



Konsep Ontologi pada Manajemen Rekayasa Perangkat Lunak: Kajian Tentang Perkembangan dan Tantangannya

Koko Wahyu Prasetyo^{1*}
¹ koko@stiki.ac.id

¹ Sekolah Tinggi Informatika & Komputer Indonesia, Sistem Informasi, Malang, Indonesia

Kata Kunci

rekayasa perangkat lunak; ontologi; sistem informasi; kajian literatur

***) Author Korespondensi**
koko@stiki.ac.id

Abstrak

Dalam era digital yang terus berkembang, tuntutan untuk menciptakan aplikasi perangkat lunak yang dinamis dan inovatif mendorong perkembangan signifikan di bidang ilmu rekayasa perangkat lunak. Mengintegrasikan potensi pengetahuan yang terkandung dalam perangkat lunak yang dikembangkan dengan ekosistem pengetahuan lain yang lebih luas menjadi sangat penting. Namun, masih ada celah literatur yang belum terpenuhi terkait analisis dan evaluasi terhadap pemanfaatan konsep ontologi sepanjang siklus hidup rekayasa perangkat lunak. Kajian ini bertujuan untuk mengidentifikasi perspektif konsep ontologi dalam sejumlah aspek rekayasa perangkat lunak dan potensinya untuk mengembangkan bidang ini lebih lanjut. Dengan mengkaji literatur dari ScienceDirect, IEEEExplore, ACM Digital Library, dan Google Scholar, serta menerapkan kriteria inklusi dan eksklusi yang relevan, penelitian ini menganalisis fenomena, tantangan, dan peluang yang dihadapi terkait konsep ontologi dalam rekayasa perangkat lunak. Hasil kajian diharapkan membantu industri mengintegrasikan pengetahuan dalam aplikasi perangkat lunak yang dikembangkan, serta memperkaya literatur di bidang manajemen rekayasa perangkat lunak, sehingga dapat mendorong inovasi lebih lanjut.

Abstract

In the ever-growing digital era, the demand to create dynamic and innovative software applications is driving significant developments in the field of software engineering. Integrating the potential knowledge contained in the software being developed with the wider ecosystem of other knowledge becomes very important. However, there is still an unfilled literature gap regarding the analysis and evaluation of the use of ontology concepts throughout the software engineering life cycle. This study aims to identify perspectives of the ontology concept in a number of aspects of software engineering and its potential to develop this field further. By reviewing literature from ScienceDirect, IEEEExplore, ACM Digital Library, and Google Scholar, and applying relevant inclusion and exclusion criteria, this research analyzes the phenomena, challenges, and opportunities faced regarding the concept of

ontology in software engineering. The results of the study are expected to help industry integrate knowledge in the software applications being developed, as well as enrich literature in the field of software engineering management, so as to encourage further innovation.

1. Pendahuluan

Perkembangan pesat dalam bidang teknologi informasi dan komunikasi mendorong terciptanya perubahan pola kebutuhan informasi secara berkelanjutan bagi pengguna teknologi informasi. Tuntutan untuk terus menciptakan layanan perangkat lunak yang dinamis dan inovatif turut mendorong perkembangan bidang ilmu rekayasa perangkat lunak secara signifikan. Hal ini memicu kebutuhan akan metode pengembangan perangkat lunak yang dapat merespon perubahan kebutuhan pengguna secara lebih efektif, sekaligus untuk meningkatkan produktivitas para pengembang perangkat lunak dalam waktu yang bersamaan (Gasevic et al., 2009). Meski demikian, tantangan yang muncul terkait situasi ini ternyata masih belum banyak berubah. Data statistik yang dirilis oleh Standish Group pada tahun 2020 menyebutkan bahwa sekitar 66% dari proyek aplikasi perangkat lunak berakhir dengan kegagalan (El-Deeb, 2022). Statistik ini menunjukkan kebutuhan krusial akan pendekatan inovatif dalam rekayasa perangkat lunak.

