

## PENERAPAN METODOLOGI *EXTREME PROGRAMMING* STUDI KASUS PENGEMBANGAN MAINTENANCE DIGITAL SYSTEM (MDS) DI PT JKL

Sasmito Budi Utomo<sup>\*</sup>, Ardianto Patrick Santoso, Efata Winapul Marcelita Tarigan

Program Studi Manajemen Informatika, Jurusan Informatika, Politeknik Astra

E-mail : sasmibu@gmail.com<sup>\*</sup>

**Abstract**—The Maintenance Department at PT JKL faces challenges in manual reporting, which impacts operational efficiency and data accuracy. This study aims to develop a Maintenance Digital System (MDS) using the Extreme Programming (XP) methodology, which emphasizes iterative cycles, intensive user collaboration, and continuous testing. The system was developed through four main phases: planning, analysis, design, and implementation, utilizing PHP, MySQL, and the CodeIgniter 3 framework. The results show that MDS successfully reduced reporting time by up to 3,360 minutes per month and improved the effectiveness of monthly data recap and departmental performance transparency. This research demonstrates the effectiveness of XP in building adaptive information systems capable of responding quickly and efficiently to the needs of the manufacturing industry. These findings are expected to serve as a reference for agile-based digital system development in the industrial sector.

**Keywords:** Maintenance Digital System, Extreme Programming, Agile Methodology, Information System, Operational Efficiency

**Abstrak**—Departemen Maintenance di PT JKL menghadapi tantangan dalam pelaporan manual yang berdampak pada efisiensi operasional dan akurasi data. Penelitian ini bertujuan untuk mengembangkan Maintenance Digital System (MDS) dengan pendekatan metodologi Extreme Programming (XP), yang mengedepankan siklus iteratif, kolaborasi intensif dengan pengguna, dan pengujian berkelanjutan. Sistem dikembangkan melalui empat tahap utama: perencanaan, analisis, perancangan, dan implementasi, menggunakan teknologi PHP, MySQL, dan framework CodeIgniter 3. Hasil menunjukkan bahwa MDS berhasil menurunkan waktu pelaporan hingga 3.360 menit per bulan serta meningkatkan efektivitas rekap data bulanan dan transparansi performa departemen. Penelitian ini memperlihatkan efektivitas XP dalam membangun sistem informasi adaptif yang mampu merespons kebutuhan industri manufaktur secara cepat dan efisien. Temuan ini diharapkan menjadi referensi dalam pengembangan sistem digital berbasis agile di sektor industri.

**Kata Kunci:** Maintenance Digital System, Extreme Programming, Agile Methodology, Sistem Informasi, Efisiensi Operasional

### I. PENDAHULUAN

Di tengah tuntutan efisiensi dan kecepatan pengambilan keputusan di sektor industri manufaktur, kebutuhan akan sistem informasi yang andal semakin mendesak. PT JKL melalui Departemen Maintenance, menghadapi kompleksitas dalam pengelolaan laporan pemeliharaan yang masih bersifat manual. Hal ini menimbulkan berbagai tantangan seperti keterlambatan pelaporan, duplikasi data, dan keterbatasan visibilitas terhadap progres perbaikan.

Digitalisasi proses pelaporan dianggap sebagai solusi strategis yang dapat menjawab tantangan tersebut. Oleh karena itu, penelitian ini berfokus pada pengembangan Maintenance Digital System (MDS) dengan menggunakan metodologi Extreme Programming (XP), salah satu pendekatan agile yang memungkinkan pengembangan sistem secara fleksibel, cepat, dan terstruktur.

### II. TINJAUAN PUSTAKA

Transformasi digital di sektor manufaktur menuntut adopsi sistem informasi yang mampu menjawab dinamika operasional [1]. Paul et al. (2024) menekankan bahwa organisasi harus membangun sistem yang tidak hanya efisien tetapi juga adaptif terhadap perubahan [2]. Plekhanov et al. (2023) menyatakan bahwa perusahaan yang sukses dalam digitalisasi cenderung menggunakan platform fleksibel yang memungkinkan iterasi cepat [3].

