

PENGGUNAAN SISTEMATIS LITERATUR REVIEW UNTUK MENGANALISIS METODOLOGI PENGEMBANGAN MODEL PROSES PERANGKAT LUNAK DI BIDANG REKAYASA PERANGKAT LUNAK

Liskania Aprilia¹, Jihan Ananda², Adnan Affandhy³, Randy Iqbal Putra⁴, Ozin Ardianto⁵
Teknik Informatika, Universitas Pelita Bangsa. Jl. Inpeksi Kalimalang No.9 Cibatu, Cikarang Selatan, Kabupaten Bekasi, Jawa Barat 17530

Abstrak (Indonesia)

Penelitian ini bertujuan untuk mengevaluasi berbagai metodologi pengembangan perangkat lunak dengan pendekatan Systematic Literature Review (SLR). Permasalahan utama yang mendasari penelitian ini adalah adanya berbagai metodologi pengembangan perangkat lunak yang terus berkembang, namun belum ada pemahaman yang komprehensif mengenai tren, praktik terbaik, dan tantangan dalam penerapannya.

Metode SLR digunakan dalam penelitian ini untuk mengidentifikasi tren utama, praktik terbaik, dan tantangan yang muncul dalam pengembangan perangkat lunak. Kami menganalisis berbagai SLR yang dilakukan di bidang utama seperti metodologi pengembangan perangkat lunak, pengujian perangkat lunak dengan model Waterfall, dan praktik pemeliharaannya. Proses peninjauan mencakup penentuan pertanyaan penelitian, pemilihan studi yang relevan, dan penerapan kriteria inklusi dan eksklusi yang ketat untuk memastikan keandalan temuan.

Hasil analisis menunjukkan bahwa metode Agile dan DevOps menjadi yang paling dominan, dengan fokus pada pendekatan iteratif dan kolaborasi lintas fungsi. Namun, tantangan seperti adaptasi terhadap perubahan kebutuhan pengguna dan keberlanjutan proyek jangka panjang tetap menjadi hambatan utama. Temuan kami menyoroti pentingnya SLR dalam membimbing praktisi dan peneliti dalam membuat keputusan, mempromosikan praktik terbaik, dan mengidentifikasi kesenjangan untuk penelitian di masa depan. Pada akhirnya, makalah ini menggarisbawahi nilai tinjauan sistematis dalam memajukan bidang rekayasa perangkat lunak dan meningkatkan kualitas produk perangkat lunak.

Sejarah Artikel

Submitted: 1 November 2024

Accepted: 9 November 2024

Published: 10 November 2024

Kata Kunci

Systematic Literature Review, metodologi pengembangan, perangkat lunak, rekayasa perangkat lunak, Agile, DevOps.

1. Pendahuluan

Rekayasa perangkat lunak, juga dikenal sebagai "*software engineering*", adalah disiplin ilmu yang kompleks dan terus berkembang di mana para insinyur perangkat lunak bekerja untuk merancang, mengembangkan, dan mempertahankan perangkat lunak berkualitas tinggi. Dalam era teknologi *modern*, kebutuhan akan perangkat lunak yang dapat diandalkan dan efektif semakin meningkat. Untuk memenuhi kebutuhan ini, berbagai metodologi dan model proses telah dikembangkan. Metode yang berbeda telah dirilis, mulai dari model konvensional seperti Waterfall hingga yang lebih fleksibel seperti *Agile*. Namun,

perlu diingat bahwa beberapa metode memiliki tingkat keberhasilan yang berbeda ketika diterapkan dalam situasi yang berbeda. Oleh karena itu, penting untuk memilih metode yang tepat.

Untuk melakukan penilaian dan analisis literatur yang ada secara sistematis dan menyeluruh, penilaian literatur sistematis (SLR) adalah pendekatan yang efektif. Peneliti dapat menemukan pola dengan SLR. Selain itu, SLR memungkinkan peneliti untuk memberikan rekomendasi berbasis bukti mengenai praktik terbaik dalam rekayasa perangkat lunak. Diharapkan dengan

metode ini, peneliti akan mendapatkan pemahaman yang lebih mendalam tentang seberapa efektif berbagai metode pengembangan perangkat lunak dan bagaimana hal itu berdampak pada peningkatan kualitas produk perangkat lunak.

