EFEKTIVITAS MANAJEMEN PROYEK PERANGKAT LUNAK BERBASIS CAPABILITY MATURITY MODEL INTEGRATION PADA PENERAPAN SECURE SOFTWARE DEVELOPMENT LIFE CYCLE
(Studi Pada PT Magicsoft Teknologi Indonesia)

TESIS

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Magister Manajemen

Oleh :

MUHAMMAD HANIFUDIN
NPM 22302081057

UNIVERSITAS ISLAM MALANG
PASCASARJANA
MALANG
2026

Abstrak

Penelitian ini bertujuan mengevaluasi penerapan Secure Software Development Life Cycle (S-SDLC) dan menilai efektivitas manajemen proyek perangkat lunak berbasis *Capability Maturity Model Integration* (CMMI) versi 3 pada PT Magicsoft Teknologi Indonesia. Penelitian menggunakan pendekatan kualitatif dengan teknik pengumpulan data berupa wawancara mendalam, observasi, dan studi dokumentasi terhadap proses pengembangan perangkat lunak dalam satu konteks proyek. Analisis dilakukan menggunakan triangulasi sumber serta pemetaan temuan terhadap indikator S-SDLC dan 12 *Practice Area* (PA) CMMI versi 3 yang dikelompokkan ke dalam Doing, Managing, dan Enabling. Hasil penelitian menunjukkan bahwa S-SDLC telah diterapkan secara terintegrasi sepanjang siklus pengembangan perangkat lunak. Keamanan diposisikan sebagai proses bawaan (built-in) melalui praktik perencanaan berbasis risiko, validasi kebutuhan lintas peran, code review berlapis, pengujian terstruktur, dokumentasi berkelanjutan, dan deployment terkontrol. Penerapan ini mendukung peningkatan kualitas dan stabilitas sistem, serta memperkuat tata kelola keamanan secara preventif. Temuan mengenai efektivitas manajemen proyek menunjukkan bahwa sebagian besar PA mencapai *Capability Level* 3 (Defined), sehingga proses dapat dikategorikan matang pada level proyek. Namun, efektivitas tersebut lebih dominan pada dimensi kualitas, keamanan, dan keterkendalian, sementara kinerja waktu menghadapi tekanan akibat beban kerja berlebihan dan keterbatasan kapasitas sumber daya. Temuan ini menegaskan bahwa efektivitas manajemen proyek bersifat multidimensional dan tidak hanya

dapat diukur dari pencapaian jadwal. Penelitian ini berkontribusi pada pengembangan teori dengan menunjukkan bahwa CMMI versi 3 dapat digunakan sebagai kerangka evaluatif yang kompatibel dengan S-SDLC, serta memberikan kontribusi praktis berupa kerangka evaluasi internal yang fit-for-purpose bagi organisasi pengembang perangkat lunak.

Kata kunci: CMMI v3; Secure Software Development Life Cycle; Manajemen Proyek Perangkat Lunak; Kematangan Proses

Abstract

This study evaluates the implementation of the Secure Software Development Life Cycle (S-SDLC) and examines the effectiveness of software project management based on the Capability Maturity Model Integration (CMMI) version 3 at PT Magicsoft Teknologi Indonesia. A qualitative research approach was employed using in-depth interviews, observations, and documentation analysis. Empirical findings were triangulated and mapped against S-SDLC indicators and twelve CMMI v3 Practice Areas (PAs) grouped into Doing, Managing, and Enabling. The findings indicate that S-SDLC has been integrated throughout the software development lifecycle, positioning security as a built-in process rather than a reactive add-on. Security-relevant practices collectively enhance software quality, system stability, and preventive security governance. With regard to project management effectiveness, the study shows that most PAs reached Capability Level 3 (Defined), reflecting a mature level of process standardization at the project level. However, effectiveness was more prominent in terms of quality, security, and process control, while schedule performance faced pressure due to workload imbalances and resource capacity constraints. These findings highlight that software project effectiveness is inherently multidimensional and cannot be assessed solely through schedule adherence. The study contributes to theory by demonstrating that CMMI v3 can serve as an evaluative framework compatible with S-SDLC and enriches project management discourse by positioning security as an integral dimension of process effectiveness.

Practically, the study provides a fit-for-purpose internal evaluation framework for software development organizations seeking to improve process maturity without formal appraisal.

Keywords: *CMMI v3; Secure Software Development Life Cycle; Software Project Management; Process Maturity*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perangkat lunak kini menjadi tulang punggung berbagai sektor industri, mulai dari keuangan, kesehatan, hingga pemerintahan. Keberhasilan implementasi sistem perangkat lunak menentukan efektivitas proses bisnis dan daya saing organisasi. Meski demikian, tingkat keberhasilan proyek perangkat lunak masih rendah. Standish Group (2020) melalui CHAOS *Report* mencatat hanya 31% proyek yang selesai tepat waktu, sesuai anggaran, dan memenuhi spesifikasi, sementara sisanya mengalami penundaan, pembengkakan biaya, atau kegagalan. Kondisi ini menunjukkan bahwa manajemen proyek perangkat lunak merupakan kebutuhan strategis dalam mendukung transformasi digital organisasi.

Kegagalan produk perangkat lunak merupakan fenomena yang signifikan dalam manajemen proyek teknologi informasi dan sering kali dikaitkan dengan lemahnya pengelolaan proses pengembangan serta kurangnya integrasi praktik terbaik dalam siklus hidup perangkat lunak. Studi oleh Standish Group melalui laporan CHAOS *Report* menunjukkan bahwa proporsi proyek perangkat lunak yang gagal atau mengalami tantangan serius tetap tinggi, dengan penyebab utama meliputi ketidakjelasan kebutuhan, kurangnya keterlibatan pengguna, dan manajemen proyek yang tidak efektif. Selain itu, kegagalan produk tidak hanya diukur dari tidak tercapainya spesifikasi teknis, tetapi juga dari ketidakmampuan sistem memenuhi aspek keamanan dan keandalan, yang

semakin krusial dalam konteks *secure software development* (Standish Group, 2020; McGraw, 2006).

Selain kegagalan produk, keterlambatan penyelesaian proyek (*schedule delay*) menjadi indikator utama ketidakefektifan manajemen proyek perangkat lunak. Keterlambatan sering disebabkan oleh estimasi waktu yang tidak akurat, perubahan kebutuhan (*scope creep*), serta kurangnya koordinasi antar tim pengembang. Menurut Project Management Institute (PMI), ketidaktepatan dalam perencanaan dan pengendalian waktu merupakan salah satu faktor dominan yang menyebabkan proyek tidak mencapai target yang telah ditetapkan. Boehm (1981) dalam *Software Engineering Economics* menekankan bahwa kompleksitas perangkat lunak dan ketidakpastian dalam proses pengembangan meningkatkan risiko deviasi jadwal secara signifikan. Oleh karena itu, keterlambatan proyek mencerminkan kegagalan dalam fungsi *planning* dan *controlling*, yang secara langsung berdampak pada efektivitas implementasi proyek secara keseluruhan.

Inefisiensi penggunaan sumber daya juga menjadi permasalahan krusial dalam manajemen proyek perangkat lunak, terutama terkait alokasi tenaga kerja, waktu, dan biaya. Inefisiensi ini sering muncul akibat kurangnya standar proses, rendahnya tingkat kematangan organisasi, serta tidak optimalnya mekanisme monitoring dan evaluasi. Menurut Sommerville (2016), organisasi dengan proses pengembangan yang tidak terstruktur cenderung mengalami pemborosan sumber daya karena aktivitas yang redundan, kesalahan berulang, dan kurangnya dokumentasi yang

sistematis. Hal ini diperkuat oleh Paulk et al. (1993) dalam kerangka CMM, yang menunjukkan bahwa organisasi dengan tingkat kematangan proses rendah (level awal) cenderung bersifat reaktif dan tidak efisien dalam penggunaan sumber daya. Oleh karena itu, inefisiensi tidak hanya berdampak pada peningkatan biaya proyek, tetapi juga menurunkan kualitas produk dan keberhasilan implementasi sistem secara keseluruhan.

Kompleksitas proyek perangkat lunak bersumber dari keterlibatan banyak pemangku kepentingan, dinamika kebutuhan bisnis, dan ketidakpastian teknologi. Tanpa manajemen proyek yang terstruktur, kompleksitas tersebut dapat menghasilkan konflik, mis-komunikasi, dan pemborosan sumber daya. Sommerville (2016) menegaskan bahwa kegagalan proyek perangkat lunak lebih sering dipicu kelemahan perencanaan, koordinasi, dan pengendalian, bukan keterbatasan teknis. Dengan demikian, persoalan utama dalam proyek perangkat lunak bersifat manajerial.

Manajemen proyek perangkat lunak memberikan kerangka kerja untuk menyeimbangkan dimensi waktu, biaya, dan kualitas. Metodologi yang matang membantu organisasi meningkatkan produktivitas tim dan mengurangi *rework* akibat perubahan kebutuhan yang tidak terdokumentasi. Kerzner (2017) menyatakan bahwa kematangan manajemen proyek memungkinkan optimalisasi penggunaan sumber daya dan meminimalkan pemborosan, terutama pada perusahaan dengan banyak proyek berjalan secara paralel.

Urgensi manajemen proyek juga dipengaruhi meningkatnya tuntutan kualitas, keandalan, dan keamanan sistem. Sektor kesehatan dan keuangan, misalnya, memerlukan kepatuhan terhadap regulasi keamanan data yang ketat. Pressman dan Maxim (2020) menyatakan bahwa kualitas perangkat lunak bergantung pada proses manajerial yang disiplin sejak perencanaan hingga pemeliharaan. Dengan demikian, manajemen proyek tidak hanya berfungsi mengontrol biaya dan jadwal, tetapi juga memastikan keandalan dan keberlanjutan sistem.

Seiring meningkatnya ketergantungan organisasi terhadap sistem perangkat lunak, risiko keamanan menjadi tantangan semakin dominan. Kegagalan mengantisipasi risiko keamanan berpotensi memicu pencurian data, gangguan layanan, dan kerugian finansial. IBM (2022) melaporkan bahwa rata-rata biaya pelanggaran data global mencapai USD 4,35 juta, di mana sebagian kelemahan bersumber dari proses pengembangan perangkat lunak. Temuan ini menegaskan bahwa keamanan tidak dapat dipisahkan dari manajemen proyek perangkat lunak dan harus menjadi perhatian sejak tahap perencanaan.