Dalam konteks pengembangan perangkat lunak, metodologi agile seperti XP telah terbukti mendukung sistem yang responsif [4][5]. Dennis et al. (2020) menyatakan bahwa XP cocok diterapkan pada proyek yang memiliki tingkat perubahan tinggi karena nilai-nilai dasarnya: komunikasi, kesederhanaan, umpan balik, dan keberanian [5]. Dalam studi oleh Alam et al. (2023), sistem pelaporan digital berbasis web menunjukkan peningkatan signifikan dalam efisiensi

pelaporan harian pegawai kontrak [6]. Beberapa studi lainnya menunjukkan bahwa digitalisasi pelaporan dapat meningkatkan efisiensi dan akurasi laporan, serta memperbaiki proses yang sebelumnya dilakukan secara manual [7][8].

Perbandingan metodologi juga penting untuk menentukan pendekatan terbaik [9]. XP memiliki keunggulan dalam hal iterasi cepat dan partisipasi aktif pengguna dibandingkan waterfall yang kaku dan scrum yang membutuhkan peran scrum master [10][11]. Tabel 1 menunjukkan perbandingan metodologi XP dengan metodologi lain seperti scrum dan waterfall [4][12]. Metodologi scrum, yang membutuhkan peran scrum master, tidak sesuai dengan kebutuhan proyek ini, sementara waterfall membutuhkan analisis yang mendalam dan waktu yang lebih lama untuk setiap fase [13]. Oleh karena itu, XP dipilih dalam pengembangan MDS untuk menjawab kebutuhan akan sistem pelaporan yang fleksibel dan efisien [14].

Tabel 1. Perbandingan Metodologi

Kemampuan Mengembangkan Sistem	Metodologi		
	Waterfall	XP	Scrum
Dengan Pengguna yang Tidak Jelas	Buruk	Sangat Baik	Baik
Dengan Teknologi yang Tidak Dikenal	Buruk	Baik	Baik
Yang Rumit	Baik	Baik	Baik
Yang Andal	Baik	Sangat Baik	Sangat Baik
Dengan Jadwal Waktu yang Singkat	Buruk	Sangat Baik	Sangat Baik
Dengan Visibilitas Jadwal	Buruk	Sangat Baik	Sangat Baik

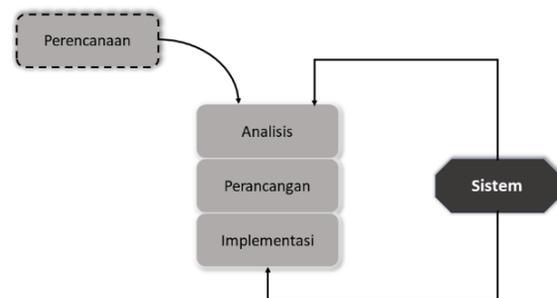
**III. METODOLOGI PENELITIAN**

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak berbasis metodologi *Extreme Programming* (XP). Metodologi XP dipilih karena kemampuannya untuk menyesuaikan dengan kebutuhan pengguna yang belum sepenuhnya jelas dan untuk mengimplementasikan perubahan bertahap selama pengembangan sistem [15][16]. Studi dilakukan dalam kurun waktu enam bulan di Departemen *Maintenance* di PT JKL. Pengembangan dilakukan dengan interaksi yang erat antara pengembang dan pengguna akhir, yang meningkatkan komunikasi dan memastikan bahwa sistem yang dibangun sesuai dengan kebutuhan pengguna [17][18]. Proses pengembangan MDS menggunakan langkah-langkah sebagai berikut:

1. Perencanaan: Identifikasi kebutuhan pengguna melalui observasi dan wawancara, menghasilkan dokumen *system request* dan *activity plan*.
2. Analisis: Pengumpulan *user stories*, pembuatan *use case diagram*, *activity diagram*, dan *class diagram* untuk mendefinisikan alur sistem.

3. Perancangan: Pembuatan antarmuka pengguna berbasis web, desain basis data (*physical data model*), dan arsitektur sistem berbasis *three-tier*.
4. Implementasi: Pengkodean menggunakan PHP dan MySQL dalam *framework* CodeIgniter 3, diikuti pengujian *black box* dan *User Acceptance Testing* (UAT) setiap akhir iterasi.