Selain aspek manfaat teknisnya, aplikasi perangkat lunak saat ini juga telah menjadi entitas sosial yang mempengaruhi hampir setiap aspek kehidupan manusia. Aplikasi perangkat lunak tidak hanya berfungsi sebagai alat, namun juga sebagai gudang pengetahuan yang terkait erat dengan domain ilmu tertentu. Untuk dapat menggali potensi sebuah aplikasi perangkat lunak seutuhnya, sangat penting untuk mengintegrasikan pengetahuan yang terkandung dalam perangkat lunak tersebut dengan ekosistem pengetahuan yang lebih luas (Triandini et al., 2021). Aspek-aspek seperti lingkup domain pengetahuan, dinamika kebutuhan yang senantiasa berubah, dan pengaruh faktor kontekstual menjadi penting untuk dikaitkan satu sama lain. Untuk dapat tetap mempertahankan keterhubungan pengetahuan tersebut secara efektif, maka diperlukan sebuah persyaratan dasar: rumusan definisi secara eksplisit dari lingkup pengetahuan tersebut (Isotani et al., 2015). Faktor inilah yang mendorong komunitas rekayasa perangkat lunak untuk mempertimbangkan konsep ontologi sebagai alternatif solusi yang menjanjikan untuk tantangan yang masih ada di bidang ini.

Sejumlah studi telah mencoba untuk menjelajahi berbagai jenis sinergi antara ontologi dan rekayasa perangkat lunak. Studi tersebut telah membahas aplikasi ontologi dalam konteks rekayasa kebutuhan (Dermeval et al., 2016), pemodelan perangkat lunak (Vujasinovic et al., 2015), transformasi model proses dan metodologi (Machado et al., 2016), hingga terkait pemeliharaan (Serna et al., 2017) dan penjaminan mutu perangkat lunak (Hovorushchenko & Pomorova, 2016). Kajian terhadap sinergi tersebut juga telah menarik perhatian badan-badan standarisasi untuk menghasilkan model-model yang dikembangkan secara berkelanjutan seperti Ontology-Driven Architecture (ODA) dan Ontology Definition Metamodel (ODM) (Haav, 2018).

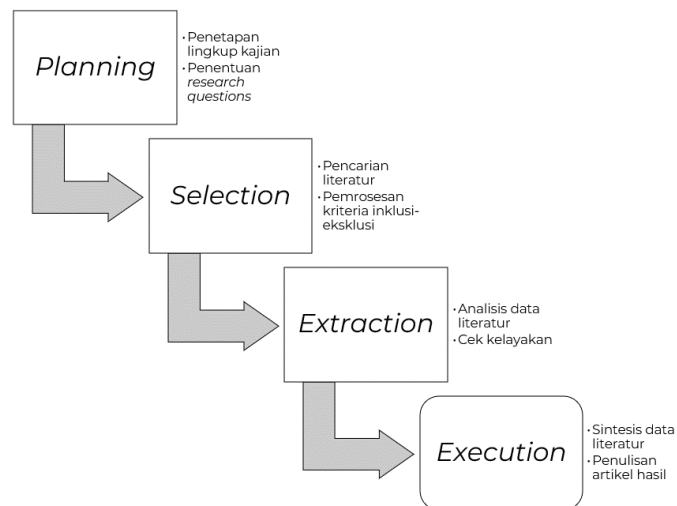
Meskipun terdapat kemajuan yang cukup signifikan, masih terdapat celah kebutuhan yang belum terpenuhi terkait analisis dan evaluasi menyeluruh konsep ontologi dalam rangkaian siklus hidup rekayasa perangkat lunak. Oleh karena itu, kajian ini akan menyelidiki sejumlah aspek untuk dirumuskan sebagai pertanyaan penelitian berikut:

- **RQ1:** Bagaimana keterkaitan konsep ontologi dalam bidang manajemen rekayasa perangkat lunak?
- **RQ2:** Bagaimana tantangan dan keterbatasan pemanfaatan aspek ontologi dalam bidang rekayasa perangkat lunak?
- **RQ3:** Bagaimana peluang dan tren penelitian terkait aspek ontologi dalam rekayasa perangkat lunak?

Dengan memperhatikan pertanyaan penelitian di atas, maka kajian ini bertujuan untuk menyajikan informasi terkait aplikasi konsep ontologi dalam berbagai aspek manajemen rekayasa perangkat lunak dan potensinya untuk mengembangkan bidang ini lebih lanjut.

