

# TINJAUAN ARTIKEL JURNAL

---

Nama Mahasiswa	Denis Erlangga
NIM	19650012

## A. Meninjau Jurnal

Nama Jurnal	IEEE Transactions on Software Engineering
Nama Penerbit	Institute of Electrical and Electronics Engineers Inc.
Tahun Cakupan	1999-2021
<i>Quartile Scopus</i>	Q1
<i>CiteScore</i>	11.4
<i>CiteScore Rank</i>	34
<i>SJR</i>	2.027
<i>H-Index</i>	173

## B. Meninjau Judul Artikel

Judul asli (termasuk subjudul jika ada)	Automated Prototype Generation From Formal Requirements Model
Judul terjemahan	Pembuatan Prototipe Otomatis Dari Model Persyaratan Formal
Jumlah kata	7
Metode/ Standar	Tidak
<i>Knowledge Area</i>	Software Engineering Models and Methods
Jumlah Sitasi	10

## C. Meninjau Pengarang

Nama pengarang 1	Yilong Yang
Total sitasi (kutipan)	62
<i>H-index</i>	10
Jumlah makalah	13

Nama pengarang 2	Xiaoshan Li
Total Sitasi (Kutipan)	26
<i>H-index</i>	48
Jumlah makalah	21

Nama pengarang 3	Wei Ke
Total Sitasi (Kutipan)	21
<i>H-index</i>	2
Jumlah makalah	6

Nama pengarang 4	Zhiming Liu
Total Sitasi (Kutipan)	2970
<i>H-index</i>	27
Jumlah makalah	103

#### D. Meninjau Abstrak

Abstrak	<p>Prototyping adalah cara yang dan efisien untuk validasi kebutuhan untuk menghindari kesalahan pada tahap awal pengembangan perangkat lunak. Namun, mengembangkan prototipe sistem perangkat lunak secara manual memerlukan upaya tambahan, yang akan meningkatkan biaya pengembangan perangkat lunak secara keseluruhan. Dalam artikel ini, kami menyajikan pendekatan dengan alat yang dikembangkan RM2PT untuk pembuatan prototipe otomatis dari model persyaratan formal untuk validasi persyaratan. Sebuah model persyaratan terdiri dari diagram use case, diagram kelas konseptual, definisi use case yang ditentukan oleh diagram urutan sistem, dan kontrak operasi sistem mereka. Kontrak sistem operasi secara formal ditentukan oleh sepasang pra dan pascakondisi dalam bahasa objek. Kami mengusulkan metode-metode aturan transformasi untuk menguraikan kontrak menjadi bagian yang dapat dieksekusi dan bagian yang tidak dapat dieksekusi. Bagian yang dapat dieksekusi dapat otomatis diubah menjadi urutan operasi awal dengan menerapkan aturan yang sesuai, dan bagian yang tidak dapat dieksekusi tidak dapat diubah dengan aturan. Alat RM2PT menyediakan mekanisme bagi pengembang untuk mengembangkan program bagian untuk setiap bagian yang tidak dapat dieksekusi secara manual, yang dapat dicolokkan ke kode sumber prototipe yang secara otomatis. Kami telah melakukan empat kasus studi dengan lebih dari 50 kasus penggunaan. Hasil eksperimen menunjukkan bahwa 93,65% operasi sistem dapat dieksekusi, dan hanya 6,35% yang tidak dapat dieksekusi, yang dapat diimplementasikan oleh pengembang secara manual atau menggunakan antarmuka pemrograman aplikasi (API) pihak ketiga. Secara keseluruhan, hasilnya memuaskan. Setiap prototipe yang dihasilkan selama 1 detik dari empat studi kasus memerlukan implementasi manual satu hari oleh programmer yang terampil. Pendekatan yang diusulkan dengan alat rekayasa perangkat lunak berbantuan komputer yang dikembangkan dapat diterapkan pada industri perangkat lunak untuk rekayasa persyaratan.</p>
Tujuan	Pendekatan dengan alat yang dikembangkan RM2PT untuk pembuatan prototipe otomatis dari model persyaratan formal untuk validasi persyaratan
Desain/metodologi/pendekatan	Mengusulkan metode-metode aturan transformasi untuk menguraikan kontrak menjadi bagian yang dapat dieksekusi dan bagian yang tidak dapat dieksekusi. Bagian yang dapat dieksekusi dapat otomatis diubah menjadi urutan operasi awal dengan menerapkan aturan yang sesuai, dan bagian yang tidak dapat dieksekusi tidak dapat diubah dengan aturan.
Temuan	Pendekatan yang diusulkan dengan alat rekayasa perangkat lunak berbantuan komputer yang dikembangkan dapat diterapkan pada industri perangkat lunak untuk rekayasa persyaratan.
Orisinalitas/nilai	Hasil eksperimen menunjukkan bahwa 93,65% operasi sistem dapat dieksekusi, dan hanya 6,35% yang tidak dapat dieksekusi, yang dapat diimplementasikan oleh pengembang secara manual atau menggunakan antarmuka pemrograman aplikasi (API) pihak ketiga.
Keterbatasan/implikasi penelitian (jika ada)	bagian yang tidak dapat dieksekusi tidak dapat diubah dengan aturan
Jenis Makalah	Makalah penelitian (Research paper)