Gambar 1 menyajikan tahapan dalam XP. Validasi dilakukan dengan membandingkan efektivitas waktu dan efisiensi biaya sebelum dan sesudah implementasi, serta pengukuran kepuasan pengguna terhadap sistem baru[4][12].



Gambar 1. Metodologi *Extreme Programming* (XP)

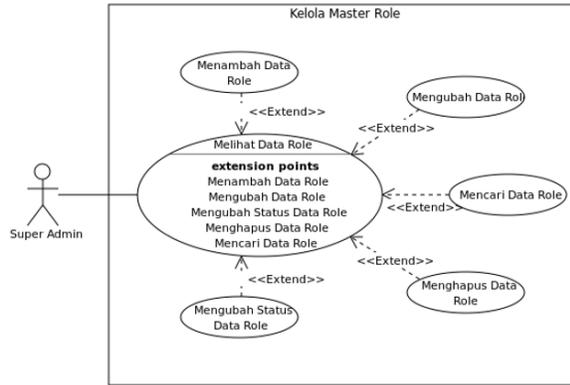
**IV. HASIL DAN PEMBAHASAN**

Pengembangan *Maintenance Digital System* (MDS) untuk Departemen *Maintenance* PT JKL telah mengadopsi metodologi *Extreme Programming* (XP) sebagai pendekatan utama untuk menggantikan sistem pelaporan manual yang ada. Pada tahap perencanaan, telah dilakukan di awal pengembangan dengan menghasilkan dua dokumen utama yaitu:

1. *System Request*: Dokumen ini mengidentifikasi kebutuhan dasar dari sistem MDS. Proses ini melibatkan wawancara mendalam dengan tim Departemen *Maintenance* untuk menangkap semua kebutuhan fungsional dan nonfungsional. Identifikasi kebutuhan yang akurat sangat penting untuk mencegah perubahan besar di kemudian hari, yang sesuai dengan prinsip XP mengenai perencanaan yang fleksibel namun terstruktur.
2. *Activity Plan*: Dokumen ini merinci langkah-langkah pengembangan yang diperlukan, termasuk jadwal dan sumber daya yang diperlukan. *Activity plan* memastikan bahwa pengembangan sistem mengikuti jalur yang terencana dan efisien, selaras dengan prinsip XP mengenai perencanaan yang dapat disesuaikan dengan kebutuhan yang berubah.

Selanjutnya, pada tahap analisis memanfaatkan model-model UML untuk mendokumentasikan dan memahami sistem secara mendalam. Pada tahap ini dilaksanakan dengan menghasilkan analisis dalam model-model sebagai berikut:

1. *Use Case Diagram*: Diagram ini, seperti yang dicontohkan pada gambar 2, menggambarkan interaksi antara pengguna misalnya *super admin* dan sistem. Diagram ini membantu dalam memahami fungsi utama sistem dari perspektif pengguna dan memastikan bahwa semua fungsionalitas yang diperlukan dapat diakomodasi.



Gambar 2. Use Case Diagram

2. *Use Case Description*: tabel 2 memberikan rincian spesifik mengenai setiap *use case*, termasuk aktivitas yang terlibat dan alur kerja detail. Dokumentasi ini memudahkan pemahaman tentang bagaimana fungsi-fungsi sistem diimplementasikan dan diintegrasikan.

