

EVALUASI ALAT PENDUKUNG PADA MANAJEMEN PROYEK PERANGKAT LUNAK : UNTUK PENGEMBANGAN PERANGKAT LUNAK

Kiangeng Nasrokh Mangkunegara Putra¹, Muhammad Andrean Hidayatullah², Muhammad Ainul Yaqin³

^{1,2,3}Teknik Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang,
Email: ¹220605110109@student.uin-malang.ac.id, ²220605110126@student.uin-malang.ac.id,
³yaqinov@ti.uin-malang.ac.id
*Penulis Korespondensi

Abstrak

Manajemen proyek perangkat lunak merupakan aspek krusial dalam pengembangan perangkat lunak yang efektif dan efisien. Salah satu tantangan utama dalam manajemen proyek perangkat lunak adalah pemilihan alat pendukung yang tepat untuk memastikan keberhasilan proyek. Penelitian ini mengevaluasi berbagai alat pendukung manajemen proyek perangkat lunak dengan tujuan mengidentifikasi alat yang paling efektif berdasarkan kriteria tertentu seperti kemudahan penggunaan, fitur yang disediakan, integrasi dengan alat lain, dan dukungan teknis. Penelitian ini mengusulkan solusi berupa penggunaan metodologi evaluasi berbasis kriteria yang telah ditentukan, dan kemudian menguji beberapa alat populer seperti Jira, Trello, Asana, Docker dan Microsoft Project. Hasil eksperimen menunjukkan bahwa Jira telah terbukti menjadi pilihan terbaik dengan skor tertinggi secara keseluruhan. Namun, Asana dan Microsoft Project juga menunjukkan performa yang baik dalam beberapa kategori, sementara Docker dan Trello cenderung memiliki skor lebih rendah. Tafsiran hasil ini menunjukkan Pemilihan alat manajemen proyek harus didasarkan pada analisis yang cermat terhadap kebutuhan spesifik proyek dan kriteria yang paling penting bagi tim yang terlibat. Oleh karena itu, keberhasilan implementasi alat bantu bergantung pada pemahaman yang mendalam tentang proses pengembangan perangkat lunak, dinamika tim, dan keterampilan manajemen yang kuat.

Kata kunci: *Manajemen Proyek Perangkat Lunak, Alat Pendukung, Evaluasi Alat, Pengembangan Perangkat Lunak.*

Abstract

Software project management is a crucial aspect of effective and efficient software development. One of the main challenges in software project management is selecting the right support tools to ensure project success. This study evaluates various software project management support tools to identify the most effective tools based on specific criteria such as ease of use, features provided, integration with other tools, and technical support. The study proposes a solution by using a criteria-based evaluation methodology and then testing several popular tools such as Jira, Trello, Asana, Docker, and Microsoft Project. The experimental results show that Jira proved to be the best choice with the highest overall score. However, Asana and Microsoft Project also performed well in certain categories, while Docker and Trello tended to score lower. These results indicate that the selection of project management tools should be based on a careful analysis of the specific needs of the project and the most important criteria for the team involved. Therefore, the successful implementation of support tools depends on a deep understanding of the software development process, team dynamics, and strong management skills.

Keywords: *Software Project Management, Support Tools, Tool Evaluation, Software Development.*

1. PENDAHULUAN

Manajemen proyek perangkat lunak bukanlah sekadar pengelolaan waktu dan sumber daya. Ini adalah perpaduan kompleks antara seni dan ilmu yang membutuhkan pemahaman yang mendalam tentang proses pengembangan perangkat lunak, dinamika tim, dan keterampilan manajemen yang kuat[1]. Dengan proyek-proyek yang semakin besar dan kompleksitas yang meningkat, manajemen proyek perangkat lunak menjadi semakin penting dalam

memastikan pengiriman produk perangkat lunak yang berkualitas, tepat waktu, dan dalam batas biaya yang ditetapkan[2].

Manajemen proyek perangkat lunak adalah disiplin yang penting dalam pengembangan produk perangkat lunak yang kompleks. Ini melibatkan proses perencanaan, pengorganisasian, pengelolaan, dan pengendalian sumber daya untuk mencapai tujuan proyek dalam batasan waktu, anggaran, dan kualitas yang ditetapkan. Sebuah pendekatan yang

sistematis dan terstruktur diperlukan untuk mengelola proyek perangkat lunak dengan efektif[3]. Definisi dan konsep dasar tentang manajemen proyek perangkat lunak telah didefinisikan oleh para ahli seperti Pressman dan Pinto, serta praktisi industri seperti Kerzner dan Wysocki.

Sebagai bagian integral dari manajemen proyek perangkat lunak, penggunaan alat bantu menjadi semakin diperlukan. Alat-alat ini, yang mencakup berbagai aplikasi perangkat lunak dan platform kolaboratif, dirancang untuk membantu manajer proyek dan tim pengembang dalam merencanakan, mengatur, dan mengendalikan proyek dengan lebih efisien. Dari pelacakan tugas hingga manajemen risiko, alat bantu ini menyediakan berbagai fitur yang memungkinkan manajer proyek untuk memantau progres proyek secara real-time dan membuat keputusan yang terinformasi.

Namun, memilih alat bantu manajemen proyek yang tepat tidak selalu mudah. Setiap proyek memiliki kebutuhan uniknya sendiri, dan tidak semua alat bantu cocok untuk setiap situasi. Kriteria seperti skala proyek, kompleksitas, kebutuhan kolaborasi, dan anggaran harus dipertimbangkan dengan cermat saat memilih alat yang sesuai[4]. Terlebih lagi, lingkungan pengembangan perangkat lunak yang terus berubah dan berkembang memperumit proses pemilihan.

Evaluasi alat pendukung dalam manajemen proyek perangkat lunak adalah langkah penting untuk meningkatkan efisiensi dan efektivitas proyek. Alat manajemen proyek yang efektif dapat membantu tim perangkat lunak dalam berbagai aspek, mulai dari perencanaan, pelacakan, hingga pelaporan. Dengan memahami kriteria evaluasi dan best practices yang ada, pengembang perangkat lunak dapat memilih dan mengimplementasikan alat yang paling sesuai dengan kebutuhan mereka, yang pada akhirnya berkontribusi pada kesuksesan proyek[3].

Masalah utama yang dihadapi oleh manajer proyek adalah kekurangan fleksibilitas dan skalabilitas dalam alat bantu yang ada. Beberapa alat mungkin terlalu kompleks untuk proyek-proyek kecil, sementara yang lain mungkin tidak memiliki fitur yang memadai untuk proyek-proyek besar dan kompleks. Selain itu, biaya dan ketersediaan dukungan teknis juga menjadi faktor yang harus dipertimbangkan[5].