Model pengembangan perangkat lunak tradisional (SDLC) umumnya menempatkan aspek keamanan sebagai aktivitas tambahan pada tahap akhir, terutama dalam fase pengujian. Pendekatan ini terbukti tidak efektif karena kerentanan baru teridentifikasi ketika sistem hampir selesai, sehingga membutuhkan biaya dan waktu tambahan untuk perbaikan. McGraw (2006) menyebut praktik "*security as an afterthought*" sebagai salah satu penyebab kegagalan sistem memenuhi standar keamanan.

Kondisi ini menegaskan perlunya integrasi keamanan sejak tahap awal pengembangan.

Secure Software Development Life Cycle (S-SDLC) hadir sebagai kerangka kerja yang memasukkan praktik keamanan ke setiap fase, mulai dari analisis kebutuhan hingga pemeliharaan. Dengan demikian, S-SDLC meningkatkan kualitas perangkat lunak sekaligus berfungsi sebagai strategi manajemen risiko yang bersifat proaktif. Microsoft (2018) melaporkan bahwa penerapan S-SDLC dapat mengurangi hingga 50 persen kerentanan yang lolos ke tahap produksi, sehingga menekan risiko keamanan dan menurunkan biaya perbaikan.

Dalam perspektif manajemen proyek, S-SDLC sejalan dengan prinsip manajemen risiko yang menuntut identifikasi, analisis, mitigasi, dan monitoring risiko sejak awal siklus. NIST (2020) menegaskan bahwa integrasi keamanan ke dalam SDLC merupakan praktik terbaik dalam manajemen risiko organisasi karena menghubungkan kontrol teknis dengan tata kelola manajerial. Hal ini menjadi kritis bagi perusahaan pengembang perangkat lunak yang melayani sektor-sektor berisiko tinggi seperti keuangan dan kesehatan.

Integrasi keamanan melalui S-SDLC juga memberikan manfaat efisiensi. Penemuan kerentanan lebih awal mengurangi biaya *rework* dan mempercepat proses rilis. Boehm dan Turner (2004) menunjukkan bahwa biaya memperbaiki kerentanan pada tahap desain hanya sepersepuluh dari biaya perbaikan setelah implementasi. Dari perspektif manajerial, S-SDLC

berkontribusi langsung pada efektivitas alokasi sumber daya dan keberlanjutan proyek.

Urgensi penerapan S-SDLC semakin menguat seiring munculnya regulasi dan standar industri yang mewajibkan keamanan aplikasi sebagai bagian dari siklus pengembangan. ISO/IEC 27034 menekankan integrasi keamanan dalam SDLC, sedangkan GDPR di Eropa mensyaratkan perlindungan data sejak tahap perancangan (*privacy by design*). Kepatuhan terhadap regulasi ini hanya dapat dicapai apabila manajemen proyek mengadopsi S-SDLC sebagai bagian dari kebijakan dan prosedur operasional.

Efektivitas manajemen proyek perangkat lunak tidak cukup hanya dinilai dari keberhasilan penyelesaian proyek tepat waktu dan sesuai anggaran, tetapi juga dari kemampuan organisasi menciptakan proses yang berulang, terukur, dan dapat ditingkatkan. Evaluasi semacam ini memerlukan alat ukur yang baku agar penilaian tidak bersifat subjektif. Kerzner (2017) menegaskan bahwa organisasi membutuhkan kerangka pengukuran formal untuk menilai tingkat kematangan manajemen proyek dan mengambil keputusan peningkatan berbasis data.

Capability Maturity Model Integration (CMMI) menyediakan kerangka komprehensif untuk menilai kematangan proses organisasi melalui *practice areas* yang berfokus pada konsistensi, prediktabilitas, dan efektivitas proses. Pendekatan ini sejalan dengan pandangan Chrissis, Konrad, dan Shrum (2011) yang menekankan bahwa CMMI meningkatkan kapabilitas organisasi secara sistematis melalui evaluasi proses. Versi terbaru, CMMI

v3, memperkuat keterkaitan antara praktik manajemen dan kinerja bisnis serta memberikan fleksibilitas dalam appraisal (CMMI Institute, 2018), sehingga organisasi tidak hanya mengukur kepatuhan proses, tetapi juga dampaknya terhadap output.

Bagi PT Magicsoft Teknologi Indonesia, penggunaan CMMI v3 berfungsi sebagai instrumen untuk menilai kematangan penerapan S-SDLC, mengidentifikasi gap dalam manajemen risiko, dan merumuskan langkah perbaikan yang terukur. Penilaian ini penting untuk meningkatkan efisiensi internal sekaligus memperkuat daya saing perusahaan di pasar jasa teknologi.

Penelitian ini relevan dalam disiplin manajemen karena fokus utamanya adalah menilai efektivitas manajemen proyek perangkat lunak melalui pendekatan berbasis proses. Dalam konteks manajemen, proyek perangkat lunak dipandang sebagai kegiatan terstruktur yang memerlukan perencanaan, pengorganisasian, dan pengendalian untuk mencapai tujuan bisnis. Dengan menggunakan CMMI v3 sebagai kerangka pengukuran, penelitian ini memberikan kontribusi pada manajemen proyek dengan menyediakan pendekatan sistematis untuk menilai kapabilitas organisasi dalam mengelola proyek teknologi informasi.

Selain itu, penelitian ini memiliki relevansi dengan manajemen risiko dan manajemen kualitas. Penerapan S-SDLC dikaji bukan semata sebagai praktik teknis, tetapi sebagai bagian dari strategi organisasi dalam mengelola risiko keamanan perangkat lunak. Dengan demikian, keamanan dan kualitas produk dapat diposisikan sebagai hasil dari praktik manajerial

yang efektif. Studi ini menunjukkan bahwa pendekatan berbasis kapabilitas proses melalui CMMI v3 dapat mendukung pengambilan keputusan manajerial serta mendorong perbaikan berkelanjutan dalam organisasi teknologi.

Hasil observasi awal peneliti menunjukkan bahwa tantangan utama dalam manajemen proyek perangkat lunak di PT Magicsoft Teknologi Indonesia terletak pada lemahnya dokumentasi teknis yang berhubungan dengan pengelolaan perubahan (*change management*). Setiap kali terjadi perubahan kebutuhan proyek atau rotasi anggota tim teknis, perusahaan menghadapi hambatan adaptasi karena tidak tersedianya dokumentasi memadai terkait scope teknis, arsitektur sistem, perilaku modul, maupun algoritma yang telah dikembangkan selama lebih dari sepuluh tahun.

Kondisi tersebut menimbulkan implikasi manajerial yang signifikan. Pertama, tenaga teknis baru membutuhkan waktu onboarding yang panjang untuk memahami sistem yang kompleks, sehingga menurunkan produktivitas tim. Kedua, ketidakjelasan dokumentasi berdampak pada keterlambatan penyelesaian pekerjaan, baik dalam pengembangan fitur baru, peningkatan (*enhancement*), maupun perbaikan *bug*. Keterlambatan ini berpotensi mengganggu jadwal rilis proyek serta pemenuhan *Service Level Agreement* (SLA) dengan klien.

Lebih jauh, masalah ini mempengaruhi kepercayaan klien dan citra perusahaan. Dalam industri perangkat lunak yang kompetitif, kualitas layanan yang tidak konsisten dan keterlambatan penyampaian hasil dapat menurunkan kepuasan pengguna dan melemahkan reputasi perusahaan.

Standish Group (2020) menegaskan bahwa kegagalan dalam manajemen dokumentasi dan komunikasi merupakan salah satu faktor penyebab keterlambatan proyek perangkat lunak. Dengan demikian, kasus yang terjadi di Magicsoft bukan sekadar kendala operasional, tetapi menunjukkan adanya kesenjangan manajerial dalam pengelolaan proyek.

Secara konseptual, masalah dokumentasi tersebut menandakan bahwa organisasi belum sepenuhnya mencapai kapabilitas manajemen proyek yang matang. Chrissis, Konrad, dan Shrum (2011) menyatakan bahwa salah satu indikator penting dalam CMMI adalah adanya *defined process* yang terdokumentasi dengan baik sehingga proses dapat berulang, terukur, dan berkelanjutan. Ketiadaan dokumentasi yang sistematis di Magicsoft menunjukkan bahwa proses pengelolaan pengetahuan dan perubahan masih berada pada level rendah, sehingga sulit mendukung efektivitas jangka panjang.

Kondisi ini memperkuat urgensi penelitian untuk menggunakan CMMI v3 sebagai kerangka evaluasi dan menilai penerapan S-SDLC sebagai praktik mitigasi risiko. Melalui pendekatan ini, perusahaan diharapkan dapat mengidentifikasi *gap process*, meningkatkan efektivitas manajemen proyek, serta memperkuat daya saingnya di pasar jasa teknologi.

Penelitian ini diposisikan pada persilangan antara manajemen proyek perangkat lunak, manajemen risiko, dan pengukuran kapabilitas organisasi. Penelitian terdahulu banyak mengkaji efektivitas manajemen proyek perangkat lunak melalui kerangka CMMI (Chrissis, Konrad, &

Shrum, 2011), dan sebagian lain membahas penerapan keamanan melalui S-SDLC (McGraw, 2006; NIST, 2020). Namun, masih terbatas studi yang secara eksplisit menghubungkan CMMI sebagai model evaluasi efektivitas dengan S-SDLC sebagai objek penerapan keamanan dalam konteks organisasi nyata. Dengan demikian, penelitian ini mengisi kesenjangan tersebut, khususnya dalam konteks industri perangkat lunak di Asia Tenggara.

PT Magicsoft Teknologi Indonesia dipilih sebagai objek penelitian karena memiliki kompleksitas organisasi yang tinggi dalam pengembangan perangkat lunak lintas sektor dan lintas negara (Indonesia, Singapura, dan Tiongkok). Secara empiris, perusahaan menghadapi kendala dokumentasi teknis, transfer pengetahuan antar tim, keterlambatan pengiriman fitur dan perbaikan *bug*, serta meningkatnya risiko keamanan. Kondisi ini menunjukkan adanya kebutuhan evaluasi efektivitas manajemen proyek sekaligus integrasi praktik keamanan dalam siklus pengembangan perangkat lunak. Akses peneliti yang dekat dengan perusahaan memungkinkan pengumpulan data primer secara mendalam melalui wawancara, observasi, dan dokumentasi.

Selain alasan akademis dan empiris, pemilihan objek penelitian juga didukung pertimbangan praktis. Peneliti merupakan praktisi teknologi dengan pengalaman lebih dari sepuluh tahun di industri perangkat lunak dan saat ini berkarier sebagai *General Lead Software Engineer* di PT Magicsoft Teknologi Indonesia. Posisi ini memberikan pemahaman kontekstual dan akses data yang relevan untuk mengkaji fenomena secara

komprehensif. Dengan demikian, penelitian menawarkan kontribusi teoritis sekaligus manfaat praktis dalam meningkatkan efektivitas proses internal organisasi.

