

Implementasi Metodologi *Agile Software Development* pada Proyek Perangkat Lunak

Handrie Noprisson

Universitas Dian Nusantara, Jakarta, Indonesia

handrie.noprisson@dosen.undira.ac.id

Abstrak— Penelitian ini adalah tinjauan literatur sistematis yang menggunakan Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) sebagai metode untuk mengumpulkan literatur mengenai metodologi agile software development dari Google Scholar. Hasil dari penelitian ini adalah ringkasan dari 11 artikel yang berkaitan dengan Agile software development. Rekomendasi dari penelitian ini adalah penggunaan GLUX framework. GLUX framework mengintegrasikan Lean UX ke Scrum untuk meningkatkan keberlanjutan proses pengembangan perangkat lunak yang cepat. Hal ini bertujuan untuk mempromosikan pola pikir yang berpusat pada pengguna dan aktivitas UX kolaboratif selama proses pengembangan dengan menggunakan teknik gamifikasi. GLUX membahas tentang tim mandiri, menciptakan lingkungan yang memotivasi, dan membina kerja sama tim.

Abstract—This research is a systematic literature review using Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) as a method to gather literature on agile software development methodology from Google Scholar. The result of this study is a summary of 11 articles related to Agile software development. The recommendation of this study is the use of the GLUX framework. GLUX framework integrates Lean UX into Scrum to improve the sustainability of the rapid software development process. It aims to promote a user-centric mindset and collaborative UX activities during the development process using gamification techniques. GLUX is about self-reliant teams, creating a motivating environment, and fostering teamwork.

Keywords— PRISMA, GLUX framework, systematic review, agile software development

1 Pendahuluan

Dalam beberapa tahun terakhir, metodologi pengembangan perangkat lunak berkembang dari pendekatan tradisional menjadi pengembangan perangkat lunak yang fokus pada perubahan waktu dan *requirement*. Pendekatan yang akhir-akhir ini banyak digunakan adalah *agile software development*. Metodologi *agile software development* digunakan dalam proyek perangkat lunak untuk beradaptasi dengan persyaratan yang berubah dengan cepat dan memberikan perangkat lunak berkualitas tinggi dengan cepat. Metodologi ini, seperti *extreme programming* (XP), menekankan komunikasi, keterlibatan pelanggan, dan pengujian yang ringkas dan cepat untuk memastikan keberhasilan proyek [1]–[9].

Pendekatan *agile software development* memungkinkan tim proyek untuk beradaptasi lebih cepat dan lebih mudah dibandingkan dengan metodologi manajemen proyek tradisional, sehingga membuat pendekatan ini cocok untuk proyek dengan *requirement* yang berubah-ubah. Prinsip-prinsip utama *agile*

software development adalah pelaporan hasil perangkat lunak kerja cepat, kolaborasi antara pengembang dan *stakeholder* yang terus-menerus untuk memantau hasil teknis dan kualitas desain. Metode *agile software development* menguntungkan proyek dalam hal meningkatkan tingkat keberhasilan proyek, penerimaan pengguna, manajemen risiko, dan kemampuan untuk penyesuaian dengan perubahan *requirement* [1], [2], [10]–[12].

Semua jenis pengembangan perangkat lunak hampir memiliki siklus hidup yang serupa. Langkah pertama dari metodologi biasanya adalah elisitasi persyaratan (*requirement elicitation*). Pada langkah ini, pengembang dan pemangku kepentingan membuat dokumen *backlog*. *Backlog* ini diekstraksi untuk mencapai fungsionalitas sistem. Langkah selanjutnya adalah fase iterasi, termasuk perencanaan proyek, pengembangan, pengujian, dan langkah pengembangan dan rilis [6], [13]–[20].