Dalam menghadapi tantangan kompleksitas pemilihan alat bantu manajemen proyek perangkat lunak, pendekatan metodologis yang terstruktur diperlukan untuk memastikan evaluasi yang komprehensif dan akurat. Oleh karena itu, penelitian ini akan mengadopsi metode penelitian yang mengacu pada protokol yang telah ditetapkan oleh Kitchenham untuk survei dalam bidang rekayasa perangkat lunak.

Protokol ini, yang terutama digunakan dalam melakukan Tinjauan Literatur Sistematis (SLR), menyediakan kerangka kerja yang kokoh untuk

melakukan evaluasi menyeluruh terhadap berbagai alat bantu manajemen proyek perangkat lunak[4]. Salah satu langkah awal dalam implementasi protokol ini adalah merancang strategi pencarian yang tepat untuk mengidentifikasi literatur yang relevan dan terkait dengan subjek penelitian.

Strategi pencarian akan didasarkan pada kata kunci yang relevan, yang akan digunakan untuk menyaring artikel-artikel yang relevan dari berbagai sumber seperti jurnal ilmiah, konferensi, dan buku referensi[6]. Selanjutnya, artikel-artikel yang telah diidentifikasi akan dinilai secara kualitatif untuk mengevaluasi kualitas metodologi penelitian dan relevansi dengan topik penelitian ini. Selain itu, penelitian ini akan melibatkan perancangan skema karakterisasi. Skema karakterisasi ini akan mencakup variabel-variabel yang relevan untuk evaluasi alat bantu manajemen proyek, seperti fitur-fitur yang disediakan, fleksibilitas, skalabilitas, biaya, dan ketersediaan dukungan teknis[7].

Dengan mengadopsi metode penelitian yang terstruktur ini, diharapkan penelitian ini dapat memberikan pemahaman yang lebih dalam tentang berbagai alat bantu manajemen proyek perangkat lunak yang tersedia, serta memberikan dasar yang kuat untuk evaluasi yang obyektif dan komprehensif.

2. PENELITIAN TERKAIT

Dalam beberapa tahun terakhir, berbagai karya telah disajikan untuk menganalisis dan membandingkan beberapa alat pendukung pada manajemen proyek perangkat lunak.

Penelitian terdahulu yang dilakukan oleh Sutanto (2017) menunjukkan bahwa penggunaan alat pendukung pada manajemen proyek perangkat lunak dapat meningkatkan efisiensi dalam pengembangan perangkat lunak. Penelitian tersebut juga menyoroti pentingnya pemilihan alat yang sesuai dengan kebutuhan proyek untuk mencapai kesuksesan proyek perangkat lunak.

Selain itu, penelitian yang dilakukan oleh Wijaya (2018) menunjukkan bahwa evaluasi alat pendukung pada manajemen proyek perangkat lunak dapat meningkatkan efektivitas dalam pengelolaan proyek. Penelitian tersebut juga menyoroti pentingnya komunikasi yang baik antara tim pengembang dalam penggunaan alat pendukung.

Miguel dan Luisa (2023) melakukan penelitian untuk menganalisis dan mengevaluasi alat manajemen proyek dari kuadran Gartner Leader (2019 Gartner Magic Quadrant) terkait kemampuannya dalam menangkap, menyimpan, berbagi, dan menerapkan pengetahuan sesuai dengan artefak di PMBOK. Mereka membandingkan alat-alat pemimpin Gartner dengan Confluence, yang dianggap sebagai pilihan unggul untuk manajemen pengetahuan dan dokumen proyek. Metodologi yang digunakan untuk mengembangkan artefak adalah Design Science Research (DSR). Hasil penelitian

menunjukkan bahwa aplikasi dengan skor tertinggi adalah Targetprocess[8].

Muhammad Sajad dan Muhammad Sadiq (2016) melakukan penelitian yang berfokus pada rekayasa perangkat lunak, dengan penekanan pada pengembangan sistem perangkat lunak yang besar dan kompleks. Penelitian ini menyoroti pentingnya penggunaan alat perangkat lunak otomatis dalam perencanaan dan manajemen proyek yang sukses. Studi ini mengevaluasi berbagai alat perangkat lunak otomatis yang dikembangkan untuk tujuan manajemen proyek yang berbeda. Makalah ini memprediksi kualitas alat manajemen proyek dan menyarankan standarisasi untuk pengembangan perangkat lunak di masa depan dalam manajemen proyek. Penelitian ini bertujuan untuk meningkatkan pemahaman tentang penilaian kualitas alat manajemen proyek perangkat lunak dan standarisasi untuk pengembangan masa depan, mengatasi kebutuhan kritis akan alat yang efektif dalam mengelola proyek perangkat lunak modern[9].

Setelah meninjau literatur penelitian sebelumnya, pembaruan dalam penelitian ini adalah mengevaluasi beberapa alat pendukung manajemen proyek yang mendukung pengembangan perangkat lunak. Evaluasi ini akan fokus pada kemampuan alat-alat tersebut dalam berbagai aspek seperti kemudahan penggunaan, fungsionalitas, integrasi dengan sistem lain, dan dukungan pengguna. Selain itu, penelitian ini akan mempertimbangkan bagaimana alat-alat ini mendukung praktik-praktik terbaik dalam manajemen proyek perangkat lunak.

3. METODOLOGI PENELITIAN

Metodologi penelitian yang digunakan dalam penelitian ini didasarkan pada pendekatan yang terstruktur dan komprehensif, mengacu pada protokol yang ditetapkan oleh Kitchenham untuk survei dalam bidang rekayasa perangkat lunak[4]. Pendekatan ini telah terbukti efektif dalam melakukan Tinjauan Literatur Sistematis (SLR) dan memberikan dasar yang kuat untuk evaluasi menyeluruh terhadap berbagai alat bantu manajemen proyek perangkat lunak.

Langkah pertama dalam metodologi penelitian ini adalah merancang strategi pencarian yang tepat untuk mengidentifikasi literatur yang relevan dan terkait dengan subjek penelitian. Strategi pencarian

akan melibatkan pemilihan kata kunci yang sesuai dan penentuan sumber-sumber informasi yang akan diselidiki, seperti basis data jurnal, konferensi, dan buku referensi.

Setelah literatur yang relevan telah diidentifikasi, langkah selanjutnya adalah melakukan evaluasi kualitatif terhadap artikel-artikel tersebut. Evaluasi ini akan dilakukan untuk menilai kualitas metodologi penelitian yang digunakan dalam artikel, serta relevansinya dengan topik penelitian ini. Artikel-artikel yang memenuhi kriteria inklusi akan dipertimbangkan untuk analisis lebih lanjut. Skema karakterisasi ini akan mencakup variabel-variabel yang relevan untuk evaluasi alat bantu manajemen proyek perangkat lunak, seperti fitur-fitur yang disediakan, fleksibilitas, skalabilitas, biaya, dan ketersediaan dukungan teknis[10].