Kebaruan penelitian ini terletak pada pendekatan integratifnya. Pertama, penelitian menggunakan CMMI v3 sebagai kerangka pengukuran efektivitas yang mengaitkan praktik manajemen dengan kinerja bisnis. Kedua, S-SDLC diposisikan bukan hanya sebagai praktik teknis, tetapi sebagai strategi manajemen risiko yang kematangannya diukur melalui indikator CMMI. Ketiga, penelitian menghadirkan studi empiris pada perusahaan pengembang perangkat lunak di Asia Tenggara sehingga memberikan bukti aplikasi model evaluasi berbasis proses dalam konteks nyata. Kontribusi ini memperluas penerapan CMMI ke ranah keamanan perangkat lunak dan menegaskan pentingnya pengelolaan risiko digital dalam manajemen proyek modern.

1.2 Rumusan Masalah

1. Bagaimana penerapan *Secure Software Lifecycle Development* di PT Magicsoft Teknologi Indonesia?
2. Bagaimana efektifitas manajemen proyek perangkat lunak berbasis CMMI di PT Magicsoft Teknologi Indonesia?

1.3 Fokus Penelitian

1. Penerapan *Secure Software Lifecycle Development* di PT Magicsoft Teknologi Indonesia.

2. Efektifitas manajemen proyek perangkat lunak berbasis CMMI di PT Magicsoft Teknologi Indonesia.

1.4 Tujuan Penelitian

1. Menganalisis penerapan *Secure Software Lifecycle Development* di PT Magicsoft Teknologi Indonesia.
2. Menganalisis efektifitas manajemen proyek perangkat lunak berbasis CMMI di PT Magicsoft Teknologi Indonesia.

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

1. Penelitian ini memberikan kontribusi akademik dengan mengintegrasikan dua kerangka penting, yaitu CMMI v3 sebagai model evaluasi efektivitas manajemen proyek dan S-SDLC sebagai praktik pengelolaan risiko keamanan perangkat lunak. Integrasi ini memperkaya literatur mengenai pendekatan evaluatif dalam manajemen proyek perangkat lunak.
2. Dengan menempatkan S-SDLC dalam perspektif manajemen, penelitian ini memperluas pemahaman teoretis tentang bagaimana praktik keamanan dapat diposisikan sebagai strategi manajemen risiko dan manajemen kualitas. Pendekatan ini menegaskan bahwa efektivitas proyek tidak hanya ditentukan oleh waktu, biaya, dan ruang lingkup, tetapi juga oleh keberlanjutan dan keamanan sistem.
3. Penelitian ini menyediakan bukti empiris dari konteks perusahaan pengembang perangkat lunak (PT Magicsoft Teknologi Indonesia)

yang relatif jarang diteliti dalam literatur internasional. Dengan demikian, penelitian menambah variasi perspektif dalam studi manajemen proyek dan keamanan perangkat lunak, khususnya di kawasan Asia Tenggara dan negara berkembang.

1.5.2 Manfaat Praktis

1. Penelitian ini membantu PT Magicsoft Teknologi Indonesia memahami tingkat efektivitas manajemen proyek perangkat lunak yang dijalankan, khususnya dalam penerapan S-SDLC. Temuan penelitian dapat menjadi dasar perbaikan proses manajerial, peningkatan kualitas dokumentasi, serta penguatan strategi manajemen risiko yang berpotensi meningkatkan produktivitas, kepuasan klien, dan daya saing perusahaan.
2. Penelitian ini dapat menjadi panduan praktis bagi manajer proyek, QA, dan engineer dalam mengintegrasikan aspek keamanan ke dalam siklus pengembangan perangkat lunak. Selain itu, model evaluasi berbasis CMMI v3 yang digunakan dalam penelitian ini dapat direplikasi oleh organisasi lain sebagai alat ukur efektivitas manajemen proyek perangkat lunak.
3. Hasil penelitian dapat memberikan wawasan bagi regulator maupun asosiasi industri mengenai pentingnya adopsi model kematangan proses (CMMI) dan praktik keamanan (S-SDLC). Penelitian ini berpotensi mendukung penyusunan kebijakan atau standar industri yang lebih baik dalam pengelolaan proyek perangkat lunak dan keamanan sistem.

BAB V

PEMBAHASAN

5.1 Pembahasan Hasil Temuan

5.1.1 Disiplin Proses vs Kinerja dalam Perspektif Manajemen Proyek

Temuan penelitian menunjukkan bahwa penerapan *Secure Software Development Life Cycle* (S-SDLC) di PT Magicsoft Teknologi Indonesia telah menghasilkan tingkat disiplin proses (*process discipline*) yang relatif tinggi dalam pengembangan perangkat lunak. Konsistensi perencanaan berbasis risiko, validasi kebutuhan, desain terstandarisasi, *secure coding*, *code review* berlapis, serta pengujian terstruktur menandakan bahwa organisasi telah berada pada level *process capability* yang mampu mereplikasi proses secara konsisten, terdokumentasi, dan dapat dikontrol (Chrissis et al., 2011). Pencapaian ini sejalan dengan prinsip CMMI bahwa tanpa disiplin proses yang memadai, organisasi tidak dapat mengelola variabilitas proyek maupun mempertahankan kualitas hasil.

Namun demikian, temuan penelitian juga memperlihatkan bahwa disiplin proses tersebut tidak secara otomatis berkontribusi pada peningkatan kinerja proyek dalam dimensi ketepatan waktu (*schedule performance*). Dalam literatur manajemen proyek perangkat lunak, terdapat pemisahan konseptual antara *process effectiveness* (sejauh mana proses berjalan sesuai standar) dan *project performance* (sejauh mana proyek mencapai target output dalam dimensi waktu, biaya, dan kualitas). PMI (2021) menegaskan bahwa proses yang efektif adalah instrumen pengendali, bukan determinan kinerja. Dengan demikian, keberhasilan S-

SDLC dalam menjaga kualitas dan keamanan tidak serta merta menjamin keberhasilan proyek dalam memenuhi jadwal.

Dalam perspektif rekayasa perangkat lunak, *relational gap* ini dapat dijelaskan melalui dua mekanisme kausal utama. Pertama, peningkatan disiplin proses menambah pekerjaan eksplisit (*explicit workload*) dalam bentuk review, dokumentasi, pengujian, dan kontrol keamanan. Pressman dan Maxim (2020) menunjukkan bahwa proses berlapis seperti verifikasi, validasi, dan review mengurangi *rework* namun meningkatkan durasi kerja pada setiap iterasi. Tanpa penyesuaian estimasi waktu dan kapasitas, proses yang lebih matang justru dapat memperpanjang jadwal. Kedua, proyek perangkat lunak merupakan *knowledge-intensive system* dengan kompleksitas bergantung pada dinamika kebutuhan, dependensi modul, dan interaksi stakeholder. Dalam konteks seperti ini, disiplin proses hanya dapat menahan variabilitas teknis, bukan variabilitas kapasitas tim atau dinamika perubahan kebutuhan.

Argumentasi ini selaras dengan kerangka *Theory of Constraints* (Goldratt, 1997) yang menyatakan bahwa kinerja sistem ditentukan oleh bottleneck atau kapasitas terlemah dalam sistem. Meski proses telah berjalan optimal, keterbatasan kapasitas sumber daya manusia menjadi constraint utama yang menahan kinerja waktu proyek. Dengan kata lain, peningkatan *process capability* tidak diikuti peningkatan *resource throughput*, sehingga menghasilkan gap antara *process control* dan *time performance*. Hal ini menunjukkan bahwa optimasi di satu subsistem

(proses teknis) tidak otomatis mengoptimalkan sistem proyek secara keseluruhan (*suboptimization effect*).

Literatur manajemen proyek perangkat lunak modern juga menambahkan mekanisme ketiga, yaitu kesenjangan antara proses formal dan realitas kerja tim (*workload realism*). Boehm (2000) menekankan bahwa proyek perangkat lunak sering gagal bukan karena proses yang salah, tetapi karena estimasi yang optimistis dan ketidakmampuan organisasi mengendalikan usaha riil (*effort*) yang dibutuhkan. Dalam konteks penelitian ini, S-SDLC berfungsi efektif sebagai *governance mechanism* untuk kontrol teknis dan keamanan, tetapi belum dilengkapi oleh mekanisme kapasitas yang memastikan bahwa beban proses sejalan dengan kapasitas produksi.

Secara analitis, temuan ini memperkuat argumen bahwa kedisiplinan proses merupakan *enabler of control*, bukan *direct driver* kinerja proyek. Efektivitasnya lebih dominan pada dimensi kualitas dan keamanan perangkat lunak, sementara performa waktu membutuhkan intervensi tambahan berupa perencanaan kapasitas (*capacity planning*), manajemen beban kerja (*workload management*), dan mekanisme prioritisasi yang adaptif. Dalam konteks CMMI v3, hal ini konsisten dengan pandangan bahwa peningkatan kematangan proses memerlukan kesinambungan antara domain teknis, manajerial, dan organisasi, bukan semata formalitas praktik proses (CMMI Institute, 2023).

Dengan demikian, secara teoritis dan empiris dapat disimpulkan bahwa keberhasilan S-SDLC di PT Magicsoft Teknologi Indonesia terutama

berkontribusi pada stabilitas dan kualitas hasil pengembangan perangkat lunak, tetapi belum sepenuhnya terkonversi menjadi peningkatan performa jadwal proyek. Konversi tersebut memerlukan penyelarasan antara process *discipline*, *resource throughput*, dan *workload realism* agar kematangan proses dapat diterjemahkan menjadi kinerja proyek yang utuh dalam dimensi teknis maupun temporal.

5.1.2 Beban Kerja Berlebihan sebagai Masalah Manajemen Kapasitas Proyek

Temuan penelitian menunjukkan bahwa keterlambatan deliverable proyek perangkat lunak di PT Magicsoft Teknologi Indonesia lebih disebabkan oleh akumulasi beban kerja berlebihan (*over workload*) pada tim proyek dibandingkan oleh kelemahan disiplin proses. Fenomena ini merupakan persoalan klasik dalam *resource capacity management*, yakni ketidakseimbangan antara kebutuhan usaha proyek (*workload demand*) dan kapasitas sumber daya yang tersedia (*resource throughput*). Dalam perspektif manajemen proyek, ketidakseimbangan tersebut merupakan salah satu determinan utama kinerja proyek, terutama pada proyek berbasis pengetahuan yang karakteristik aktivitasnya padat kognitif, iteratif, dan sulit diprediksi (PMI, 2021).