Secara keseluruhan, metodologi *agile software development* merupakan pendekatan yang fleksibel dan efisien untuk manajemen proyek perangkat lunak, memungkinkan tim untuk memenuhi kebutuhan pelanggan dan memberikan perangkat lunak berkualitas tinggi di pasar yang berubah dengan cepat. Berdasarkan hal tersebut, penelitian ini bertujuan untuk melakukan tinjauan literatur sistematis mengenai *agile software development* untuk manajemen proyek perangkat lunak.

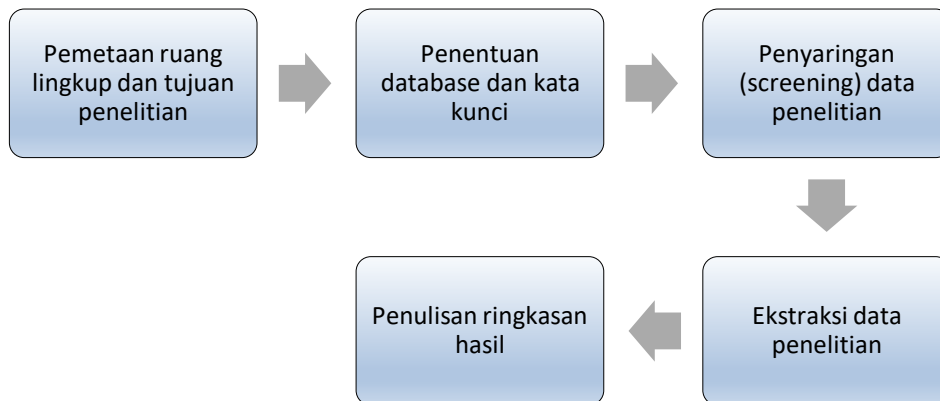
2 Metodologi Penelitian

Penelitian ini merupakan jenis penelitian *systematic literature review* (SLR) yang menggunakan metode *preferred reporting items for systematic reviews and meta-analyses* (PRISMA). Hasil dari penelitian ini adalah ringkasan peninjauan yang menggunakan beberapa protokol untuk meminimalkan bias dalam ulasan mengenai *agile software development*. Ringkasan disajikan menggunakan beberapa panduan agar lebih sistematis [21].

Langkah dalam penelitian ini adalah memetakan ruang lingkup tinjauan mengenai *agile software development*. Langkah kedua adalah melakukan penelitian dengan menggunakan kata kunci penelitian yang berhubungan dengan *agile software development*. Kata kunci ini akan direvisi jika kata kunci tidak menunjukkan hasil yang baik. Kata kunci terbaik yang diterapkan dalam SLR ini adalah (*agile* ATAU *software* ATAU *development* ATAU *software* ATAU *development* DAN (*agile software development*). Kata kunci tersebut digunakan untuk mencari data penelitian di Google Scholar.

Penilaian kualitas data penelitian dilakukan setelah tahap penyaringan. Tujuan dari penilaian kualitas ini adalah untuk memastikan bahwa artikel

sebagai data penelitian tersebut sesuai dengan topik *agile software development*. Data penelitian yang memenuhi kriteria penilaian akan diekstraksi untuk mendapatkan jawaban atas pertanyaan penelitian. Langkah selanjutnya adalah mensintesis data yang diekstraksi dan menulis hasil penelitian. Tahapan penelitian ini secara keseluruhan dapat dilihat pada **Gambar 1**.



Gambar 1. Tahapan Penelitian

3 Hasil dan Pembahasan

Metodologi agile software development telah menjadi subjek tinjauan literatur sistematis untuk memahami tantangan dan solusi dalam proyek perangkat lunak. Studi literatur digunakan untuk mengetahui tantangan yang dihadapi dalam proses *agile requirements engineering* dan bagaimana solusinya. Kombinasi antara *agile software development* (ASD) dengan *user-centered design* (UCD) telah dieksplorasi sebagai solusi untuk tantangan yang terkait dengan implementasi proses dan ruang lingkup serta persyaratan produk. Metodologi *agile software development* yang berbeda, seperti *Scrum* dan *extreme programming*, telah dianalisis untuk memahami karakteristik dan keunggulannya dibandingkan metodologi lain.

