

Laporan Kemajuan Tesis

Laporan Tesis

- Revisi berdasarkan hasil seminar:
 - (**Todo**) Cakupan dan batasan penelitian belum diperjelas (masih *source code* Java, bukan Java Web)
 - (**Todo**) Penyesuaian judul belum dilakukan. Usulan judul: Pengembangan DSL untuk Verifikasi Formal dan Pengujian pada BDD
 - (**Done**) Menambahkan penjelasan mengenai V&V pada bagian latar belakang dan tinjauan pustaka
 - (**Done**) Perbaiki rumusan masalah

Rumusan Masalah - Awal	Hasil perbaikan
Masalah yang dikaji pada penelitian ini adalah verifikasi dan validasi perangkat lunak dengan pendekatan verifikasi formal dan pengujian, yang sulit untuk diselaraskan dengan perubahan spesifikasi dan implementasi perangkat lunak. Diperlukan strategi untuk membuat spesifikasi perangkat lunak yang mencakup kebutuhan untuk verifikasi formal dan pengujian, serta strategi untuk mengekstraksi model formal dari <i>source code</i> . Dengan demikian, verifikasi formal dan pengujian akan selalu relevan dengan spesifikasi dan implementasi.	<ol style="list-style-type: none">1. Pengujian yang dilakukan pada proses pengembangan perangkat lunak dengan BDD, tidak cukup untuk menjamin perangkat lunak dari kesalahan (<i>correctness</i>). Verifikasi formal dapat melengkapi proses V&V pada BDD. Diperlukan strategi untuk membuat spesifikasi perangkat lunak yang mencakup kebutuhan untuk pengujian dan verifikasi formal.2. Model perangkat lunak yang telah terverifikasi secara formal belum dapat menjamin <i>correctness</i> perangkat lunak, karena sangat tergantung dari implementasinya. Verifikasi formal harus dikenakan pada <i>source code</i>, agar hasil verifikasi selalu relevan dan mampu menjadi parameter evaluasi <i>correctness</i> perangkat lunak.

- (**Done**) Perbaiki tujuan penelitian
Output penelitian = DSL, dan kakas (DSL *parser*) untuk mentransformasi DSL kedalam interpretasi lain yang mampu digunakan oleh kakas BDD terpilih (JBehave), dan kakas *model checking*

Tujuan Penelitian - Awal	Hasil perbaikan
Tujuan yang ingin dicapai dari penelitian ini adalah verifikasi dan validasi perangkat lunak dengan dua pendekatan, yaitu verifikasi formal dan pengujian, yang bermanfaat dalam penjaminan <i>correctness</i> implementasi perangkat lunak terhadap spesifikasinya. Tujuan spesifik dari penelitian ini adalah menghasilkan teknik yang mampu membangkitkan skrip pengujian dari spesifikasi, mampu memformalkan spesifikasi, dan memodelkan <i>source code</i> agar dapat diverifikasi secara formal.	<ol style="list-style-type: none"> 1. Menghasilkan sebuah <i>Domain-Specific Language</i> yang mampu menjadi spesifikasi perangkat lunak, dan mencakup kebutuhan untuk pengujian dan verifikasi formal 2. Menghasilkan kerangka kerja untuk melengkapi proses V&V pada pengembangan perangkat lunak <i>Behavior-Driven Development</i>, dengan verifikasi formal yang dikenakan pada level <i>source code</i>

(Done) Perbaikan metodologi penelitian

- Tinjauan Pustaka

(Todo) Belum menambahkan tulisan mengenai hasil eksplorasi kakas *model checking* (*Java Path Finder* dan *Bandera*), dan eksplorasi terkait anotasi di Java (*BSL / Bandera Specification Language*, *JML*, *extended JML*)

- Ekplorasi Kakas

- Kakas BDD (dibatasi: hanya kakas yang mendukung Java)

- Cucumber (Cucumber-JVM) → *Acceptance test*

- JBehave → *Acceptance test & code generator*

- (JBehaveCodeGenerator

- <http://www.mazataz.com/resources/org.custom.jbehave.code.generator.2.0.2.jar>)

- Code generator (stub function)* Belum dicoba.

- Stub function* untuk *test*: Baru mencoba yang sederhana

(Todo) Rencana selanjutnya: melanjutkan eksplorasi pada JBehave. Alasan pemilihan JBehave dijelaskan pada subbab II.7.3

- Kakas untuk anotasi di Java
 - Belum eksplorasi lebih dalam, masih membandingkan:
 - a. Java Annotation
 - b. (Todo) JML: JML menggunakan *hoare logic*. Tetapi penelitian (Trentelman dan Huisman, 2002) membuat *extended* JML dengan memasukkan *temporal logic* (penelitiannya dirancang untuk menspesifikasikan properti pada program Java Card)
 - c. BSL: Bandera Specification Language. melakukan ekstraksi formal model dari *source code*, dan melakukan *model checking* (dengan kakas tambahan yaitu Bogor)
- Kakas untuk *model checking*
 - Java Path Finder: verifikasi terhadap *executable* Java Bytecode. Digunakan untuk memeriksa *concurrency defect* seperti *deadlocks*, dan *unhandled exception* seperti *NullPointerException* dan *AssertionError*
<https://github.com/javapathfinder/jpf-core>
 - Bandera: tools untuk eksperimen dengan properti *model-checking* pada *source code* Java
Bandera:
 - 1. ekstraksi / konstruksi model dari *source code*
 - 2. menyediakan BSL (*Bandera Specification Language*) dengan *temporal specification language*Model yang dikonstruksi didefinisikan dalam *guarded command language* yang disebut BIR (*Bandera Intermediate Representation*). BIR language dapat ditranslasikan ke dalam *input language* pada kakas *model checking* lain (pada paper disebutkan JPF, Spin, SMV, Bogor)
- Analisis dan perancangan
(Done) Analisis terkait V&V pada existing BDD
- Rencana Eksperimen
 - 1. Kasus: Simple Withdrawal ATM
(<https://github.com/masterthought/jbehave-example>)
 - 2. ... Java Web. Kandidat: <https://github.com/mbarbuscio/bow-automation>

On going:

- Eksplorasi kakas BDD, anotasi Java, dan kakas *model checking*
 - Kasus: Simple Withdrawal ATM → dites dan diverifikasi
 - *Acceptance test* dengan JBehave
 - Verifikasi dengan Bandera
- Analisis dan perancangan

Referensi

Trentelman, K., dan Huisman, M. (2002): Extending JML Specifications with Temporal

Logic, *Algebraic Methodology and Software Technology*, 334–348, Springer,

Berlin, Heidelberg, diperoleh melalui situs internet: [https://doi.org/10.1007/3-](https://doi.org/10.1007/3-540-45719-4_23)

540-45719-4_23.