Kata Kunci (bahasa Inggris)	Formal requirements model, object constraint language (OCL), prototype, requirements, requirements model, requirements validation, unified modeling language (UML).
Kata Kunci (bahasa Indonesia)	Model persyaratan formal, bahasa kendala objek (OCL), prototipe, persyaratan, model persyaratan, validasi persyaratan, bahasa pemodelan terpadu (UML).

## E. Meninjau Pendahuluan

Latar Belakang ( <i>gap analysis</i> )	Prototyping dinilai efektif untuk validasi evolusi persyaratan melalui model yang bisa di eksekusi oleh software untuk mendemonstrasikan konsep, menemukan kesalahan dan melakukan solusi perbaikan. implementasi fungsionalitas sistem utama sebuah prototipe memiliki antarmuka pengguna yang memungkinkan klien untuk memvalidasi persyaratan mereka secara visual, memudahkan untuk menemukan sebuah kesalahan dalam persyaratan, dan kemudian diperbaiki. Dalam artikel dijelaskan bahwa disajikan pendekatan untuk menghasilkan prototipe secara otomatis dari model persyaratan formal dalam diagram UML yang dilengkapi dengan kontrak operasi sistem dalam bahasa kendala objek. Metode formal memiliki peningkatan kejelasan maupun ketepatan spesifikasi persyaratan. Hal ini sangat membantu untuk menemukan kesalahan dalam validasi persyaratan dan pengujian perangkat lunak.
Rumusan Masalah/ Pertanyaan Penelitian/ Hipotesis	Bagaimana pendekatan dengan alat yang dikembangkan RM2PT untuk pembuatan prototipe otomatis dari model persyaratan formal? Bagaimana penggunaan metode-metode aturan transformasi untuk menguraikan kontrak menjadi bagian yang dapat dieksekusi dan bagian yang tidak dapat dieksekusi?
Teori Utama (tinjauan teoretis)	Kesalahan persyaratan adalah salah satu penyebab utama kegagalan dalam proyek perangkat lunak. Spesifikasi persyaratan formal telah banyak digunakan dan terbukti berguna dalam industri perangkat lunak, terutama dalam pengembangan sistem yang kritis terhadap keselamatan . OCL adalah bahasa spesifikasi formal ringan berdasarkan logika orde pertama, yang terutama digunakan untuk menentukan batasan model UML seperti invarian, pra dan pascakondisi, serta proyek fondasi Eclipse dan lini produk IBM. Ide awal generasi prototipe otomatis dari model persyaratan disajikan dalam pekerjaan kami sebelumnya
Referensi dari Teori Utama	A. G. Sutcliffe, A. Economou, and P. Markis, "Tracing requirements errors to problems in the requirements engineering process," <i>Requirements Eng.</i> , vol. 4, no. 3, pp. 134–151, 1999. P. D. Bois, E. Dubois, and J.-M. Zeippen, "On the use of a formal requirements engineering language: The generalized railroad crossing problem," <i>Requirements Eng.</i> , vol. 2, no. 4, pp. 171–183, Dec. 1997. G. Vanwormhoudt, O. Caron, and B. Carré, "Aspectual templates in UML—Enhancing the semantics of UML templates in OCL," <i>Softw. Syst. Model.</i> , vol. 16, no. 2, pp. 469–497, 2017.