Analisis penelitian memperlihatkan bahwa praktik *Project Planning* dan *Project Monitoring and Control* telah dijalankan secara formal, namun belum sepenuhnya mampu mengantisipasi dinamika beban kerja kumulatif yang muncul dari multitasking lintas proyek dan intensitas pekerjaan operasional. Kondisi ini sejalan dengan kritik Kerzner (2017) bahwa

perencanaan proyek perangkat lunak sering kali berfokus pada estimasi aktivitas individual dan *nominal task duration*, tetapi mengabaikan konteks realisasi kerja yang melibatkan gangguan, *rework*, koordinasi lintas peran, dan aktivitas non-teknis yang tidak tercatat secara eksplisit dalam baseline rencana.

Dalam perspektif rekayasa perangkat lunak, Boehm (2000) menegaskan bahwa estimasi proyek cenderung optimistis karena tidak mempertimbangkan biaya koordinasi dan *cognitive switching* akibat multitasking. Temuan penelitian ini memperkuat argumen tersebut. Penerapan S-SDLC yang ketat menambahkan aktivitas kualitas dan keamanan (misalnya review berlapis, pengujian berulang, validasi, analisis risiko, dan dokumentasi) yang secara metodologis sah dan memberikan nilai teknis, tetapi meningkatkan beban kerja apabila tidak diimbangi dengan kapasitas sumber daya yang memadai. Dengan demikian, disiplin proses justru dapat menurunkan performa waktu apabila proses peningkatan kualitas tidak dikaitkan dengan perencanaan kapasitas.

Analisis lebih lanjut menunjukkan bahwa beban kerja berlebihan tidak hanya berdampak pada dimensi teknis, tetapi juga pada dimensi manusia. Literatur *human-centric project management* menegaskan bahwa kelelahan kerja (*work fatigue*) dan *schedule pressure* memiliki implikasi terhadap penurunan fokus, peningkatan kesalahan, dan penurunan produktivitas marginal (PMI, 2021). Temuan ini menunjukkan bahwa keterlambatan deliverable bukan akibat kegagalan individu, tetapi

merupakan konsekuensi sistemik dari tekanan kapasitas pada konteks kerja berbasis pengetahuan.

Dalam kerangka *systems thinking*, fenomena ini dapat dipahami melalui mekanisme *bottleneck*, di mana *throughput* proyek dibatasi oleh kapasitas sumber daya yang paling lemah dalam sistem (Goldratt, 1997). Meskipun S-SDLC telah mengoptimalkan proses teknis, kapasitas tim tetap menjadi constraint utama. Dengan kata lain, peningkatan kematangan proses tanpa intervensi kapasitas hanya memindahkan locus persoalan dari domain proses ke domain sumber daya.

Temuan penelitian juga mengindikasikan adanya *misalignment* antara praktik manajemen proyek di level proyek dan kebutuhan pengelolaan sumber daya di level organisasi. Beban kerja berlebihan sering kali merupakan konsekuensi dari keputusan portofolio proyek, seperti penugasan sumber daya yang sama ke beberapa proyek sekaligus dengan prioritas yang tidak disinkronkan. Kerzner (2017) menegaskan bahwa tanpa koordinasi pada level portofolio, manajemen proyek individual cenderung mengalami *capacity conflict* meskipun proses telah dikelola dengan baik. Temuan ini memperkuat argumen tersebut dan menjelaskan mengapa keterlambatan tetap terjadi walaupun praktik CMMI dan S-SDLC telah dijalankan secara konsisten pada level proyek.

Secara analitis, temuan penelitian menegaskan bahwa beban kerja berlebihan merupakan determinan krusial yang membatasi efektivitas manajemen proyek perangkat lunak di PT Magicsoft Teknologi Indonesia. Keberhasilan penerapan S-SDLC dan CMMI harus ditopang oleh praktik

manajemen kapasitas yang realistis. Tanpa pengendalian beban kerja dan penyesuaian kapasitas yang sejalan dengan peningkatan kematangan proses, disiplin proses berisiko menurunkan *schedule performance* meskipun meningkatkan dimensi kualitas dan keamanan. Dalam perspektif manajemen proyek, implikasi ini menekankan pentingnya integrasi antara *process governance* dan *resource governance* agar manfaat proses dapat dikonversi secara penuh ke dalam kinerja proyek (Boehm, 2000; Kerzner, 2017; PMI, 2021).

5.1.3 Keseimbangan antara Kualitas, Keamanan, dan Jadwal

Temuan penelitian menunjukkan bahwa penerapan *Secure Software Development Life Cycle* (S-SDLC) di PT Magicsoft Teknologi Indonesia membentuk prioritas manajerial yang kuat terhadap dimensi keamanan dan kualitas sistem. Aktivitas tambahan seperti analisis risiko keamanan, *multi-layer code review*, pengujian berulang, dan validasi lintas peran mencerminkan komitmen organisasi terhadap pencegahan cacat dan kerentanan sejak tahap awal pengembangan. Dalam perspektif manajemen proyek, keputusan ini dapat dipahami sebagai bentuk alokasi preferensi terhadap dimensi ruang lingkup dan kualitas dalam kerangka *project management triple constraint* (Kerzner, 2017), di mana setiap peningkatan kualitas dan keamanan berpotensi memengaruhi durasi dan beban kerja proyek.

Secara analitis, penerapan S-SDLC memperluas ruang lingkup aktivitas proyek secara inheren. Namun perlu dicatat bahwa perluasan ruang lingkup tersebut tidak selalu diikuti dengan penyesuaian jadwal dan

kapasitas sumber daya yang proporsional. Kondisi ini menciptakan tensi sistemik antara kualitas/keamanan dan ketepatan waktu penyelesaian, sebagaimana ditegaskan oleh PMI (2021) bahwa perubahan pada satu dimensi kendala proyek hampir selalu memengaruhi dimensi lainnya. Temuan penelitian ini memperlihatkan bahwa organisasi memilih untuk memprioritaskan keamanan dan kualitas sebagai *primary project value*, sementara dimensi waktu mengalami tekanan berupa keterlambatan deliverable.

Dalam diskursus rekayasa perangkat lunak, Pressman dan Maxim (2020) menekankan bahwa investasi pada pengendalian kualitas dan keamanan pada tahap awal merupakan strategi yang secara ekonomis menguntungkan untuk jangka panjang, karena menurunkan biaya *rework* dan risiko cacat pada fase pasca-*deployment*. Namun, dalam horizon jangka pendek manajemen proyek, aktivitas tersebut sering diasosiasikan sebagai *cost driver* tambahan yang memperluas durasi pengembangan. Temuan ini memperlihatkan adanya tensi epistemik antara perspektif rekayasa perangkat lunak yang berorientasi kualitas dan perspektif manajemen proyek yang berorientasi ketepatan waktu.

Dalam konteks keamanan perangkat lunak, McGraw (2006) menegaskan bahwa praktik *building security in* efektivitasnya berbanding lurus dengan kedalaman integrasi aktivitas keamanan, namun integrasi tersebut meningkatkan beban kerja dan kompleksitas teknis apabila tidak diiringi dengan perencanaan kapasitas yang memadai. Temuan penelitian ini memperkuat pandangan tersebut, di mana penerapan S-SDLC

meningkatkan kualitas dan keamanan, namun menimbulkan tekanan jadwal ketika kapasitas tim menjadi *bottleneck*.

Dalam perspektif manajemen proyek adaptif, situasi ini menuntut pengambilan keputusan berbasis prioritas (*priority-based decision-making*). Kerzner (2017) menyatakan bahwa manajer proyek harus mampu membuat *trade-off* yang eksplisit antara kualitas, biaya, dan waktu sesuai dengan tujuan strategis organisasi. Temuan menunjukkan bahwa PT Magicsoft Teknologi Indonesia secara implisit mengambil keputusan untuk mempertahankan kualitas dan keamanan meskipun terdapat risiko keterlambatan waktu. Namun, *trade-off* tersebut belum sepenuhnya dikomunikasikan atau dinegosiasikan sebagai strategi manajemen proyek, sehingga menciptakan kesenjangan antara prioritas aktual dan ekspektasi jadwal pemangku kepentingan.

Dari perspektif komunikasi proyek, PMI (2021) menegaskan bahwa penetapan *trade-off* seharusnya diikuti dengan penyesuaian baseline jadwal dan ekspektasi pemangku kepentingan. Tanpa pengelolaan ekspektasi yang eksplisit, *trade-off* yang sebenarnya rasional dapat dipersepsikan sebagai penurunan kinerja. Temuan penelitian ini mencerminkan kondisi tersebut: kualitas dan keamanan meningkat, tetapi dimensi waktu dipersepsikan kurang optimal.

Dengan demikian, secara analitis dapat disimpulkan bahwa penerapan S-SDLC di PT Magicsoft Teknologi Indonesia menghasilkan *trade-off* struktural antara keamanan, kualitas, dan waktu yang merupakan konsekuensi inheren dari proyek perangkat lunak berbasis pengetahuan.

Keberhasilan pengelolaan *trade-off* ini bergantung pada kemampuan organisasi dalam mengintegrasikan dimensi proses (S-SDLC), kapasitas sumber daya, dan manajemen ekspektasi proyek. Dalam perspektif manajemen proyek, S-SDLC terbukti efektif sebagai mekanisme peningkatan kualitas dan keamanan, tetapi memerlukan dukungan praktik penyesuaian jadwal dan kapasitas agar *trade-off* tersebut dapat dikelola secara transparan dan strategis (Kerzner, 2017; PMI, 2021).

5.1.4 Efektivitas Manajemen Proyek sebagai Konsep Multidimensional

Temuan penelitian menunjukkan bahwa efektivitas manajemen proyek perangkat lunak berbasis CMMI dan S-SDLC di PT Magicsoft Teknologi Indonesia tidak dapat ditentukan hanya dari satu indikator kinerja seperti kepatuhan proses atau ketepatan waktu. Dalam disiplin manajemen proyek, efektivitas dipandang sebagai konsep multidimensional yang mencakup kinerja waktu, biaya, ruang lingkup, kualitas, serta keberlanjutan sumber daya manusia (Kerzner, 2017). Temuan ini memperkuat pandangan bahwa proyek dapat dinilai efektif pada dimensi tertentu, seperti kualitas dan keamanan, namun kurang optimal pada dimensi lain, seperti ketepatan waktu penyelesaian *deliverable*.