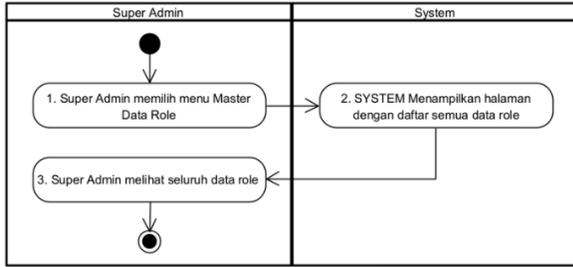
Tabel 2. Use Case Description

<i>Use case name :</i> Master Role	ID : UC001	<i>Importance level :</i> Medium
<i>Primary actor :</i> Super Admin	<i>Use case detail :</i> Detail, Real	
<i>Stakeholder and Interest :</i> Super Admin dapat mengelola data role pada aplikasi		
<i>Brief description :</i> Use case ini menjelaskan bagaimana pengguna dapat mengelola data role		
<i>Trigger :</i> Ketika pengguna ingin mengelola data role		
<i>Relationships :</i> <i>Association :</i> Super Admin <i>Include :</i> - <i>Extend :</i> Menambah Data Role, Mengubah Data Role, Mengubah Status Data Role, Menghapus Data Role, Mencari Data Role		
<i>Pre-Condition :</i> Use case dimulai ketika Super Admin ingin melihat data role		
<i>Normal Flow of Events :</i> 1. Super Admin memilih menu Master Data Role 2. SYSTEM Menampilkan halaman dengan daftar semua data role 3. Super Admin melihat seluruh data role Jika Super Admin ingin menambah data role, lihat <i>sub flows</i> S-1 Jika Super Admin ingin mengubah data role, lihat <i>sub flows</i> S-2 Jika Super Admin ingin mengubah status data role, lihat <i>sub flows</i> S-3 Jika Super Admin ingin menghapus data role, lihat <i>sub flows</i> S-4		

Jika Super Admin ingin mencari data role, lihat <i>sub flows</i> S-5
<i>Sub Flow :</i> S-1 : Menambah data role 1. Super Admin mengklik tombol "Add Data" 2. SYSTEM Menampilkan form tambah data 3. Super Admin mengisi form tambah data 4. if Apakah ada data yang belum diisi? 4.1. SYSTEM Menampilkan pesan "Please fill out this field" 5. else 5.1. SYSTEM Menyimpan data dan kembali ke halaman daftar data role 5.2. SYSTEM Menampilkan kotak pesan sukses end if S-2 : Mengubah data role 1. Super Admin memilih data yang ingin diubah 2. SYSTEM Menampilkan kotak pesan sukses 3. Super Admin mengubah data yang diperlukan dan menekan tombol "Submit" 4. if Ada data kosong? 4.1. SYSTEM Menampilkan pesan "Please fill out this field" 5. else 5.1. SYSTEM Menyimpan perubahan data dan kembali ke halaman daftar data role 5.2. SYSTEM Menampilkan kotak pesan sukses end if S-3 : Mengubah Status data role 1. Super Admin memilih data yang ingin diubah statusnya 2. Super Admin menekan tombol switch pada data yang akan diubah statusnya 3. SYSTEM Menampilkan pesan konfirmasi perubahan status 4. Super Admin menekan tombol "Yes, Update!" 5. SYSTEM Menampilkan pesan sukses 6. SYSTEM Menampilkan kembali halaman daftar data role S-4 : Menghapus data <i>role</i> 1. Super Admin memilih data employee yang ingin dihapus 2. SYSTEM Menampilkan pesan konfirmasi untuk menghapus data 3. Super Admin mengonfirmasi penghapusan dengan menekan tombol "Yes, Delete!" 4. SYSTEM Menghapus data role 5. SYSTEM Menampilkan kembali halaman daftar data role S-5 : Mencari data <i>role</i> 1. Super Admin memasukkan kata kunci pencarian 2. if Data tidak ada? 2.1. SYSTEM Menampilkan baris pesan "No matching records found" 3. else 3.1. SYSTEM Menampilkan data pada tabel data end if 4. Super Admin melihat data sesuai dengan kata kunci pencarian
<i>Alternate/Exceptional Flows :</i> -
<i>Post-Condition :</i> Use case berakhir ketika Super Admin berhasil melihat data role