## F. Meninjau Metodologi

Skenario Pengujian	Melakukan penyesuaian terhadap aturan transformasi dan algoritma bekerja. Menentukan prototype pola arsitektur GUI yang digunakan. Prototipe berisi tiga modul: tampilan, pengontrol, dan model. Pembuatan kelas entitas adalah prosedur yang diperlukan untuk mencapai autoprotyping dari model persyaratan, tidak
--------------------	---

	hanya perlu menghasilkan atribut dan asosiasi dari kelas entitas tetapi juga membutuhkan untuk menghasilkan implementasi operasi primitif untuk pengaturan dan mendapatkan atribut, menemukan objek yang ditautkan, dan menambahkan dan menghapus tautan. Selanjutnya validasi persyaratan dengan memeriksa apakah persyaratan memenuhi kebutuhan nyata pelanggan. Pemeriksaan validitas berfokus pada persyaratan mencerminkan kebutuhan nyata pemangku kepentingan serta persyaratan mengandung konflik dan kontradiktif, terutama untuk kendala kontradiktif dalam kontrak yang sama dari fungsi sistem.
Metode yang digunakan	Model persyaratan CoCoME berisi tiga aktor, 16 kasus penggunaan, 43 operasi sistem, 273 operasi primitif, 13 kelas entitas, 20 asosiasi antara objek tersebut, dan sepuluh invarian. Model persyaratan LibMS mencakup tujuh aktor, 19 kasus penggunaan, 45 operasi sistem, 433 operasi primitif, 11 kelas entitas, 17 asosiasi, dan 25 invarian. Model persyaratan ATM mencakup dua aktor, enam kasus penggunaan, 15 operasi sistem, 103 operasi primitif, tiga kelas entitas, empat asosiasi, dan lima invarian. Model persyaratan LoanPS mencakup lima aktor, sepuluh kasus penggunaan, 34 operasi sistem, 171 operasi primitif, 12 kelas entitas, delapan asosiasi, dan 12 invarian. Singkatnya, kami menyediakan 17 aktor, 51 kasus penggunaan, 137 operasi sistem, 980 operasi primitif, 39 kelas entitas, 49 asosiasi kelas entitas, dan 52 invarian.
Objek Penelitian	statistik kesalahan persyaratan dan hilang selama pemodelan persyaratan tiga putaran, pembuatan prototipe dan validasi persyaratan.
Lama Penelitian	-
Biaya Penelitian	-
Jumlah Responden	-
Referensi dari Metode	G. E. Krasner and S. T. Pope, "A cookbook for using the model-view controller user interface paradigm in Smalltalk-80," <i>J. Object-Oriented Program.</i> , vol. 1, no. 3, pp. 26–49, Aug. 1988. L. Bettini, <i>Implementing Domain-Specific Languages With Xtext and Xtend</i> . Birmingham, U.K.: Packt Publ. Ltd., 2016.