Analisis data akan dilakukan dengan cermat untuk mengidentifikasi kekuatan dan kelemahan dari masing-masing alat bantu yang dievaluasi. Hasil analisis akan disajikan secara sistematis dalam bentuk laporan, yang akan mencakup temuan-temuan utama serta rekomendasi untuk pemilihan alat bantu manajemen proyek perangkat lunak yang tepat.

Dengan menggunakan metodologi penelitian yang terstruktur ini, diharapkan penelitian ini dapat memberikan wawasan yang lebih dalam tentang berbagai alat bantu manajemen proyek perangkat lunak yang tersedia, serta memberikan dasar yang kuat untuk evaluasi yang obyektif dan komprehensif.

Pada Gambar 1 menunjukkan proses SLR yang telah kami ikuti dalam survei kami dan bagian selanjutnya menjelaskan secara rinci setiap tahapan yang terlibat. SLR ini pada dasarnya melibatkan tiga fase:

1. Perencanaan (*Planning*): Tahap ini melibatkan perencanaan tinjauan, pembuatan skema, dan karakterisasi. Ini adalah langkah awal di mana Anda menentukan topik penelitian, pertanyaan yang ingin dijawab, dan metode yang akan digunakan untuk mengumpulkan dan menganalisis data.
2. Pelaksanaan (*Implementation*): Pada fase ini, penelitian dilakukan dan data dikumpulkan. Ini termasuk pencarian literatur yang sistematis, seleksi studi yang relevan, dan ekstraksi data yang diperlukan dari studi tersebut.



Gambar 1. Tahapan Proses SLR

3. Pelaporan (*Reporting*): Langkah terakhir ini melibatkan sintesis data, penyajian hasil, dan penarikan kesimpulan. Di sini Anda akan menggabungkan temuan dari berbagai studi, menilai kualitas dan relevansi mereka, dan menyusun temuan tersebut dalam laporan yang koheren.

3.1. Perencanaan Tinjauan

Dalam penelitian ini, kami akan mengikuti langkah-langkah metodologi yang telah diuraikan sebelumnya untuk melakukan evaluasi terhadap beberapa alat bantu manajemen proyek perangkat lunak. Langkah pertama dalam proses ini adalah merancang strategi pencarian yang tepat untuk mengidentifikasi literatur yang relevan dan terkait dengan subjek penelitian[4].

Strategi pencarian akan mencakup pemilihan kata kunci yang sesuai dengan topik penelitian, seperti "manajemen proyek perangkat lunak", "alat bantu manajemen proyek", dan nama-nama alat bantu tertentu yang akan dievaluasi. Alat bantu yang akan kami evaluasi dalam penelitian ini adalah Asana, Jira, Microsoft Project, Docker, dan Trello. Pemilihan alat bantu ini didasarkan pada penilaian awal kami bahwa mereka memenuhi kriteria-kriteria yang relevan untuk digunakan dalam manajemen proyek perangkat lunak[11].

Setelah literatur yang relevan telah diidentifikasi, langkah selanjutnya adalah melakukan evaluasi kualitas terhadap artikel-artikel tersebut. Kami akan menilai kualitas metodologi penelitian yang digunakan dalam artikel, serta relevansinya dengan topik penelitian kami. Artikel-artikel yang memenuhi kriteria inklusi akan dipertimbangkan untuk analisis lebih lanjut.

Kami akan melibatkan perancangan skema karakterisasi yang akan digunakan untuk mengumpulkan dan menganalisis data dari artikel-artikel yang telah dipilih. Skema karakterisasi ini akan mencakup variabel-variabel yang relevan untuk evaluasi alat bantu manajemen proyek perangkat lunak, seperti fitur-fitur yang disediakan, fleksibilitas, skalabilitas, biaya, dan ketersediaan dukungan teknis[12].

Analisis data akan dilakukan dengan cermat untuk mengidentifikasi kekuatan dan kelemahan dari masing-masing alat bantu yang dievaluasi. Kami akan memperhatikan sejauh mana setiap alat bantu memenuhi kriteria penilaian yang telah ditetapkan, serta kecocokannya dengan kebutuhan manajemen proyek perangkat lunak. Hasil analisis ini akan menjadi dasar untuk menyusun laporan penelitian kami.

Dengan demikian, kami akan mengikuti langkah-langkah metodologi ini dengan cermat untuk memastikan evaluasi yang komprehensif dan akurat terhadap alat bantu manajemen proyek perangkat lunak yang telah dipilih.

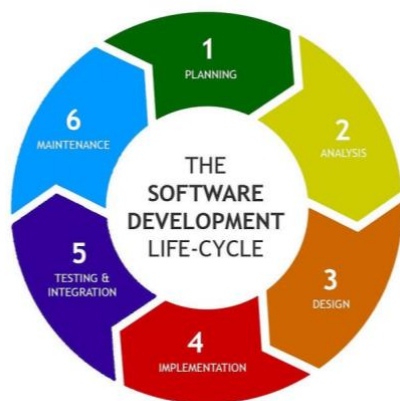
3.2. Model kriteria berdasarkan skema karakterisasi

Untuk menentukan model kriteria untuk mengevaluasi alat bantu manajemen proyek maka dibutuhkan lifecycle yang cocok untuk skema karakterisasi yang akan digunakan sebagai kriteria penilaian untuk alat bantu manajemen proyek, kita perlu memilih sebuah model lifecycle yang mencerminkan tahapan-tahapan umum dalam pengembangan perangkat lunak dan manajemen proyek[13]. Dengan demikian, sebuah model lifecycle yang umum digunakan dalam industri perangkat lunak adalah Model Pengembangan Perangkat Lunak atau Software Development Lifecycle (SDLC)[14].

SDLC menggambarkan serangkaian tahapan yang ditempuh dalam pengembangan perangkat lunak, dari perencanaan hingga pemeliharaan. Tahapan-tahapan umum dalam SDLC meliputi:

1. Perencanaan (Planning): Tahap ini melibatkan penentuan tujuan, ruang lingkup, dan sumber daya yang diperlukan untuk proyek.
2. Analisis (Analysis): Pada tahap ini, kebutuhan pengguna diidentifikasi dan dianalisis untuk membentuk dasar desain sistem.
3. Desain (Design): Desain sistem dan arsitektur perangkat lunak dikembangkan berdasarkan hasil analisis.
4. Implementasi (Implementation): Tahap ini melibatkan pembangunan perangkat lunak sesuai dengan spesifikasi yang telah dirancang sebelumnya.
5. Pengujian (Testing): Perangkat lunak diuji untuk memastikan bahwa berfungsi sesuai dengan yang diharapkan dan memenuhi kebutuhan pengguna.
6. Penerapan (Deployment): Perangkat lunak diperkenalkan ke lingkungan produksi dan siap digunakan oleh pengguna akhir.
7. Pemeliharaan (Maintenance): Perangkat lunak dipelihara untuk memperbaiki bug, meningkatkan kinerja, dan mengakomodasi perubahan kebutuhan pengguna.