Dari perspektif kematangan proses, penerapan CMMI v3 dan S-SDLC telah membangun dasar tata kelola pengembangan perangkat lunak yang kuat melalui konsistensi praktik *Doing, Managing,* dan *Enabling*. Namun, sebagaimana ditegaskan oleh Chrissis et al. (2011), kematangan proses merupakan sarana untuk mencapai kinerja proyek yang baik, bukan tujuan akhir. Nilai proses baru dapat terkonversi menjadi peningkatan

kinerja apabila berkontribusi terhadap seluruh dimensi hasil proyek, tidak hanya dimensi teknis.

Dalam proyek berbasis pengetahuan seperti pengembangan perangkat lunak, faktor manusia memegang peran penting dalam menentukan efektivitas. PMI (2021) menekankan bahwa kinerja pekerjaan dipengaruhi oleh keseimbangan beban kerja, kapasitas mental, dan keberlanjutan energi tim. Temuan mengenai beban kerja berlebihan dalam penelitian ini menunjukkan bahwa efektivitas proses yang tinggi dapat berkurang apabila tidak diimbangi dengan perhatian terhadap aspek manusia, sehingga menegaskan bahwa efektivitas memiliki dimensi sosial dan psikologis yang tidak dapat diabaikan.

Di sisi lain, dari sudut pandang *value-based project management*, efektivitas proyek ditentukan oleh nilai yang dihasilkan bagi organisasi dan pemangku kepentingan (PMI, 2021). Dalam konteks PT Magicsoft Teknologi Indonesia, keterlambatan *deliverable* dapat dianggap mengurangi efektivitas proyek dalam jangka pendek, meskipun kualitas dan keamanan produk meningkat dan menciptakan nilai jangka panjang. Hal ini memperlihatkan adanya ketegangan strategis antara nilai jangka pendek (waktu) dan nilai jangka panjang (kualitas dan keamanan), yang membutuhkan mekanisme *trade-off* yang eksplisit dalam manajemen proyek.

Secara kritis, temuan ini juga mengidentifikasi keterbatasan penggunaan indikator tradisional efektivitas proyek, seperti waktu dan biaya. Literatur kontemporer mendorong pendekatan evaluasi yang lebih

holistik melalui penambahan indikator keberlanjutan sumber daya, kepuasan pemangku kepentingan, dan pembelajaran organisasi (Kerzner, 2017; PMI, 2021). Dalam konteks penelitian ini, penerapan S-SDLC terbukti mendukung pembelajaran organisasi dalam aspek keamanan dan kualitas, namun tantangan terkait kapasitas sumber daya menunjukkan bahwa efektivitas secara keseluruhan belum tercapai secara seimbang.

Dengan demikian, efektivitas manajemen proyek perangkat lunak dalam penelitian ini dapat dipahami sebagai hasil interaksi antara disiplin proses, pengelolaan sumber daya, dan kemampuan adaptasi organisasi dalam mengelola nilai proyek. Peningkatan kematangan proses melalui CMMI dan S-SDLC perlu diiringi dengan manajemen kapasitas dan portofolio proyek agar manfaatnya tidak bersifat parsial. Temuan ini berkontribusi pada literatur manajemen proyek dengan menegaskan bahwa keberhasilan proyek perangkat lunak bersifat multidimensional dan menuntut keseimbangan antara proses, manusia, dan nilai bisnis yang dihasilkan (Chrissis et al., 2011; Kerzner, 2017; PMI, 2021).

Efektivitas manajemen proyek perangkat lunak pada dasarnya merupakan konsep multidimensional yang tidak hanya mencerminkan pencapaian target waktu, biaya, dan kualitas, tetapi juga melibatkan efisiensi proses, keberlanjutan kinerja, serta kemampuan organisasi dalam beradaptasi terhadap dinamika proyek. Dalam konteks ini, fungsi manajemen klasik yang dirumuskan dalam kerangka *Planning, Organizing, Actuating, Controlling* (POAC) memberikan landasan konseptual yang komprehensif untuk mengintegrasikan berbagai temuan penelitian ke

dalam suatu sistem pengelolaan yang utuh. Terry (1972) menegaskan bahwa keempat fungsi tersebut bersifat interdependen dan membentuk siklus manajerial yang menentukan keberhasilan organisasi dalam mencapai tujuan secara efektif dan efisien. Oleh karena itu, efektivitas manajemen proyek tidak dapat dipahami secara parsial, melainkan sebagai hasil integrasi dari keseluruhan fungsi manajemen yang berjalan secara konsisten.

Dalam dimensi *Planning*, temuan penelitian menunjukkan bahwa praktik perencanaan proyek berbasis CMMI, khususnya melalui *Project Planning* dan *Requirements Development*, telah mampu menghasilkan struktur kerja yang sistematis dan terdokumentasi dengan baik. Perencanaan ini mencakup penentuan ruang lingkup, jadwal, serta identifikasi risiko yang mendukung keterkendalian proyek. Namun, efektivitas perencanaan tidak hanya ditentukan oleh kelengkapan dokumen, tetapi juga oleh akurasi dalam mengestimasi kapasitas sumber daya dan kompleksitas pekerjaan. Menurut Kerzner (2017), perencanaan yang tidak realistis akan menciptakan deviasi sejak awal proyek, yang pada akhirnya berdampak pada inefisiensi dan keterlambatan. Dengan demikian, fungsi *Planning* berkontribusi terhadap efektivitas apabila mampu menyelaraskan antara target proyek dan kapasitas operasional secara proporsional.

Pada dimensi *Organizing*, hasil penelitian yang tercermin dalam kelompok *Enabling* menunjukkan adanya pengelolaan struktur kerja yang mendukung koordinasi antar pemangku kepentingan melalui praktik seperti

Stakeholder Engagement dan *Configuration Management*. Fungsi ini berperan dalam memastikan bahwa aliran informasi, pembagian tugas, dan keterlacakan artefak proyek berjalan secara konsisten. Efisiensi dalam organisasi proyek tercermin dari minimnya miskomunikasi dan reduplikasi pekerjaan, yang pada akhirnya mendukung stabilitas proses pengembangan. Robbins dan Coulter (2018) menyatakan bahwa struktur organisasi yang jelas dan mekanisme koordinasi yang efektif merupakan determinan utama dalam meningkatkan efisiensi operasional. Oleh karena itu, fungsi *Organizing* menjadi fondasi penting dalam mengurangi pemborosan sumber daya dan meningkatkan kualitas kolaborasi tim.

Dimensi *Actuating* atau pelaksanaan mencerminkan bagaimana rencana dan struktur yang telah disusun diwujudkan dalam aktivitas operasional proyek. Temuan pada kelompok *Doing* menunjukkan bahwa praktik seperti *Technical Solution*, *Verification*, dan *Validation* telah berjalan secara sistematis dan berorientasi kualitas. Hal ini menunjukkan adanya disiplin proses yang tinggi dalam implementasi SSDLC. Namun demikian, efektivitas pada tahap ini sangat dipengaruhi oleh faktor manusia, khususnya terkait beban kerja dan kapasitas tim. Studi dalam manajemen sumber daya manusia menunjukkan bahwa tekanan kerja yang berlebihan dapat menurunkan produktivitas dan meningkatkan risiko kesalahan (Sommerville, 2016). Dengan demikian, fungsi *Actuating* tidak hanya menuntut kepatuhan terhadap prosedur, tetapi juga keseimbangan antara tuntutan kerja dan kemampuan individu dalam mengeksekusi tugas secara berkelanjutan.

Sementara itu, dimensi *Controlling* berperan dalam memastikan bahwa pelaksanaan proyek tetap berada dalam koridor yang telah direncanakan melalui mekanisme pemantauan, evaluasi, dan tindakan korektif. Temuan penelitian menunjukkan bahwa praktik seperti *Project Monitoring & Control* serta *Decision Analysis & Resolution* telah memungkinkan organisasi untuk mendeteksi deviasi dan merespons perubahan secara adaptif. Namun, sebagaimana diidentifikasi dalam analisis PDCA, pengendalian yang efektif tidak hanya berfokus pada kepatuhan terhadap jadwal dan prosedur, tetapi juga pada indikator kinerja yang lebih luas, termasuk efisiensi sumber daya dan keberlanjutan kerja tim. Menurut PMI (2021), fungsi pengendalian yang komprehensif harus mampu mengintegrasikan data kinerja proyek dengan pengambilan keputusan strategis. Oleh karena itu, *Controlling* menjadi elemen kunci dalam menjembatani antara kepatuhan proses dan pencapaian hasil yang optimal.

Secara keseluruhan, integrasi temuan penelitian ke dalam kerangka POAC menunjukkan bahwa efektivitas dan efisiensi manajemen proyek perangkat lunak merupakan hasil dari sinergi antar fungsi manajemen yang berjalan secara simultan. CMMI menyediakan kerangka kematangan proses yang memperkuat setiap fungsi tersebut, sementara implementasi SSDLC memberikan konteks operasional dalam pengembangan perangkat lunak yang aman dan berkualitas. Namun, temuan juga menegaskan bahwa keberhasilan tidak hanya ditentukan oleh tingkat formalitas proses, melainkan oleh kemampuan organisasi dalam menyeimbangkan antara

standar prosedural, kapasitas sumber daya, dan dinamika proyek. Dengan demikian, efektivitas manajemen proyek sebagai konsep multidimensional menuntut pendekatan yang holistik, di mana efisiensi proses dan pencapaian kinerja diintegrasikan dalam satu sistem manajemen yang adaptif dan berkelanjutan.

5.1.6 CMMI dan Pergeseran Paradigma Manajemen Proyek dari Hasil ke Proses

Temuan penelitian menunjukkan bahwa adopsi CMMI v3 dalam manajemen proyek perangkat lunak di PT Magicsoft Teknologi Indonesia mencerminkan pergeseran paradigma dari orientasi hasil (*output-oriented*) menuju orientasi proses (*process-oriented*). Pergeseran ini mengindikasikan bahwa keberhasilan proyek tidak hanya diukur dari ketepatan waktu dan biaya, tetapi dari bagaimana pekerjaan dilaksanakan secara konsisten, terdokumentasi, dan terkendali. Dalam struktur CMMI, proses diperlakukan sebagai aset strategis yang memengaruhi keberlanjutan kinerja proyek (Chrissis et al., 2011).

