

PENGUKURAN TINGKAT KEMATANGAN IMPLEMENTASI SCRUM DALAM PENGEMBANGAN PERANGKAT LUNAK MENGGUNAKAN SCRUM MATURITY MODEL

Putri Larasati^{1*}, Arfive Gandhi², Nungki Selviandro³

^{1,2,3}Program Studi Rekayasa Perangkat Lunak, Universitas Telkom
email: putrilarasxx@student.telkomuniversity.ac.id^{1*}

Abstrak: Dalam beberapa tahun terakhir, *Scrum* semakin populer dan kini menjadi salah satu pendekatan pengembangan proyek yang banyak digunakan oleh organisasi di seluruh dunia. Startup XYZ telah menerapkan *Scrum* dalam pengembangan proyeknya sejak tahun 2021. Namun, hingga saat ini, proses implementasi *Scrum* tersebut masih mengalami banyak kendala. Berdasarkan identifikasi melalui diagram tulang ikan, permasalahan tersebut berasal dari penerapan *Scrum* yang belum maksimal dalam pengembangan perangkat lunak di Startup XYZ. Tujuan penelitian ini adalah mengukur tingkat kematangan implementasi *Scrum* dalam pengembangan perangkat lunak di Startup XYZ menggunakan *Scrum Maturity Model* (SMM). SMM berfokus pada evaluasi dan memberikan arahan untuk pengembangan perangkat lunak yang menerapkan *Scrum*. Kemudian digunakan *Key Process Area* (KPA) Rating untuk mendefinisikan pencapaian di setiap area proses, yang diadaptasi dari *Agile Maturity Model* (AMM). Hasil pengukuran menunjukkan bahwa tingkat kematangan *Scrum* di Startup XYZ masih berada pada level 1 (*Initial*). Hal ini dibuktikan dengan nilai KPA Rating pada level 2 adalah 85,12% (belum mencapai interpretasi "*Fully Achieved*"). Berdasarkan hasil pengukuran tersebut, diberikan rekomendasi perbaikan yang disusun berdasarkan jawaban kuesioner SMM dan mengacu pada *Scrum Guide 2020*, *SBOK Guide v3*, dan *Essential Scrum*, dengan fokus pada peningkatan level 2 dan level 3 sesuai ekspektasi dari pihak Startup XYZ.

Kata Kunci: *Agile Maturity Model* (AMM), pengukuran tingkat kematangan, *Scrum*, *Scrum Maturity Model* (SMM).

Abstract: In recent years, *Scrum* has grown in popularity and is now one of the project development approaches widely used by organizations around the world. Startup XYZ has been implementing *Scrum* in its project development since 2021. However, until now, the *Scrum* implementation process is still experiencing many obstacles. Based on identification through a fishbone diagram, these problems stem from the application of *Scrum* which has not been maximized in software development at Startup XYZ. The purpose of this research is to measure the maturity level of *Scrum* implementation in software development at Startup XYZ using the *Scrum Maturity Model* (SMM). SMM focuses on evaluating and providing direction for software development that implements *Scrum*. Then the *Key Process Area* (KPA) Rating is used to define achievements in each process area, which is adapted from the *Agile Maturity Model* (AMM). The measurement results show that the *Scrum* maturity level at Startup XYZ is still at level 1 (*Initial*). This is evidenced by the KPA Rating value at level 2 is 85.12% (has not yet reached the interpretation of "*Fully Achieved*"). Based on the measurement results, improvement recommendations are provided based on the answers to the SMM questionnaire and referring to the *Scrum Guide 2020*, *SBOK Guide v3*, and *Essential Scrum*, with a focus on increasing level 2 and level 3 according to the expectations of Startup XYZ.

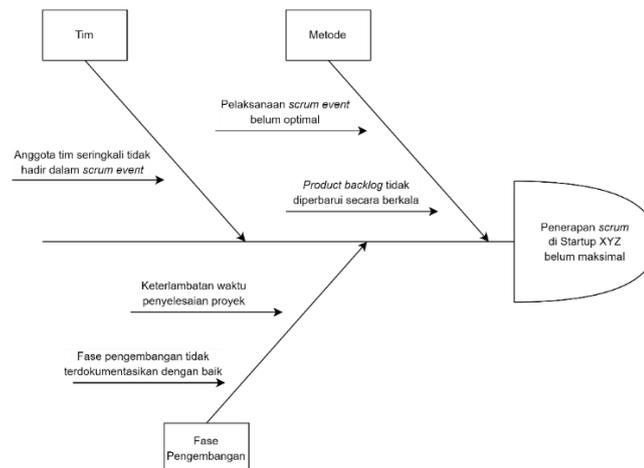
Keywords: *Agile Maturity Model* (AMM), maturity level measurement, *Scrum*, *Scrum Maturity Model* (SMM).

PENDAHULUAN

Dalam era digital yang semakin kompetitif, Teknologi Informasi (TI) menjadi komponen penting yang dimanfaatkan sebagai strategi bisnis dan inovasi perusahaan. TI dikatakan sebagai *core enabler* bagi perusahaan untuk menjalankan bisnis [1]. Untuk memastikan bahwa penerapan TI sesuai dengan strategi bisnis perusahaan, diperlukan tata kelola yang baik [2]. Tata kelola TI yang baik sangat penting untuk menjamin keberlangsungan proses bisnis dan mendapatkan manfaat optimal dari penerapan TI [3]. Tata kelola TI meliputi pengelolaan infrastruktur TI yang handal hingga pengembangan perangkat lunak yang berkualitas [3].

Dalam beberapa tahun terakhir, *Scrum* semakin populer dan kini menjadi salah satu pendekatan pengembangan proyek yang banyak digunakan oleh organisasi di seluruh dunia [4]. *Scrum* merupakan kerangka kerja sederhana yang membantu individu, tim, dan organisasi dalam menciptakan nilai melalui solusi adaptif untuk menyelesaikan permasalahan yang kompleks [5]. *Scrum* mudah untuk dipahami, tetapi sulit untuk diimplementasikan dengan benar [6].

Keberadaan startup dalam menciptakan produk dan layanan inovatif menunjukkan bahwa teknologi telah dimanfaatkan dengan baik. Startup adalah perusahaan rintisan yang bergerak dalam bidang teknologi, yang baru dibangun dan masih dalam tahap berkembang. Startup XYZ berdiri sejak Agustus 2021, memiliki visi menjadi *platform* kesehatan mental yang paling berdampak di dunia dengan menyediakan layanan kesehatan mental melalui *experience-based approaches*. Untuk mendukung tujuan bisnis perusahaan, tim pengembang Startup XYZ mengerjakan proyek pengembangan *website*. Startup XYZ telah menerapkan kerangka kerja *Scrum* dalam pengembangan proyeknya sejak tahun 2021. Namun hingga saat ini, penerapan *Scrum* dalam pengembangan perangkat lunak masih mengalami banyak kendala.



Gambar 1. Diagram Tulang Ikan analisis akar permasalahan Implementasi Scrum di Startup XYZ

Diagram tulang ikan pada Gambar 1 merupakan visualisasi hasil penggalian masalah yang terjadi dalam proses implementasi *Scrum* di tim pengembang Startup XYZ, yang diperoleh melalui wawancara dengan seorang *Product Owner* dan seorang *Business Analyst*. Permasalahan tersebut terbagi ke dalam tiga domain, yaitu tim, metode, dan fase pengembangan. Berdasarkan hasil analisis menggunakan diagram tulang ikan, telah diidentifikasi bahwa akar permasalahan tersebut berasal dari penerapan kerangka kerja *Scrum* yang belum maksimal di tim pengembang Startup XYZ.