Model SDLC memberikan kerangka kerja yang jelas untuk mengelompokkan kriteria penilaian dalam pengembangan perangkat lunak[14]. Dengan mengacu pada tahapan-tahapan ini, kita dapat mengidentifikasi karakteristik yang relevan dan penting dalam mengevaluasi alat bantu manajemen proyek. Misalnya, fitur perencanaan dan analisis alat bantu manajemen proyek dapat dinilai berdasarkan kemampuannya untuk mendukung tahapan perencanaan dan analisis dalam SDLC, sementara fitur pengujian dan pemeliharaan dapat dinilai berdasarkan kemampuannya untuk mendukung tahapan pengujian dan pemeliharaan.



Gambar 2. *Software Development Life Cycle*

3.3. Skema Karakterisasi

Dalam proses evaluasi alat bantu manajemen proyek, kami telah mengembangkan skema karakterisasi yang berdasarkan tahapan dalam *Software Development Lifecycle* (SDLC). Skema ini bertujuan untuk memudahkan penilaian terhadap berbagai aspek yang relevan dalam setiap tahapan pengembangan perangkat lunak. Mari kita bahas lebih lanjut tentang setiap bagian dari skema karakterisasi ini:

1. Perencanaan (*Planning*):

- Kemudahan Pengaturan Proyek: Kami menilai seberapa mudah alat ini memungkinkan pengguna untuk membuat dan mengatur proyek baru. Ini mencakup fitur-fitur seperti pembuatan proyek baru, menentukan ruang lingkup proyek, dan alokasi sumber daya.
- Manajemen Anggaran: Kami mengevaluasi kemampuan alat dalam membantu pengguna mengelola anggaran proyek dan memantau pengeluaran.
- Perencanaan Jadwal: Kami menilai fitur yang memungkinkan pengguna membuat dan mengelola jadwal proyek dengan mudah, termasuk penjadwalan tugas dan menetapkan batas waktu.

2. Analisis (*Analysis*):

- Manajemen Persyaratan: Kami mengevaluasi kemampuan alat dalam menangkap, menyusun, dan melacak persyaratan proyek. Ini mencakup fitur-fitur untuk mendokumentasikan persyaratan, menentukan prioritas, dan menanggapi perubahan.
- Kolaborasi Tim: Kami menilai fitur yang memfasilitasi kolaborasi antara anggota tim dalam menganalisis persyaratan dan merancang solusi. Ini bisa meliputi kemampuan untuk berbagi dokumen, memberikan umpan balik, dan melacak revisi.

3. Desain (*Design*):

- Desain Arsitektur: Kami mengevaluasi kemampuan alat dalam membantu pengguna merancang arsitektur perangkat lunak. Ini mencakup fitur untuk membuat diagram arsitektur, memodelkan struktur sistem, dan menentukan integrasi komponen.
- Dokumentasi Desain: Kami menilai fitur yang memungkinkan pengguna membuat dan mengelola dokumentasi desain proyek, termasuk spesifikasi teknis, diagram desain, dan panduan pengembangan.

4. Implementasi (*Implementation*):

- Manajemen Kode Sumber: Evaluasi dilakukan terhadap kemampuan alat dalam mengelola kode sumber proyek dan melakukan kontrol versi. Ini termasuk fitur seperti pembuatan cabang (*branching*), penggabungan (*merging*), dan manajemen konflik.
- Integrasi dengan Platform Pengembangan: Kami menilai ketersediaan integrasi alat dengan platform pengembangan yang umum digunakan seperti GitHub, GitLab, atau Bitbucket. Kemampuan untuk berinteraksi dengan repositori kode sumber eksternal menjadi pertimbangan penting.

5. Pengujian (*Testing*):

- Pembuatan dan Manajemen Uji Coba: Evaluasi dilakukan terhadap fitur yang memungkinkan pembuatan dan manajemen uji coba proyek dengan mudah. Ini mencakup pembuatan skenario uji, eksekusi otomatis, dan pelaporan hasil.
- Pelaporan Hasil Uji Coba: Kami mengevaluasi kemampuan alat untuk memberikan laporan hasil uji coba secara rinci dan terstruktur. Informasi yang diberikan harus memungkinkan tim untuk mengidentifikasi masalah dengan cepat dan meresponsnya.

6. Penerapan (*Deployment*):

- Manajemen Penerapan: Kami menilai fitur yang memfasilitasi proses penerapan perangkat lunak ke lingkungan produksi. Ini mencakup otomatisasi penerapan, manajemen konfigurasi, dan pemantauan proses.
- Monitoring Kinerja: Evaluasi dilakukan terhadap kemampuan alat untuk memantau kinerja aplikasi setelah penerapan. Ini mencakup pemantauan beban kerja, waktu tanggap, dan penggunaan sumber daya.

7. Pemeliharaan (*Maintenance*):

- Manajemen *Bug*: Kami mengevaluasi kemampuan alat untuk melacak, mengelola, dan memperbaiki *bug*. Ini termasuk fitur pelaporan *bug*, penugasan prioritas, dan jejak riwayat perbaikan.

- Manajemen Perubahan: Evaluasi dilakukan terhadap fitur yang membantu dalam mengelola perubahan ke perangkat lunak setelah penerapan. Ini mencakup manajemen versi, alur kerja perubahan, dan pemeriksaan kode.

Dengan skema karakterisasi ini, kami dapat melakukan evaluasi yang komprehensif terhadap berbagai alat bantu manajemen proyek. Evaluasi ini memungkinkan pemilihan alat yang paling sesuai dengan kebutuhan dan tujuan proyek, serta memastikan efisiensi dan efektivitas dalam pengembangan perangkat lunak.