## G. Meninjau Diskusi/Hasil

Diskusi / Analisis	<p>Pada bagian ini peneliti menunjukkan pemodelan persyaratan dan hasil validasi lebih detail. Mengukur jumlah aktor, kasus penggunaan, operasi sistem, kelas entitas dan asosiasi kelas entitas dalam model kompleksitas dan biaya persyaratan. Biaya pemodelan mempengaruhi pada penerapan pendekatan yang di usulkan. Sehingga dilakukan penyelidikan biaya untuk menentukan dan memodelkan persyaratan. Dilakukan penyelidikan terpisah mengenai diagram UML dan kontrak OCL dari model persyaratan.</p> <p>Dalam hasil pembuatan prototype, kegagalan pemodelan pembuatan persyaratan dibagi menjadi dua situasi yaitu yang pertama kontrak operasi sistem yang tidak dapat ditentukan dalam ekspresi OCL tanpa menggunakan API pihak ketiga seperti algoritme pengurutan dan pengiriman email. Yang kedua Kontrak operasi sistem dapat ditentukan dengan benar dalam ekspresi OCL, tetapi tidak ada aturan transformasi yang cocok untuk menghasilkan implementasi untuk operasi sistem dengan benar. Dalam studi kasus yang berisi kesalahan statistik persyaratan dan hilang, validasi persyaratan dapat membantu menemukan kesalahan dalam persyaratan, tetapi juga dapat membantu mendapatkan persyaratan yang hilang.</p>
Teori Pendukung (tinjauan empiris/praktis)	Model konseptual berisi model objek yang ditentukan dalam diagram kelas, model fungsional untuk atribut kelas, dan model dinamis untuk setiap kelas yang didefinisikan dalam diagram diagram status.

	Prototipe yang dihasilkan oleh organisasi spesifikasi formal yang diturunkan untuk kemajuan standar informasi terstruktur (OASIS) dari model konseptual dan model eksekusi terkait implementasi, SCORES mengusulkan pendekatan semiotomatis untuk menghasilkan prototipe dari peningkatan spesifikasi persyaratan dengan model UI di FLUID. Model pada UI mencakup spesifikasi pada widget tampilan, navigasi, dan pemilihan atau operasi primitif dari objek domain tersebut.
Referensi dari Diskusi/Hasil	<p>O. Pastor, J. Gómez, E. Insfrán, and V. Pelechano, "The OO-method approach for information systems modeling: From object-oriented conceptual modeling to automated programming," <i>Inf. Syst.</i>, vol. 26, no. 7, pp. 507–534, 2001.</p> <p>A. Homrighausen, H.-W. Six, and M. Winter, "Round-trip prototyping based on integrated functional and user interface requirements specifications," <i>Requirements Eng.</i>, vol. 7, no. 1, pp. 34–45, Apr. 2002.</p> <p>G. Kusters, H. W. Six, and J. Voss, "Combined analysis of user interface and domain requirements," in <i>Proc. 2nd Int. Conf. Requirements Eng.</i>, Apr. 1996, pp. 199–207.</p>

## H. Meninjau Kesimpulan

Kesimpulan	Dalam artikel yang ditulis oleh peneliti menyajikan pendekatan untuk pembuatan prototipe otomatis dari model persyaratan formal, dan model persyaratan dapat divalidasi oleh prototipe yang dihasilkan. Di dalamnya mencakup analisis spesifikasi formal yang dapat dieksekusi dan dirancang aturan transformasi untuk menerjemah bagian kontrak yang dieksekusi dalam kode java. Pada bagian kontrak yang tidak dapat dieksekusi diidentifikasi dan dibungkus oleh UI yang dapat dipenuhi oleh API pihak ketiga.
Implikasi teoretis	Dalam artikel pendekatan dengan RM2PT pembuatan prototipe otomatis dari persyaratan formal untuk validasi persyaratan yang terdiri dari diagram use case, diagram kelas konseptual, urutan sistem, dan kontrak operasi sistem. Dimana pada bagian yang dapat dieksekusi dapat otomatis di unduh menjadi prosedur operasi awal menetapkan aturan sesuai yang digunakan. Sedangkan pada bagian yang tidak dapat dieksekusi tidak dapat diubah dengan aturan.
Implikasi praktis	Implikasi praktis dalam artikel ini melakukan pengujian terhadap Empat studi kasus, yaitu LibMS, ATM, CoCoME, dan LoanPS, telah diselidiki, dan hasil eksperimen memuaskan bahwa 93% operasi sistem dari kasus penggunaan dapat dihasilkan dengan sukses. Alat CASE: RM2PT dan tutorialnya tersedia untuk umum di GitHub