2. Metode Penelitian

Penelitian ini menggunakan metode kajian literatur untuk mengeksplorasi dan menganalisis keterkaitan konsep ontologi dalam ranah manajemen rekayasa perangkat lunak. Metode tersebut dipilih karena dapat mengidentifikasi, menilai, dan menginterpretasikan seluruh penelitian yang relevan dengan topik manajemen rekayasa perangkat lunak, sehingga dapat memberikan pemahaman yang komprehensif dan holistik. Secara terstruktur, Gambar 1 menunjukkan tahapan-tahapan penelitian yang dilakukan.



Gambar 1. Tahapan Metode Kajian Literatur

Tahapan pertama yang dilakukan adalah perencanaan. Pada tahap ini, peneliti menetapkan lingkup kajian yang akan dilakukan serta menentukan pertanyaan penelitian (*research questions*) yang menjadi fokus utama pada kajian literatur ini. Lingkup kajian ditentukan secara spesifik dengan memanfaatkan kata kunci "*ontology*", "*software engineering*", "*software development*", dan "*software project*". Lingkup tersebut didefinisikan terlebih dahulu untuk memastikan bahwa kajian literatur yang dilakukan tetap terfokus dan relevan dengan topik yang ingin dieksplorasi.

Setelah tahap perencanaan selesai, langkah selanjutnya adalah proses seleksi. Pada tahap ini, pencarian literatur dilakukan dengan menggunakan kata kunci tersebut di atas pada sejumlah koleksi literatur digital berikut: ScienceDirect, IEEEExplore, ACM Digital Library, dan Google Scholar. Literatur yang diperoleh kemudian diproses melalui kriteria inklusi dan eksklusi sebagaimana tersaji pada Tabel 1 untuk memastikan bahwa hanya studi yang relevan dan memenuhi standar kualitas yang dimasukkan dalam kajian. Kriteria inklusi dan eksklusi membantu menyaring literatur yang kurang relevan atau tidak sesuai dengan tujuan penelitian, sehingga menjaga integritas dan kualitas kajian.

Tabel 1. Kriteria Inklusi dan Eksklusi Literatur

Tindakan	Uraian Kriteria
inklusi	<ul style="list-style-type: none"> • Sesuai dengan kata kunci dan lingkup kajian • Konteks penelitian bidang ilmu komputer, sistem informasi, maupun perangkat lunak • Berkontribusi terhadap satu RQ atau lebih • Dipublikasikan tidak lebih dari 10 tahun terakhir • Dilengkapi data empiris atau hasil eksperimen yang relevan
eksklusi	<ul style="list-style-type: none"> • Artikel versi lengkap tidak tersedia • Ditulis selain dalam Bahasa Indonesia atau Bahasa Inggris • Dipublikasikan tanpa melewati proses <i>peer-review</i>

Pada tahapan selanjutnya, peneliti melakukan analisis terhadap data literatur yang telah diperoleh di tahap sebelumnya. Setiap literatur yang terpilih dianalisis secara komprehensif untuk mengidentifikasi informasi yang relevan dengan pertanyaan penelitian. Selain itu, dilakukan pula cek kelayakan untuk memastikan bahwa data yang diekstraksi valid dan dapat diandalkan. Proses ekstraksi ini penting untuk mengumpulkan bukti-bukti yang mendukung dan menjawab pertanyaan penelitian secara obyektif.

Pada tahapan terakhir, data yang telah diekstraksi disintesis untuk menjawab tiga buah pertanyaan penelitian yang telah dirumuskan. Sintesis data literatur dilakukan dengan mengelompokkan temuan-temuan berdasarkan tema utama yang muncul, serta membandingkan dan mengontraskan berbagai perspektif yang ditemukan dalam literatur. Hasil sintesis ini kemudian dituliskan dalam format artikel ilmiah yang menyajikan temuan-temuan utama, diskusi, dan implikasi dari kajian literatur yang telah dilakukan.

Dengan mengikuti tahapan-tahapan ini, kajian literatur ini diharapkan dapat menghasilkan temuan yang valid, reliabel, dan memberikan kontribusi berarti bagi ilmu pengetahuan bidang rekayasa perangkat lunak.