Dalam pelaksanaan *Scrum Event*, *Product Owner* mengatakan bahwa pelaksanaan *Sprint Review* dan *Sprint Retrospective* belum dilakukan dengan baik, karena ada beberapa *Sprint* yang terlewat tidak dilakukan. *Product Owner* juga mengatakan bahwa *Daily Scrum* hanya sebagai formalitas dan belum dilakukan secara maksimal. *Product Owner* dan *Scrum Master* juga seringkali tidak hadir pada pelaksanaan *Scrum Event*. Ketidakhadiran tersebut berpotensi menghambat komunikasi dan kolaborasi dengan anggota tim, yang pada akhirnya dapat mengganggu proses pencapaian tujuan *Sprint*.

Startup XYZ tidak mendokumentasikan proses pengembangan perangkat lunak dengan baik. Selain itu, *Product Backlog* tidak diperbarui secara berkala sesuai dengan kemajuan proyek. Startup XYZ mengalami keterlambatan waktu penyelesaian proyek, sehingga tidak sesuai dengan penjadwalan yang telah ditetapkan. Permasalahan tersebut membuktikan bahwa praktik-praktik dalam kerangka kerja *Scrum* belum sepenuhnya dilaksanakan secara konsisten dan berkelanjutan oleh Startup XYZ.

Penelitian [3] menjelaskan bahwa dengan mengevaluasi tingkat kematangan *Scrum*, dapat dijadikan bahan evaluasi untuk mengatasi permasalahan yang dihadapi saat ini dan dapat meningkatkan kualitas perangkat lunak pada proyek pengembangan mendatang. Temuan ini sejalan dengan penelitian [2], yaitu melakukan evaluasi pengembangan perangkat lunak dan dapat memberikan arahan untuk meningkatkan kinerja di masa yang akan datang. Berdasarkan penelitian [7], hasil dari evaluasi tersebut menggambarkan kesenjangan antara praktik yang ada dengan prosedur yang seharusnya dilakukan. Untuk mengatasi permasalahan yang teridentifikasi melalui analisis diagram tulang ikan, seharusnya temuan keberhasilan pengukuran kematangan pada penelitian [2], [3], [7] relevan untuk diimplementasikan pada penelitian ini.

Penulis menggunakan *Scrum Maturity Model* (SMM) untuk melakukan pengukuran tingkat kematangan implementasi *Scrum* di Startup XYZ. SMM berfokus pada evaluasi dan memberikan arahan untuk proses pengembangan perangkat lunak yang mengadopsi *Scrum* [7]. Digunakan juga *Key Process Area* (KPA) Rating untuk mendefinisikan pencapaian di setiap area proses, yang diadaptasi dari *Agile Maturity Model* (AMM). Hal ini akan membantu organisasi untuk mengevaluasi tingkat kematangan implementasi *Scrum* dalam proyek pengembangan perangkat lunak dan akan menjadi dasar dimana organisasi dapat memulai fokus perbaikannya [8].

Sebagai upaya mengatasi permasalahan dan mendukung peningkatan penerapan *Scrum* di Startup XYZ, berdasarkan hasil pengukuran tingkat kematangan, penulis mengusulkan rekomendasi perbaikan. Rekomendasi perbaikan disusun berdasarkan jawaban kuesioner SMM yang bernilai "Sebagian", "Tidak", atau "Tidak Dapat Diterapkan (N/A)", untuk mengidentifikasi setiap target perbaikan. Dari setiap jawaban kuesioner tersebut, kemudian dipetakan dengan praktik-praktik yang seharusnya dilakukan sesuai *Scrum Guide 2020*, *SBOK Guide v3*, dan *Essential Scrum*, dengan berfokus pada level SMM yang ditargetkan.

TINJAUAN PUSTAKA

Scrum

Scrum pertama kali diperkenalkan oleh Jeff Sutherland pada tahun 1993, sebagai kerangka kerja pengembangan perangkat lunak yang didasarkan pada prinsip-prinsip metodologi *agile*. *Scrum* merupakan

kerangka kerja sederhana yang membantu individu, tim, dan organisasi dalam menciptakan nilai melalui solusi adaptif untuk menyelesaikan permasalahan yang kompleks [5]. Nilai-nilai *Scrum* seperti komitmen, fokus, keterbukaan, rasa hormat, dan keberanian harus dijunjung tinggi oleh masing-masing peran dalam tim *Scrum* untuk keberhasilan penerapan *Scrum*.

Menurut Schwaber & Sutherland [5], terdapat tiga peran utama dalam *Scrum*, yaitu *Scrum Master*, *Product Owner*, dan *Development Team*. Setiap anggota tim *Scrum* memiliki tanggung jawabnya masing-masing. Terdapat tiga artefak *Scrum*, yaitu *Product Backlog*, *Sprint Backlog*, dan *increment* [5]. Artefak-arterfak tersebut membantu tim *Scrum* untuk mengelola dan melacak pekerjaan selama siklus pengembangan perangkat lunak. *Scrum* melibatkan serangkaian acara yang berulang di setiap *Sprint*, yaitu *Sprint Planning*, *Sprint Review*, *Sprint Retrospective*, dan *Daily Scrum*.

Pengukuran Tingkat Kematangan

Pengukuran tingkat kematangan pengembangan perangkat lunak bertujuan untuk mengevaluasi dan meningkatkan proses pengembangan dengan menerapkan praktik-praktik terbaik kerangka kerja perangkat lunak yang telah diadopsi dari suatu organisasi. Tingkat kematangan mencerminkan kemampuan manajemen dan mendorong siklus pembelajaran berkelanjutan untuk mencapai hasil pengembangan perangkat lunak yang lebih baik. Dalam upaya mengukur tingkat kematangan pengembangan perangkat lunak dan meningkatkan proses tersebut, digunakan metode yang dikenal sebagai model kematangan. Model kematangan memungkinkan organisasi untuk mengevaluasi dan mengklasifikasikan kapabilitasnya, serta mendorong organisasi untuk meningkatkan kapabilitas secara bertahap [7].

Agile Maturity Model

Patel & Ramachandran [9] mengembangkan *Agile Maturity Model* (AMM) yang terinspirasi dari *Capability Maturity Model Integration* (CMMI) [7]. AMM bertujuan untuk meningkatkan praktik pengembangan perangkat lunak yang menerapkan *agile*. Dalam penelitian ini, untuk tujuan analisis data kuesioner dan mendefinisikan pencapaian di setiap area proses, digunakan *Key Process Area* (KPA) Rating yang diadopsi dari AMM. Menurut Yin dkk. [13], mengimplementasikan praktik-praktik CMMI dalam metodologi *agile* tidak menyelesaikan masalah, karena cakupannya yang begitu luas. Selain itu, Startup XYZ tidak memiliki sumber daya yang cukup untuk mengimplementasikan CMMI secara maksimal. Oleh karena itu, dalam penelitian ini, penulis menerapkan SMM dan KPA Rating dari AMM yang paling berfokus pada peningkatan implementasi *Scrum* pada pengembangan perangkat lunak di Startup XYZ.