4. METODE PENILAIAN

Setelah kami mengirimkan setiap fitur model kualitas, kami menetapkan metode pemeringkatan untuk mengevaluasi setiap alat bantu manajemen proyek secara konsisten. Kami merancang metode evaluasi untuk mendapatkan evaluasi kuantitatif terhadap setiap kriteria, kelompok kriteria, dan tools secara keseluruhan[15]. Metode ini memungkinkan perbandingan yang mudah dan homogen antara tools di berbagai tingkatan:

- **Tingkat Kriteria:** Poin diberikan pada setiap kriteria berdasarkan skala dari nol hingga empat [0–4], dengan interpretasi sebagai berikut:
 - 4 poin menunjukkan tools memberikan dukungan penuh terhadap kriteria yang dievaluasi.
 - 3 poin menunjukkan dukungan sebagian terhadap kriteria yang dievaluasi oleh tools.
 - 2 poin menunjukkan tools menyediakan antarmuka pemrograman yang mendukung pengembangan kriteria yang dievaluasi.
 - 1 poin menunjukkan diperlukan komponen pihak ketiga untuk mendukung kriteria.
 - 0 poin menunjukkan tools tidak mendukung kriteria tersebut.
- **Skor Parsial (PS):** Diperoleh per kelompok kriteria dan dihitung dengan menjumlahkan skor resultan dari semua subkriteria kelompok, kemudian dibagi dengan skor maksimum kelompok kriteria, dan hasilnya dikalikan dengan 10. Persamaan untuk perhitungan ini adalah:

$$PS = \frac{\sum_{i=1}^n S_i}{max} \times 10$$

Persamaan 1. Skor Parsial

Di mana n mewakili jumlah sub-kriteria dalam kelompok, S mewakili skor sub-kriteria, dan max mewakili skor maksimum kelompok.

- **Skor Akhir (FS):** Merupakan skor akhir tools dan diperoleh dengan menjumlahkan sebagian

skor dari seluruh kelompok kriteria. Persamaan untuk perhitungan ini adalah:

$$FS = \frac{\sum_n PS}{ts} \times 10$$

Persamaan 2. Skor Akhir

Di mana n mewakili jumlah kelompok dan ts mewakili total skor maksimum semua kelompok kriteria.

5. EVALUASI

5.1. Peninjauan Setiap Alat

Bagian ini membahas evaluasi skema karakterisasi kami untuk setiap alat bantu manajemen proyek yang dipilih dalam bagian sebelumnya. Proses evaluasi melibatkan pengujian setiap alat dan tinjauan sistematis terhadap semua dokumentasi resmi, komunitas resmi, dan forum yang disediakan oleh alat bantu manajemen proyek tersebut guna mengevaluasi kriteria yang didukung. Bagian-bagian berikut secara singkat menjelaskan setiap alat bantu manajemen proyek yang sedang diteliti dan merangkum hasil evaluasi.

5.1.1. Asana

Aplikasi berikut merupakan alat bantu manajemen proyek dan kolaborasi yang memungkinkan tim untuk mengatur, melacak, dan menyelesaikan tugas-tugas. Ini membantu tim untuk berkoordinasi lebih baik dengan menyediakan alat untuk mengatur proyek dalam bentuk daftar, tugas, jadwal, dan lainnya. Asana menawarkan antarmuka yang intuitif dan mudah digunakan, memungkinkan pengguna untuk dengan cepat memulai dan mengelola proyek mereka tanpa memerlukan pelatihan khusus. Pengguna dapat membuat, mengorganisir, dan melacak tugas-tugas proyek dengan mudah, menetapkan tenggat waktu, tanggapan, lampiran, dan komentar untuk setiap tugas. Selain itu, Asana menyediakan fitur penjadwalan yang memungkinkan pengguna untuk membuat jadwal proyek, menetapkan tanggal mulai dan selesai, serta melacak kemajuan proyek dengan menggunakan kalender dan diagram Gantt. Kolaborasi tim yang efisien juga dimungkinkan dengan fitur-fitur seperti berbagi tugas, diskusi, pembaruan status, dan notifikasi secara real-time. Terakhir, Asana dapat diintegrasikan dengan berbagai alat dan layanan lainnya seperti Gmail, Google Drive, Slack, dan banyak lagi, untuk meningkatkan produktivitas dan keterhubungan dalam lingkungan kerja.

Berikut aspek peninjauan evaluasi yang kami sorot :

- **Pemodelan:** Asana memungkinkan pengguna untuk membuat struktur proyek dengan

membuat proyek baru dan menetapkan tugas-tugas di dalamnya. Pengguna dapat mengorganisir tugas-tugas ke dalam bagian-bagian yang berbeda, seperti bagian, sub-bagian, dan sub-sub-bagian.

- **Desain:** Fitur-fitur seperti pemberian label, penugasan tugas, dan penjadwalan memungkinkan pengguna untuk merancang strategi pengembangan proyek dengan detail. Pengguna dapat menetapkan tenggat waktu untuk tugas, memprioritaskan tugas, dan mengatur tugas berdasarkan kategori tertentu.
- **Penyebaran:** Kolaborasi tim terjadi melalui komentar, pembaruan status, dan notifikasi. Setiap anggota tim dapat melihat pembaruan tugas secara real-time dan berpartisipasi dalam diskusi yang relevan.
- **Eksekusi dan Operasi:** Asana memungkinkan pengguna untuk melaksanakan tugas dengan menandai mereka sebagai selesai, memperbarui status proyek, dan mengubah rencana jika diperlukan. Pengguna juga dapat mengelola sumber daya proyek, seperti anggaran dan tenaga kerja, melalui fitur-fitur yang disediakan.
- **Pemantauan dan Pengendalian:** Alat pemantauan proyek Asana memungkinkan pengguna untuk melacak kemajuan proyek, mengidentifikasi risiko, dan mengelola perubahan yang terjadi selama proses pengembangan.
- **Analisis:** Asana menyediakan analisis kinerja proyek yang meliputi grafik Gantt, diagram burndown, dan laporan proyek. Analisis ini membantu pengguna dalam mengevaluasi kinerja proyek dan membuat perubahan jika diperlukan.

5.1.2. Jira

Jira adalah platform manajemen proyek yang sering digunakan oleh tim pengembangan perangkat lunak. Ini memungkinkan tim untuk melacak pekerjaan, mengelola backlog, melaporkan bug, dan berkolaborasi dalam siklus pengembangan perangkat lunak dengan fitur-fitur seperti sprint planning, scrum boards, dan lainnya.

Platform ini menyediakan manajemen tugas yang luas, termasuk pembuatan, pelacakan, dan penugasan tugas kepada anggota tim. Selain itu, Jira memungkinkan pengguna untuk membuat dan mengelola jadwal proyek, mengatur tugas berdasarkan prioritas, dan melacak kemajuan proyek dengan diagram burndown dan sprint. Fitur kolaborasi yang kuat, termasuk diskusi, komentar, dan pembaruan status tugas secara real-time, memungkinkan kolaborasi tim yang efektif. Jira juga dapat diintegrasikan dengan berbagai alat pengembangan seperti Bitbucket, GitHub, dan Jenkins, serta alat-alat manajemen proyek dan komunikasi lainnya, untuk meningkatkan integrasi

dan keterhubungan dalam lingkungan pengembangan perangkat lunak.

