

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Agile dalam pengembangan perangkat lunak menawarkan pendekatan yang lebih fleksibel, iteratif dan bertahap dalam manajemen proyek (Lawong dan Akanfe, 2024). *Agile* merupakan metodologi pengembangan perangkat lunak ringan era 1990-an (Abtokhi dkk., 2023). Metodologi ini dapat diterapkan melalui berbagai kerangka kerja adaptif seperti *Scrum*, *Lean*, *Kanban*, dan *eXtreme Programming*. Namun, kerangka kerja *Agile* yang paling banyak digunakan dan terbukti paling efektif adalah *Scrum* (Lawong dan Akanfe, 2024).

Scrum telah menjadi tren sebagai kerangka kerja untuk pengembangan dan manajemen proyek perangkat lunak (Prasetyaa dkk., 2021). Perkembangan *Scrum* saat ini terus meningkat, terutama di industri TI. Tetapi dalam beberapa tahun terakhir penerapannya telah meluas ke sektor ekonomi lainnya, baik swasta maupun publik (Quintero, 2021). Peningkatan alat bantu digital, pembaruan *Scrum Guide*, serta adopsi metodologi skala besar menunjukkan bahwa *Scrum* terus berkembang dan beradaptasi sesuai dengan kebutuhan organisasi modern.

Tabel 2.1 menjelaskan penelitian-penelitian terdahulu yang telah menerapkan SMM sebagai metode untuk mengevaluasi kinerja *Scrum* serta dapat memberikan rekomendasi perbaikan bagi organisasi:

Tabel 2.1 Penelitian Terdahulu

No	Penelitian	Tujuan	Hasil
1	Ridha, dkk (2020)	Menganalisis tingkat kematangan <i>Scrum</i> pada manajemen proyek pengembangan perangkat lunak.	Hasil pengukuran tingkat kematangan manajemen proyek pengembangan perangkat lunak ada pada <i>level 2</i> . Dengan Peringkat KPA < 86%.

No	Penelitian	Tujuan	Hasil
2	Setiyawan, dkk (2019)	Mengevaluasi dan melakukan perbaikan terhadap praktik <i>Scrum</i> pada perusahaan telekomunikasi	Penilaian menunjukkan praktik <i>scrum</i> berada pada <i>level 1 (Initial)</i> . Terdapat 31,76% praktik <i>scrum</i> di divisi IT dan 36,25% praktik <i>scrum</i> di divisi <i>Corporate Strategy</i> .
3	Patara dan Raharjo (2023)	Menilai tingkat kematangan <i>Scrum</i> dan mengusulkan rekomendasi untuk proses pengembangan perangkat lunak Bank XYZ	Grp-DGT mencapai tingkat kematangan <i>Scrum level 5 (Optimizing)</i> , dengan tujuan KPA <i>Fully Achieved</i> . Grp-BPR berada pada <i>level 1 (Initial)</i> , dengan beberapa tujuan KPA dinilai sebagai <i>Largely Achieved</i> .
4	Arifin, dkk (2020)	Memberikan rekomendasi untuk meningkatkan praktik <i>Scrum</i>	Penilaian menunjukkan bahwa tim <i>Scrum</i> mendapatkan hasil KPA dinilai " <i>Largely Achieved</i> ". Namun, tingkat kematangannya masih berada pada <i>level 1 (Initial)</i> .
5	Ayunda dan Budiardjo (2020)	Menilai seberapa baik kerangka <i>Scrum</i> yang diterapkan dalam organisasi	Fokus penelitian mendapatkan 3 hasil yang melibatkan aspek-aspek utama yaitu: <i>Scrum roles</i> : 75.56 dari 100, <i>Scrum events</i> : 82 dari 100, <i>Scrum artifacts</i> : 75 dari 100
6	Abimaulana, dkk (2021)	Menilai kematangan proses pengembangan perangkat lunak di PT XYZ	Tingkat kematangan organisasi ditemukan berada pada <i>level 1 (Initial)</i> menurut SMM, dengan semua tujuan umum mencapai <i>Large Rating</i> .
7	Panjaitan dan Hardian (2023)	Mengukur tingkat kematangan implementasi <i>Scrum</i> di <i>Startup XYZ</i> serta memberikan rekomendasi perbaikan	Hasil penelitian menunjukkan bahwa <i>Startup XYZ</i> berada pada <i>level 2</i> dengan kategori (<i>Largely Achieved</i>)
8	Panjaitan dan Legowo (2022)	Menilai tingkat kematangan <i>Scrum</i> dalam pengembangan perangkat lunak di PT XYZ	Penelitian menemukan bahwa praktik <i>Scrum</i> PT XYZ berada pada tingkat kematangan 1 (<i>Initial</i>).

No	Penelitian	Tujuan	Hasil
9	Jalaluddin, dkk (2024)	Menilai tingkat kematangan implementasi <i>Scrum</i> di <i>Software House XYZ</i>	Tingkat kematangan implementasi <i>Scrum</i> di <i>Software House XYZ</i> berada pada <i>level 1 (Initial)</i> .
10	Simanjuntak, dkk (2023)	Meningkatkan implementasi <i>Scrum</i> dalam pengembangan aplikasi perbankan digital yang disebut ABC.	Hasil penelitian menunjukkan bahwa implementasi <i>Scrum</i> di ABC berada pada <i>level 2 (Defined)</i> .

