGRAM STUDI METODOLOGI DESAIN PERANGKAT LUNAK PRAKTIK
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
YOGYAKARTA
2021

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa. Atas rahmat dan hidayah-Nya, kami dapat menyelesaikan tugas makalah yang berjudul "MODEL PROSES AGILE - TEST DRIVEN DEVELOPMENT" dengan tepat waktu.

Makalah disusun untuk memenuhi tugas Mata Kuliah Metodologi Desain Perangkat Lunak Praktik. Selain itu, makalah ini bertujuan menambah wawasan tentang model proses Agile Test Driven Development bagi kelompok lain dan juga bagi kelompok kami.

Penulis mengucapkan terima kasih kepada Endang Anggiratih selaku Dosen pengampu mata kuliah Metodologi Desain Perangkat Lunak Praktik. Ucapan terima kasih juga disampaikan kepada semua pihak yang telah membantu diselesaikannya makalah ini. Kami menyadari makalah ini masih jauh dari sempurna. Oleh sebab itu, saran dan kritik yang membangun diharapkan demi kesempurnaan makalah ini.

DAFTAR ISI

Kata Pengantar	ii
Daftar Isi	iii
Bab I Pendahuluan	1
A. Latar Belakang	1
B. Rumusan Masalah	1
C. Tujuan Penelitian	1
Bab II Pembahasan	2
A. Pengertian	2
B. Tujuan	2
C. Jenis - Jenis Test Driven Development.....	2
D. Manfaat Penggunaan Test Driven Development	2
E. Kelebihan dan Kekuarang Test Driven Development	2
F. Prinsip Test Driven Development.....	4
Bab III Contoh Metode	5
Bab IV Penutup.....	7
A. Waterfall :	7
B. Prototype :	7
C. RAD :	8
Referensi	9

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan perangkat lunak saat ini telah berkembang sangat pesat. Dikarenakan adanya perkembangan zaman yang mana menyebabkan perkembangan teknologi di berbagai bidang telah berkembang dengan sangat pesat, hal ini menyebabkan penggunaan teknologi khususnya penggunaan teknologi perangkat lunak sangat dibutuhkan dalam kegiatan di berbagai bidang. Dengan adanya teknologi ini dapat mempermudah kita dalam melakukan beberapa tugas yang memungkinkan kita dapat melakukan beberapa tugas dengan lebih cepat dan efisien.

Metode Agile jenis Test Driven Development merupakan salah satu jenis metode dalam pengembangan perangkat lunak. Metode ini merupakan salah satu cabang dari metode Agile. Konsep sederhana dari Test Driven Development adalah merancang dan menjalankan tes sebelum kode baru ditulis. Ini mencegah penulisan kode duplikat. Ini karena hanya jumlah kode terkecil yang ditulis, yang cukup untuk lulus tes. Jika pengujian kecil berhasil, proses pengujian berjalan pada pesan kesalahan berikut yang kode harus ditulis lagi. Siklus ini berlanjut hingga pengujian selesai tanpa kesalahan. Faktanya, tes tidak lebih dari persyaratan aplikasi yang harus kita tentukan apakah sudah terpenuhi.

B. Rumusan Masalah

1. Pengertian test driven development
2. Tujuan test driven development
3. Jenis- jenis test driven development
4. Manfaat penggunaan test driven development
5. Kelebihan dan kekuarang test driven development
6. Prinsip test driven development

C. Tujuan Penelitian

1. Mengetahui pengertian test driven development
2. Mengetahui Tujuan test driven development
3. Mengetahui Jenis- jenis test driven development
4. Mengetahui Manfaat penggunaan test driven development
5. Mengetahui Kelebihan dan kekuarang test driven development
6. Mengetahui Prinsip test driven development

BAB II

PEMBAHASAN

A. Pengertian

Test Driven Development (TDD) adalah skema pengembangan perangkat lunak / aplikasi dengan melakukan unit test terlebih dahulu sebelum melakukan pemrograman lebih lanjut atau production. Metode pengembangan yang “dilandasi” proses pengujian.

B. Tujuan

Tujuan test driven development(TDD)adalah untuk membuat kode lebih jelas, sederhana dan bebas bug. Pengembangan Test-Driven dimulai dengan merancang dan mengembangkan tes untuk setiap fungsionalitas kecil aplikasi.

Test Driven Development menginstruksikan pengembang untuk menulis kode baru hanya jika tes otomatis gagal. Ini menghindari duplikasi kode. Bentuk penuh TDD adalah pengembangan yang digerakkan oleh tes.

C. Jenis - Jenis Test Driven Development

- Buat Kode Pengujian (Add Test).
- Jalankan test, hasilnya pasti fail (Watch Test Fail).
- Tulis Kode Fitur Sistem (Write Code).
- Jalankan test (Run Tests) hingga testing passed.
- Refactor Kode Fitur Sistem (Refactor).