3. Hasil dan Pembahasan

RQ1: Ontologi dan Rekayasa Perangkat Lunak

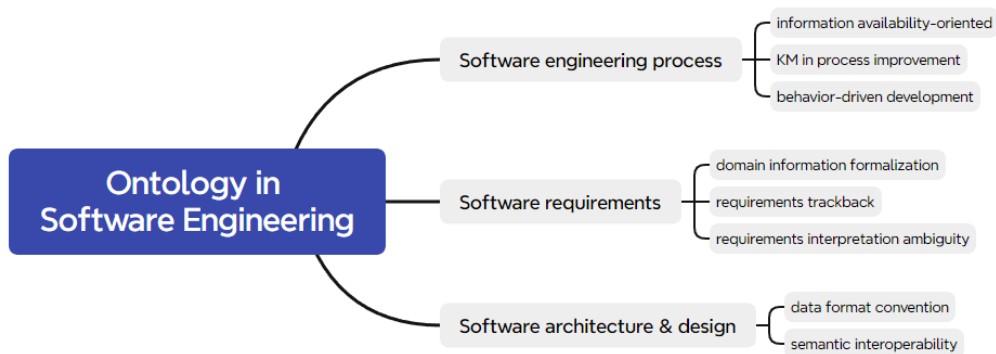
Dalam bidang ilmu komputer, istilah "ontologi" mengacu pada model metadata yang berisi kumpulan gagasan-gagasan atau konsep-konsep dalam domain tertentu, termasuk keterkaitan antara konsep-konsep tersebut (Triandini et al., 2021). Dalam konteks rekayasa perangkat lunak, ontologi memiliki berbagai tujuan. Salah satunya adalah analisis domain, di mana ontologi membantu dalam memahami dan struktur pengetahuan yang terkait dengan area subyek tertentu. Selain itu, ontologi memainkan peran penting dalam mengidentifikasi dan mengurangi ambiguitas yang melekat dalam praktik-praktik rekayasa perangkat lunak (Bhatia & Beniwal, 2016).

Konsep rekayasa perangkat lunak berbasis ontologi (*ontology-driven software engineering*) merupakan sebuah sub area dalam bidang rekayasa perangkat lunak yang mulai diperbincangkan dalam beberapa tahun terakhir. Konsep tersebut bertujuan untuk mengeksplorasi berbagai alternatif cara di mana rekayasa ontologi dan teknologi lain yang berasal dari web semantik dapat dimanfaatkan untuk meningkatkan proses pengembangan perangkat lunak (Henderson-Sellers, 2011). Poin yang menjadi perhatian utama adalah adanya pergeseran fokus dari praktik pengembangan yang berorientasi pada proses menjadi praktik yang berorientasi pada ketersediaan informasi. Hal ini menjadi relevan terutama pada lingkungan pengembangan perangkat lunak yang berjauhan, di mana kesenjangan keberadaan informasi dan sumber daya dapat menghasilkan ambiguitas dan interpretasi yang salah terhadap kebutuhan perangkat lunak. Untuk mengatasi tantangan ini, pendekatan

berbasis informasi yang menggunakan rekayasa ontologi diajukan untuk membuat, memformalkan, dan menerapkan pengetahuan tentang suatu domain tertentu.

Cakupan pengetahuan bidang rekayasa perangkat lunak dapat diformalkan menggunakan klasifikasi ontologi domain (Isotani et al., 2015) yang secara lebih rinci akan disajikan melalui Gambar 2:

- domain pengetahuan tentang proses rekayasa perangkat lunak; termasuk di dalamnya berbagai model proses pengembangan termasuk berbagai jenis artefak dan asosiasinya
- domain analisis permasalahan dan kebutuhan perangkat lunak
- domain konsep desain dan perancangan arsitektural perangkat lunak



Gambar 2. Kerangka Konsep Ontologi dalam Rekayasa Perangkat Lunak

Ketika sejumlah informasi pada domain-domain tersebut dapat diformalkan ke dalam tatanan ontologi, maka berbagai pihak pemangku kepentingan yang ada dalam sebuah proyek perangkat lunak dapat saling berbagi informasi secara lebih efektif. Penerapan pendekatan ontologis dapat mengurangi ambiguitas dan memudahkan pertukaran informasi di antara para pemangku kepentingan yang meliputi: klien, pengguna, dan tim pengembang. Selain itu, pendekatan ini dapat memfasilitasi adanya rekam jejak balik antar artefak yang dihasilkan sepanjang siklus pengembangan perangkat lunak (Fitsilis et al., 2014). Pendekatan ontologi dapat memfasilitasi keterkaitan seluruh informasi yang terkait kebutuhan, desain, implementasi, dan pengujian perangkat lunak sehingga dapat diakses secara efisien dan akuntabel.