Ridha dan Hegarini (2020) menganalisis tingkat kematangan manajemen proyek pengembangan perangkat lunak *Tribe Enterprise Digitization* PT. XYZ. AMM digunakan untuk menganalisis data dari kuesioner SMM. Untuk mengidentifikasi area perbaikan demi mencapai tingkat kematangan yang lebih tinggi, analisis menggunakan KPA. Namun, peringkat KPA dalam penelitian ini belum tercapai sepenuhnya. Selain itu, keterbatasan subjek penelitian yang hanya melibatkan *Scrum Master* di *Tribe Enterprise* PT. XYZ berpotensi memengaruhi akurasi penilaian tingkat kematangan.

Dalam penelitian Setiyawan dkk (2020), *Scrum* digunakan sebagai kerangka utama untuk mengelola pengembangan proyek di PT XYZ, sebuah perusahaan telekomunikasi yang sedang menjalani transformasi digital. SMM memainkan peran penting dalam menilai praktik *Scrum* yang ada di perusahaan. Penelitian ini melakukan evaluasi penerapan *Scrum* pada divisi *Corporate Strategy* (CS) dan *Information Technology* (IT) dengan mengkategorikan masalah ke dalam empat domain, yaitu proses, teknologi, manusia dan organisasi. Hasil tingkat kematangan pada penelitian ini masih berada pada *level* awal, sehingga masih perlu ditingkatkan untuk mencapai *level 5 (Optimizing)*. Model ini membantu menilai kapabilitas *Scrum* organisasi, memandu upaya perbaikan, dan memberikan rekomendasi untuk mencapai kelincahan dan efisiensi optimal dalam manajemen proyek.

Patara dan Raharjo (2023), membandingkan tingkat kematangan *Scrum* antara dua grup di Bank milik negara di Indonesia (Bank XYZ), yaitu Grp-DGT yang mengembangkan proyek digital dan Grp-BPR yang mengembangkan proyek

Business Process Reengineering (BPR). Penelitian ini mengevaluasi seberapa efektif kedua kelompok menerapkan *roles*, *events* dan *artifacts Scrum* dengan menggunakan SMM untuk menilai praktik dan AMM untuk menghitung peringkat kematangan. Grup Grp-DGT di Bank XYZ telah mencapai tingkat kematangan *Scrum level 5 (optimizing)*, sedangkan Grp-BPR berada di *level 1 (initial)*. Sehingga, disusun 15 rekomendasi untuk meningkatkan praktik *Scrum* di Grp-BPR dengan tujuan “*Basic Scrum management*” dinilai *Largely Achieved* (80,97%), “*Standardized Project Management*” dan “*Performance management*” yang berada di peringkat 75%.

Arifin dkk (2020), mengembangkan metode *hybrid* SMM untuk evaluasi praktik *Scrum* pada perusahaan telekomunikasi yang meluncurkan produk baru untuk kaum muda. Penilaian menunjukkan bahwa tingkat yang telah dicapai adalah 16 praktik “*Fully Achieved*”, 49 praktik “*Largely Achieved*”, dan 14 praktik “*Partially Achieved*”. Setelah tingkat kematangan dan masalah yang ada diidentifikasi, rekomendasi berdasarkan komponen *Scrum* (*Scrum roles*, *Scrum artifacts* dan *Scrum events*) diusulkan. Rekomendasi perbaikan implementasi *Scrum* diperoleh dengan cara memetakan praktik-praktik yang perlu diperbaiki dengan teori dari *Scrum Guide 2017*, *SBOK v3* dan pengalaman para ahli *Scrum*. Tujuannya adalah untuk meningkatkan praktik *Scrum* yang belum sepenuhnya dicapai oleh tim.

Ayunda dan Budiardjo (2020), memberikan kerangka kerja yang jelas untuk mengevaluasi *Scrum* secara terstruktur pada perusahaan bisnis terkemuka dalam pengembangan aplikasi, dengan menilai *roles*, *events*, dan *artifacts Scrum* untuk memastikan kepatuhan terhadap praktik terbaik *Agile*. Penelitian ini menggunakan *Scrum Maturity Assessment* dari Vyacheslav Moskalenko. Kerangka ini membantu tim merencanakan langkah-langkah proaktif guna mengatasi potensi risiko dari praktik *Scrum* yang kurang efektif. Namun, pendekatan ini cenderung menekankan kepatuhan terhadap *Scrum Guide*, yang mungkin tidak selalu mencerminkan keberhasilan proyek. Akibatnya, metode ini kurang fleksibel untuk organisasi yang memiliki kebutuhan atau tantangan khusus, karena bisa membatasi efektivitasnya.