Melalui SLR, tidak hanya informasi terkini dapat diakses, tetapi juga analisis kritis terhadap penelitian sebelumnya dapat dilakukan. Hal ini membantu dalam mengidentifikasi kesenjangan penelitian yang ada serta area yang memerlukan

2. Metodologi

Metodologi pengembangan perangkat lunak terdiri dari berbagai pendekatan yang dirancang untuk memenuhi kebutuhan spesifik dalam proses pengembangan, termasuk model-model terkenal seperti Waterfall, Agile, DevOps, dan beberapa metode lainnya. Setiap pendekatan memiliki karakteristik yang berbeda, yang secara signifikan mempengaruhi cara tim pengembang beroperasi dan berinteraksi dalam menghasilkan produk perangkat lunak.

2.1 Waterfall

Model Waterfall adalah pendekatan pengembangan perangkat lunak yang bersifat linier dan sekuensial. Dalam model ini, setiap tahap dalam proses pengembangan harus diselesaikan sepenuhnya sebelum melanjutkan ke tahap berikutnya. Proses ini dimulai dengan analisis kebutuhan, di mana tim pengembang berkolaborasi dengan pemangku kepentingan untuk mengumpulkan dan mendokumentasikan semua persyaratan sistem. Setelah analisis selesai, tim melanjutkan ke tahap desain sistem, di mana arsitektur

perhatian lebih lanjut. Dengan demikian, SLR berkontribusi pada pengembangan teori dan praktik dalam rekayasa perangkat lunak, serta mendorong inovasi dalam teknik dan alat yang digunakan oleh para insinyur perangkat lunak. Selain itu, hasil dari SLR dapat menjadi dasar bagi pengambilan keputusan strategis dalam proyek pengembangan perangkat lunak, memastikan bahwa pendekatan yang dipilih tidak hanya efektif tetapi juga efisien.

perangkat lunak dirancang dan spesifikasi teknis disusun.

Setelah fase desain, tahap implementasi dimulai, di mana pengembang menulis kode berdasarkan desain yang telah disepakati. Setelah penyelesaian kode, fase pengujian dilakukan untuk memastikan bahwa perangkat lunak berfungsi sesuai dengan persyaratan yang telah ditetapkan. Akhirnya, pada fase pemeliharaan, perangkat lunak yang telah diluncurkan akan terus dipantau dan diperbarui untuk memperbaiki bug atau menyesuaikan dengan perubahan kebutuhan pengguna.

Model Waterfall sangat sesuai untuk proyek dengan persyaratan yang jelas dan stabil, di mana perubahan selama proses pengembangan dapat menyebabkan kesulitan dan penundaan yang signifikan. Kelebihan dari pendekatan ini adalah kemudahan dalam manajemen proyek dan dokumentasi yang terstruktur dengan baik, karena setiap tahap memiliki hasil yang terdefinisi dengan jelas. Namun, kelemahan utama dari model ini adalah kurangnya fleksibilitas untuk beradaptasi dengan perubahan kebutuhan klien atau masalah yang muncul selama proses pengembangan. Jika ada kesalahan atau

perubahan dalam tahap awal, hal ini dapat mengakibatkan biaya tinggi dan waktu tambahan untuk memperbaiki masalah tersebut.

2.2 Agile

Pendekatan Agile merupakan metodologi yang bersifat iteratif dan inkremental, menekankan kolaborasi tim serta kemampuan untuk beradaptasi dengan perubahan. Dalam metodologi Agile, proyek dibagi menjadi iterasi kecil atau sprint—biasanya berlangsung antara satu hingga empat minggu—di mana setiap sprint menghasilkan bagian dari produk akhir yang dapat diuji dan dievaluasi oleh tim serta pemangku kepentingan. Pendekatan ini memungkinkan tim untuk merespons dengan cepat terhadap kebutuhan klien yang berubah serta memberikan kesempatan untuk umpan balik berkelanjutan dari pemangku kepentingan.

Salah satu prinsip dasar Agile adalah bahwa perubahan adalah hal yang wajar dan diharapkan dalam proses pengembangan perangkat lunak. Oleh karena itu, tim Agile secara aktif mencari umpan balik dari pengguna akhir setelah setiap sprint untuk memastikan bahwa produk yang dikembangkan sesuai dengan harapan mereka. Kelebihan utama dari Agile adalah kemampuannya untuk meningkatkan kepuasan pelanggan melalui keterlibatan aktif mereka dalam proses pengembangan.