Dalam beberapa tahun terakhir, adopsi teknologi Internet of Things (IoT) telah meningkat pesat dan mengubah pola interaksi manusia dengan lingkungannya dan mendorong lahirnya solusi inovatif di berbagai sektor industri. Salah satu tantangan yang dihadapi terkait pengembangan aplikasi berbasis IoT adalah beragamnya format data yang dihasilkan oleh perangkat IoT yang seringkali tidak mengikuti standar semantik tertentu. Hal ini menunjukkan belum adanya pola yang akurat untuk menggambarkan makna dan konteks data IoT akibat adanya inkonsistensi, konvensi penamaan yang tidak standar, dan kosakata yang beragam. Studi yang dilakukan oleh Gyrard (2018) mencoba mengkaji permasalahan interoperabilitas semantik pada domain IoT dengan pendekatan ontologi pada sejumlah proyek perangkat lunak. Studi tersebut bertujuan meningkatkan kualitas interoperabilitas data semantik melalui kajian aspek ontologis. Model dan teknik yang ditawarkan pada hasil studi tersebut disebut dapat mengurangi kurva pembelajaran pengguna dan memfasilitasi upaya pengembang dalam mencapai interoperabilitas semantik yang lebih baik.

Tidak hanya terkait aspek teknis dan perangkat teknologi, pendekatan konsep ontologi juga diterapkan pada aspek proses dan metodologi pengembangan perangkat lunak. Studi yang dilakukan oleh Lopes de Souza et al. (2021) mempelajari sejumlah tantangan yang ditemui dalam penerapan metodologi pengembangan perangkat lunak secara *agile*, khususnya Scrum dan Behavior-Driven Development (BDD). Kedua model pendekatan tersebut mensyaratkan pendefinisian skenario penggunaan aplikasi secara eksplisit untuk dapat memahami perilaku pengguna perangkat lunak. Pada studi tindakan yang dilakukan pada proyek pengembangan Learning Management System (LMS) EAMS-CBALM tersebut, pendekatan ontologi digunakan untuk menanggulangi masalah yang berkaitan dengan pendefinisian kebutuhan fungsional perangkat lunak, rangkaian pengujian, dan validasi terhadap ambiguitas yang muncul pada penyusunan *user story* oleh Product Owner. Studi tersebut mengkonfirmasi bahwa model ScrumOntoBDD - sebagai pendekatan yang menggabungkan Scrum, konsep ontologi, dan BDD, dapat berkontribusi untuk mengurangi tingkat permasalahan yang disebutkan di atas.

Salah satu pendekatan kunci bagi praktisi pengembang perangkat lunak untuk meningkatkan produktivitasnya adalah melalui implementasi perbaikan proses perangkat lunak (*software process improvement*, SPI). Seringkali perbaikan proses perangkat lunak menimbulkan berbagai hambatan karena kurangnya pengetahuan dalam memilih implementasi yang tepat. Masalah yang paling umum terjadi terletak pada pemilihan dan penerapan referensi model referensi yang sesuai dengan karakter sebuah organisasi. Studi yang dilakukan oleh Mejia (2016) mengusulkan pendekatan multi-model yang memungkinkan adopsi praktik terbaik dari berbagai referensi model pengembangan. Studi tersebut mengadopsi konsep manajemen pengetahuan (*knowledge management*) dalam kaitannya dengan praktik berbagi pengetahuan (*knowledge sharing*) dan integrasi sistem. Hasil penelitian tersebut menyajikan kerangka ontologi berdasarkan pendekatan multi-model untuk peningkatan SPI dalam perusahaan kecil dan menengah, dengan bertumpu pada perbaikan proses siklus hidup pengembangan perangkat lunak. Model ini memberikan perhatian lebih pada aspek manajemen pengetahuan berbasis kerangka ontologi sehingga dapat memberikan wawasan berharga dalam meningkatkan proses SPI.