Penelitian lain Abimaaulana dkk (2021), melakukan evaluasi tingkat kematangan proses pengembangan perangkat lunak organisasi di bidang Teknologi Pendidikan dan merekomendasikan perbaikan dengan menggunakan kombinasi kerangka kerja SMM dan AMM. Terdapat permasalahan yang dihadapi oleh organisasi, seperti adanya ketidaksesuaian *Sprint Goal* dibandingkan dengan *Objective Key Result* (OKR) yang ditetapkan. Sehingga penelitian ini memberikan rekomendasi untuk menerapkan 14 dari 69 praktik dalam SMM yang belum diterapkan.

Panjaitan dan Hardian (2023) melakukan penelitian pada *Startup XYZ*, yang merupakan pelopor aplikasi *Earned Wage Access* (EWA) dengan visinya membangun kesejahteraan finansial bagi pekerja berpenghasilan rendah dan menengah di Asia. Dalam menerapkan kerangka kerja *Scrum*, ditemukan beberapa masalah yang terjadi yaitu proses pengembangan yang belum optimal sehingga menyebabkan rencana rilis dibatalkan. Rekapitulasi perhitungan tingkat kematangan metodologi *scrum* diketahui bahwa startup XYZ masih berada pada *level 2* dengan kategori "*Largely Achieved*". Setelah mendapatkan tingkat kematangan masing-masing *level*, maka dibuat rekomendasi berdasarkan hasil observasi, *Scrum Guide* 2020, *Scrum Essential* dan *SBOK v3*. Ditemukan sebanyak 24 rekomendasi untuk meningkatkan tingkat kematangan *Startup XYZ*.

Panjaitan dan Legowo (2022), mengembangkan evaluasi untuk mengukur praktik *Scrum* di PT XYZ, perusahaan pemantauan dan analisis media dengan menggunakan model kematangan *Scrum*. Sasaran manajemen *scrum* dasar pada penelitian ini masih berada pada *level "Largely Achieved"* dengan rating KPA sebesar 75,41%, sedangkan sasaran rekayasa persyaratan perangkat lunak juga berada pada kategori "*Largely Achieved*" dengan interpretasi sebesar 73,61%. Saran perbaikan yang disusun hanya ditetapkan pada *level 2* dan *level 3*, hasil penelitian dapat digunakan untuk mengevaluasi pengembangan perangkat lunak serta dapat digunakan sebagai panduan bagi organisasi lain yang mengadopsi *Scrum* dalam proses implementasi proyek perangkat lunak sesuai dengan kebutuhan organisasi.

Jalaluddin dkk (2024), melakukan penilaian tingkat kematangan *Scrum Software House XYZ*. Perusahaan telah menerapkan *Scrum at Scale* sebagai kerangka kerja untuk memecahkan masalah. SMM, yang diadaptasi ke *Scrum Guide 2020*, dipilih sebagai metode untuk mengevaluasi seberapa efektif perusahaan menerapkan *Scrum*. Berdasarkan penilaian menggunakan SMM, kematangan implementasi *Scrum* saat ini di nilai pada *level 1 (Initial)*. Sebanyak 61 praktik diusulkan untuk perbaikan dalam proses *Scrum*. *Scrum at Scale* dapat diimplementasikan setelah perbaikan proses *Scrum* yang disarankan telah dilakukan. Rekomendasi yang disusun mengikuti kerangka kerja yang diuraikan dalam *Scrum at Scale Guide 2022*.

Simanjuntak dkk (2023), melakukan penelitian metode campuran SMM dan AMM untuk memberikan rekomendasi perbaikan implementasi *Scrum* pada salah satu *digital banking* di Indonesia (Bank XYZ) yang memiliki produk *digital ABC* bagi nasabah untuk menyelesaikan transaksi perbankan *online*. Penelitian berbasis kasus ini menunjukkan bahwa Bank XYZ mencapai tingkat kematangan 2 untuk pengembangan ABC. Bank XYZ telah mengimplementasikan 78 dari 79 praktik. Namun, 28 praktik perlu perbaikan dan 1 praktik perlu diterapkan.

Berdasarkan penelitian-penelitian terdahulu, terlihat bahwa implementasi *Scrum* dalam berbagai industri masih menunjukkan variasi tingkat kematangan yang berbeda-beda. Hal ini menunjukkan pentingnya penelitian lebih lanjut untuk memahami faktor-faktor yang mempengaruhi tingkat kematangan *Scrum* dalam berbagai konteks organisasi. Selain itu, perlu dikaji lebih mendalam mengenai efektivitas rekomendasi perbaikan yang diusulkan serta dampaknya terhadap peningkatan kualitas pengembangan perangkat lunak. Penelitian lanjutan ini diharapkan dapat memberikan panduan yang lebih komprehensif dalam mengoptimalkan penerapan *Scrum* sesuai dengan kebutuhan spesifik setiap organisasi serta memvalidasi model kematangan yang digunakan, sehingga implementasi *Scrum* dapat mencapai hasil yang maksimal dan berkelanjutan.