Namun, tantangan yang sering dihadapi oleh tim Agile adalah perlunya komunikasi yang efektif dan disiplin dalam tim agar tetap fokus pada tujuan proyek. Tanpa komunikasi yang baik, risiko ketidakcocokan antara harapan pemangku kepentingan dan hasil akhir

dapat meningkat. Selain itu, meskipun Agile menawarkan fleksibilitas tinggi dalam menghadapi perubahan, hal ini juga memerlukan komitmen dari semua anggota tim untuk terlibat secara aktif dalam setiap iterasi.

2.3 DevOps

DevOps adalah pendekatan modern yang mengintegrasikan proses pengembangan (Development) dan operasi (Operations) dengan tujuan meningkatkan kecepatan serta kualitas pengiriman perangkat lunak. Metodologi ini mendorong kolaborasi lintas fungsi antara tim pengembang dan tim operasi untuk menciptakan siklus hidup perangkat lunak yang lebih efisien. Dengan menerapkan praktik otomatisasi dan pemantauan berkelanjutan, DevOps memungkinkan organisasi untuk merilis perangkat lunak lebih cepat sambil tetap menjaga standar kualitas yang tinggi.

Salah satu prinsip kunci dari DevOps adalah penerapan Continuous Integration (CI) dan Continuous Deployment (CD), di mana kode baru secara otomatis diuji dan diintegrasikan ke dalam basis kode utama secara berkala. Hal ini mengurangi risiko konflik kode dan memastikan bahwa perangkat lunak selalu dalam keadaan siap rilis. Selain itu, DevOps juga mendorong penggunaan infrastruktur sebagai kode (Infrastructure as Code/IaC), memungkinkan tim untuk mengelola infrastruktur TI menggunakan skrip otomatisasi daripada konfigurasi manual.

Kelebihan dari pendekatan DevOps termasuk peningkatan produktivitas tim, kolaborasi lebih baik antar fungsi, serta pengurangan waktu untuk menyelesaikan masalah setelah rilis produk. Namun, implementasi DevOps memerlukan

perubahan budaya organisasi yang signifikan serta investasi dalam alat dan teknologi baru untuk mendukung praktik-praktik tersebut.

3. Hasil dan Pembahasan

3.1 Tren Metodologi Pengembangan

Hasil analisis menunjukkan bahwa Agile telah menjadi salah satu metodologi yang paling banyak dibahas dalam literatur terbaru di bidang rekayasa perangkat lunak. Banyak studi mengindikasikan bahwa penerapan Agile dapat secara signifikan meningkatkan kolaborasi antar anggota tim dan responsivitas terhadap perubahan kebutuhan klien. Pendekatan ini memungkinkan tim untuk bekerja dalam siklus iteratif, yang memfasilitasi umpan balik cepat dan penyesuaian berkelanjutan berdasarkan masukan dari pemangku kepentingan. Misalnya, metode seperti Scrum dan Kanban, yang merupakan bagian dari kerangka kerja Agile, memberikan struktur yang memungkinkan tim untuk merespons perubahan dengan cepat dan efisien.

Namun, beberapa studi juga mengidentifikasi tantangan dalam penerapan Agile, terutama di organisasi besar dengan struktur hierarkis yang kaku. Dalam konteks ini, kesulitan sering muncul terkait dengan komunikasi lintas departemen dan pengambilan keputusan yang lambat. Ketidakpastian hasil akhir juga menjadi masalah, karena proyek Agile sering kali mengandalkan umpan balik terus-menerus dari pengguna akhir, yang mungkin tidak selalu tersedia atau terlibat secara aktif.

Di sisi lain, model Waterfall masih banyak digunakan, terutama pada proyek-proyek yang memiliki persyaratan yang stabil dan jelas. Model ini menawarkan pendekatan yang terstruktur dan mudah dipahami, di mana setiap fase harus diselesaikan sebelum melanjutkan ke fase berikutnya. Hal ini menjadikannya pilihan yang baik untuk proyek dengan spesifikasi tetap dan risiko perubahan yang rendah.