RQ2: Tantangan dan Keterbatasan

Saat ini, aplikasi perangkat lunak yang diterapkan pada organisasi menjadi semakin kompleks dan beragam. Dengan demikian, pendekatan ontologi dalam bidang rekayasa perangkat lunak turut menghadapi sejumlah tantangan dan keterbatasan sebagai dampak dari kompleksitas tersebut sebagaimana disajikan pada Tabel 1.

Tabel 2. Tantangan Potensial Konsep Ontologi pada Rekayasa Perangkat Lunak

Domain	Tantangan Potensial
<i>software engineering process</i>	<ul style="list-style-type: none"> • model proses organik yang sesuai dengan karakteristik organisasi • metrik kuantitatif untuk menilai praktikabilitas proses
<i>software requirements</i>	<ul style="list-style-type: none"> • definisi ontologi pada notasi non-tekstual • inkonsistensi dan ambiguitas pada notasi terstruktur
<i>software architecture design</i>	<ul style="list-style-type: none"> • standarisasi dan konvensi notasi semantik • interoperabilitas semantik

Sumber: data diolah

Pada konteks ontologi dalam manajemen proses rekayasa perangkat lunak, sebuah kajian yang membahas tentang Ontology-based Software Process Assessment (SPA) mengungkapkan beberapa tantangan utama yang dihadapi. Sebagian besar studi tentang SPA pada dasarnya berkaitan dengan proses perangkat lunak yang bersifat generik (Tarhan & Giray, 2017). Sehingga salah satu keterbatasan yang muncul adalah kurangnya variasi mekanisme validasi terkait studi tersebut. CMMI (Capability Maturity Model Index) adalah model proses yang paling sering diteliti, sedangkan OWL (W3C Web Ontology Language) adalah bahasa representasi ontologi yang paling sering digunakan dalam konteks tersebut. Representasi tersebut seringkali mendeskripsikan manfaat secara kualitatif, akan tetapi tidak banyak studi yang berkontribusi terhadap aspek pengukuran kuantitatif untuk dapat menilai praktikabilitas dari representasi pengetahuan tersebut.

Berdasarkan pernyataan yang dikemukakan oleh Dermeval (2016), konsep ontologi masih dimanfaatkan secara terbatas pada fase rekayasa kebutuhan perangkat lunak. Studi tersebut menunjukkan bahwa ontologi biasanya lebih banyak digunakan bersama dengan rekayasa kebutuhan yang bersifat tekstual, yang karakteristiknya lebih rentan terhadap inkonsistensi dan ambiguitas daripada notasi pemodelan yang terstruktur. Dalam hal ini, salah satu keterbatasan pemanfaatan ontologi dalam ranah rekayasa perangkat lunak terletak pada keterbatasan kemampuan ontologi untuk beradaptasi dengan notasi pemodelan kebutuhan yang beragam, terutama yang bersifat non-tekstual.

Tantangan lain yang terkait dengan pengembangan perangkat lunak terletak pada kenyataan bahwa tim pengembang perangkat lunak seringkali tidak mengikuti rencana proyek yang telah ditetapkan. Hal ini cenderung menghasilkan produk aplikasi perangkat lunak yang tidak sesuai dengan kebutuhan pengguna. Salah satu pendekatan yang diusulkan terkait permasalahan tersebut adalah rumusan Ontology Development Life Cycle (ODLC) yang terdefinisi dengan baik (Triandini et al., 2021). Model tersebut berupaya untuk mengintegrasikan pengetahuan sistem ontologi, pengetahuan domain, dan praktik terbaik pengembangan perangkat lunak. Namun demikian, model yang dihasilkan akan sangat tergantung dari kesesuaian dari metode, teknik, proses, dan praktik terhadap karakteristik organisasi yang berbeda.

RQ3: Tren Penelitian Masa Depan

Berdasarkan sejumlah studi yang telah dikaji di atas, model pengembangan perangkat lunak berbasis ontologi akan semakin diterima secara luas di masa depan. Salah satu area studi yang dapat ditelusuri lebih lanjut adalah pemanfaatan ontologi dalam pengembangan sistem multi-agen (*multi-agent system*, MAS) menggunakan arsitektur berbasis layanan (*service-oriented architecture*, SOA) (Pileggi et al., 2018). Pendekatan ontologi dapat digunakan untuk memfasilitasi proses verifikasi dan validasi secara otomatis, sehingga dapat memastikan konsistensi artefak sepanjang siklus pengembangan perangkat lunak.