2.2 Dasar Teori

2.2.1. Manajemen Proyek

Manajemen proyek merupakan praktik umum dalam manajemen organisasi (Azanha dkk., 2017). Manajemen proyek adalah penerapan pengetahuan, keterampilan, alat dan teknik pada aktivitas proyek untuk memenuhi kebutuhan proyek. Dalam manajemen proyek terdapat manajer proyek yang bertugas memastikan terpenuhinya persyaratan ruang lingkup, jadwal, biaya, sumber daya, risiko, dan kualitas proyek. Selain itu, manajer proyek juga harus memfasilitasi seluruh proses agar memenuhi kebutuhan serta harapan orang-orang yang terlibat atau terpengaruh oleh aktivitas proyek (Schwalbe, 2021).

2.2.2. Pengembangan Perangkat Lunak

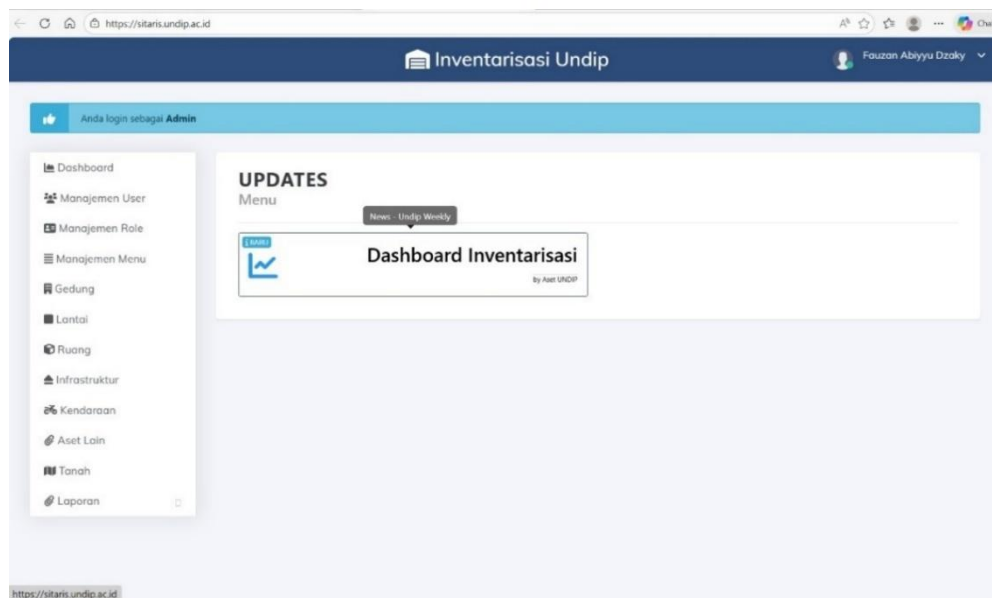
Pengembangan perangkat lunak adalah disiplin dalam rekayasa perangkat lunak yang berfokus pada pembuatan, pengembangan, dan pemeliharaan perangkat lunak berkualitas tinggi. Proses ini mencakup berbagai aktivitas seperti analisis kebutuhan, desain, pengkodean, pengujian, dan pemeliharaan. Tujuannya adalah menghasilkan perangkat lunak yang memenuhi kebutuhan pengguna dengan efisien dan efektif (Asari, 2024). Proses yang melibatkan serangkaian tahapan terstruktur dikenal sebagai *Software Development Life Cycle* (SDLC). SDLC adalah kerangka kerja yang mendefinisikan langkah-langkah yang diperlukan dalam pengembangan perangkat lunak, mulai dari perencanaan hingga pemeliharaan. Ada berbagai model SDLC yang dapat digunakan, seperti *Waterfall* yang berurutan, *V-Model* yang menekankan pengujian di setiap tahap, Iteratif yang memungkinkan pengembangan bertahap, *Agile* yang fleksibel dan cepat beradaptasi, serta *Hybrid* yang menggabungkan beberapa pendekatan (Hossain, 2023).

2.2.3. Sistem Manajemen Aset

SIMASET (Sistem Informasi Manajemen Aset Terpadu) milik Universitas Diponegoro adalah sistem yang digunakan untuk mengelola aset dan barang inventaris di lingkungan kampus. Sistem ini memiliki tiga proyek utama, yaitu Inventarisasi, Pengelolaan Barang dan Pengadaan. Inventarisasi berfungsi untuk mencatat dan mendata seluruh aset yang dimiliki oleh universitas, termasuk peralatan, fasilitas, dan barang inventaris lainnya. Pengelolaan Barang digunakan

untuk mengatur penggunaan, pemeliharaan, dan distribusi barang dalam sistem manajemen aset kampus. Pengadaan berfungsi untuk mengelola proses pengadaan barang dan aset baru untuk kebutuhan universitas.

SIMASET UNDIP merupakan sistem terintegrasi yang mendukung pengelolaan aset secara menyeluruh, mulai dari perencanaan kebutuhan, pelaksanaan pengadaan, hingga pencatatan dan penghapusan aset. Proses inventarisasi dilakukan dengan mendaftarkan Barang Milik Universitas Diponegoro (BMU) ke dalam sistem inventaris agar seluruh aset tercatat secara resmi dan terkontrol (Dzaky dan Kurniawan, 2023).