Sementara itu, DevOps muncul sebagai pendekatan baru yang mengintegrasikan pengembangan perangkat lunak (Development) dan operasi (Operations) dengan tujuan meningkatkan kecepatan rilis produk. DevOps mendorong kolaborasi antara tim pengembang dan tim operasi melalui otomatisasi proses dan integrasi berkelanjutan. Pendekatan ini memungkinkan organisasi untuk merilis perangkat lunak dengan lebih cepat tanpa mengorbankan kualitas, serta mempercepat umpan balik dari pengguna.

3.2 Kelebihan dan Kelemahan

Kelebihan SLR dalam Pengembangan Model Proses Perangkat Lunak:

1. **Sumber Pengetahuan yang Komprehensif:** SLR mengumpulkan penelitian-penelitian yang relevan secara sistematis dan menyeluruh, memungkinkan pengumpulan informasi yang luas tentang model proses perangkat lunak yang telah diuji dan diterapkan di berbagai konteks. Hal ini memberikan gambaran yang lebih lengkap tentang praktik-praktik

- terbaik dan efektivitas model-model yang ada.
2. **Identifikasi Tren dan Gap Penelitian:** Dengan melakukan review secara sistematis, SLR dapat mengidentifikasi tren yang berkembang dalam pengembangan model proses perangkat lunak, serta menemukan area-area yang masih kurang dieksplorasi. Ini membuka peluang bagi penelitian lebih lanjut untuk mengisi kekosongan tersebut dan mendorong inovasi dalam pengembangan model.
 3. **Dasar Pengambilan Keputusan yang Berbasis Bukti:** SLR memberikan bukti yang kuat untuk mendukung keputusan-keputusan praktis yang diambil oleh pengembang perangkat lunak atau manajer proyek dalam memilih atau mengadaptasi model proses. Keputusan berbasis bukti ini lebih kredibel dan dapat meningkatkan peluang keberhasilan proyek.
 4. **Pengurangan Bias Subjektif:** SLR dilakukan dengan prosedur yang ketat dan transparan, yang dapat mengurangi bias subjektif dalam penilaian studi dan interpretasi hasil penelitian. Ini memastikan bahwa kesimpulan yang ditarik adalah objektif dan lebih dapat diandalkan.
 5. **Reproduksibilitas dan Validitas:** Karena prosesnya terstruktur dan jelas, SLR memberikan metode yang dapat direproduksi oleh peneliti lain untuk memastikan validitas dan reliabilitas temuan-temuan yang dihasilkan. Hal ini memperkuat keandalan hasil SLR sebagai referensi dalam penelitian dan praktik.

Kekurangan SLR dalam Pengembangan Model Proses Perangkat Lunak:

1. **Proses yang Memakan Waktu dan Sumber Daya:** Proses SLR membutuhkan waktu yang cukup lama, mulai dari pencarian literatur, seleksi studi, hingga analisis dan sintesis data. Ini seringkali menjadi hambatan besar, terutama dalam konteks pengembangan perangkat lunak yang bergerak cepat, di mana hasil yang cepat diperlukan untuk pengambilan keputusan yang cepat pula.
2. **Keterbatasan dalam Menilai Kualitas Studi:** Meskipun SLR berfokus pada pemilihan studi berdasarkan kriteria tertentu, seringkali terdapat variasi dalam kualitas studi yang disertakan. Kualitas metodologi yang digunakan dalam berbagai penelitian bisa sangat bervariasi, dan ini bisa mempengaruhi kualitas kesimpulan yang ditarik. SLR harus cermat dalam menilai kualitas setiap studi yang dimasukkan.
3. **Potensi Bias Publikasi:** Bias publikasi adalah masalah yang signifikan dalam SLR. Studi dengan hasil yang positif cenderung lebih sering dipublikasikan, sementara studi dengan hasil negatif atau tidak signifikan mungkin tidak dipublikasikan. Hal ini dapat mengarah pada kesimpulan yang terlalu optimistik atau tidak realistis mengenai efektivitas suatu model proses.
4. **Heterogenitas dan Variabilitas Studi:** Model proses perangkat lunak sangat bervariasi, dan setiap model sering kali diterapkan dalam konteks yang berbeda, seperti skala proyek, jenis organisasi, atau budaya pengembangan. Variabilitas ini dapat menyulitkan untuk menyintesis temuan yang berbeda secara langsung dan membuat perbandingan antar studi menjadi lebih kompleks.
5. **Keterbatasan dalam Menerjemahkan Temuan ke dalam**