D. Manfaat Penggunaan Test Driven Development

Manfaat memakai test driven development(TDD) Membuat kode yang mudah untuk di-maintain Membuat kode yang modular karena tes biasanya cenderung kecil. Dalam membuat tes tidak perlu memikirkan arsitektur, jadi pengembangannya bisa sangat cepat

E. Kelebihan dan Kekuarang Test Driven Development

1. Kelebihan TDD

- Simplicity artinya bahwa kode program yang dibuat harus sederhana, tidak perlu membuat kode panjang jika masih bisa ditulis lebih singkat serta tidak terjadi pengulangan kode program yang sama.

- Readability artinya bahwa kode program yang dibuat dapat dibaca dan dimengerti orang lain yang membaca kode program tersebut.
- Modularity artinya kode program ditulis dengan membagi-baginya ke dalam bentuk yang lebih kecil sehingga tidak menggabungkan semua kode dalam satu page. Kode program dapat dibagi berdasarkan model, view, controller.
- Design artinya bahwa tim pembangun harus memikirkan atau membayangkan terlebih dahulu kode program yang akan dibangun sebelum mulai membangunnya. Tim pembangun harus memikirkan spesifikasi fungsi yang akan dibangun. Test-Driven Development menuntut tim pembangun memahami setiap fungsi yang akan dibangun karena tim pembangun tidak bisa membuat kode program tanpa mengerti bagaimana hasil yang akan dicapai dan bagaimana membuat unit test-nya[8]. Test-Driven Development menuntun tim pembangun untuk mendapatkan desain yang lebih baik, karena tim pembangun bebas membuat desain seiring dilakukannya proses refactoring karena tidak dituntut untuk melakukan perancangan desain di awal pembangunan aplikasi[8].
- Efficiency artinya bahwa performansi program harus baik contohnya adalah
- Clarity artinya kode program, desain, dan tujuan kode program tersebut harus jelas dan benar-benar dimengerti apa yang dilakukan pada setiap level.

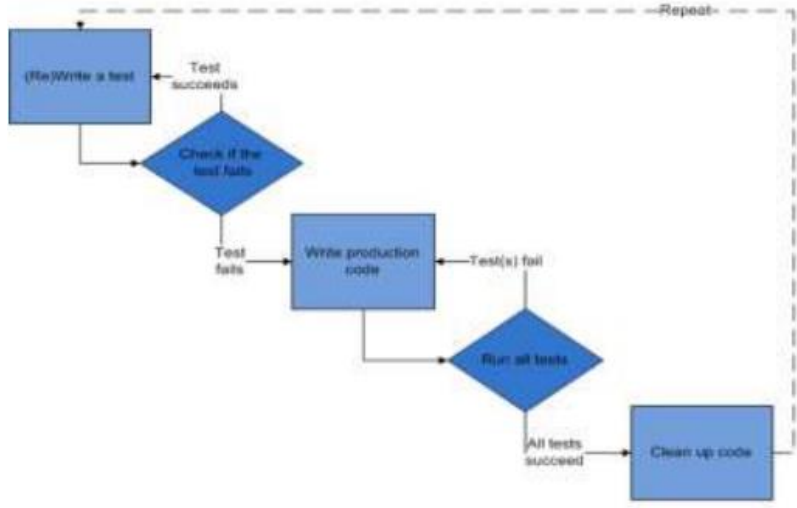
2. Kekurangan TDD

- Sulit menentukan unit test yang benar dalam sebuah studi kasus karena tidak ada ukuran yang menunjukkan kebenaran dari suatu unit test yang dibangun. *soalnya kita kan buat test dulu baru buat code nya
- Tim pembangun sulit membuat abstraksi dari sebuah interface karena tidak ada physical design yang dilakukan di awal proses pembangunan aplikasi. Tim pembangun hanya membangun aplikasi dengan design thinking.
- Dalam pembangunan aplikasi dengan metodologi ini dibutuhkan tim pembangun yang sudah berpengalaman dan memahami bagaimana menulis test yang baik dan juga mengerti sedikit tentang arsitektur yang baik sehingga tidak cocok bagi tim pembangun pemula.

F. Prinsip Test Driven Development

Prinsip dari test driven development adalah merupakan sebuah proses pada pembuatan perangkat lunak yang sangat bergantung kepada uji coba yang berubah-ubah kasusnya sesuai kebutuhan dengan harapan program yang di bangun dapat memenuhi uji coba