Sumber: Management Aset UNDIP

Gambar 2.1 Dashboard SITARIS UNDIP

Gambar 2.1 merupakan sistem manajemen aset modul inventarisasi UNDIP berbasis web. Pada modul inventarisasi, sistem menyediakan fitur pendataan aset berdasarkan kategori, seperti Gedung, Lantai, Ruang, Infrastruktur, Kendaraan, Tanah, dan Aset Lainnya. Sistem juga dilengkapi dengan *dashboard* yang menampilkan ringkasan data aset dan dapat memantau aset berdasarkan kategori serta fitur laporan inventarisasi yang dapat digunakan sebagai bahan pemantauan dan pengambilan keputusan.

2.2.4. Agile

Agile merupakan salah satu metode pengembangan perangkat lunak secara bertahap sehingga setiap perubahan kebutuhan dapat diterima dengan cepat (Ayunda dan Budiardjo, 2020). Konsep *Agile software development* dicetuskan oleh Kent Beck bersama 16 rekannya, yang menyatakan bahwa *Agile Software Development* adalah pendekatan dalam membangun perangkat lunak dengan fokus pada pengembangan perangkat lunak itu sendiri sekaligus mendukung kolaborasi antar individu yang terlibat dalam proses tersebut (Pambudi, 2021). Pada tahun 2001, para pelopor metodologi *Agile* berkumpul di Snowbird Ski Resort di Utah dan merumuskan *manifesto* bersama mengenai metodologi ini. *Manifesto* ini kemudian menjadi prinsip dasar *Agile*, yang menekankan komunikasi dan kolaborasi, fungsionalitas perangkat lunak, dan fleksibilitas untuk beradaptasi dengan realitas bisnis yang muncul (Panjaitan dan Hardian, 2023).

Ke-12 prinsip ini berfungsi sebagai panduan bagi mereka yang ingin berhasil dalam mengimplementasikan *Agile Software Development*. Pada penelitian (Patara dan Raharjo, 2023), menjelaskan prinsip-prinsip tersebut, adalah sebagai berikut: (1) memprioritaskan kepuasan pelanggan, (2) menerima perubahan kebutuhan, (3) secara konsisten memberikan perangkat lunak yang berfungsi dengan baik, (4) interaksi dan kolaborasi harian antara orang bisnis dan pengembang, (5) memotivasi individu untuk membangun proyek, (6) menggunakan percakapan tatap muka untuk berbagi informasi ke dalam tim pengembang, (7) kemajuan proyek dievaluasi melalui perangkat lunak yang berfungsi dengan baik, (8) keberlanjutan pengembangan dicapai ketika sponsor, pengembang dan pengguna mempertahankan ritme mereka secara konsisten, (9) fokus konstan pada keunggulan teknis dan desain yang baik, (10) menyederhanakan hal-hal untuk memaksimalkan hasil dan dampak, (11) tim yang mengorganisasi sendiri mempromosikan desain, spesifikasi dan arsitektur terbaik, dan (12) tim mempertimbangkan cara untuk lebih efektif, kemudian mengkalibrasi dan menyesuaikan perilakunya sesuai dengan itu.

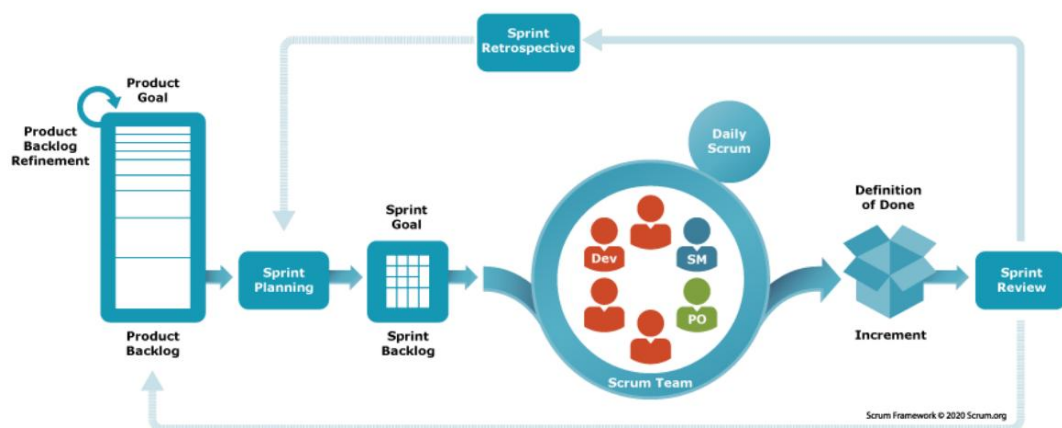
2.2.3. *Scrum Framework*

Scrum merupakan salah satu framework bagian dari *Agile Software Development Methodology* yang dikembangkan oleh Jeff Sutherland dan Ken Schwaber pada awal tahun 1990-an (Maulana dan Raharjo, 2021). Saat ini, *Scrum* banyak diimplementasikan sebagai alternatif terhadap metodologi *waterfall* yang bersifat prediktif (Setiyawan dkk., 2020). *Scrum* didefinisikan sebagai kerangka kerja yang digunakan oleh tim untuk mengatasi masalah adaptif yang kompleks sambil menghasilkan produk dengan nilai maksimal pada tingkat produktivitas dan kreativitas tertinggi (Schwaber dan Sutherland, 2020).