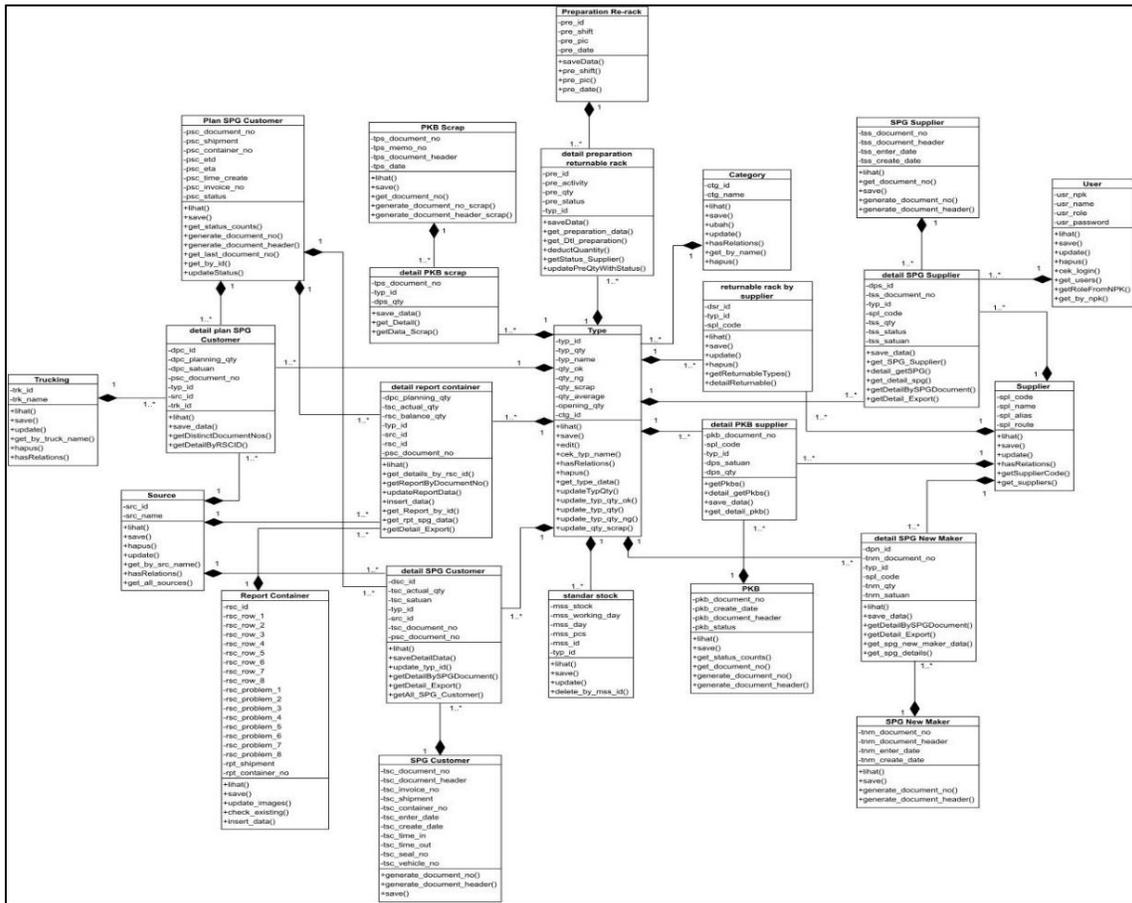
3. *Activity Diagram*: Diagram ini, seperti yang dicontohkan pada gambar 3, menggambarkan interaksi antara pengguna misalnya *super admin* dan sistem. Diagram ini membantu dalam memahami fungsi utama sistem dari perspektif

pengguna dan memastikan bahwa semua fungsionalitas yang diperlukan dapat diakomodasi.



Gambar 3. Activity Diagram

4. **CRC Cards:** Kartu CRC digunakan untuk mendokumentasikan tanggung jawab dan kolaborasi setiap kelas dalam sistem. Hal ini membantu dalam merancang struktur sistem yang efektif dan memahami interaksi antar kelas.
5. **Class Diagram:** Gambar 4 menunjukkan struktur kelas dalam sistem, termasuk hubungan antar kelas seperti agregasi dan asosiasi. Diagram ini memberikan gambaran menyeluruh tentang struktur sistem dan memfasilitasi perancangan yang kohesif.

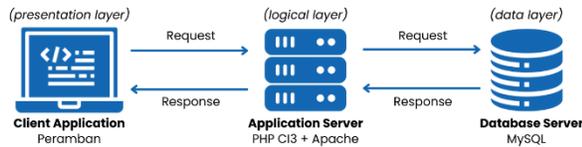


Gambar 4. Class Diagram

Tahap perancangan berfokus pada penerapan umpan balik dari analisis untuk mengembangkan desain sistem yang optimal, yaitu dengan menghasilkan luaran:

1. **Desain Antarmuka Pengguna atau User Interface (UI):** Desain UI dikembangkan pada pembangunan sistem MDS untuk memudahkan transisi dari laporan Microsoft Excel ke antarmuka berbasis web, memastikan akses yang mudah dan efisiensi pengguna.