Secara teoretis, orientasi proses tersebut selaras dengan pandangan Kerzner (2017) yang menegaskan bahwa prediktabilitas dan pengendalian kinerja proyek memerlukan pengendalian proses, bukan semata hasil akhir. Temuan penelitian memperlihatkan bahwa praktik CMMI pada level proyek berperan sebagai mekanisme institusionalisasi disiplin proses, sehingga aktivitas pengembangan tidak lagi bergantung pada praktik individual, tetapi pada prosedur yang dapat direplikasi lintas pekerjaan maupun lintas tim.

Namun demikian, pergeseran orientasi ini membawa implikasi manajerial yang tidak dapat diabaikan. Peningkatan formalitas proses, dokumentasi, dan review berlapis secara praktis dapat menambah kompleksitas pengelolaan proyek dan menurunkan kelincahan organisasi dalam menghadapi perubahan kebutuhan atau tekanan jadwal. PMI (2021) mengingatkan bahwa orientasi proses yang tidak disesuaikan dengan konteks dapat menghasilkan *trade-off* yang tidak disengaja antara disiplin proses dan kecepatan *delivery*. Kondisi ini relevan dengan temuan lapangan, di mana kualitas dan keterkendalian meningkat, namun tidak selalu sejalan dengan percepatan penyelesaian proyek.

Dalam perspektif rekayasa perangkat lunak, Pressman dan Maxim (2020) menyatakan bahwa proses yang kuat merupakan fondasi untuk menjaga kualitas dan mengurangi risiko kegagalan teknis. Temuan penelitian di PT Magicsoft Teknologi Indonesia mengonfirmasi relevansi pandangan tersebut, karena praktik CMMI terbukti membantu menjaga konsistensi teknis dan kualitas implementasi melalui keterpaduan kebutuhan, desain, implementasi, dan pengujian. Namun, manfaat ini memerlukan koordinasi dan kontrol proses yang lebih intensif, yang berimplikasi pada peningkatan beban manajerial.

Lebih jauh, orientasi proses dalam CMMI dapat dipahami sebagai respons metodologis terhadap kompleksitas proyek perangkat lunak modern. Sommerville (2016) menegaskan bahwa kompleksitas dan ketergantungan antar komponen sistem menuntut struktur proses yang mampu mengelola perubahan secara sistematis. Dalam konteks ini, CMMI

berfungsi sebagai alat reduksi ketidakpastian, meskipun tidak secara otomatis menghilangkan tekanan operasional yang terkait dengan sumber daya, jadwal, maupun dinamika kebutuhan pengguna.

Secara kritis, temuan ini menegaskan bahwa pergeseran dari orientasi hasil ke orientasi proses memerlukan kejelasan tujuan manajerial agar proses diperlakukan sebagai sarana untuk mencapai nilai, bukan tujuan itu sendiri. Dalam perspektif *value-based project management*, keberhasilan proses diukur dari kontribusinya terhadap nilai bisnis, bukan dari tingkat kepatuhan prosedural (PMI, 2021). Tanpa kejelasan ini, organisasi berisiko terjebak dalam fenomena *process compliance* yang tinggi tetapi kehilangan efisiensi dan ketepatan waktu proyek.

Dengan demikian, secara analitis dapat disimpulkan bahwa relevansi CMMI v3 dalam diskursus manajemen proyek terletak pada kemampuannya memperkuat disiplin proses dan keterkendalian proyek, sekaligus memindahkan orientasi dari hasil akhir menuju proses kerja yang lebih terstruktur. Namun, efektivitasnya sangat bergantung pada kemampuan organisasi menjaga keseimbangan antara struktur proses dan fleksibilitas operasional. Temuan penelitian menegaskan bahwa CMMI efektif sebagai kerangka penguatan proses selama diadopsi secara adaptif dan selaras dengan tujuan strategis organisasi (Chrissis et al., 2011; Kerzner, 2017; PMI, 2021).

Pergeseran paradigma dalam manajemen proyek dari orientasi hasil (*output-oriented*) menuju orientasi proses (*process-oriented*) menempatkan efisiensi dan efektivitas sebagai indikator keberhasilan yang lebih

komprehensif dan berkelanjutan. Dalam pendekatan tradisional, keberhasilan proyek umumnya diukur berdasarkan pencapaian batasan waktu, biaya, dan ruang lingkup (*iron triangle*). Namun, pendekatan ini cenderung mengabaikan kualitas proses yang menghasilkan output tersebut. CMMI hadir sebagai kerangka yang menekankan bahwa kualitas hasil proyek merupakan konsekuensi langsung dari kematangan proses yang mendasarinya. Chrissis et al. (2011) menegaskan bahwa organisasi dengan proses yang terdefinisi dan terkelola dengan baik cenderung menghasilkan kinerja proyek yang lebih konsisten, sehingga efektivitas tidak hanya dilihat dari hasil akhir, tetapi dari stabilitas dan keterkendalian proses yang menghasilkan hasil tersebut.

Dalam konteks efisiensi, penerapan CMMI mendorong organisasi untuk meminimalkan pemborosan melalui standardisasi proses, dokumentasi yang sistematis, serta pengendalian aktivitas proyek secara berkelanjutan. Temuan penelitian menunjukkan bahwa praktik dalam kelompok *Doing*, *Managing*, dan *Enabling* telah mendukung terciptanya alur kerja yang terstruktur dan terkoordinasi, sehingga mengurangi redundansi aktivitas dan meningkatkan pemanfaatan sumber daya secara optimal. Hal ini sejalan dengan pandangan Sommerville (2016) yang menyatakan bahwa organisasi dengan proses pengembangan yang terstandarisasi cenderung lebih efisien karena mampu mengurangi kesalahan berulang dan meningkatkan kejelasan alur kerja. Dengan demikian, efisiensi dalam manajemen proyek berbasis CMMI tidak hanya tercermin pada

penghematan biaya atau waktu, tetapi juga pada optimalisasi proses yang mendukung produktivitas tim.

Sementara itu, efektivitas dalam manajemen proyek berbasis CMMI berkaitan dengan sejauh mana tujuan proyek dapat dicapai secara konsisten dalam berbagai kondisi operasional. Temuan penelitian menunjukkan bahwa mekanisme seperti *Project Monitoring & Control* dan *Risk Management* memungkinkan organisasi untuk mengidentifikasi potensi deviasi sejak dini dan melakukan penyesuaian secara adaptif. Hal ini mencerminkan pergeseran dari pendekatan reaktif menuju proaktif dalam pengelolaan proyek. Menurut PMI (2021), efektivitas proyek tidak hanya diukur dari pencapaian target, tetapi juga dari kemampuan organisasi dalam merespons perubahan dan mempertahankan kinerja di tengah ketidakpastian. Dengan demikian, efektivitas dalam paradigma berbasis proses menekankan pada kemampuan sistem manajemen proyek untuk menghasilkan hasil yang dapat diprediksi dan berkelanjutan.

Namun demikian, temuan penelitian juga menunjukkan bahwa efisiensi dan efektivitas tidak sepenuhnya ditentukan oleh tingkat kepatuhan terhadap proses formal. Meskipun praktik CMMI telah diimplementasikan dengan baik, masih terdapat indikasi keterlambatan delivery yang berkaitan dengan faktor kapasitas kerja dan beban operasional tim. Hal ini mengindikasikan bahwa keberhasilan manajemen proyek merupakan hasil dari interaksi antara kualitas proses dan kondisi nyata sumber daya manusia. Oleh karena itu, pergeseran paradigma dari hasil ke proses perlu dilengkapi dengan perspektif yang lebih holistik, di

mana efisiensi dan efektivitas tidak hanya dihasilkan dari standardisasi proses, tetapi juga dari kemampuan organisasi dalam mengelola dinamika tim dan kapasitas operasional. Dengan demikian, CMMI berfungsi sebagai fondasi struktural, sementara keberhasilan implementasinya tetap bergantung pada integrasi antara proses, manusia, dan konteks proyek secara menyeluruh.

5.1.7 Adaptabilitas CMMI dalam Konteks *Fit-for-Purpose Project Management*

Temuan penelitian menunjukkan bahwa adopsi CMMI v3 di PT Magicsoft Teknologi Indonesia tidak dijalankan sebagai model yang kaku dan menyeluruh, tetapi diadaptasi secara selektif sesuai konteks proyek. Pendekatan ini mencerminkan prinsip *fit-for-purpose project management* yang menegaskan bahwa efektivitas suatu kerangka manajemen sangat ditentukan oleh kesesuaiannya dengan tujuan, kompleksitas, serta keterbatasan proyek, bukan oleh tingkat kepatuhan formal terhadap seluruh elemen kerangka (PMI, 2021). Dengan kata lain, relevansi CMMI justru meningkat ketika model tersebut dioperasionalisasikan secara kontekstual.

Secara konseptual, CMMI memang dirancang sebagai model referensi (*reference model*), bukan metodologi operasional yang wajib diterapkan secara penuh pada setiap organisasi (CMMI Institute, 2023). Keputusan penelitian untuk memfokuskan analisis pada 12 Practice Area yang dapat diamati pada level proyek menunjukkan pemahaman terhadap karakter CMMI sebagai kerangka yang fleksibel dan dapat ditailor.

Pendekatan ini sejalan dengan prinsip *contingency theory* yang menolak keberadaan "*one best way*" dalam manajemen proyek (Kerzner, 2017).

Dalam konteks pengembangan perangkat lunak, adaptabilitas CMMI menjadi penting mengingat tingginya variabilitas kompleksitas teknis, dinamika kebutuhan pengguna, dan keterbatasan sumber daya tim. Sommerville (2016) mengingatkan bahwa penerapan proses yang terlalu formal tanpa mempertimbangkan konteks dapat mengurangi responsivitas dan memperlambat pengambilan keputusan. Temuan penelitian ini menunjukkan bahwa selektivitas dalam adopsi CMMI memberikan manfaat pengendalian proses tanpa menghadirkan beban birokrasi berlebihan yang sering dikritik dalam model proses yang *heavyweight*.

Pendekatan *fit-for-purpose* ini juga menantang kritik klasik terhadap CMMI sebagai kerangka yang *rigid* dan tidak adaptif. Pressman dan Maxim (2020) menegaskan bahwa kerangka proses tetap relevan selama diterapkan secara pragmatis dan proporsional terhadap risiko proyek. Dalam kasus PT Magicsoft Teknologi Indonesia, CMMI berfungsi sebagai *process backbone* yang mendukung stabilitas proses dan konsistensi teknis tanpa menghambat fleksibilitas perangkat lunak.