Pendekatan manajemen proyek dalam pengembangan perangkat lunak juga telah dipelajari, dengan fokus pada pengelolaan jadwal, kualitas, dan komunikasi. Selain itu, penelitian sebelumnya membahasnya merekomendasikan agile planning tools yang bertujuan untuk mengatasi tantangan perencanaan proyek di lingkungan terdistribusi. Hasil studi kelayakan artikel yang dikumpulkan dan relevan dengan ruang lingkup penelitian yang berkaitan dengan *agile software development* dapat dilihat pada **Tabel 1**.

Tabel 1. Hasil Data Penelitian

Sumber	Tahun	Penulis
[22]	2015	Larusdottir et al.
[23]	2015	Choma et al.
[24]	2015	Graphenthin et al.
[25]	2016	Sfetsos et al.
[26]	2016	Matthies et al.
[27]	2016	Lai et al.
[28]	2017	Dragicevic et al.
[29]	2017	Jabar et al.
[30]	2022	Sharma et al.
[31]	2022	Alhammad et al.
[32]	2023	Jain et al.

Metodologi *agile software development* telah banyak digunakan dalam proyek pengembangan perangkat lunak, khususnya di industri perangkat lunak, karena kemampuan metode ini untuk meningkatkan produktivitas, mengurangi risiko, dan meningkatkan kepuasan pelanggan. Metodologi *agile software development* dipandang lebih cocok untuk proyek kecil dan menengah, yang seringkali memiliki siklus pengiriman pendek dan membutuhkan fleksibilitas [29], [30], [32], [33].

Metodologi agile software development selain memiliki kelebihan dalam peningkatan kepuasan klien dan kerja sama yang lebih baik, namun metodologi ini memiliki tantangan dalam pelaksanaannya, termasuk dalam hal mengelola permintaan dari stakeholder dan memastikan partisipasi dan kualitas tim bekerja sesuai dengan target seharusnya [29], [30], [32], [33].

Penggunaan metodologi *agile software development*, seperti *Scrum* dan *Lean UX*, dalam kombinasi dengan desain yang berpusat pada pengguna dan desain pengalaman pengguna, telah diusulkan sebagai pendekatan yang efektif untuk pengembangan perangkat lunak. Pendekatan ini memungkinkan untuk produksi prototipe yang berguna difokuskan pada pengguna, dengan umpan balik konstan, dan telah berhasil diterapkan dalam proyek-proyek perangkat lunak [29], [30], [32], [33]. Salah satu penelitian yang menggunakan *Scrum* dan *Lean UX* adalah penelitian oleh Alhammad et al. (2022) seperti yang ditunjukkan pada **Gambar 2**.

4 Kesimpulan dan Saran

Metodologi *agile software development* adalah telah banyak dilakukan untuk mengeksplorasi isu-isu yang berkaitan komunikasi tim, pengembangan *agile*, dan proses pengembangan perangkat lunak. Penelitian ini merupakan jenis penelitian Systematic Literature Review Method (SLR) yang menggunakan metode *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA). Hasil dari penelitian ini adalah ringkasan peninjauan yang menggunakan beberapa protokol untuk meminimalkan bias dalam ulasan mengenai *agile software development*. Rekomendasi dari penelitian ini adalah penggunaan *GLUX framework*. *GLUX framework* mengintegrasikan *Lean UX* ke *Scrum* untuk meningkatkan keberlanjutan proses pengembangan perangkat lunak yang cepat. Hal ini bertujuan untuk mempromosikan pola pikir yang berpusat pada pengguna dan aktivitas UX kolaboratif selama proses pengembangan dengan menggunakan teknik gamifikasi. *GLUX* membahas tentang tim mandiri, menciptakan lingkungan yang memotivasi, dan membina kerja sama tim. Penerapan *Lean UX* dalam proyek skala besar memberikan manfaat namun terdapat tantangan penerapan terutama pada lingkungan kolaboratif pengembangan perangkat lunak yang kompleks.