2. **Physical Data Model (PDM):** PDM menggambarkan struktur basis data yang diperlukan, yang dirancang untuk mendukung integrasi dan efisiensi data yang lebih baik.
3. **Arsitektur Sistem:** Gambar 5 menunjukkan desain arsitektur sistem dengan model *three-tier*, yang mengikuti prinsip XP untuk memastikan iterasi yang cepat dan umpan balik yang terus menerus.



Gambar 5. Desain Arsitektur

Pada tahap implementasi, sistem MDS dikembangkan dan diuji secara iteratif. Proses pengembangan dilakukan melalui empat iterasi yang dapat dilihat pada tabel 3.

Tabel 3. Riwayat Iterasi

Iterasi	Tanggal	Kegiatan Utama
1	22 Januari 2024 – 13 Februari 2024	Pembuatan pangkalan data, template antarmuka, dan pengkodean master data.
2	1 Maret 2024 – 26 Maret 2024	Analisis dan pengkodean transaksi daily report dan follow up pending problem.
3	17 April 2024 – 9 Mei 2024	Analisis dan pengkodean transaksi problem report dan production unit.
4	13 Mei 2024 – 14 Juni 2024	Analisis dan pengkodean menu rekap data, dashboard, landing page, dan menu performance.

Setiap iterasi mencakup pengembangan fitur, pengujian, dan perbaikan berdasarkan umpan balik, sesuai dengan prinsip XP yang mendorong siklus pengembangan yang berulang dan adaptif.

Setelah implementasi, sistem MDS diuji lebih lanjut menggunakan metode *black box* dan *User Acceptance Testing* (UAT) untuk memastikan semua fitur berfungsi dengan baik. Hasil pengujian menunjukkan bahwa sistem MDS secara signifikan mengurangi waktu proses pelaporan dan rekap data.

- Penghematan Waktu: Tabel 4 menunjukkan penurunan durasi waktu proses pelaporan harian setelah implementasi, yang menghilangkan beberapa proses manual yang ada sebelumnya.

Tabel 4. Penghematan Waktu Setelah Implementasi MDS

No	Rincian Proses	Reason	Durasi Sebelum & Setelah Ada Sistem					
			Sebelum	Satuan	Sesudah	Satuan	Seluruh	Satuan
1	Rekap Data Daily Report	Data yang diperlukan setiap bulan terkait rapat bulanan	480	Menit	0	Menit	480	Menit
2	Perhitungan Performance	Data yang diperlukan setiap bulan terkait rapat bulanan	2400	Menit	0	Menit	2400	Menit
3	Rekap Data Man Hour	Data yang diperlukan setiap bulan terkait rapat bulanan	240	Menit	0	Menit	240	Menit
4	Rekap Data Consumable & Sparepart	Data yang diperlukan setiap bulan terkait rapat bulanan	240	Menit	0	Menit	240	Menit
Total			3360	Menit	0	Menit	3360	Menit

- Analisis *Net Quality Income* (NQI): Meskipun biaya pembangunan sistem sebesar Rp 38.640.000, sistem ini menghemat biaya operasional sebesar Rp 820.893.546 dan efisiensi waktu yang setara dengan Rp 573.013.659 dalam setahun.

Implementasi XP dalam pengembangan MDS menghasilkan sistem pelaporan yang efisien, adaptif, dan *user-friendly*. Analisis hasil menunjukkan:

1. Penurunan waktu pelaporan bulanan dari 3.360 menit menjadi 0 menit.
2. Efisiensi biaya operasional sebesar Rp820.893.546/tahun.
3. Efektivitas pelaporan meningkat ditunjukkan dari rekap performa yang lebih cepat dan akurat.

Lebih lanjut, hasil UAT menunjukkan tingkat kepuasan pengguna sebesar 92% dari total responden. Hal ini mencerminkan bahwa keterlibatan pengguna sejak awal pengembangan berdampak signifikan terhadap penerimaan sistem.

Jika dibandingkan dengan metode tradisional seperti waterfall, XP memungkinkan pengembangan yang lebih fleksibel dan cepat, sangat sesuai dengan kebutuhan industri yang dinamis. Keunggulan XP dalam kolaborasi pengguna terbukti mendukung adopsi teknologi digital secara optimal di lingkungan kerja manufaktur.