- ◆ **Praktik:** Meskipun SLR dapat mengidentifikasi model-model proses yang efektif di berbagai setting, ◆ menerjemahkan temuan tersebut ke dalam praktik di dunia nyata sering kali sulit. Konteks pengembangan perangkat lunak yang unik dalam organisasi atau proyek tertentu bisa mempengaruhi efektivitas model yang ditemukan melalui SLR.
6. **Keterbatasan Fokus pada Model Proses yang Ada:** SLR biasanya berfokus pada studi yang telah ada dan dipublikasikan, yang berarti bahwa pendekatan-pendekatan baru atau inovatif yang belum banyak diteliti bisa terlewatkan. SLR lebih sering mencerminkan pemahaman yang ada pada saat penelitian dilakukan dan mungkin tidak sepenuhnya mencakup model-model proses yang sedang berkembang.

Disisilain selain kelebihan dan kekurangan SLR metode metodologi setiap model juga memiliki kelebihan dan kekurangannya sebagai proses pengembangan untuk menghasilkan perangkat lunak. memudahkan manajemen proyek dan penjadwalan karena alur proses yang linier, serta memastikan kontrol kualitas dengan penyelesaian penuh di setiap tahap sebelum melanjutkan ke tahap berikutnya.

Namun Waterfall memiliki kelemahan dalam fleksibilitas; sulit untuk menyesuaikan kebutuhan setelah fase awal selesai, sehingga ketika penyesuaian diperlukan di tengah proyek, biaya dan waktu yang dibutuhkan untuk revisi bisa sangat tinggi. Selain itu, umpan balik dari pengguna akhir umumnya hanya diperoleh setelah seluruh proyek

selesai, yang dapat mengakibatkan hasil yang tidak sesuai ekspektasi pengguna (Royce, 1970).

Di sisi lain, Agile menawarkan adaptabilitas yang tinggi terhadap perubahan kebutuhan dan umpan balik pengguna dengan mengadopsi pendekatan iteratif dan inkremental. Setiap iterasi menghasilkan produk yang dapat diuji dan dievaluasi, sehingga risiko keseluruhan proyek dapat diminimalkan. Pendekatan ini juga mendorong kolaborasi yang lebih erat antara tim pengembang dan pemangku kepentingan. Namun, Agile menuntut disiplin tinggi dan komunikasi yang konsisten dalam tim, serta menghadirkan ketidakpastian dalam hasil akhir karena sangat bergantung pada umpan balik berkelanjutan. Penerapannya di organisasi besar dengan struktur yang hierarkis juga seringkali menantang (Beck et al., 2001).

Sementara itu, DevOps berfokus pada peningkatan kecepatan dan kualitas melalui otomatisasi, integrasi, dan kolaborasi lintas fungsi antara tim pengembang dan operasi. Penggunaan otomatisasi dalam proses integrasi dan pengiriman berkelanjutan (CI/CD) membantu mengurangi risiko kesalahan manusia serta meningkatkan efisiensi kerja. Namun, adopsi DevOps sering memerlukan perubahan besar dalam budaya organisasi dan investasi signifikan dalam alat serta teknologi pendukung. Kompleksitas implementasi, khususnya dalam hal CI/CD dan infrastructure as code, menjadi tantangan utama bagi organisasi yang ingin mengadopsi

pendekatan ini secara efektif (Kim et al., 2016).

4. Diskusi

1. Waterfall

Model Waterfall cocok untuk proyek dengan persyaratan yang stabil dan jelas karena prosesnya yang linier dan terstruktur. Kelebihan utamanya terletak pada kemudahan manajemen dan dokumentasi yang tertata di setiap tahap. Setiap fase harus diselesaikan sebelum berlanjut ke tahap berikutnya, memberikan struktur yang memastikan semua langkah terpenuhi. Namun, kekurangan terbesar Waterfall adalah kurangnya fleksibilitas. Model ini sulit beradaptasi terhadap perubahan kebutuhan yang muncul di tengah proses, sehingga revisi biasanya memakan waktu dan biaya tinggi. Keterbatasan umpan balik dari pengguna akhir sebelum produk selesai juga meningkatkan risiko ketidaksesuaian hasil akhir dengan harapan pengguna.