BAB III CONTOH METODE

Judul Artikel	Implementasi Test Driven Development Pada Pengembangan Aplikasi Android Untuk Mahasiswa Universitas Ahmad Dahlan
Penulis	Jamalludin1 , Herman Yuliansyah1 , Sri Winiati1 , Imam Riadi2
Nama Jurnal	Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI) Vol. 4, No. 1, Juni 2018
Media Publikasi	PDF / pdfs.semanticscholar.org
Link Artikel	https://www.semanticscholar.org/paper/Implementasi-Test-Driven-Development-Pada-Aplikasi-Jamalludin-Yuliansyah/809daf8c87245e4caf4b58361a5fdddb222f7cc3
Tahun, halaman	Juni 2018, Halaman 43
Tujuan Penelitian	Tujuan penelitian ini adalah pengembangan aplikasi mobile dengan menggunakan pendekatan Test Driven Development yang dapat melakukan kegiatan perwalian dan memberikan informasi perkuliahan, jadwal dosen, jadwal program studi, dan jadwal ruang untuk mahasiswa Universitas Ahmad Dahlan
Metode Penelitian	Test Driven Development
Alur Kerja	 <pre> graph TD Start(()) --> WriteTest[Write a test] WriteTest --> CheckFail{Check if the test fails} CheckFail -- "Test succeeds" --> WriteTest CheckFail -- "Test fails" --> WriteCode[Write production code] WriteCode --> RunTests{Run all tests} RunTests -- "Test(s) fail" --> WriteCode RunTests -- "All tests succeed" --> CleanUp[Clean up code] CleanUp -- "Repeat" --> WriteTest </pre> <p>The flowchart illustrates the Test Driven Development (TDD) cycle. It begins with a process box 'Write a test'. This leads to a decision diamond 'Check if the test fails'. If the test succeeds, the flow loops back to 'Write a test'. If the test fails, the flow proceeds to 'Write production code'. From there, it goes to another decision diamond 'Run all tests'. If any test fails, it loops back to 'Write production code'. If all tests succeed, it proceeds to 'Clean up code'. A dashed line labeled 'Repeat' connects the end of the 'Clean up code' box back to the start of the 'Write a test' box, indicating the iterative nature of the process.</p>

Kesimpulan	Berdasarkan hasil pembahasan, maka telah dikembangkan aplikasi android perwalian dan simeru dengan pendekatan Test Driven Development yang dapat digunakan mahasiswa Universitas Ahmad Dahlan untuk melakukan kegiatan perwalian, mengecek informasi perkuliahan, jadwal dosen, jadwal program studi dan jadwal ruang. Hasil pengujian usability yaitu pengujian menggunakan System Usability Scale (SUS) yang menghasilkan nilai 69,7 menunjukan system dapat diterima.
------------	--

BAB IV

PENUTUP

Dari sekian banyak model metode Agile, salah satunya yaitu Test Driven Development (TDD) merupakan skema pengembangan perangkat lunak atau aplikasi dengan melakukan unit test sebelum melakukan pemrograman lanjut. Tujuan – tujuan dari Test Driven Development (TDD) :

1. Membuat kode lebih jelas, sederhana, dan bebas bug.
2. Menginstruksikan pengembang untuk menulis kode hanya jika tes otomatis gagal.

Dalam TDD memiliki kelebihan seperti *simplicity*, *readability*, *modularity*, *design*, *efficiency*, dan *clarity*. Untuk kekurangan sulit menentukan unit test, tim pembangun sulit membuat abstraksi dari sebuah interface, serta dibutuhkan tim pembangun yang berpengalaman dan memahami menulis test yang baik serta mudah dimengerti.

Untuk perbandingan dengan metode lain, seperti Waterfall, Prototype, dan RAD, setiap metode memiliki tujuan, kelebihan, dan kekurangan masing – masing.

A. Waterfall :

- Kelebihan
 1. Mudah dimengerti
 2. Desain sederhana sehingga mudah ditesting dan dianalisis
- Kekurangan
 1. Tidak cocok untuk proyek jangka Panjang
 2. Tidak fleksibel

B. Prototype :

- Kelebihan
 1. Komunikasi client dan tim pengembang yang intens
 2. Memberi client experience lebih awal
- Kekurangan
 1. Prototyping cukup mahal
 2. Terlalu banyak modifikasi

C. RAD :

- Kelebihan
 1. Feedback yang berkala
 2. Mengurangi risiko
- Kekurangan
 1. Butuh kerja sama tim yang tinggi
 2. Tidak cocok untuk proyek budget rendah

Setiap metodologi atau metode pengembangan perangkat lunak berusaha untuk memastikan bahwa software yang dikembangkan sesuai dengan requirement yang dibutuhkan.

REFERENSI

- Jamalludin, Yuliansyah, H., Winiati, S., & Riadi, I. (2018). Implementasi Test Driven Development Pada Pengembangan Aplikasi Android Untuk Mahasiswa Universitas Ahmad Dahlan. *Jurnal Ilmu Teknik Elektro Komputer dan Informatika (JITEKI)* Vol. 4, No. 1, Juni 2018, 43.
- Yuwono, J. (2020, April 14). *Kelebihan dan Kekurangan dari Berbagai Metode Pengembangan Software*. Retrieved from Badr Interactive: <https://badr.co.id/id/kelebihan-dan-kekurangan-dari-berbagai-metode-pengembangan-software>