Metodologi ini membantu menyelesaikan masalah jangka pendek, mengurangi risiko proyek, meningkatkan keterlibatan pelanggan dan memberikan produk yang lebih fungsional. Fokus utama *Scrum* adalah memfasilitasi kerja sama tim, meningkatkan komunikasi dan mendorong kolaborasi yang lebih baik (Sassa dkk., 2023). Tim *Scrum* sendiri terdiri dari *Scrum Master* (SM), *Product Owner* (PO), dan *Development Team* (DT). Setiap bagian dalam tim memiliki tugas yang mendasar (Quintero, 2021). SM bertugas mempromosikan *Scrum* di dalam tim, mendukung dan menyebarkan konsep *Scrum* di tingkat tim dan organisasi. SM memastikan terlaksananya semua *ceremonial Scrum* dan membantu tim menghindari kendala selama *sprint*. PO bertanggung jawab untuk memaksimalkan nilai produk yang dihasilkan. PO memiliki tanggung jawab penuh atas pengelolaan *backlog*. DT adalah kelompok yang mengembangkan inkrementasi produk jadi. Tim ini harus mampu mengorganisasi diri sendiri dan bersifat multifungsi. DT idealnya terdiri dari 3 hingga 9 orang agar tetap produktif dan sesuai dengan prinsip *Agile* (Schwaber dan Sutherland, 2020).

Gambar 2.2 merupakan kegiatan pertemuan/*ceremonial Scrum* yang terdiri dari beberapa bagian. (Quintero, 2021) merangkum karakteristik utama *ceremonial* yang terdiri dari *Sprint Planning* yaitu merencanakan pekerjaan yang akan dilakukan dalam *sprint*, mendiskusikan tujuan *sprint* dan *backlog*, serta waktu yang dibutuhkan untuk setiap item *backlog*. Dilakukan maksimal 8 jam untuk *sprint* 1 bulan oleh seluruh tim *Scrum*: PO, SM dan DT. *Daily Scrum* adalah pertemuan singkat selama lima belas menit. Dimana SM memastikan tim dapat mengadakan

pertemuan yang dilaksanakan oleh DT untuk mendiskusikan rencana kerja untuk 24 jam ke depan, mengevaluasi kemajuan terhadap tujuan *sprint* dan memastikan komunikasi dalam tim berjalan lebih baik. *Sprint Review* adalah kegiatan yang dilakukan diakhir *sprint*. PO menjelaskan backlog yang selesai dan tertunda, DT membahas kinerja selama *sprint* serta mendemonstrasikan hasilnya dan seluruh grup menganalisis langkah perencanaan *sprint* berikutnya. Durasi yang dibutuhkan maksimal 4 jam untuk *sprint* satu bulan. *Sprint Retrospective* adalah melakukan evaluasi mandiri terkait pelaksanaan *sprint* terakhir, mencakup aspek orang, hubungan, proses dan alat yang digunakan. Mengidentifikasi keberhasilan, potensi perbaikan dan menyusun rencana untuk meningkatkan cara kerja tim di *sprint* berikutnya. *Sprint Retrospective* dilakukan maksimal 3 jam untuk *sprint* 1.



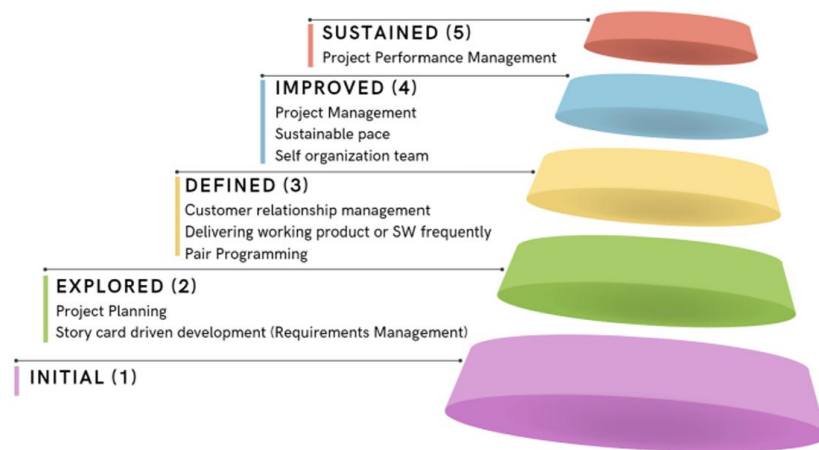
Sumber: scrum.org

Gambar 2.2 Scrum Framework

Scrum juga mengidentifikasi empat artifak yang dioperasikan oleh seluruh tim *Scrum* pada siklus pengembangan yaitu *Product Backlog* yang merupakan sebuah daftar prioritas mengenai segala sesuatu yang diperlukan untuk menyimpulkan suatu produk. *Sprint Backlog* yaitu merupakan daftar tugas yang harus dilakukan selama *sprint*. *Release Burndown Charts* yang berupa grafik yang menunjukkan kemajuan proyek dari waktu ke waktu dan *Sprint Burndown Charts* yaitu grafik yang menampilkan kemajuan *sprint* dari waktu ke waktu (Pambudi, 2021).