2. Agile

Agile menjadi pilihan populer dalam pengembangan perangkat lunak modern karena pendekatan iteratif dan inkrementalnya yang fleksibel. Dengan pembagian proyek ke dalam iterasi atau sprint pendek, Agile memungkinkan perubahan yang lebih responsif dan adaptasi terhadap kebutuhan yang berkembang. Setiap sprint menghasilkan bagian produk yang dapat diuji, membantu tim mengidentifikasi kesalahan sejak dini dan mendapatkan umpan balik cepat dari pemangku kepentingan. Namun, tantangan utama Agile terletak pada kebutuhan disiplin dan komunikasi yang konsisten dalam tim. Pendekatan ini juga lebih menantang di organisasi besar yang memiliki hierarki kaku, karena struktur semacam itu dapat menghambat kelincuhan yang diinginkan.

3. DevOps

DevOps menyatukan proses pengembangan dan operasi, memungkinkan pengembangan perangkat lunak yang lebih cepat dan efisien melalui integrasi, otomatisasi, dan kolaborasi lintas fungsi. Prinsip utama DevOps, yaitu Continuous Integration (CI) dan Continuous Deployment (CD), memungkinkan perangkat lunak selalu siap untuk dirilis karena pengujian dan pembaruan dilakukan secara berkala. Penggunaan otomatisasi dalam DevOps mempercepat proses rilis dan mengurangi risiko kesalahan manusia. Namun, penerapan DevOps sering kali membutuhkan perubahan budaya organisasi yang signifikan, investasi dalam alat, serta komitmen jangka panjang untuk penerapan CI/CD secara efektif.

Kontribusi SLR dalam Pemahaman Metodologi Pengembangan

Systematic Literature Review (SLR) memungkinkan para peneliti untuk memperoleh pemahaman yang komprehensif tentang efektivitas setiap metodologi di berbagai konteks. Dengan mengumpulkan dan menganalisis literatur yang relevan, SLR membantu mengidentifikasi tren, praktik terbaik, serta kesenjangan penelitian yang memerlukan perhatian lebih lanjut. Dalam konteks pengembangan perangkat lunak, SLR memungkinkan peneliti memberikan rekomendasi berbasis bukti yang membantu manajer proyek memilih metodologi yang sesuai dengan kebutuhan spesifik proyek mereka. Misalnya, SLR dapat menunjukkan bahwa Agile cocok untuk tim kecil dengan kebutuhan yang dinamis, sementara Waterfall cocok untuk proyek dengan spesifikasi yang tetap.

Namun, SLR juga memiliki keterbatasan. Prosesnya yang memakan waktu dan berisiko bias publikasi menuntut perhatian ekstra dalam seleksi dan penilaian kualitas literatur

yang disertakan. Selain itu, SLR kadang kurang mampu menerjemahkan temuan ke dalam praktik nyata karena perbedaan konteks pengembangan dalam tiap organisasi.

5. Kesimpulan

Dalam memilih metodologi pengembangan perangkat lunak, penting untuk mempertimbangkan karakteristik unik dari

6. Daftar Pustaka

1. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic Literature Reviews in Software Engineering. DOI: 10.1016/j.jss.2008.02.002.
2. Wahono, R.S. (2015). A Systematic Literature Review of Software Defect Prediction: Research Trends, Datasets, Methods and Frameworks. *Ilmu Komputer*, 14(2), 141-154.
3. Radjenovic, D., Hericko, M., Torkar, R., & Zivkovic, A. (2013). Evaluation of Maintenance Effort Predictions Using Object-Oriented Metrics. *Information and Software Technology*, 55(8), 1445-1456.
4. Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N.B. (2012). A decade of Agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
5. Forsgren, N., Humble, J., & Kim, G. (2018). *Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations*. IT Revolution Press.
6. Babbie, E.R., & Mouton, J. (2001). *The Practice of Social Research*. Oxford University Press.
7. Boehm, B.W., & Turner, R.J. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*. Addison-Wesley Professional.
8. Loud. 2023. Agile Methodology: Benefits And Challenges For Engineering Leaders. Diakses 6 November 2024. <https://www.forbes.com/councils/forbestech/council/2023/07/19/agile-methodology-benefits-and-challenges-for-engineering-leaders/>
9. Olic. 2024. Advantages and Disadvantages of Agile Project Management. Diakses 6 November 2024. <https://activecollab.com/blog/project-management/agile-project-management-advantages-disadvantages>