## I. Meninjau Referensi

Jumlah Artikel	30
Jumlah Prosiding	19
Jumlah Buku	0
Jumlah Website	0

Jumlah Lain-lain	2
------------------	---

#### J. Meninjau *State of the Art* Artikel

Apakah penelitian ini " <i>reasonable</i> "?	Berdasarkan tinjauan artikel yang telah dilakukan diketahui bahwa Kesalahan persyaratan adalah salah satu penyebab utama kegagalan dalam proyek perangkat lunak. Untuk mengatasi permasalahan tersebut penelitian ini menyajikan pendekatan untuk menghasilkan prototipe secara otomatis dari model persyaratan formal dalam diagram UML yang dilengkapi dengan kontrak operasi sistem dalam bahasa kendala objek. Dalam praktiknya prototipe memiliki antarmuka pengguna (UI) yang memungkinkan klien untuk memvalidasi persyaratan mereka secara visual, memudahkan untuk menemukan kesalahan dalam persyaratan, dan kemudian memperbaikinya sehingga melalui penelitian ini diharapkan untuk memiliki alat yang menghasilkan prototipe langsung dari persyaratan secara otomatis
Apakah penelitian ini " <i>complicated</i> "?	Selain itu dibandingkan dengan pekerjaan terkait lainnya, pendekatan dalam penelitian ini tidak memerlukan model desain tetapi mengandalkan model persyaratan formal yang mencakup Diagram use case, Diagram urutan system, Kontrak operasi system, dan Diagram kelas konseptual. Penelitian ini memperluas karya asli dengan sepenuhnya mempertimbangkan algoritme teoritis transformasi model, dan merancang dan mengimplementasikan alat prototipe yang berlaku dengan fungsionalitas penuh pemodelan persyaratan, pembuatan prototipe otomatis serta validasi persyaratan, dan alat tersebut telah diterapkan pada empat studi kasus yang ekstensif.
Apakah penelitian ini " <i>imperative</i> "?	Berdasarkan tinjauan artikel yang telah dilakukan selain implementasi fungsionalitas sistem utama, prototipe memiliki antarmuka pengguna (UI) yang memungkinkan klien untuk memvalidasi persyaratan mereka secara visual, memudahkan untuk menemukan kesalahan dalam persyaratan dan berpotensi untuk diperbaiki sehingga melalui penelitian yang dilakukan ini diharapkan mampu meminimalisir kesalahan dalam persyaratan.
Apakah penelitian ini " <i>useful</i> "?	Ya, karena dapat mengetahui pembuatan prototipe otomatis dari model persyaratan formal dan model persyaratan pada validasi oleh prototipe yang dihasilkan. Hal ini mencakup keseluruhan analisis dari spesifikasi formal. Dengan spesifikasi formal yang dapat dieksekusi untuk merancang aturan transformasi bagian kontrak ke dalam sumber java.
Penelitian lebih lanjut	Prototipe pengujian e-learning : studi kasus metode belajar siswa
Artikel jurnal terkait	<ol style="list-style-type: none"> <li>1. Integrating design thinking into instructional design : the #openTeach case study.</li> <li>2. Enablers, Difficulties and Features to Implement E-Learning Technology in Conventional Higher Education: Case Study and Prototype Implication</li> <li>3. Deep learning for detecting pulmonary tuberculosis via chest radiography : an international study across 10 menit.</li> </ol>