Sejumlah penelitian saat ini masih berfokus pada model pengembangan tradisional dengan fase-fase yang berurutan (*waterfall*), sementara industri telah mengadopsi praktik pengembangan berbasis *agile*. Perlu penelitian lebih lanjut untuk menunjukkan manfaat yang jelas dari penggunaan ontologi dalam konteks *agile* tanpa menimbulkan kendala yang signifikan dalam proses pengembangan perangkat lunak. Lebih lanjut, para peneliti dapat melakukan studi eksperimental yang lebih luas dan komprehensif menggunakan kasus-kasus nyata di dunia industri. Upaya ini diharapkan dapat memberikan pemahaman yang lebih dalam kepada pengembang perangkat lunak mengenai peran ontologi dalam pemodelan pengetahuan sepanjang proses pengembangan perangkat lunak (Triandini et al., 2021). Untuk mencapai hal ini, terdapat kebutuhan untuk mengintegrasikan ontologi dengan berbagai metode dan aplikasi di berbagai domain dan organisasi.

Terkait aspek dalam manajemen pengetahuan dan kebutuhan dalam rekayasa perangkat lunak, sejumlah studi telah mencoba mengekstraksi informasi dari dokumen spesifikasi kebutuhan aplikasi perangkat lunak (*software requirements specification*, SRS). Namun, data yang tersedia untuk pelabelan semantik di domain kebutuhan perangkat lunak masih sangat sedikit (Wang, 2016). Dukungan ontologi dapat dimanfaatkan untuk memperkaya sumber daya kerangka semantik dalam domain identifikasi kebutuhan perangkat lunak. Dengan sumber daya data yang lebih besar, efektivitas metode ontologis dalam meningkatkan ekstraksi kebutuhan fungsional dapat diukur lebih lanjut. Selain itu, kerangka ontologi dapat dimanfaatkan untuk bekerja dengan struktur kalimat yang lebih kompleks dalam dokumen SRS memperkaya domain pengetahuan dalam konteks rekayasa kebutuhan perangkat lunak.

4. Kesimpulan

Bidang rekayasa perangkat lunak menghadapi tantangan berkelanjutan untuk dapat menemukan model metodologi yang cukup fleksibel untuk memenuhi perubahan kebutuhan pengguna, namun juga untuk meningkatkan produktivitas pengembang di saat yang sama. Dalam rangka mengoptimalkan aktivitas rekayasa perangkat lunak, menghubungkan pengetahuan dalam perangkat lunak dengan jaringan pengetahuan yang lebih luas menjadi penting. Konsep penggunaan ontologi dan web semantik menawarkan sejumlah cara untuk meningkatkan efisiensi pengembangan perangkat lunak. Pendekatan berbasis ontologi dapat membantu mengurangi ambiguitas dan meningkatkan efisiensi pertukaran informasi di antara pemangku kepentingan. Integrasi sistem ontologi, pengetahuan domain, dan praktik pengembangan perangkat lunak juga dapat disesuaikan lebih lanjut berdasarkan kebutuhan organisasi. Dalam hal ini, pendekatan ontologi dapat meningkatkan konsistensi informasi dan pengetahuan yang tersimpan di sepanjang siklus pengembangan perangkat lunak. Penelitian lebih lanjut masih diperlukan untuk dapat membuktikan kontribusi konkret penggunaan ontologi dalam penentuan model metodologi pengembangan perangkat lunak di berbagai bentuk organisasi. Hal ini menunjukkan pentingnya pemahaman terhadap perspektif ontologis dalam lanskap pengembangan perangkat lunak yang terus berubah.