Scrum Maturity Model

Dalam penelitiannya pada tahun 2011, Yin dkk. [10] memperkenalkan *Scrum Maturity Model* (SMM). SMM adalah kerangka kerja hasil dari pengembangan AMM [7] yang berfokus pada evaluasi dan memberikan arahan bagi organisasi yang mengadopsi *Scrum*. Model ini tidak hanya berfokus pada praktik pengembangan perangkat lunak secara umum, tetapi juga pada penerapan prinsip-prinsip dan nilai-nilai *Scrum* secara khusus. Selain itu, model ini juga membantu organisasi yang belum terbiasa menerapkan *Scrum*, tetapi ingin mengadopsinya [10]. Dalam penelitian ini, penulis menggunakan SMM untuk menyusun pertanyaan kuesioner berdasarkan indikator-indikator yang telah ditetapkan pada SMM. SMM memiliki lima tingkat kematangan [10], yaitu level 1 (*initial*), level 2 (*managed*), level 3 (*defined*), level 4 (*quantitatively managed*), dan level 5 (*optimizing*).

Studi Terkait

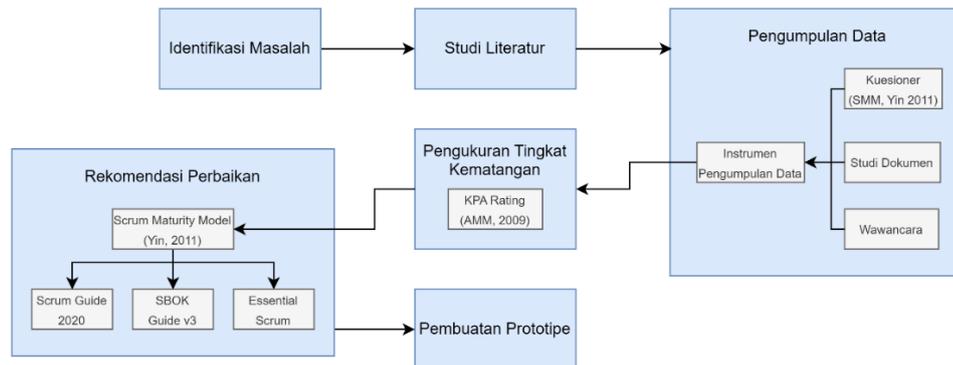
Penelitian mengenai pengukuran tingkat kematangan *Scrum* pernah dilakukan sebelumnya oleh beberapa penulis, diantaranya adalah penelitian oleh Ridha & Hegarini pada tahun 2020 [2], yaitu melakukan analisis tingkat kematangan *Scrum* pada Suku Enterprise PT. XYZ. Rekomendasi perbaikan didapatkan dari target spesifik dari setiap sasaran umum yang KPA Ratingnya masih belum sepenuhnya tercapai. Ridha & Hegarini menyebutkan bahwa evaluasi pengembangan perangkat lunak dapat memberikan arahan untuk meningkatkan kinerja di masa yang akan datang.

Pada tahun 2023, Patara & Rahardjo [11] melakukan penelitian untuk membandingkan kematangan *Scrum* antara dua grup di Bank XYZ, yaitu Grup Digital (Grp-DGT) dan Grup Rekayasa Ulang Proses Bisnis (Grp-BPR). Rekomendasi perbaikan diusulkan untuk meningkatkan praktik-praktik *Scrum* pada proyek atau *Sprint* berikutnya di kedua grup tersebut.

Pada tahun 2022, Panjaitan & Legowo [7] melakukan penelitian untuk mengevaluasi proses penerapan *Scrum* pada pengembangan proyek perangkat lunak di PT. XYZ. Rekomendasi perbaikan diberikan berdasarkan jawaban pada setiap pertanyaan yang didefinisikan pada level 2 dan level 3 dan dikelompokkan berdasarkan *Scrum Role*, *Scrum Event*, dan *Scrum Artifact*. Panjaitan dan Legowo menyebutkan hasil dari evaluasi tersebut menggambarkan kesenjangan antara praktik yang ada dengan prosedur yang seharusnya dilakukan.

METODE

Pada penelitian ini, penulis menggunakan metode campuran sekuensial eksplanatoris, yaitu menggabungkan metode kuantitatif dan kualitatif secara berurutan. Penulis mengumpulkan data kuantitatif terlebih dahulu melalui kuesioner untuk mengukur tingkat kematangan implementasi *Scrum* berdasarkan indikator-indikator yang telah ditetapkan pada *Scrum Maturity Model* (SMM). Setelah itu, diikuti dengan pengumpulan data kualitatif untuk memvalidasi hasil data kuantitatif, serta menggali pemahaman lebih dalam tentang bagaimana implementasi *Scrum* diterapkan di Startup XYZ melalui studi dokumentasi proyek pengembangan perangkat lunak dan wawancara. Adapun alur penelitian yang dilakukan terlihat pada Gambar 2.



Gambar 2. Alur Penelitian

Identifikasi Masalah

Identifikasi masalah dilakukan untuk mendefinisikan latar belakang masalah yang mendasari penelitian ini. Proses penggalian masalah dalam penelitian ini dilakukan melalui wawancara semi-terstruktur dengan seorang *Product Owner* dan seorang *Business Analyst* dari Startup XYZ. Selanjutnya hasil wawancara tersebut dianalisis menggunakan diagram tulang ikan untuk mengidentifikasi akar permasalahan yang terjadi di Startup XYZ.

Studi Literatur

Studi literatur dilakukan untuk mendapatkan landasan teori dan usulan solusi yang menjawab rumusan masalah yang telah didefinisikan sebelumnya. Landasan teori diperlukan untuk memberikan penjelasan dan pemahaman yang lebih mendalam tentang masalah yang diteliti.

Pengumpulan Data

Dalam pengumpulan data, digunakan teknik *Purposive Sampling* untuk menentukan subjek penelitian. *Purposive Sampling* adalah metode pengumpulan data di mana penulis secara sengaja memilih partisipan berdasarkan kriteria tertentu yang dianggap relevan dengan tujuan penelitian [12]. Subjek penelitian yang dipilih adalah *Scrum Master*, *Product Owner*, dan *Development Team*. Ketiga peran tersebut dipilih dengan tujuan mencakup individu yang terlibat langsung dalam proyek pengembangan perangkat lunak di Startup XYZ. Dengan demikian, diharapkan dapat memberikan informasi yang kaya dan representatif terkait permasalahan yang sedang diteliti. Penulis menggunakan tiga teknik untuk mengumpulkan data dari responden, sebagai berikut:

1. Kuesioner

Penilaian praktik-praktik *Scrum* dilakukan dengan menggunakan kuesioner untuk mengetahui bagaimana praktik-praktik tersebut diimplementasikan dalam proyek pengembangan perangkat lunak di Startup XYZ. Pertanyaan kuesioner dalam penelitian ini disusun berdasarkan indikator-indikator SMM oleh Yin dkk. [10]. Kuesioner yang telah disusun, selanjutnya disebar dengan menggunakan Google Form kepada responden. Terdapat total 80 praktik *Scrum* yang akan dinilai. Semua praktik diubah menjadi pertanyaan yang dapat dijawab dengan beberapa pilihan, antara lain: Ya (telah diimplementasikan sepenuhnya); Tidak (tidak diterapkan atau tidak berlaku sama sekali); Sebagian (diimplementasikan secara parsial); N/A (tidak dapat diterapkan).