5 Ucapan Terima Kasih

Terima kasih kepada Universitas Dian Nusantara yang telah mendukung pelaksanaan penelitian ini dan Deki Satria yang telah mendukung pengumpulan data penelitian ini.

6 Daftar Pustaka

- [1] T. Liu, "Application of Agile Project Management in Software R&D Management of M Enterprise," *Front. Comput. Intell. Syst.*, vol. 1, no. 3, pp. 85–87, 2022.
- [2] K. Pal, "eXtreme Programming, Agile Methodologies, Software Project Management, Customer Role, Rigorous Testing," pp. 163–182, 2022.
- [3] W. Gunawan, R. A. Wiradiputra, A. P. Sari, D. Prayama, and E. R. Nainggolan, "Prediction of Cross-Platform and Native Apps Technology Opportunities for Beginner Developers Using C 4.5 and Naive Bayes Algorithms," *JOIV Int. J. Informatics Vis.*, vol. 7, no. 4, pp. 2145–2153, 2023.
- [4] H. D. Wijaya and W. Gunawan, "Implementation of Analytic Network Process Algorithm in E-Lowker System," *J. Syst. Eng. Inf. Technol.*, vol. 1, no. 1, pp. 18–26, 2022.

-
- [5] H. D. Wijaya, W. Gunawan, R. Avrizal, and S. M. Arif, "Designing chatbot for college information management," *IJISCS (International J. Inf. Syst. Comput. Sci.)*, vol. 4, no. 1, pp. 8–13, 2020.
- [6] P. Sukmasetya, F. Nurhidayati, I. Permatasari, A. Rahmah, D. I. Sensuse, and H. Noprisson, "Developing mobile expert web-based system using brainstorming method: Case: Tetanus and botulism diagnosis and treatment in goat," in *2017 International Conference on Information Technology Systems and Innovation (ICITSI)*, 2017, pp. 303–308.
- [7] N. Ani, H. Noprisson, and N. M. Ali, "Measuring usability and purchase intention for online travel booking: A case study," *Int. Rev. Appl. Sci. Eng.*, vol. 10, no. 2, pp. 165–171, 2019.
- [8] V. Ayumi *et al.*, "Transfer Learning for Medicinal Plant Leaves Recognition: A Comparison with and without a Fine-Tuning Strategy," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 9, 2022.
- [9] M. Sadikin, D. Ramayanti, and A. P. Indrayanto, "The Graded CNN Technique to Identify Type of Food as The Preliminary Stages to Handle the Issues of Image Content Abundant," in *Proceedings of the 2020 12th International Conference on Computer and Automation Engineering*, 2020, pp. 108–113.
- [10] A. Khovrat and D. Teslenko, "The concept of agile software development and its main features," 2022.
- [11] R. Biloskursky, "Conceptual fundamentals of agile project management of development and implementation of information systems," *Nauk. visnik Užgorods'kogo nacional'nogo universitetu*, no. 41, 2022.
- [12] N. H. M. Ali *et al.*, "Agile Project Management Software for Construction and Management Industries," pp. 101–111, 2021.
- [13] A. Read, N. Arreola, and R. O. Briggs, "The Role of the Story Master : A Case Study of the Cognitive Load of Story Management Tasks," pp. 1–10, 2011.
- [14] D. Avison and G. Fitzgerald, *Information Systems Development Advances in Methodologies , Components , and Management*, 1st ed. 2002.
- [15] C. Schmidt, *Agile Software Development Teams*. 2016.
- [16] C. Glennon, "An Overview of Agile Development Methods for Engineers and Tech Professionals," *ieeusa*, 2012. .
- [17] M. Utami, E. D. Putra, and Y. Apridiansyah, "Survey Paper: Perbandingan Agile Process Development Method (SCRUM dan RUP)," *JSAI (Journal Sci. Appl. Informatics)*, vol. 5, no. 2, pp. 113–116, 2022.
- [18] D. Ramayanti, Y. Jumaryadi, D. M. Gufron, and D. D. Ramadha, "Sistem Keamanan Perumahan Menggunakan Face Recognition," *TIN Terap. Inform. Nusant.*, vol. 3, no. 12, pp. 486–496, 2023.