Berikut aspek peninjauan evaluasi yang kami sorot :

- **Pemodelan:** Jira memungkinkan pengguna untuk membuat struktur proyek dengan membuat proyek baru, menetapkan sprint, dan membuat daftar tugas. Pengguna dapat menyesuaikan papan kerja mereka dengan menambahkan kolom-kolom baru dan menyesuaikan aliran kerja.
- **Desain:** Fitur-fitur seperti manajemen persyaratan, penjadwalan sprint, dan diagram burndown membantu pengguna merancang strategi pengembangan proyek dengan detail. Pengguna dapat mengelola backlog produk, mengatur sprint, dan menetapkan persyaratan untuk setiap tugas.
- **Penyebaran:** Kolaborasi tim terjadi melalui komentar, pembaruan status, dan pemantauan aktivitas tim. Setiap anggota tim dapat melihat perubahan yang dilakukan oleh anggota tim lainnya dan berpartisipasi dalam diskusi yang relevan.
- **Eksekusi dan Operasi:** Jira memungkinkan pengguna untuk melaksanakan tugas dengan menyeret dan menjatuhkan tugas di papan kerja, mengubah status tugas, dan melacak waktu yang dihabiskan untuk setiap tugas. Pengguna juga dapat melihat riwayat perubahan untuk setiap tugas.
- **Pemantauan dan Pengendalian:** Alat pemantauan proyek Jira memungkinkan pengguna untuk melacak kemajuan proyek, mengidentifikasi hambatan, dan mengelola risiko. Pengguna juga dapat mengatur notifikasi untuk mendapatkan pemberitahuan tentang perubahan yang terjadi dalam proyek.
- **Analisis:** Jira menyediakan analisis kinerja proyek yang meliputi grafik burndown, diagram burnup, dan laporan proyek. Analisis ini membantu pengguna dalam mengevaluasi kinerja proyek dan membuat perubahan jika diperlukan.

5.1.3. Microsoft Project

Microsoft Project menyediakan fitur perencanaan proyek yang lengkap, pembuatan jadwal, penugasan sumber daya, dan manajemen anggaran. Selain itu, Microsoft Project memungkinkan pengguna untuk membuat dan mengelola jadwal proyek dengan menggunakan berbagai alat seperti diagram Gantt dan jadwal tugas. Pengguna juga dapat mengelola sumber daya proyek seperti tenaga kerja, peralatan, dan bahan, serta mengoptimalkan penggunaan sumber daya untuk meningkatkan efisiensi proyek. Fitur pelaporan yang komprehensif, termasuk laporan status proyek, analisis biaya, dan grafik kinerja, memungkinkan

pengguna untuk melacak kemajuan proyek dan mengambil tindakan yang sesuai. Integrasi dengan aplikasi Microsoft Office lainnya dan layanan cloud seperti Microsoft 365, memperkuat keterhubungan dan kolaborasi dalam lingkungan kerja.

Berikut aspek peninjauan evaluasi yang kami sorot :

- **Pemodelan:** Microsoft Project memungkinkan pengguna untuk membuat struktur proyek dengan menetapkan tugas, sumber daya, dan tenggat waktu. Pengguna dapat membuat jadwal proyek dan mengatur tugas ke dalam folder dan sub-folder.
- **Desain:** Fitur-fitur seperti manajemen sumber daya, penjadwalan, dan pelaporan membantu pengguna merancang strategi pengembangan proyek dengan detail. Pengguna dapat mengelola sumber daya proyek, menetapkan tenggat waktu untuk tugas, dan mengatur prioritas tugas.
- **Penyebaran:** Kolaborasi tim terjadi melalui berbagi jadwal, pembaruan status, dan diskusi tentang tugas dan proyek. Setiap anggota tim dapat melihat jadwal proyek, status tugas, dan komentar yang relevan.
- **Eksekusi dan Operasi:** Microsoft Project memungkinkan pengguna untuk melaksanakan tugas dengan memperbarui status tugas, menetapkan kemajuan, dan mengelola perubahan yang terjadi selama proses pengembangan.
- **Pemantauan dan Pengendalian:** Alat pemantauan proyek Microsoft Project memungkinkan pengguna untuk melacak kemajuan proyek, mengidentifikasi hambatan, dan mengelola risiko. Pengguna juga dapat mengatur peringatan untuk mendapatkan pemberitahuan tentang batas waktu dan perubahan yang terjadi dalam proyek.
- **Analisis:** Microsoft Project menyediakan analisis kinerja proyek yang meliputi laporan proyek, diagram Gantt, dan grafik sumber daya. Analisis ini membantu pengguna dalam mengevaluasi kinerja proyek dan membuat perubahan jika diperlukan.

5.1.4. Docker

Docker memungkinkan pengguna untuk merancang arsitektur perangkat lunak dengan menggunakan kontainer, menyediakan lingkungan terisolasi yang dapat digunakan untuk menjalankan aplikasi dan layanan tanpa mengganggu lingkungan host. Fitur lengkap untuk manajemen kontainer, pembuatan, penghapusan, dan penjadwalan kontainer, memudahkan pengguna mengelola dan mengontrol kontainer Docker. Docker juga memungkinkan penerapan aplikasi dengan cepat dan mudah menggunakan kontainer, yang menyediakan cara konsisten untuk menjalankan aplikasi di

berbagai lingkungan. Fleksibilitas lingkungan pengembangan yang diberikan oleh Docker memungkinkan pengguna membuat, mengubah, dan menghapus kontainer Docker untuk menguji aplikasi dalam berbagai konfigurasi dan skenario. Integrasi dengan berbagai alat pengembangan dan infrastruktur seperti Jenkins, Kubernetes, dan Amazon Web Services (AWS), meningkatkan alur kerja pengembangan dan implementasi aplikasi.