2. Studi Dokumen

Setelah pengumpulan data kuantitatif, dilanjutkan dengan pengumpulan data kualitatif, yaitu melalui studi dokumen dan wawancara. Studi dokumen dilakukan dengan memeriksa dokumen pendukung yang relevan untuk memvalidasi jawaban dari setiap pertanyaan kuesioner. Dokumen pendukung yang menjadi sumber data utama yang relevan dengan penelitian ini adalah dokumen artefak *Scrum*.

3. Wawancara

Jika tidak ada dokumen pendukung atau dokumen pendukungnya kurang lengkap, maka dilakukan wawancara untuk menentukan nilai yang lebih sesuai dari pertanyaan kuesioner tersebut. Wawancara juga dilakukan untuk memastikan kesesuaian jawaban kuesioner dengan bukti dokumentasi proyek pengembangan perangkat lunak [2].

Pengukuran Tingkat Kematangan

Data yang diperoleh dari tahap sebelumnya, selanjutnya akan dianalisis, yaitu mengukur tingkat kematangan penerapan *Scrum* dalam proses pengembangan perangkat lunak di Startup XYZ. Metode SMM tidak menjelaskan bagaimana cara menganalisis data kuesioner secara detail. Oleh karena itu, untuk tujuan analisis data kuesioner [2], [7], [8] dan mendefinisikan pencapaian di setiap area proses, digunakan metode statistik KPA Rating yang diadaptasi dari *Agile Maturity Model (AMM)*.

Perhitungan data kuesioner menggunakan rumus KPA Rating, dihitung dengan persamaan (1) sebagai berikut [9].

$$\frac{\sum(Yn) + \frac{1}{2}\sum(Pn)}{\sum(Tn) - \sum(NAn)} * 100\% \quad (1)$$

Keterangan:

Yn = Jumlah jawaban "Ya";

Pn = Jumlah jawaban "Sebagian";

Tn = Total jumlah pertanyaan;

NAn = Jumlah jawaban "N/A."

Setelah mendapatkan nilai KPA Rating, nilai tersebut dikategorikan seperti yang terlihat pada Tabel 1.

Tabel 1. Interpretasi Nilai KPA Rating

No.	KPA Rating Percentage	Interpretation	Description
1.	86% - 100%	<i>Fully Achieved</i> (Tercapai Sepenuhnya)	Terdapat bukti pendekatan yang lengkap dan sistematis, serta pencapaian penuh atas praktik-praktik utama yang telah ditetapkan dalam KPA yang dinilai [9].
2.	51% - 85%	<i>Largely Achieved</i> (Sebagian Besar Tercapai)	Terdapat bukti pendekatan sistematis yang baik dan pencapaian yang signifikan terhadap praktik-praktik utama yang telah ditetapkan dalam KPA yang dinilai [9].
3.	16% - 50%	<i>Partially Achieved</i> (Tercapai Sebagian)	Terdapat bukti pendekatan sistematis yang baik dan pencapaian praktik-praktik utama yang ditetapkan dalam KPA yang dinilai [9].
4.	0% - 15%	<i>Not Achieved</i> (Tidak tercapai)	Hanya ada sedikit atau tidak ada bukti pencapaian praktik-praktik utama yang ditetapkan dalam KPA yang dinilai [9].

Tingkat kematangan *Scrum* dianggap telah mencapai level tersebut apabila nilai KPA Rating tercapai sepenuhnya (Nilai KPA Rating \geq 86%) [2], [3], [13].

Rekomendasi Perbaikan

Berdasarkan hasil pengukuran tingkat kematangan pada tahap sebelumnya, langkah selanjutnya adalah menyusun rekomendasi perbaikan. Rekomendasi bertujuan untuk meningkatkan proses implementasi *Scrum* dalam pengembangan perangkat lunak di Startup XYZ, yang dapat diaplikasikan pada proyek pengembangan berikutnya. Rekomendasi perbaikan dalam penelitian ini, disusun berdasarkan jawaban kuesioner SMM yang bernilai "Sebagian", "Tidak", atau "Tidak Dapat Diterapkan (N/A)" [2], [3], [13], untuk mengidentifikasi setiap target perbaikan. Setiap jawaban kuesioner akan dipetakan pada praktik-praktik *Scrum* yang belum diterapkan dan seharusnya dilakukan sesuai dengan *Scrum Guide 2020*, *SBOK Guide v3*, dan *Essential Scrum*, dengan berfokus pada level SMM yang ditargetkan.

HASIL DAN PEMBAHASAN

Hasil Pengumpulan Data

Evaluasi tingkat kematangan *Scrum* dalam pengembangan perangkat lunak di Startup XYZ dimulai dengan mengumpulkan data yang diperoleh dari hasil pengisian kuesioner oleh 1 *Product Owner*, 2 *Scrum Master*, dan 1 anggota tim pengembang. Adapun dokumen pendukung yang didapatkan, yaitu *screen capture* Trello yang berisi *Product Backlog*, *Sprint Backlog*, *Timebox*, *Story Point*, *Acceptance Criteria*, *User Story*, *Task Owner*, jumlah *Sprint*, dan status dari setiap *Backlog*. Selain itu, terdapat foto demonstrasi pada saat pelaksanaan *Sprint Review* dan struktur tim proyek Startup XYZ. Dikarenakan dokumen pendukung yang didapatkan kurang lengkap, maka dilakukan wawancara dengan 1 *Product Owner* dan 1 *Scrum Master* untuk menentukan nilai yang lebih sesuai dari pertanyaan kuesioner tersebut dan memastikan kesesuaian jawaban kuesioner dengan bukti-bukti yang ada.

Hasil Pengukuran Tingkat Kematangan *Scrum*

Level 1 (*Initial*)

Pengembangan produk hingga saat ini telah dilakukan dalam 11 *Sprint* dengan total penyelesaian 56 *task*, sebagaimana ditunjukkan pada Tabel 2. *Timebox* untuk setiap *Sprint* memiliki waktu selama empat hari. Hasil ini menunjukkan pencapaian level 1 dalam model kematangan *Scrum*, karena tim telah menerapkan *Sprint* sebagai bagian dari praktik *Scrum*.

Tabel 2. Rekapitulasi *Sprint* yang telah dilaksanakan oleh Startup XYZ

<i>Sprint</i>	<i>Timebox</i>	<i>Task</i>
<i>Sprint 1</i>	25 Desember – 29 Desember 2023	1
<i>Sprint 2</i>	15 Januari – 19 Januari 2024	1
<i>Sprint 3</i>	22 Januari – 26 Januari 2024	3
<i>Sprint 4</i>	29 Januari – 03 Februari 2024	1
<i>Sprint 5</i>	05 Februari – 09 Februari 2024	11
<i>Sprint 6</i>	12 Februari – 16 Februari 2024	10
<i>Sprint 7</i>	19 Februari – 23 Februari 2024	5
<i>Sprint 8</i>	26 Februari – 1 Maret 2024	5
<i>Sprint 9</i>	4 Maret – 8 Maret 2024	10
<i>Sprint 10</i>	11 Maret – 15 Maret 2024	6
<i>Sprint 11</i>	18 Maret – 22 Maret 2024	3
Total		56

Level 2 (Managed)

Tingkat kematangan *Scrum* level 2 terdiri dari dua tujuan, yaitu *Basic Scrum Management* dan *Software Requirements Engineering*. Tabel 3 menunjukkan rekapitulasi KPA Rating tingkat kematangan *Scrum* level 2.