Secara kritis, temuan penelitian juga mengindikasikan bahwa kemampuan men-*tailor* CMMI menuntut kapabilitas manajerial yang memadai. Pemilihan *Practice Area* yang relevan, penentuan tingkat formalitas proses, serta integrasi dengan praktik manajemen proyek lain memerlukan pemahaman konseptual yang kuat mengenai tujuan dan batasan proyek. PMI (2021) menegaskan bahwa *tailoring* merupakan

kompetensi kunci yang membedakan organisasi yang matang secara manajerial dengan organisasi yang sekadar patuh standar. Dengan demikian, adaptabilitas CMMI bukan fenomena otomatis, tetapi merupakan fungsi dari kualitas keputusan manajerial.

Dalam diskursus yang lebih luas, temuan penelitian ini memperkuat posisi CMMI sebagai kerangka pendukung (*supporting framework*) yang kompatibel dengan pendekatan manajemen proyek modern, termasuk *Agile* dan model *hybrid*. Kerzner (2017) menekankan bahwa organisasi proyek modern semakin bergerak menuju kombinasi kerangka kerja yang saling melengkapi, bukan adopsi tunggal yang absolut. Dalam konteks ini, CMMI berfungsi sebagai lapisan pengendalian proses yang stabil di atas praktik pengembangan yang lebih lincah.

Dengan demikian, secara analitis dapat disimpulkan bahwa relevansi CMMI v3 dalam manajemen proyek terletak pada kapasitasnya untuk diadaptasi secara *fit-for-purpose*. Dalam konteks PT Magicsoft Teknologi Indonesia, adopsi selektif terhadap CMMI memungkinkan terciptanya keseimbangan antara disiplin proses dan fleksibilitas operasional. Temuan ini memberikan kontribusi konseptual bagi diskursus manajemen proyek dengan menegaskan bahwa CMMI tetap relevan sebagai kerangka tata kelola selama diadopsi secara kontekstual, adaptif, dan berorientasi pada tujuan strategis organisasi (CMMI Institute, 2023; Kerzner, 2017; PMI, 2021).

5.2. Implikasi Penelitian

5.2.1 Implikasi Teoretis

Hasil penelitian ini memberikan kontribusi teoretis yang memperkuat sekaligus memperluas diskursus mengenai efektivitas manajemen proyek perangkat lunak berbasis CMMI. Sejalan dengan Asmy dan Hasugian (2021) serta Wibisono (2020), penelitian ini mengonfirmasi bahwa dokumentasi dan standardisasi proses merupakan determinan utama dalam pembentukan kematangan proses. Namun, penelitian ini menunjukkan bahwa ketika standar proses diterapkan secara konsisten lintas proyek, kapabilitas proses dapat mencapai level *Defined* (*Capability* Level 3) sebagaimana dimodelkan dalam CMMI v3. Dengan demikian, penelitian ini mempertegas posisi teoritis CMMI yang menempatkan standardisasi proses sebagai fondasi utama peningkatan kinerja proyek perangkat lunak.

Dari perspektif *Secure Software Development Life Cycle* (S-SDLC), penelitian ini memperkuat teori *security by design* (McGraw, 2006) dan *shift-left security* dengan menunjukkan bahwa integrasi keamanan ke dalam setiap fase SDLC tidak hanya berdampak pada kualitas teknis perangkat lunak, tetapi juga pada efektivitas manajemen proyek. Penelitian terdahulu cenderung memisahkan evaluasi keamanan dari evaluasi proses, sementara penelitian ini menunjukkan bahwa keamanan dapat diposisikan sebagai variabel integral dalam evaluasi efektivitas manajemen proyek berbasis proses, khususnya dalam konteks CMMI v3 yang lebih fleksibel dan kontekstual.

Selain itu, penelitian ini memperluas temuan Tsai (2021) mengenai peran faktor manusia dan kolaborasi dalam implementasi CMMI. Temuan empiris menunjukkan bahwa efektivitas penerapan S-SDLC dan CMMI v3 tidak semata-mata bergantung pada instrumen teknis, prosedur formal, dan dokumentasi standar, tetapi juga pada mekanisme koordinasi, komunikasi lintas peran, serta budaya kerja yang mendukung keamanan dan kualitas. Secara teoretis, hal ini menegaskan bahwa model kematangan proses perlu dipahami sebagai sistem sosio-teknis, bukan sekadar kerangka prosedural.

Penelitian ini juga memberikan kontribusi terhadap kajian integrasi CMMI dan pendekatan *Agile* sebagaimana dibahas oleh Yilmaz dan Ozcan (2023). Temuan bahwa praktik Agile dapat memenuhi sebagian besar *Practice Area* CMMI v3 yang relevan, dan sekaligus memperkuat pengendalian keamanan melalui S-SDLC, memberikan implikasi bahwa pendekatan proses tidak harus dipertentangkan dengan fleksibilitas teknis. Secara teoretis hal ini membuka ruang integrasi konseptual antara CMMI, Agile, dan S-SDLC dalam kerangka manajemen proyek adaptif dan aman.

5.2.2 Implikasi Praktis

Secara praktis, hasil penelitian ini memberikan implikasi langsung bagi PT Magicsoft Teknologi Indonesia dalam peningkatan efektivitas pengelolaan proyek perangkat lunak. Temuan bahwa sebagian besar *Practice Area* mencapai *Capability Level* 3 (*Defined*) menunjukkan bahwa IT Development Documentation 2.0 telah berfungsi sebagai instrumen standardisasi yang mampu menciptakan konsistensi proses lintas proyek.

Konsistensi ini berdampak pada peningkatan kejelasan peran, pengendalian perubahan, keterlacakan artefak, serta penurunan risiko kesalahan yang bersumber dari variasi praktik antar tim.

Implikasi praktis berikutnya adalah bahwa organisasi tidak selalu memerlukan appraisal CMMI formal untuk memperoleh manfaat dari kerangka CMMI v3. Penelitian ini menunjukkan bahwa evaluasi internal berbasis bukti, yang dikombinasikan dengan indikator S-SDLC, dapat digunakan sebagai mekanisme refleksi proses dan perbaikan berkelanjutan. Pendekatan ini relevan bagi organisasi dengan keterbatasan sumber daya atau batasan kerahasiaan dokumen, tetapi tetap ingin meningkatkan kualitas, keamanan, dan keterkendalian proyek perangkat lunak.

Dari perspektif manajemen sumber daya manusia dan kerja tim, penelitian ini menegaskan bahwa kolaborasi, komunikasi lintas peran, dan koordinasi teknis merupakan faktor kunci keberhasilan implementasi S-SDLC. Oleh karena itu, organisasi perlu mempertahankan dan memperkuat mekanisme kolaboratif seperti pertemuan rutin, review lintas peran, serta keterlibatan aktif stakeholder teknis dalam pengambilan keputusan. Penguatan aspek kolaborasi ini memiliki implikasi langsung terhadap kecepatan penyelesaian masalah, pengurangan *rework*, dan integrasi keamanan dalam siklus pengembangan.

Implikasi praktis terakhir berkaitan dengan strategi peningkatan berkelanjutan. Dengan telah tercapainya *Capability Level* 3 pada sebagian besar *Practice Area*, organisasi memiliki dasar yang memadai untuk naik

menuju level berikutnya, khususnya pada penguatan pengukuran kinerja proses, penggunaan data sebagai dasar pengambilan keputusan, serta pengembangan metrik keamanan. Meskipun *Practice Area Measurement and Analysis* tidak dievaluasi dalam penelitian ini, temuan memberikan arah strategis bahwa pengembangan indikator kuantitatif kinerja proyek dan keamanan dapat menjadi langkah penting untuk meningkatkan kematangan proses pengembangan perangkat lunak.

5.3. Kontribusi Penelitian

Penelitian ini memberikan kontribusi penting bagi pengembangan teori dan praktik manajemen proyek perangkat lunak, khususnya dalam konteks integrasi *Capability Maturity Model Integration* (CMMI) v3 dan *Secure Software Development Life Cycle* (S-SDLC).

a) Penelitian ini memperluas diskursus mengenai CMMI dengan menunjukkan bahwa CMMI v3 dapat berfungsi tidak hanya sebagai model peningkatan proses (*process improvement model*), tetapi juga sebagai kerangka evaluatif untuk menilai efektivitas manajemen proyek perangkat lunak yang mengintegrasikan keamanan sebagai dimensi proses. Pendekatan ini melampaui penelitian terdahulu yang cenderung memisahkan analisis proses dan keamanan perangkat lunak, sehingga menawarkan perspektif teoretis baru bahwa keamanan dapat diposisikan sebagai komponen integral dalam evaluasi proses manajemen proyek.

b) Penelitian ini memperkuat teori kematangan proses (*process maturity theory*) dengan menunjukkan bahwa capaian *Capability*

Level 3 pada sebagian besar *Practice Area* tidak hanya ditentukan oleh keberadaan aktivitas teknis, tetapi terutama oleh standardisasi proses lintas proyek dan konsistensi implementasinya. Temuan ini mendukung asumsi teoretis CMMI bahwa proses yang terdefinisi (*defined process*) merupakan fondasi utama kinerja proyek yang terkendali, sekaligus memperluas pemahaman bahwa penilaian kapabilitas proses dapat dilakukan melalui pendekatan internal berbasis bukti tanpa appraisal formal, selama selaras dengan prinsip *fit-for-purpose*.

c) Penelitian ini berkontribusi pada pengembangan teori S-SDLC dengan memberikan bukti empiris bahwa integrasi keamanan dalam seluruh fase SDLC tidak hanya berdampak pada kualitas teknis sistem, tetapi juga pada efektivitas manajemen proyek. Dengan menghubungkan indikator S-SDLC dan *Practice Area* CMMI v3, penelitian ini menawarkan landasan teoretis untuk memandang keamanan perangkat lunak sebagai dimensi evaluasi manajemen proyek, bukan sekadar isu teknis atau operasional.

d) Penelitian ini membuka ruang sintesis teoretis antara CMMI, *Agile*, dan S-SDLC. Temuan menunjukkan bahwa pendekatan *Agile* dapat dipadukan dengan struktur proses CMMI v3 dan praktik keamanan S-SDLC tanpa menimbulkan kontradiksi teoretis. Hal ini memperkuat arah literatur terbaru mengenai integrasi proses formal dan pendekatan adaptif dalam proyek perangkat lunak modern.

e) Penelitian ini memberikan nilai langsung bagi PT Magicsoft Teknologi Indonesia sebagai lokasi penelitian. Hasil penelitian menunjukkan bahwa IT Development Documentation 2.0 berfungsi sebagai instrumen strategis dalam menciptakan konsistensi lintas proyek, memperjelas peran dan akuntabilitas, serta mengurangi risiko variasi proses dan cacat sistem. Temuan ini menyediakan dasar empiris bagi manajemen untuk mempertahankan dan memperkuat kebijakan standardisasi proses dalam pengembangan perangkat lunak.

f) Penelitian ini menghasilkan kerangka evaluasi internal berbasis CMMI v3 dan S-SDLC yang dapat digunakan tanpa appraisal formal. Kerangka ini memungkinkan organisasi melakukan refleksi proses berbasis bukti (*evidence-based reflection*), mengidentifikasi area perbaikan secara sistematis, serta merancang inisiatif peningkatan berkelanjutan yang realistis dan kontekstual. Pendekatan ini relevan terutama bagi organisasi yang memiliki keterbatasan sumber daya atau sensitivitas kerahasiaan dokumen.