5. Referensi

- Bhatia, M. P. S., & Beniwal, A. K. and R. (2016). Ontologies for Software Engineering: Past, Present and Future. *Indian Journal of Science and Technology*, 9(9), 1–16. <https://doi.org/10.17485/ijst/2016/v9i9/71384>
- Dermeval, D., Vilela, J., Bittencourt, I. I., Castro, J., Isotani, S., Brito, P., & Silva, A. (2016). Applications of ontologies in requirements engineering: A systematic review of the literature. *Requirements Engineering*, 21(4), 405–437. <https://doi.org/10.1007/s00766-015-0222-6>
- El-Deeb, A. (2022). Major Challenges Currently Facing the Software Industry. *ACM SIGSOFT Software Engineering Notes*, 47(3), 14–15. <https://doi.org/10.1145/3539814.3539818>
- Fitsilis, P., Gerogiannis, V., & Anthopoulos, L. (2014). Ontologies for Software Project Management: A Review. *Journal of Software Engineering and Applications*, 07(13), Article 13. <https://doi.org/10.4236/jsea.2014.713097>
- Gasevic, D., Kaviani, N., & Milanović, M. (2009). Ontologies and Software Engineering. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 593–615). Springer. https://doi.org/10.1007/978-3-540-92673-3_27
- Gyrard, A., Datta, S. K., & Bonnet, C. (2018). A survey and analysis of ontology-based software tools for semantic interoperability in IoT and WoT landscapes. *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 86–91. <https://doi.org/10.1109/WF-IoT.2018.8355091>
- Haav, H.-M. (2018). A Comparative Study of Approaches of Ontology Driven Software Development. *Informatica*, 29(3), 439–466.
- Henderson-Sellers, B. (2011). Bridging metamodels and ontologies in software engineering. *Journal of Systems and Software*, 84(2), 301–313. <https://doi.org/10.1016/j.jss.2010.10.025>
- Isotani, S., Ibert Bittencourt, I., Francine Barbosa, E., Dermeval, D., & Oscar Araujo Paiva, R. (2015). Ontology Driven Software Engineering: A Review of Challenges and Opportunities. *IEEE Latin America Transactions*, 13(3), 863–869. <https://doi.org/10.1109/TLA.2015.7069116>

- Lopes de Souza, P., Lopes de Souza, W., & Ferreira Pires, L. (2021). ScrumOntoBDD: Agile software development based on scrum, ontologies and behaviour-driven development. *Journal of the Brazilian Computer Society*, 27(1), 10. <https://doi.org/10.1186/s13173-021-00114-w>
- Machado, J. B., Isotani, S., Barbosa, A., Bandeira, J., Alcantara, W., Bittencourt, I., & Barbosa, E. F. (2016). OntoSoft Process: Towards an Agile Process for Ontology-Based Software. *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5813–5822. <https://doi.org/10.1109/HICSS.2016.719>
- Mejia, J., Muñoz, E., & Muñoz, M. (2016). Reinforcing the applicability of multi-model environments for software process improvement using knowledge management. *Science of Computer Programming*, 121, 3–15. <https://doi.org/10.1016/j.scico.2015.12.002>
- Pileggi, S. F., Lopez-Lorca, A. A., & Beydoun, G. (2018, January 1). Ontology in Software Engineering. *ACIS 2018 Proceedings*. Australasian Conferences on Information Systems (ACIS), Sydney, Australia. <https://aisel.aisnet.org/acis2018/92>
- Serna, E., Bachiller, O., & Serna, A. (2017). Knowledge meaning and management in requirements engineering. *International Journal of Information Management*, 37(3), 155–161. <https://doi.org/10.1016/j.ijinfomgt.2017.01.005>
- Tarhan, A., & Giray, G. (2017). On the Use of Ontologies in Software Process Assessment: A Systematic Literature Review. *Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering*, 2–11. <https://doi.org/10.1145/3084226.3084261>
- Triandini, E., Kristyanto, M. A., Rishika, R. V., & Rawung, F. (2021). A Systematic Literature Review of The Role of Ontology in Modeling Knowledge in Software Development Processes. *IPTEK The Journal for Technology and Science*, 32(3), Article 3. <https://doi.org/10.12962/j20882033.v32i3.12998>
- Vujasinovic, M., Ivezic, N., & Kulvatunyou, B. (2015). A survey and classification of principles for domain-specific ontology design patterns development. *Applied Ontology*, 10(1), 41–69. <https://doi.org/10.3233/AO-150140>
- Wang, Y. (2016). Automatic semantic analysis of software requirements through machine learning and ontology approach. *Journal of Shanghai Jiaotong University (Science)*, 21(6), 692–701. <https://doi.org/10.1007/s12204-016-1783-3>