2.2.4. *Agile Maturity Model*

Agile Maturity Model (AMM) diperkenalkan oleh dua peneliti di sebuah Universitas IT di Leeds, dan dirancang untuk memberikan peneliti masa depan model kematangan *Agile* yang lebih mendalam sebagai dasar penelitian mereka (Yin dkk., 2011). Tujuan utama model AMM adalah menyesuaikan program perbaikan proses perangkat lunak untuk lingkungan *Agile*. Oleh karena itu, mengidentifikasi tingkat kematangan praktik *Agile* merupakan aktivitas krusial dalam model AMM (Patel dan Ramachandran, 2009). Gambar 2.3 menjelaskan mengenai lima definisi *level* kematangan dalam AMM.



Sumber: Patara dan Raharjo (2023)

Gambar 2.3 *Agile maturity model*

Menurut (Arifin dkk., 2020), penilaian AMM dibagi menjadi dua tahap. Tahap pertama menilai kesesuaian dan kemampuan beradaptasi tim. Penilaian dilakukan dengan menjawab kuesioner yang dibagi menjadi lima bagian. Kusioner tersebut terdiri dari: 1) metode pengembangan perangkat lunak yang dipilih atau yang ingin dipilih; 2) masalah yang diidentifikasi selama pengembangan perangkat lunak dan yang diadopsi atau direncanakan untuk diadopsi untuk memecahkan masalah; 3) ketersediaan dan hubungan pelanggan; 4) pengetahuan pengembang dan manajer tentang metode *Agile* dan kualitas kerja tim; 5) ukuran proyek.

Tahap kedua adalah melakukan penilaian untuk mengidentifikasi area yang perlu ditingkatkan. Semua tujuan dalam AMM akan diterjemahkan menjadi pertanyaan-pertanyaan. Setiap pertanyaan dapat dijawab dengan “ya”, “sebagian”,

“tidak” dan “tidak berlaku (N/A)”. Perhitungan data dari kuesioner menggunakan persamaan yang diadopsi dari AMM seperti berikut (Chetankumar & Ramachandran, 2009):

$$KPA = \frac{\sum(Yn) + \frac{1}{2}\sum(Pn)}{\sum(Tn) - \sum(NAn)} \times 100 \quad (2.1)$$

(Yn) = Jumlah jawaban “Ya”;

(Pn) = Jumlah jawaban “Berlaku Sebagian”;

(Tn) = Jumlah pertanyaan;

(NAn) = Jumlah Jawaban “Tidak berlaku [N/A].

Berdasarkan perhitungan rating KPA, beberapa kategori penilaian akan diperoleh sebagai berikut:

1. *Fully Achieved* (Tercapai Sepenuhnya), yang berkisar antara 86% hingga 100%. Tidak ada kelemahan signifikan di seluruh unit organisasi.
2. *Largely Achieved* (Tercapai Sebagian Besar), yang berkisar antara 51% hingga 85%. Terdapat bukti pendekatan yang baik dan sistematis serta pencapaian praktik-praktik kritis yang ditetapkan dalam KPA yang dinilai.
3. *Partially Achieved* (Tercapai Sebagian), yang berkisar antara 16% hingga 50%. Terdapat bukti pendekatan yang baik dan sistematis serta pencapaian praktik-praktik utama yang ditetapkan dalam KPA yang dinilai. Beberapa aspek pencapaian mungkin tidak dapat diprediksi.
4. *Not Achieved* (Tidak Tercapai), yang berkisar antara 0% hingga 15%. Terdapat sedikit bukti pencapaian praktik-praktik utama yang ditetapkan dalam KPA yang dinilai.

2.2.5. Scrum Maturity Model

SMM merupakan pendekatan yang dapat diterapkan dan dimanfaatkan untuk mengurangi kegagalan pengembangan proyek organisasi, yang bertujuan untuk membantu organisasi dalam mengadopsi *Scrum* dan mendorong peningkatan dengan berfokus pada peran klien dan pengguna (Panjaitan dan Hardian, 2023). Tabel 2.2. merupakan 5 *Level Scrum Maturity Model* yang dikembangkan oleh (Yindkk., 2011).