- [19] Y. Jumaryadi, D. Firdaus, B. Priambodo, and Z. P. Putra, "Determining the Best Graduation Using Fuzzy AHP," in *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)*, 2020, pp. 59–63.
- [20] H. Noprisson and Budiyarti, "Aplikasi Manajemen Pemeliharaan Produk Perangkat Lunak," *J. Sci. Appl. Informatics*, vol. 1, no. 2, pp. 41–45, 2018.
- [21] J. K. Jesson, L. Matheson, and F. M. Lacey, *Doing Your Literature Traditional and Systematic Techniques*. London: SAGE, 2011.
- [22] M. Larusdottir and J. Gulliksen, "A license to kill – Improving UCSD in Agile development," *J. Syst. Softw.*, vol. 123, pp. 214–222, 2017.
- [23] J. Choma, L. A. . Zaina, and T. S. da Silva, "Towards an Approach Matching CMD and DSR to Improve the Academia-Industry Software Development Partnership," in *Brazilian Symposium on Software Engineering*, 2015, pp. 51–60.
- [24] S. Grapenthin, S. Poggel, M. Book, and V. Gruhn, "Improving task breakdown comprehensiveness in agile projects with an Interaction Room," *Inf. Softw. Technol.*, vol. 67, pp. 254–264, 2015.
- [25] P. Sfetsos, L. Angelis, I. Stamelos, and P. Raptis, "Integrating User-Centered Design Practices into Agile Web Development: A Case Study," in *Information, Intelligence, Systems & Applications (IISA)*, 2016, pp. 13-15 July 2016.
- [26] C. Matthies, T. Kowark, K. Richly, M. Uflacker, and H. Plattner, "ScrumLint: Identifying Violations of Agile Practices Using Development Artifacts," pp. 40–43, 2016.
- [27] S. Lai, "A Version Control-based Continuous Testing Frame for Improving the IID Process Efficiency and Quality," pp. 464–469, 2016.
- [28] S. Dragicevic, S. Celar, and M. Turic, "The Journal of Systems and Software Bayesian network model for task effort estimation in agile software development," *J. Syst. Softw.*, vol. 127, pp. 109–119, 2017.
- [29] M. A. Jabar, N. M. Ali, Y. Y. Jusoh, S. Abdullah, S. Mohanarajah, and N. M. Nawati, "Agile Methodologies and The Use of Its Waterscrumfall Derivative For Software Project Development," vol. 1, no. 2, pp. 11–12, 2017.
- [30] P. Sharma, "Impact of Agile Methodologies in Project Success," *Int. J. Sci. Technol. Eng.*, vol. 10, no. 5, pp. 3728–3730, 2022.
- [31] M. M. Alhammad and A. Moreno, "Integrating User Experience into Agile : An Experience Report on Lean UX and Scrum," pp. 146–157, 2022.
- [32] S. Jain, "Software Engineering: Agile Software Development," *Int. J. Adv. Res. Sci. Commun. Technol.*, pp. 101–104, 2023.
- [33] C. S. González-González, P. Toledo-Delgado, and V. Muñoz-Cruz, "Agile human centered methodologies to develop educational software," *Dyna*, vol. 82, no. 193, pp. 187–194, 2015.

7 Penulis



Handrie Noprisson adalah dosen Teknik Informatika di Fakultas Teknik dan Informatika, Universitas Dian Nusantara Jakarta. Beliau meraih gelar magister dari Universitas Indonesia. Bidang penelitian beliau adalah *Image Processing* dan *Software Development*. Beliau dapat dihubungi di alamat email: handrie.noprisson@dosen.undira.ac.id