Berikut aspek peninjauan evaluasi yang kami sorot :

- **Pemodelan:** Docker memungkinkan pengguna untuk membuat struktur proyek dengan menentukan layanan, kontainer, dan jaringan yang diperlukan. Pengguna dapat membuat gambar Docker yang berisi semua dependensi dan konfigurasi yang diperlukan untuk aplikasi.
- **Desain:** Fitur-fitur seperti manajemen gambar, pengaturan jaringan, dan pengelolaan kontainer membantu pengguna merancang strategi pengembangan proyek dengan detail. Pengguna dapat mengelola gambar Docker, membuat jaringan khusus untuk aplikasi, dan menjalankan kontainer dalam mode yang diisolasi.
- **Penyebaran:** Kolaborasi tim terjadi melalui berbagi gambar Docker, pembaruan konfigurasi, dan diskusi tentang kebutuhan aplikasi. Setiap anggota tim dapat melihat gambar Docker, konfigurasi jaringan, dan diskusi yang terkait dengan proyek.
- **Eksekusi dan Operasi:** Docker memungkinkan pengguna untuk melaksanakan aplikasi dengan menjalankan kontainer Docker, memperbarui konfigurasi, dan mengelola sumber daya yang digunakan oleh aplikasi.
- **Pemantauan dan Pengendalian:** Alat pemantauan Docker memungkinkan pengguna untuk melacak kinerja aplikasi, memonitor penggunaan sumber daya, dan mengelola perubahan yang terjadi dalam konfigurasi aplikasi.
- **Analisis:** Docker menyediakan analisis kinerja aplikasi dan penggunaan sumber daya yang membantu pengguna dalam memperbaiki dan meningkatkan performa aplikasi.

5.1.5. Trello

Trello adalah alat manajemen proyek berbasis papan yang menggunakan konsep kartu untuk mengorganisir tugas dan proyek. Dengan Trello, pengguna dapat membuat daftar tugas, menetapkan tugas kepada anggota tim, menambahkan *deadline*, lampiran, dan lainnya. Ini sangat cocok untuk proyek-proyek yang membutuhkan visualisasi yang jelas dan fleksibilitas dalam mengelola tugas.

Trello menyediakan papan kerja kolaboratif yang dapat digunakan untuk mengorganisir tugas dan proyek secara visual. Setiap tugas atau pekerjaan direpresentasikan sebagai kartu di dalam papan kerja

Trello, yang dapat berisi informasi detail tentang tugas, termasuk deskripsi, tenggat waktu, daftar tugas, lampiran, dan komentar. Pengguna dapat menugaskan tugas kepada anggota tim dan berkolaborasi dalam menyelesaikan pekerjaan. Organisasi dan label memungkinkan pengguna untuk mengorganisir kartu dan papan kerja dengan lebih baik. Trello juga dapat diintegrasikan dengan berbagai alat dan layanan lainnya seperti Google Drive, Slack, GitHub, dan banyak lagi, meningkatkan produktivitas kolaborasi.

Berikut aspek peninjauan evaluasi yang kami sorot :

- **Pemodelan:** Trello memungkinkan pengguna untuk membuat struktur proyek dengan membuat papan kerja, daftar tugas, dan kartu tugas. Pengguna dapat mengatur tugas-tugas ke dalam daftar yang berbeda dan menyesuaikan kategorinya.
- **Desain:** Fitur-fitur seperti label, penugasan tugas, dan penjadwalan membantu pengguna merancang strategi pengembangan proyek dengan detail. Pengguna dapat menetapkan tenggat waktu untuk tugas, menandai tugas yang penting, dan mengatur tugas berdasarkan prioritas.
- **Penyebaran:** Kolaborasi tim terjadi melalui komentar, pembaruan status, dan pemantauan aktivitas papan kerja. Setiap anggota tim dapat melihat perubahan yang dilakukan oleh anggota tim lainnya dan berpartisipasi dalam diskusi yang relevan.
- **Eksekusi dan Operasi:** Trello memungkinkan pengguna untuk melaksanakan tugas dengan menyeret dan menjatuhkan kartu tugas, mengubah status tugas, dan mengatur waktu untuk tugas-tugas tersebut.
- **Pemantauan dan Pengendalian:** Alat pemantauan proyek Trello memungkinkan pengguna untuk melacak kemajuan proyek, mengidentifikasi hambatan, dan mengelola risiko yang terkait dengan proyek.
- **Analisis:** Trello menyediakan analisis sederhana tentang kinerja proyek dan penggunaan papan kerja yang membantu dalam evaluasi proyek dan perbaikan berkelanjutan.

5.2. Hasil dan Analisis

Setelah melakukan penilaian dan peninjauan berbagai fitur dari alat bantu manajemen proyek yang telah disebutkan sebelumnya, kami telah menjalani proses yang cermat untuk menilai skornya. Dari evaluasi yang telah dilakukan, kami telah menganalisis setiap aspek dari setiap alat berdasarkan skema penilaian yang telah ditentukan. Hasil dari penilaian ini kemudian dipersempit menjadi skor yang merefleksikan kinerja dan kemampuan relatif dari masing-masing alat.

	Asana	Jira	Microsoft Project	Docker	Trello
1. Perencanaan (Planning):	9,17	9,17	10,00	5,83	7,50
1.1. Kemudahan Pengaturan Proyek	4	4	4	3	4
1.2. Manajemen Anggaran	3	3	4	2	2
1.3. Perencanaan Jadwal	4	4	4	2	3
2. Analisis (Analysis):	8,75	10,00	7,50	6,25	7,50
2.1. Manajemen Persyaratan	3	4	3	2	2
2.2. Kolaborasi Tim	4	4	3	3	4
3. Desain (Design):	6,25	6,25	6,25	8,75	5,00
3.1. Desain Arsitektur	2	2	2	4	2
3.2. Dokumentasi Desain	3	3	3	3	2
4. Implementasi (Implementation):	5,00	10,00	3,75	7,50	5,00
4.1. Manajemen Kode Sumber	1	4	1	3	1
4.2. Integrasi dengan Platform Pengembangan	3	4	2	3	3
5. Pengujian (Testing):	6,25	10,00	3,75	6,25	3,75
5.1. Pembuatan dan Manajemen Uji Coba	3	4	2	3	2
5.2. Pelaporan Hasil Uji Coba	2	4	1	2	1
6. Penerapan (Deployment):	3,75	7,50	2,50	8,75	2,50
6.1. Manajemen Penerapan	2	3	1	4	1
6.2. Monitoring Kinerja	1	3	1	3	1
7. Pemeliharaan (Maintenance):	6,25	10,00	6,25	7,50	5,00
7.1. Manajemen Bug	2	4	2	3	2
7.2. Manajemen Perubahan	3	4	3	3	2
Skor Akhir :	6,49	8,99	5,71	7,26	5,18

Tabel 1. Ringkasan evaluasi seluruh alat bantu

Dari lima alat manajemen proyek yang dievaluasi—Asana, Jira, Microsoft Project, Docker, dan Trello—terlihat bahwa **Jira** memiliki skor tertinggi secara keseluruhan dengan **8,99**. Ini menunjukkan bahwa Jira mungkin merupakan pilihan terbaik untuk manajemen proyek berdasarkan kriteria yang dinilai, seperti Perencanaan, Analisis, Desain, Implementasi, Pengujian, Pengerahan, dan Pemeliharaan.