Tabel 3. Rekapitulasi KPA Rating SMM Level 2

<i>Level</i>	<i>Goals</i>	<i>Objectives</i>	<i>KPA Rating</i>	<i>Average Goals</i>
2 (Managed)	<i>Basic Scrum Management</i>	Adanya Peran <i>Scrum</i>	100%	75%
		Adanya Artefak <i>Scrum</i>	55,56%	
		Adanya <i>Scrum Meeting</i> dan dipartisipasi	44,44%	
	<i>Software Requirements Engineering</i>	Sprint dilaksanakan dengan benar	100%	95,24%
		Definisi <i>Product Owner</i> Jelas	85,71%	
		Manajemen <i>Product Backlog</i>	100%	
		<i>Sprint Planning Meeting</i> yang sukses	100%	
Average			85,12%	Largely Achieved
Interpretation				Largely Achieved

Tabel 3 menunjukkan nilai rata-rata KPA Rating untuk level 2 adalah 85,12% (*Largely Achieved*). Hasil ini membuktikan bahwa praktik-praktik *Scrum* pada level 2 sebagian besar telah tercapai. Dalam pengembangan perangkat lunak, Startup XYZ memiliki tim *Scrum* yang terdiri dari 1 *Product Owner*, 2 *Scrum Master*, dan 4 anggota tim pengembang. Sementara itu, untuk *Scrum Artifacts*, tim *Scrum* menggunakan Trello untuk mengelola *Product Backlog* dan *Sprint Backlog*. Beberapa praktik *Scrum* yang tidak dilaksanakan pada level 2, yaitu tim *Scrum* tidak menerapkan *Sprint Burndown Chart* dan *Release Burndown Chart*.

Tim *Scrum* tidak pernah melaksanakan *Daily Scrum*, sehingga kurang memiliki kesempatan untuk berdiskusi dan menyampaikan perkembangan, hambatan, dan rencana harian selama pengembangan berlangsung. Di sisi lain, pelaksanaan *Scrum Event* lainnya seperti *Release Planning* telah sepenuhnya dilaksanakan. *Sprint Planning*, *Sprint Review*, *Sprint Retrospective* masih dilakukan secara sebagian, karena ada beberapa *Sprint* yang terlewat tidak dilakukan. Sementara itu, *Product Owner* dan *Scrum Master* tidak selalu hadir pada setiap pertemuan yang diselenggarakan. Ketidakhadiran *Product Owner* dan *Scrum Master*, serta pelaksanaan *Scrum Event* yang masih sebagian dilakukan dapat meningkatkan risiko miskomunikasi, duplikasi pekerjaan, kesenjangan informasi antar anggota tim *Scrum*, dan pada akhirnya dapat mengganggu proses pencapaian tujuan *Sprint*.

Level 3 (Defined)

Tingkat kematangan *Scrum* level 3 terdiri dari dua tujuan, yaitu *Customer Relationship Management* dan *Iteration Management*. Tabel 4 menunjukkan rekapitulasi KPA Rating tingkat kematangan *Scrum* level 3.

Tabel 4. Rekapitulasi KPA Rating SMM Level 3

<i>Level</i>	<i>Goals</i>	<i>Objectives</i>	<i>KPA Rating</i>	<i>Average Goals</i>
3 (Defined)	<i>Customer Relationship Management</i>	Adanya <i>Definition of Done</i>	0%	58,33%
		Ketersediaan <i>Product Owner</i>	75%	
		<i>Sprint Review Meeting</i> yang sukses	100%	
	<i>Iteration Management</i>	Manajemen <i>Sprint Backlog</i>	85,71%	41,74%
		Perencanaan Iterasi	62,50%	
		<i>Velocity</i> yang terukur	18,75%	
Efektivitas <i>Burndown Chart</i>			0%	

Level	Goals	Objectives	KPA Rating	Average Goals
		Average		50,04%
		Interpretation		Partially Achieved

Tabel 4 menunjukkan nilai rata-rata KPA Rating untuk level 3 adalah 50,04% (*Partially Achieved*). Hasil ini membuktikan bahwa praktik-praktik *Scrum* pada level 3 telah tercapai sebagian. Dalam pengembangan perangkat lunak yang dilakukan, tidak ada *Definition of Done* yang dibuat oleh tim *Scrum*, sehingga tim *Scrum* tidak memiliki standar kualitas yang disepakati untuk menentukan kapan suatu *Backlog* dianggap selesai dan menjadi *increment*. Sementara itu, *Product Owner* tidak selalu hadir untuk tim *Scrum*. *Product Owner* hanya hadir pada acara-acara penting saja, seperti pada saat *Sprint Planning*, *Sprint Review*, dan *Sprint Retrospective*.

Pada pengelolaan *Sprint Backlog* dan estimasi untuk setiap *task* yang sedang berjalan tidak diperbarui setiap hari. Selain itu, *Scrum Master* tidak melakukan analisis secara berkala terhadap kemajuan *Sprint*. *Sprint* yang tidak tercapai sesuai jadwal tidak dibatalkan segera, sehingga pekerjaan yang seharusnya diselesaikan dalam *Sprint* tersebut menumpuk dan tim menjadi terbiasa melanjutkan *Sprint* yang tidak tercapai tepat waktu. Akibatnya, masalah yang sama berpotensi terulang kembali pada *Sprint-Sprint* berikutnya. Dalam *Release Planning*, *Product Owner* tidak menggunakan metrik *velocity*, sehingga sulit untuk memperkirakan berapa banyak *Sprint* yang dibutuhkan untuk menyelesaikan sebuah rilis.

Rekapitulasi KPA Rating untuk Level 2 dan Level 3 SMM

Tabel 5 menunjukkan hasil rekapitulasi penilaian tingkat kematangan *Scrum* untuk setiap tujuan pada level 2 dan level 3 SMM yang telah diperoleh sebelumnya.

Tabel 5. Rekapitulasi KPA Rating Level 2 dan Level 3 SMM

Levels	Goals	KPA Rating	Interpretation
2 (Managed)	<i>Basic Scrum Management</i>	85,12%	<i>Largely Achieved</i>
	<i>Software Requirements Engineering</i>		
3 (Defined)	<i>Customer Relationship Management</i>	50,04%	<i>Partially Achieved</i>
	<i>Iteration Management</i>		

Dapat disimpulkan bahwa tingkat kematangan implementasi *Scrum* dalam pengembangan perangkat lunak di Startup XYZ masih berada pada level 1 (*Initial*) dan belum mencapai tingkat kematangan level 2. Hal ini dibuktikan dengan nilai KPA Rating pada level 2 masih dibawah 86% atau dengan kata lain belum mencapai interpretasi *Fully Achieved*. Rekomendasi perbaikan difokuskan pada level 2 dan level 3 sesuai ekspektasi dari pihak Startup XYZ yang diperoleh melalui wawancara dengan *Product Owner*.