Tabel 2.2 *Scrum Maturity Level*

Level	Maturity	Tingkat	Objective
1	<i>Initial</i>	-	-
2	<i>Managed</i>	<i>Basic Scrum Management (BSM)</i>	Adanya peran <i>Scrum</i>
			Adanya <i>Scrum Artifacts</i>
			Adanya <i>Scrum meeting</i> dan dipartisipasi
			Alur proses <i>Scrum</i> dihormati
3	<i>Defined</i>	<i>Software Requirement Engineering (SRE)</i>	Definisi yang jelas tentang <i>Product owner</i>
			Manajemen <i>Product backlog</i>
			<i>Sprint Planning Meeting</i> yang sukses
		<i>Customer Relationship Management (CRM)</i>	Adanya <i>Definition of "Done"</i>
			<i>Product owner</i> tersedia
			<i>Sprint Review Meeting</i> yang sukses
4	<i>Quantitatively Managed</i>	<i>Iteration Management (IM)</i>	Manajemen <i>Sprint backlog</i>
			Iterasi direncanakan
		<i>Standardized Project Management (SPM)</i>	<i>Velocity</i> yang terukur
			<i>Daily Scrum</i> yang sukses
5	<i>Optimizing</i>	<i>Performance Management (PM)</i>	Manajemen proyek terpadu
			Analisis Manajemen
			<i>Process Performance Management (PPM)</i>
5	<i>Optimizing</i>	<i>Performance Management (PM)</i>	<i>Sprint retrospective</i> yang sukses
			Analisis penyebab dan penyelesaian
			Indikator positif

Pada tahapan ini, pengumpulan data dilakukan melalui kuesioner sesuai *Scrum Assessment Questionnaire* (SAQ). Penilaian SMM digunakan untuk mengetahui tingkat kematangan pada area proses *Scrum*, sedangkan AMM digunakan untuk menilai tingkat kematangan pada masing-masing area proses. KPA

Rating digunakan untuk menghitung dan menilai penerapan praktik-praktik yang terdapat dalam AMM. Respons kuesioner terdiri dari beberapa pilihan jawaban, antara lain:

1. Ya, artinya menunjukkan bahwa pertanyaan kuesioner telah diimplementasikan sepenuhnya.
2. Tidak, artinya menunjukkan bahwa pertanyaan kuesioner tidak dilaksanakan atau tidak berlaku sama sekali.
3. Terlaksana sebagian, artinya menunjukkan bahwa pertanyaan kuesioner terlaksana sebagian.
4. Tidak Berlaku (N/A), yang menunjukkan bahwa pertanyaan kuesioner tidak dapat dijawab.

Scrum Maturity Model adalah kerangka kerja yang dikembangkan dari *Agile Maturity Model*, yang dapat digunakan oleh organisasi dalam pengembangan perangkat lunak menggunakan *Scrum*. SMM bertujuan membantu membimbing proses pengembangan atau peningkatan *Scrum* dalam pengembangan perangkat lunak secara bertahap dan mandiri. SMM memiliki lima tingkat kematangan dengan tujuan dan sasaran pada setiap *level* sebagai berikut (Yin dkk., 2011):

1. *Level 1 – Initial*

Proses umumnya berjalan lambat dan tim menghadapi berbagai masalah dalam memberikan nilai. Keberhasilan bergantung pada kompetensi anggota tim. *Level 1* diberikan jika sebuah organisasi tidak memenuhi tujuan yang didefinisikan pada *level 2*.

2. *Level 2 – Defined*

Karakteristik organisasi *level 2* adalah mengikuti praktik *Scrum* dalam versi yang disesuaikan. Terdapat dua tujuan utama pada *level 2*: manajemen dasar *Scrum* dan rekayasa kebutuhan perangkat lunak. *Basic Scrum Management* mengevaluasi peran tim *Scrum*, keberadaan *Scrum Artifacts*, adanya *Scrum Meeting*, serta pelaksanaan *sprint* yang benar. *Software Requirement Engineering* mengevaluasi peran *Product owner* yang jelas, Manajemen *product backlog* dan keberhasilan *sprint planning meeting*.

3. *Level 3 – Managed*

Pada tahap ini, unit telah menguasai kerangka kerja *Scrum* yang benar berdasarkan prinsip dan nilai *Agile* yang tepat. Terdapat dua tujuan utama pada *level 3: Customer Relationship Management* dan *Iteration Management*. Manajemen hubungan pelanggan mengevaluasi ketersediaan *Product owner*, keberhasilan *sprint review meeting*, serta kejelasan *definition of “Done”*. Manajemen iterasi mengevaluasi pengelolaan *sprint backlog*, perencanaan iterasi, keberhasilan *daily Scrum* dan pengukuran kecepatan.

4. *Level 4 – Quantitatively Managed*

Pada tahap ini, *Scrum* telah diinstitusionalisasi. Tujuan utama *level 4* meliputi *Standardized Project Management* dan *Process Performance Management*. Tujuan utamanya adalah mengevaluasi manajemen proyek terpadu dan analisis manajemen.

5. *Level 5 – Optimizing*

Level 5 adalah tingkat tertinggi dalam SMM dengan peningkatan diri yang berkelanjutan. Tujuan utama untuk *level 5* adalah *Performance Management* untuk mengukur dan menganalisis semua proses peningkatan diri.

SEKOLAH PASCASARJANA