## V. KESIMPULAN

Penerapan metodologi *Extreme Programming* (XP) dalam pengembangan *Maintenance Digital System* (MDS) di PT JKL terbukti efektif dalam mengatasi tantangan pelaporan manual. Sistem yang dikembangkan tidak hanya menghemat waktu dan biaya tetapi juga meningkatkan transparansi dan akurasi data.

Kontribusi utama penelitian ini adalah bukti empiris bahwa XP dapat diterapkan secara efektif di lingkungan industri, khususnya dalam pengembangan sistem pelaporan digital yang dinamis. Studi ini juga memberikan model pengembangan yang dapat direplikasi di sektor industri lain yang menghadapi tantangan serupa.

Disarankan agar penelitian lanjutan dapat difokuskan pada perbandingan XP dengan metode *agile* lainnya dalam pengembangan sistem berskala besar serta eksplorasi dampak penerapan MDS terhadap produktivitas jangka panjang.

## VI. DAFTAR PUSTAKA

- [1] F. P. Appio et al., "Digital Transformation and Innovation Management: A Synthesis of Existing Research and an Agenda for Future Studies," *Journal of Product Innovation Management*, vol. 38, no. 1, pp. 4–20, 2021.
- [2] J. Paul et al., "Digital transformation: A multidisciplinary perspective and future research agenda," *International Journal of Consumer Studies*, vol. 48, no. 2, pp. 1–28, 2024.D.
- [3] Plekhanov et al., "Digital transformation: A review and research agenda," *European*

- Management Journal*, vol. 41, no. 6, pp. 821–844, 2023.
- [4] Dingsøy, 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] A. Dennis et al., *System Analysis and Design: An Object-Oriented Approach with UML*, 6th ed., Lise Johnson, 2020.
- [6] A. Alam et al., "Sistem Informasi Laporan Harian Kinerja Pegawai Kontrak berbasis Web Pada Kantor Satpol PP dan WH Aceh," *Jurnal Sistem Komputer*, vol. 3, no. 2, pp. 65–73, 2023.
- [7] J. Y. Ma et al., "The Effect of Digital Transformation on the Pharmaceutical Sustainable Supply Chain Performance," *Sustainability*, vol. 15, no. 1, pp. 1–21, 2023.
- [8] A. I. Stoumpos et al., "Digital Transformation in Healthcare: Technology Acceptance and Its Applications," *International Journal of Environmental Research and Public Health*, 2023.
- [9] Wang, X., Conboy, K., & Cawley, O. (2012). "Leagile" software development: An experience report analysis of the application of lean approaches in agile software development. *Journal of Systems and Software*, 85(6), 1287–1299.
- [10] Beck, K., & Andres, C. (2004). *Extreme Programming Explained: Embrace Change* (2nd ed.). Addison-Wesley.
- [11] Mahnic, V. (2015). *A case study on agile software development using Scrum. Elektronika ir Elektrotechnika*, 21(3), 103–108.
- [12] Boehm, B., & Turner, R. (2005). *Management challenges to implementing agile processes in traditional development organizations. IEEE Software*, 22(5), 30–39.
- [13] Kropp, M., Meier, A., & Biddle, R. (2018). *Agile practices and success in student software projects. Proceedings of the 40th International Conference on Software Engineering: Software Engineering Education and Training*, 153–162.
- [14] Basri, S., & O'Connor, R. V. (2010). *Understanding the perception of agile software development characteristics in academia: A Malaysian perspective. Proceedings of the 2010 International Symposium on Information Technology*, 1, 1–6.
- [15] R. S. Pressman, *Software Quality Engineering: A Practitioner's Approach*, 7th ed., 2010.
- [16] Misra, S. C., Kumar, V., & Kumar, U. (2009). *Identifying some important success factors in adopting agile software development practices. Journal of Systems and Software*, 82(11), 1869–1890.
- [17] Serrador, P., & Pinto, J. K. (2015). *Does Agile work?—A quantitative analysis of agile project success. International Journal of Project Management*, 33(5), 1040–1051.
- [18] Moe, N. B., Šmite, D., Ågerfalk, P. J., & Jabangwe, R. (2021). *Understanding the dynamics of distributed agile teams: A case study of team health over time. Information and Software Technology*, 136, 106583.