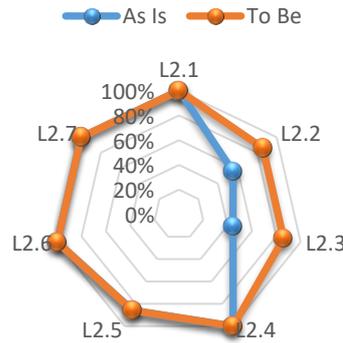
Rekomendasi Perbaikan

Untuk melihat sejauh mana Startup XYZ telah mencapai tujuan yang ditetapkan pada level 2 dan 3 SMM. Penulis melakukan perbandingan persentase antara tingkat kematangan nilai KPA Rating pada setiap objektif saat ini (*as is*) dengan persentase yang ingin dicapai oleh Startup XYZ (*to be*) pada level 2 yang terlihat pada Tabel 6.

Tabel 6. Kematangan Saat Ini vs Ekspektasi pada Level 2 SMM

Code	Objectives	As Is (%)	To Be (%)
L2.1	Adanya Peran <i>Scrum</i>	100%	100%
L2.2	Adanya Artefak <i>Scrum</i>	55,56%	86%
L2.3	Adanya <i>Scrum Meeting</i> dan dipartisipasi	44,44%	86%
L2.4	<i>Sprint</i> dilaksanakan dengan benar	100%	100%
L2.5	Definisi <i>Product Owner</i> Jelas	85,71%	86%
L2.6	Manajemen <i>Product Backlog</i>	100%	100%
L2.7	<i>Sprint Planning Meeting</i> yang sukses	100%	100%

Pada level 2 SMM, terdapat empat objektif yang telah mencapai nilai KPA Rating 100% (nilai maksimum "*Fully Achieved*"). Sementara itu, tiga objektif lainnya perlu meningkatkan nilai KPA Rating hingga mencapai persentase 86% (nilai minimum "*Fully Achieved*"). Perbandingan antara nilai KPA Rating saat ini dengan target KPA Rating yang ingin dicapai pada level 2 SMM diilustrasikan dalam diagram radar pada Gambar 3.



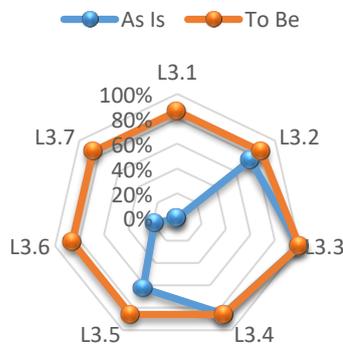
Gambar 3. Representasi Perbandingan SMM Level 2

Tabel 7 menunjukkan perbandingan persentase antara tingkat kematangan nilai KPA Rating pada setiap objektif saat ini (*as is*) dengan persentase yang ingin dicapai oleh Startup XYZ (*to be*) pada level 3 SMM.

Tabel 7. Kematangan Saat Ini vs Ekspektasi pada Level 3 SMM

Code	Objectives	As Is (%)	To Be (%)
L3.1	Adanya <i>Definition of Done</i>	0%	86%
L3.2	Ketersediaan <i>Product Owner</i>	75%	86%
L3.3	<i>Sprint Review Meeting</i> yang sukses	100%	100%
L3.4	Manajemen <i>Sprint Backlog</i>	85,71%	86%
L3.5	Perencanaan Iterasi	62,50%	86%
L3.6	<i>Velocity</i> yang terukur	18,75%	86%
L3.7	Efektivitas <i>Burndown Chart</i>	0%	86%

Pada level 3 SMM, terdapat satu objektif yang telah mencapai nilai KPA Rating 100% (nilai maksimum "*Fully Achieved*"). Sementara itu, enam objektif lainnya perlu meningkatkan nilai KPA Rating hingga mencapai persentase 86% (nilai minimum "*Fully Achieved*"). Perbandingan antara nilai KPA Rating saat ini dengan target KPA Rating yang ingin dicapai pada level 3 SMM diilustrasikan dalam diagram radar pada Gambar 4.



Gambar 4. Representasi Perbandingan SMM Level 3

Rekomendasi Perbaikan untuk *Scrum Role*

Tim *Scrum* Startup XYZ terdiri dari 1 *Product Owner*, 2 *Scrum Master*, dan 4 anggota tim pengembang. *Scrum Master* utama memiliki pekerjaan dan tanggung jawab diluar Startup XYZ, sehingga ditunjuk salah satu anggota tim *Scrum* untuk dijadikan *backup* ketika *Scrum Master* utama sedang berhalangan. Rekomendasi perbaikan untuk *Scrum Role* ditunjukkan pada Tabel 8.

Tabel 8. Rekomendasi Perbaikan untuk *Scrum Role*

No.	Scrum Roles	Challenges	Improvements Recomendations
1.	<i>Scrum Master</i>	<i>Scrum Master</i> tidak melakukan analisis secara berkala terhadap kemajuan <i>Sprint</i> .	<ol style="list-style-type: none"> <i>Scrum Master</i> melakukan analisis secara berkala terhadap kemajuan <i>Sprint</i> dan mengevaluasi implementasi <i>Scrum</i> dari kinerja <i>Development Team</i>. <i>Scrum Master</i> setiap hari memperbarui <i>Sprint</i> dan <i>Burndown Chart</i>, membantu anggota tim untuk meningkatkan penerapan <i>Scrum</i>, atau berdiskusi dengan anggota tim <i>non-Scrum</i> [14].

No.	Scrum Roles	Challenges	Improvements Recommendations
2.	Product Owner	<ol style="list-style-type: none"> Product Owner tidak selalu hadir untuk tim Scrum, hanya tersedia pada acara-acara penting dalam Scrum. Product Owner kurang memiliki pengetahuan dalam menentukan prioritas. 	<ol style="list-style-type: none"> Product Owner harus berkomitmen dan tersedia untuk tim Scrum dan Stakeholder. Jika Product Owner tidak tersedia setiap hari, maka Product Owner harus mendelegasikan tugas kepada orang yang tepat, sehingga tim tidak terhalang dalam menyelesaikan tugasnya [14]. Product Owner secara aktif berkolaborasi dengan Scrum Master dan Development Team. Product Owner harus tersedia untuk menjawab pertanyaan segera setelah pertanyaan diajukan [14]. Product Owner perlu mengikuti pelatihan mengenai praktik-praktik Scrum, terutama pada pengetahuan metode atau teknik dalam menentukan prioritas.
3.	Development Team	Hasil yang dikirimkan oleh Development Team tidak sesuai dengan jadwal yang telah direncanakan, sehingga tanggal rilis produk tidak sesuai dengan rencana awal.	<ol style="list-style-type: none"> Development Team harus meningkatkan komunikasi dan kolaborasi dengan Scrum Master dan Product Owner. Hadir dalam Scrum Event untuk mengidentifikasi apa yang perlu diperbaiki. Menggunakan alat visualisasi proyek, seperti Scrum Board dan Burndown Chart untuk melacak kemajuan proyek.