Asana dan **Microsoft Project** juga menunjukkan performa yang baik dalam beberapa kategori, namun tidak secara konsisten seperti **Jira**. **Docker** dan **Trello**, sementara itu, memiliki skor yang lebih rendah dibandingkan dengan alat lainnya, yang mungkin menunjukkan bahwa mereka kurang cocok untuk kebutuhan manajemen proyek

yang kompleks atau tidak memenuhi semua kriteria yang dinilai dengan baik.

Secara keseluruhan, penilaian ini menunjukkan bahwa pemilihan alat manajemen proyek harus didasarkan pada kebutuhan spesifik proyek dan kriteria yang paling penting bagi tim yang terlibat. Jira tampaknya menjadi pilihan yang kuat untuk proyek yang memerlukan perencanaan yang rinci dan analisis yang mendalam, sementara alat lain mungkin lebih cocok untuk proyek dengan kebutuhan yang berbeda.

6. KESIMPULAN

Manajemen proyek perangkat lunak memegang peran krusial dalam memastikan pengiriman produk perangkat lunak yang berkualitas, tepat waktu, dan dalam batas biaya yang ditetapkan. Dalam konteks ini, penggunaan alat bantu menjadi semakin penting untuk meningkatkan efisiensi dan efektivitas proyek. Penelitian ini bertujuan untuk mengevaluasi berbagai alat manajemen proyek dengan pendekatan yang terstruktur dan komprehensif. Berdasarkan hasil evaluasi, Jira telah terbukti menjadi pilihan terbaik dengan skor tertinggi secara keseluruhan. Namun, Asana dan Microsoft Project juga menunjukkan performa yang baik dalam beberapa kategori, sementara Docker dan Trello cenderung memiliki skor lebih rendah.

Pemilihan alat manajemen proyek harus didasarkan pada analisis yang cermat terhadap kebutuhan spesifik proyek dan kriteria yang paling penting bagi tim yang terlibat. Oleh karena itu, keberhasilan implementasi alat bantu bergantung pada pemahaman yang mendalam tentang proses pengembangan perangkat lunak, dinamika tim, dan keterampilan manajemen yang kuat. Dengan demikian, evaluasi berkala dan peningkatan kolaborasi antara tim pengembangan dan tim manajemen proyek menjadi kunci dalam memastikan penggunaan alat bantu yang efektif.

Untuk masa depan, langkah-langkah penting yang dapat diambil untuk meningkatkan manajemen proyek perangkat lunak adalah sebagai berikut. Pertama, perlu dilakukan implementasi alat manajemen proyek yang dipilih, seperti Jira, dalam lingkungan proyek yang sesuai. Ini memerlukan pelatihan tim untuk menggunakan alat tersebut dan integrasi yang baik dengan praktik manajemen proyek yang telah ada.

Selanjutnya, evaluasi berkala terhadap penggunaan alat manajemen proyek sangat penting. Hal ini dilakukan untuk memastikan bahwa alat tersebut masih relevan dan efektif sesuai dengan perkembangan kebutuhan proyek dan teknologi yang terus berubah.

Peningkatan kolaborasi antara tim pengembangan dan tim manajemen proyek juga menjadi fokus utama. Dengan meningkatkan kolaborasi, diharapkan pemahaman yang lebih baik tentang kebutuhan dan tujuan proyek dapat tercapai.

Langkah-langkah ini dapat dilakukan melalui rapat proyek rutin, komunikasi yang terbuka, dan penggunaan alat kolaborasi yang efektif.

Terakhir, penting untuk terus melakukan penelitian dan inovasi dalam bidang manajemen proyek perangkat lunak. Dengan memahami tren baru dan teknologi yang muncul, diharapkan efisiensi dan efektivitas proyek dapat ditingkatkan di masa depan.

Dengan mengambil langkah-langkah ini, diharapkan manajemen proyek perangkat lunak dapat terus berkembang dan memberikan kontribusi yang signifikan terhadap kesuksesan proyek-proyek yang kompleks.

7. REFERENSI

- [1] A. Nagari and U. Djuanda, "MANAJEMEN PROYEK Hilda Yuliasuti Awa Awa," 2023. [Online]. Available: <https://www.researchgate.net/publication/375792159>
- [2] A. Rachman, E. Yochnan, A. I. Samanlagi, and H. Purnomo, *METODE PENELITIAN KUANTITATIF KUALITATIF DAN R&D*. 2024.
- [3] R. S. Pressman and B. R. Maxim, "Software Engineering EIGHTH EDITION."
- [4] B. Kitchenham, D. Budgen, O. P. Brereton, M. Turner, J. Bailey, and S. Linkman, "Guidelines for performing Systematic Literature Reviews in Software Engineering," 2007.
- [5] A. I. L. Paz and R. I. López, "Recommendation method for customized IT project management," *Procedia Comput Sci*, vol. 219, pp. 1938–1945, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.493.
- [6] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, Apr. 2007, doi: 10.1016/J.JSS.2006.07.009.
- [7] H. Kerzner, *Project management : case studies*. 2017.
- [8] M. Clemente and L. Domingues, "Analysis of Project Management Tools to support Knowledge Management," *Procedia Comput Sci*, vol. 219, pp. 1769–1776, Jan. 2023, doi: 10.1016/J.PROCS.2023.01.472.
- [9] M. Shahid Iqbal, M. Sajad, M. Sadiq, and K. Naveed, "Software Project Management: Tools assessment, Comparison and suggestions for future development," 2016. [Online]. Available: <https://www.researchgate.net/publication/295080231>
- [10] N. Alawi, K. A. Alwaly, and N. Abdulmajid Alawi, "Factors Affecting the Application of Project Management Knowledge Guide

- (PMBOK ® GUIDE) in Construction Projects in Yemen,” *International Journal of Construction Engineering and Management*, 2020, doi: 10.5923/j.ijcem.20200903.01.
- [11] E. Verzuh, “THE FAST FORWARD MBA IN PROJECT MANAGEMENT FIFTH EDITION,” 2015.
- [12] P. A. Laplante, *What every engineer should know about software engineering*. Taylor & Francis, 2007.
- [13] N. Takagi and J. Varajão, “Success Management and the Project Management Body of Knowledge (PMBOK): An Integrated Perspective-research-in-progress,” 2020. [Online]. Available: <https://www.researchgate.net/publication/347949303>
- [14] N. Tri and S. Saptadi, “Rekayasa Perangkat Lunak,” 2024. [Online]. Available: <https://www.researchgate.net/publication/378622156>
- [15] A. Meidan, J. A. García-García, M. J. Escalona, and I. Ramos, “A survey on business processes management suites,” *Comput Stand Interfaces*, vol. 51, pp. 71–86, Mar. 2017, doi: 10.1016/J.CSI.2016.06.003.