5.4. Keterbatasan Penelitian

Penelitian ini memiliki sejumlah keterbatasan yang perlu diperhatikan agar interpretasi hasil penelitian dapat dilakukan secara proporsional.

a) Ruang lingkup penelitian dibatasi pada divisi IT PT Magicsoft Teknologi Indonesia, khususnya pada proses pengembangan perangkat lunak yang terkait dengan penerapan *Secure Software*

Development Life Cycle (S-SDLC). Pembatasan ini menyebabkan penelitian tidak mencakup fungsi organisasi lain, seperti manajemen strategis, sumber daya manusia, keuangan, maupun layanan pasca-*deployment*. Dengan demikian, penelitian ini tidak dimaksudkan untuk merepresentasikan tingkat kematangan organisasi secara keseluruhan maupun menilai efektivitas tata kelola TI pada level korporat.

b) Meskipun penelitian ini mengonfirmasi keberadaan standardisasi proses lintas proyek melalui IT Development Documentation 2.0, peneliti tidak dapat menyajikan artefak dokumen tersebut secara penuh karena alasan kerahasiaan perusahaan. Keterbatasan ini menyebabkan bukti implementasi hanya dapat disajikan melalui deskripsi naratif, konfirmasi wawancara, dan observasi terbatas, sehingga pembaca tidak dapat menilai detail teknis dokumentasi secara langsung.

c) Penelitian ini tidak melakukan appraisal CMMI formal (misalnya SCAMPI Class A), sehingga hasil penilaian terhadap *Practice Area* dan *Capability Level* bersifat evaluatif dan diagnostik, bukan penilaian yang menghasilkan tingkat kematangan organisasi secara resmi. Implikasi metodologisnya adalah bahwa hasil penelitian tidak dapat dibandingkan secara langsung dengan hasil appraisal CMMI pada organisasi lain.

Meskipun terdapat keterbatasan tersebut, penelitian ini tetap memberikan kontribusi konseptual dan empiris yang signifikan mengenai

integrasi S-SDLC dan CMMI v3 dalam pengembangan perangkat lunak. Keterbatasan ini sekaligus membuka peluang bagi penelitian selanjutnya untuk memperluas cakupan variabel, memperdalam konteks organisasi, dan memperkaya pendekatan metodologis.

DAFTAR PUSTAKA

Agile Alliance. (2001). *Manifesto for agile software development*. https://agilemanifesto.org

Asmy, Y. Y., & Hasugian, L. P. (2021). Penilaian maturity level perangkat lunak menggunakan CMMI-DEV 1.3 pada aplikasi Manans MINT. *Jurnal Manajemen Informatika dan Komputerisasi Akuntansi (JAMIKA), 21*(1), 35–42. https://ojs.unikom.ac.id/index.php/jamika/article/download/5523/2678

Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., … Thomas, D. (2001). *Manifesto for agile software development*. Agile Alliance. https://agilemanifesto.org

Behl, A., & Behl, K. (2017). *Cyberwar: The next threat to national security and what to do about it*. Oxford University Press.

Boehm, B. W. (1981). *Software engineering economics*. Prentice Hall.

Boehm, B. W. (2000). *Software cost estimation with COCOMO II*. Prentice Hall.

Boehm, B. W., & Basili, V. R. (2001). Software defect reduction top 10 list. *Computer, 34*(1), 135–137. https://doi.org/10.1109/2.962984

Boehm, B. W., & Turner, R. (2004). *Balancing agility and discipline: A guide for the perplexed*. Addison-Wesley.

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI for development: Guidelines for process integration and product improvement* (3rd ed.). Addison-Wesley.

CMMI Institute. (2018). *CMMI® development v3.0: Improving processes for better performance*. ISACA/CMMI Institute.

CMMI Institute. (2023). *CMMI model version 3.0: Improving performance and outcomes*. CMMI Institute.

Creswell, J. W., & Creswell, J. D. (2018). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). SAGE Publications.

De Win, B., Scandariato, R., Buyens, K., Grégoire, J., & Joosen, W. (2009). On the secure software development process: CLASP, SDL and

Touchpoints compared. *Information and Software Technology, 51*(7), 1152–1171. https://doi.org/10.1016/j.infsof.2008.01.010

Deming, W. E. (1986). *Out of the crisis*. MIT Press.

Drucker, P. F. (2007). *Management challenges for the 21st century*. Routledge.

Gibson, D., Goldenson, D., & Kost, K. (2006). *Performance results of CMMI-based process improvement* (CMU/SEI-2006-TR-004). Software Engineering Institute. https://doi.org/10.1184/R1/6582011.v1

Goldenson, D. R., & Gibson, D. L. (2003). *Demonstrating the impact and benefits of CMMI* (CMU/SEI-2003-SR-009). Software Engineering Institute.

Goldratt, E. M. (1997). *Critical chain*. North River Press.

Hillson, D. (2017). *The risk management handbook: A practical guide to managing the multiple dimensions of risk*. Kogan Page.

Humble, J., & Farley, D. (2010). *Continuous delivery: Reliable software releases through build, test, and deployment automation*. Addison-Wesley.

IBM Security. (2022). *Cost of a data breach report 2022*. IBM Corporation.

ISACA. (2023). *CMMI V3: Building capability and improving performance*. ISACA.

Jiang, J. J., Klein, G., & Chen, H. G. (2001). The relative influence of IS project implementation policies and project leadership on eventual outcomes. *Project Management Journal, 32*(3), 49–55. https://doi.org/10.1177/875697280103200308

Kerzner, H. (2017). *Project management: A systems approach to planning, scheduling, and controlling* (12th ed.). Wiley.

Lincoln, Y. S., & Guba, E. G. (1985). *Naturalistic inquiry*. Sage Publications.

Maxwell, J. A. (2013). *Qualitative research design: An interactive approach* (3rd ed.). SAGE Publications.

McConnell, S. (2004). *Code complete: A practical handbook of software construction* (2nd ed.). Microsoft Press.

McGraw, G. (2006). *Software security: Building security in*. Addison-Wesley.

Meyer, A. N., Murphy, G. C., & Zimmermann, T. (2019). Software developers' perceptions of productivity. *Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering*, 19–29.

Microsoft. (2018). *Security development lifecycle (SDL) process guidance*. Microsoft Corporation.

Miles, M. B., Huberman, A. M., & Saldaña, J. (2018). *Qualitative data analysis: A methods sourcebook* (4th ed.). SAGE Publications.

National Institute of Standards and Technology. (2020). *Systems security engineering: Considerations for a multidisciplinary approach* (SP 800-160 Vol. 1). U.S. Department of Commerce.

Niazi, M., Babar, M. A., & Verner, J. M. (2010). Software process improvement barriers: A cross-cultural comparison. *Information and Software Technology, 52*(11), 1204–1216. https://doi.org/10.1016/j.infsof.2010.06.005

Paulk, M. C., Curtis, B., Chrissis, M. B., & Weber, C. V. (1993). Capability maturity model for software, version 1.1. *IEEE Software*, 10(4), 18–27.

Pohl, K., & Hof, H. (2015). *Requirements engineering: Fundamentals, principles, and techniques*. Springer.

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (9th ed.). McGraw-Hill Education.

Project Management Institute. (2021). *A guide to the project management body of knowledge (PMBOK guide)* (7th ed.). PMI.

Ramasubbu, N., Krishnan, M. S., & Kompalli, P. (2005). Leveraging global resources: A process maturity framework. *IEEE Software, 22*(3), 80–86. https://doi.org/10.1109/MS.2005.69

Robbins, S. P., & Coulter, M. (2018). *Management* (14th ed.). Pearson.

Royce, W. W. (1970). Managing the development of large software systems. *Proceedings of IEEE WESCON*, 1–9.

Schwalbe, K. (2019). *Information technology project management* (9th ed.). Cengage Learning.

Shelat, A., Kumar, S., & Ganesh, R. (2025). Assessing CMMI Level 3 adoption. *International Journal of Software Engineering and Applications, 14*(2), 45–56.

Shostack, A. (2014). *Threat modeling: Designing for security*. Wiley.

Silva, F. S. C., Soares, F. S. F., França, A. C. C., & Monteiro, C. V. F. (2015). Using CMMI together with agile software development. *Information and Software Technology, 58*, 20–43. https://doi.org/10.1016/j.infsof.2014.09.002

Slack, N., & Brandon-Jones, A. (2022). *Operations management* (10th ed.). Pearson.

Sommerville, I. (2016). *Software engineering* (10th ed.). Pearson Education.

Souppaya, M., Scarfone, K., & Dodson, D. (2022). *Secure software development framework (SSDF) version 1.1* (NIST SP 800-218). NIST. https://doi.org/10.6028/NIST.SP.800-218

Standish Group. (2020). *CHAOS report 2020*. Standish Group International.

Staples, M., & Niazi, M. (2008). Organizational motivations for adopting CMM-based SPI. *Information and Software Technology, 50*(7–8), 605–620. https://doi.org/10.1016/j.infsof.2007.07.003

Terry, G. R. (1972). *Principles of management* (6th ed.). Richard D. Irwin.

Tsai, W. (2021). The impact of project teams on CMMI implementations. *Systems Research and Behavioral Science, 34*(2), 239–252. https://doi.org/10.1007/s11213-020-09531-y

Turner, J. R. (2016). *Gower handbook of project management* (5th ed.). Routledge.

Viega, J., & McGraw, G. (2011). *Building secure software: How to avoid security problems the right way*. Addison-Wesley Professional.

Wibisono, M. I. (2020). Penilaian kematangan proses pengembangan perangkat lunak. *Jurnal Teknologi Informasi dan Ilmu Komputer, 7*(5), 975–984. https://media.neliti.com/media/publications/338058

Yilmaz, M., & Ozcan, T. (2023). Mapping CMMI-DEV v2 with Scrum. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4310530