Rekomendasi Perbaikan untuk Scrum Event

Dalam pelaksanaan Scrum Event, Startup XYZ telah menyelenggarakan Release Planning yang bertujuan untuk menyusun rencana perilis produk atau fitur baru. Untuk Scrum Event yang lain, seperti Daily Scrum, belum pernah diselenggarakan selama proses pengembangan. Sementara itu, Sprint Planning, Sprint Review, dan Sprint Retrospective masih dijalankan sebagian. Rekomendasi perbaikan untuk Scrum Event terlihat pada Tabel 9.

Tabel 9 Rekomendasi Perbaikan untuk Scrum Event

No.	Scrum Events	Challenges	Improvements Recommendations
1.	Release Planning Meeting	<ol style="list-style-type: none"> Tanggal rilis produk tidak sesuai dengan rencana yang telah disepakati. Product Owner tidak menggunakan velocity untuk Release Planning. 	<ol style="list-style-type: none"> Hasil pertemuan harus terdokumentasi dengan baik. Hal ini bertujuan agar seluruh anggota tim Scrum memiliki pemahaman yang sama mengenai target rilis yang telah disepakati dan bertanggung jawab dalam menyelesaikan target rilis tepat waktu. Velocity digunakan untuk dua tujuan penting dalam Scrum, salah satunya memperkirakan berapa banyak Sprint yang dibutuhkan untuk menyelesaikan sebuah rilis. Velocity diukur di setiap akhir Sprint dengan menjumlahkan ukuran dari setiap Backlog yang diselesaikan selama satu Sprint, biasanya diukur dalam bentuk Story Point. Jika sebuah Backlog tidak selesai, maka tidak dihitung ke dalam velocity [14].
2.	Daily Scrum Meeting	Tim Scrum tidak pernah menyelenggarakan Daily Scrum selama periode Sprint berlangsung.	<ol style="list-style-type: none"> Daily Scrum dilaksanakan 15 menit di waktu dan tempat yang sama setiap hari kerja selama periode Sprint berlangsung [5]. Daily Scrum minimal dihadiri oleh Scrum Master dan Development Team. Pada saat Daily Scrum, anggota tim Scrum berfokus pada kemajuan menuju Sprint Goal dan mengidentifikasi kemajuan, rencana, dan hambatan untuk hari itu [5].
3.	Sprint Planning Meeting	<ol style="list-style-type: none"> Sprint Planning tidak secara konsisten dilakukan di setiap Sprint, karena ada beberapa Sprint yang terlewat tidak dilakukan Sprint Planning. Scrum Master dan Product Owner tidak selalu hadir di setiap pelaksanaan Sprint Planning. 	<ol style="list-style-type: none"> Setiap Sprint dimulai dengan Sprint Planning, di mana tim Scrum berkumpul untuk menyetujui Sprint Goal dan menentukan apa yang dapat dikerjakan selama Sprint [14]. Sprint Planning dihadiri oleh Scrum Master, Product Owner, dan Development Team.
4.	Sprint Review Meeting	1. Sprint Review tidak secara konsisten dilakukan di setiap Sprint, karena ada beberapa	1. Sprint Review dilakukan di setiap akhir Sprint, sebelum Sprint Retrospective.

No.	Scrum Events	Challenges	Improvements Recommendations
		<p><i>Sprint</i> yang terlewat tidak dilakukan <i>Sprint Review</i>.</p> <p>2. <i>Scrum Master</i> dan <i>Product Owner</i> tidak selalu hadir di setiap pelaksanaan <i>Sprint Review</i>.</p>	<p>2. Semua anggota tim <i>Scrum</i> hadir di setiap <i>Sprint Review</i>, untuk menjelaskan apa yang telah dicapai, menjawab pertanyaan, dan mendapatkan manfaat umpan balik secara langsung. <i>Stakeholder</i> juga harus hadir untuk memberikan umpan balik [14].</p> <p>3. Hasil pertemuan <i>Sprint Review</i> harus terdokumentasi, yang memuat <i>feedback</i> yang diberikan oleh <i>Stakeholder</i> dan <i>Product Owner</i>.</p>
5.	<i>Sprint Retrospective Meeting</i>	<p>1. <i>Sprint Retrospective</i> tidak secara konsisten dilakukan di setiap <i>Sprint</i>, karena ada beberapa <i>Sprint</i> yang terlewat tidak dilakukan <i>Sprint Retrospective</i>.</p> <p>2. <i>Scrum Master</i> dan <i>Product Owner</i> tidak selalu hadir di setiap pelaksanaan <i>Sprint Retrospective</i>.</p>	<p>1. <i>Sprint Retrospective</i> dilakukan setelah <i>Sprint Review</i> dan sebelum <i>Sprint Planning</i> berikutnya [14].</p> <p>2. Semua anggota tim <i>Scrum</i> menghadiri pertemuan <i>Sprint Retrospective</i>, yang difasilitasi atau dimoderatori oleh <i>Scrum Master</i>. <i>Product Owner</i> disarankan untuk hadir, tetapi tidak diwajibkan [4].</p> <p>3. Salah satu anggota tim bertindak sebagai pencatat dan mendokumentasikan diskusi dan hal-hal yang perlu ditindaklanjuti di masa mendatang [4].</p>

Rekomendasi Perbaikan untuk *Scrum Artifact*

Artefak *Scrum* merupakan hasil kerja yang memberikan transparansi informasi dan pemahaman bersama tentang tujuan proyek, pekerjaan yang dilakukan, dan hasil yang telah dicapai oleh tim *Scrum*. Artefak *Scrum* harus disusun sebaik mungkin dan diperbarui secara berkala, agar setiap orang yang terlibat dalam proyek dengan cepat dan mudah memperoleh informasi. Tabel 10 menyajikan rekomendasi perbaikan untuk *Scrum Artifact*.

Tabel 10. Rekomendasi Perbaikan untuk *Scrum Artifact*

No.	Scrum Artifacts	Challenges	Improvements Recommendations
1.	<i>Product Backlog</i>	<p>1. <i>Product Owner</i> tidak melakukan pembaruan <i>Product Backlog</i> sesuai kemajuan proyek.</p> <p>2. <i>Product Owner</i> tidak mendefinisikan visi produk secara tertulis.</p>	<p>1. <i>Product Owner</i> melakukan pembaruan <i>Product Backlog</i> sesuai dengan kemajuan proyek.</p> <p>2. <i>Product vision</i> adalah pernyataan singkat tentang keadaan masa depan yang akan dicapai melalui pengembangan dan perilsan suatu produk. <i>Product Owner</i> adalah seorang visioner yang merumuskan visi produk dan memimpin tim untuk mencapai visi tersebut [14]. <i>Product Owner</i> harus mendefinisikan visi produk secara tertulis.</p>
2.	<i>Sprint Backlog</i>	<p>1. <i>Sprint Backlog</i> tidak diperbarui setiap hari.</p> <p>2. Estimasi untuk setiap <i>task</i> yang berjalan tidak diperbarui setiap hari.</p> <p>3. Tim <i>Scrum</i> tidak menerapkan <i>Release Burndown Chart</i> dan <i>Sprint Burndown Chart</i> selama pengembangan berlangsung.</p> <p>4. <i>Sprint</i> tidak selalu berakhir sesuai jadwal.</p> <p>5. <i>Sprint</i> yang tidak tercapai sesuai jadwal tidak harus dibatalkan sesegera mungkin.</p>	<p>1. <i>Sprint Backlog</i> diperbarui secara berkala selama <i>Sprint</i> berlangsung, seiring dengan semakin banyaknya informasi yang diperoleh [5].</p> <p>2. Setiap hari selama periode <i>Sprint</i>, anggota tim memperbarui estimasi berapa banyak usaha yang tersisa untuk setiap tugas yang belum selesai [14].</p> <p>3. Tim <i>Scrum</i> menerapkan <i>Release Burndown Chart</i> untuk mengetahui jumlah total pekerjaan yang tersisa di setiap <i>Sprint</i> untuk mencapai tujuan rilis. Tim <i>Scrum</i> memperbarui grafik ini di setiap akhir <i>Sprint</i> [14].</p> <p>4. <i>Sprint Burndown Chart</i> berguna untuk melacak kemajuan dan juga dapat digunakan sebagai indikator utama untuk memprediksi kapan pekerjaan akan selesai. Setiap hari tim <i>Scrum</i> memperbarui grafik ini untuk menunjukkan total estimasi upaya yang tersisa di semua tugas yang belum selesai [14].</p> <p>5. Jika tujuan <i>Sprint</i> tidak valid, tim <i>Scrum</i> dapat menyarankan <i>Product Owner</i> untuk menghentikan <i>Sprint</i> yang tidak normal. Setelah itu, dilakukan <i>Sprint Retrospective</i> untuk merencanakan <i>Sprint</i> berikutnya, dengan tujuan dan <i>Product Backlog</i> yang berbeda [14].</p>
3.	<i>Increment</i>	Tidak ada <i>Definition of Done</i> yang dibuat oleh tim <i>Scrum</i> .	<p>1. Tim <i>Scrum</i> harus mendefinisikan <i>Definition of Done</i> dalam <i>Backlog</i>. Tujuannya agar tim memiliki standar kualitas yang disepakati untuk menentukan kapan suatu <i>Backlog</i> dianggap selesai dan menjadi <i>increment</i>.</p> <p>2. <i>User Story</i> dianggap selesai ketika disetujui oleh <i>Product Owner</i> berdasarkan <i>Definition of Done</i> dan <i>User Story Acceptance Criteria</i> [4].</p>

No.	Scrum Artifacts	Challenges	Improvements Recommendations
			3. <i>Definition of Done</i> harus disepakati oleh semua anggota tim <i>Scrum</i> dan dicapai pada setiap iterasi.

KESIMPULAN DAN SARAN

Berdasarkan hasil penelitian, dapat disimpulkan bahwa tingkat kematangan implementasi kerangka kerja *Scrum* dalam pengembangan perangkat lunak di Startup XYZ masih berada pada level 1 (*Initial*) dalam SMM. Hal ini dibuktikan dengan nilai KPA Rating pada level 2 adalah 85,12% atau dengan kata lain masih dibawah 86% (belum mencapai interpretasi "*Fully Achieved*"). Rekomendasi perbaikan disusun berdasarkan jawaban kuesioner SMM ("Sebagian" dan "Tidak") dan mengacu pada *Scrum Guide 2020*, *SBOK Guide v3*, dan *Essential Scrum*. Rekomendasi perbaikan difokuskan pada level 2 dan level 3 sesuai ekspektasi pihak Startup XYZ dan dikelompokkan ke dalam tiga komponen dasar *Scrum*, yaitu *Scrum Role*, *Scrum Event*, dan *Scrum Artifact*.

Penulis memberikan beberapa saran untuk penelitian selanjutnya, yaitu dapat memperluas cakupan penelitian dengan melibatkan seluruh tim proyek dalam pengembangan perangkat lunak. Objek penelitian ini hanya berfokus pada satu tim proyek saja, sehingga kemungkinan informasi yang diperoleh kurang mendalam. Penelitian selanjutnya dapat melakukan observasi dalam proses pengumpulan data, agar dapat meningkatkan kualitas dan akurasi dari hasil pengukuran tingkat kematangan. Rekomendasi perbaikan pada penelitian selanjutnya dapat disusun berdasarkan panduan *Scrum* terbaru dan diperkaya dengan rekomendasi dari ahli *Scrum*. Penelitian selanjutnya juga dapat melakukan analisis terhadap pencapaian target peningkatan perusahaan, setelah menerapkan rekomendasi perbaikan dalam rentang waktu tertentu.

DAFTAR PUSTAKA

- [1] L. M. Applegate, R. D. Austin and F. W. McFarlan, *Corporate Information Strategy and Management: Text and Cases*, New York: McGraw-Hill Irwin, 2009.
- [2] F. Ridha and E. Hegarini, "Analysis of Maturity Level Project Management of Software Development In Scrum Framework: Case Research On Tribe Enterprise PT. XYZ," *IT Journal Research and Development*, vol. 5, 2020.
- [3] E. S. Pambudi, "Analisis Tingkat Kematangan Implementasi Scrum Menggunakan Scrum Maturity Model pada Instansi Pemerintah," *Jurnal Ilmiah Komputasi*, vol. 20, 2022.
- [4] VMEdu, *A Guide to the Scrum Body of Knowledge (SBOK™ Guide) – 3rd edition*, SCRUMstudy™, 2016.
- [5] K. Schwaber and J. Sutherland, *Panduan Scrum. Panduan Definitif untuk Scrum: Aturan Permainan*, 2020.
- [6] K. Schwaber and J. Sutherland, *Panduan Scrum. Panduan Definitif untuk Scrum: Aturan Main*, 2017.
- [7] I. Panjaitan and N. Legowo, "Measuring Maturity Level of Scrum Practices in Software Development Using Scrum Maturity Model," *Journal of System and Management Sciences*, vol. 12, 2022.
- [8] K. C. Abimaulana, E. K. Budiardjo, K. Mahatma and A. Hidayati, "Evaluation of Scrum-Based Software Development Process Maturity using the SMM and AMM: A Case of Education Technology Startup," in *International Conference on Advanced Computer Science and Information Systems*, 2021.
- [9] C. Patel and M. Ramachandran, "Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices," in *International Journal of Software Engineering*, 2009.
- [10] A. Yin, S. Figueiredo and M. M. da Silva, "Scrum Maturity Model," in *Instituto Superior Técnico*, Lisboa, 2011.
- [11] G. S. Patara and T. Raharjo, "Comparing Scrum Maturity of Digital and Business Process Reengineering Groups: A Case Study at an Indonesia's State-Owned Bank," *International Journal of Advanced Computer Science and Applications*, vol. 14, 2023.
- [12] P. Pandey and M. M. Pandey, *Research Methodology: Tools & Techniques*, Bridge Center, 2015.
- [13] N. Panjaitan and B. Hardian, "Maturity Level Analysis in Software Development Using Scrum Methodology: XYZ Startup Case Study," *Asian Journal of Social and Humanities*, vol. 1, 2023.
- [14] K. Rubin, *Essential Scrum: A Practical Guide to the Most Popular Agile Process*, New York: Addison-Wesley Professional, 2019.
- [15] R. Umar, I. Riadi and E. Handoyo, "Analisis Keamanan Sistem Informasi Berdasarkan Framework COBIT 5 Menggunakan Capability Maturity Model Integration (CMMI)," *Jurnal Sistem Informasi Bisnis*, 2019.