

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pesatnya perkembangan teknologi saat ini, sangat mempermudah pekerjaan dan memenuhi kebutuhan manusia dengan lebih cepat (Panjaitan dan Hardian, 2023). Proyek-proyek Teknologi Informasi (TI) modern ditandai dengan tingkat kompleksitas yang tinggi, dengan penggunaan teknologi inovatif yang terintegrasi antara aspek teknologi dan organisasi (Prokopenko dkk., 2021). Dalam konteks pengembangan perangkat lunak, salah satu metodologi yang banyak digunakan saat ini adalah *Agile framework Scrum* yang telah terbukti efektif dalam mengelola tim dan proses pengembangan secara iteratif dan fleksibel (Wonohardjo dkk., 2019). *Scrum* dirancang sebagai *framework* yang fleksibel dan sederhana, di mana anggota tim bekerja sama untuk menyelesaikan permasalahan yang kompleks dengan hasil yang terbukti mampu meningkatkan produktivitas (Schwaber dan Sutherland, 2020).

Salah satu organisasi yang telah menerapkan *Scrum* sebagai kerangka kerjanya yaitu Universitas Diponegoro dengan mengimplementasi *Agile framework Scrum* dalam pengembangan Sistem Informasi Manajemen Aset (SIMASET) pada modul inventarisasi (Dzaky dan Kurniawan, 2023). Aplikasi ini mendukung pengelolaan aset universitas secara menyeluruh, mulai dari perencanaan kebutuhan hingga penghapusan aset. Pengelolaan aset yang efisien sangat penting untuk institusi besar seperti universitas, karena berhubungan langsung dengan penggunaan sumber daya yang optimal. Sebelumnya, Universitas Diponegoro menggunakan metode *Big Bang* dan *Use Case Point* dalam pengembangan sistem informasinya, namun kini telah beralih ke penerapan *Scrum*.

Dalam pengembangan perangkat lunak, pada SIMASET Modul Inventarisasi, implementasi *Scrum* belum sepenuhnya optimal. Beberapa permasalahan utama meliputi pengetahuan yang belum mendalam mengenai konsep *Scrum*, keterbatasan jumlah anggota tim yang memengaruhi alokasi tugas, serta tantangan dalam beradaptasi terhadap perubahan metodologi. Kondisi ini dapat berdampak pada

efektivitas kolaborasi, pengelolaan *backlog*, dan perencanaan *sprint*, serta menimbulkan ketidakefisienan dan hambatan dalam koordinasi tim. Selain itu, belum adanya evaluasi terstruktur mengenai tingkat kematangan praktik *Scrum* dapat membuat instansi kesulitan untuk mengidentifikasi area yang perlu perbaikan.

Proses pengembangan perangkat lunak perlu dievaluasi untuk memastikan bahwa proses tersebut telah memenuhi kriteria pengembangan perangkat lunak yang baik dan sesuai dengan praktik yang telah teruji (Anggraeny dkk., 2023). Implementasi *Scrum* dapat memunculkan beberapa risiko krusial jika tidak ditangani dengan tepat, seperti kurangnya koordinasi, komunikasi dan integrasi tugas dalam tim. Hal ini dapat menyebabkan kesulitan dalam mengelola peran dan memastikan kolaborasi yang efisien. Di sisi lain, keberhasilan *Scrum* sangat bergantung pada keterampilan dan pengetahuan anggota tim (Wonohardjo dkk., 2019). Hambatan lain juga perlu diperhatikan, yaitu kesulitan dalam menemukan solusi paling efektif, yang dapat mengakibatkan keterlambatan penyelesaian tugas sesuai jadwal yang ditetapkan dan memperpanjang durasi proses pengembangan (Morandini dkk., 2021). Transisi menuju *Scrum* sering kali terhambat oleh kesulitan dalam estimasi selama perencanaan *sprint*. Hal ini dapat menyebabkan ritme *sprint* tidak sesuai dengan praktik standar (Setiyawan dkk., 2020). Oleh karena itu, evaluasi implementasi *Scrum* dalam organisasi diperlukan untuk memberikan rekomendasi perbaikan di masa mendatang.

Dalam mengukur tingkat pengembangan manajemen proyek, digunakan sebuah metode yang disebut *maturity model*. Beberapa model kematangan yang sering diterapkan dalam pengembangan proyek antara lain: *Capability Maturity Model Integration for Development* (CMMI-Dev), *Agile Maturity Model* (AMM), dan *Scrum Maturity Model* (SMM) (Abimaaulana dkk., 2021). CMMI dikenal sebagai salah satu metode yang digunakan untuk mengukur tingkat kematangan dan memberikan panduan untuk memastikan proses yang stabil, mumpuni dan matang. Namun, model ini tidak sepenuhnya mampu mendukung implementasi proses *Agile* secara memadai (Panjaitan dan Legowo, 2022). CMMI merupakan model yang tergolong rumit dalam melakukan evaluasi tingkat kematangan karena harus

mengevaluasi keseluruhan dari organisasi. Sifat *Agile* yang iteratif dan fleksibel sangat bertentangan dengan dengan struktur CMMI yang lebih memiliki format formal dan terstruktur (Febiana dkk., 2024).

Kerangka *Agile Maturity Model* dikembangkan oleh Patel dan Ramachandran pada tahun 2009, menurutnya AMM menghubungkan praktik pengembangan perangkat lunak dengan metode *Agile* dengan konsep kematangan dalam CMMI (Panjaitan dan Hardian, 2023). AMM memberikan pendekatan pertama untuk mengklasifikasikan kematangan proses pengembangan *Agile*, yang hakikatnya terdiri dari praktik-praktik dari berbagai metodologi *Agile* (Atissalam dan Aji, 2024). Pada tahun 2011 Yin, Figueiredo dan Miguel memperkenalkan konsep *Scrum Maturity Model* (Ridha dan Hegarini, 2020). SMM dikembangkan dari CMMI dan AMM dengan fokus pada kerangka *Scrum* yang bertujuan untuk membantu serta membimbing organisasi dalam mengembangkan proyek teknologi informasi dengan mendorong peningkatan berkelanjutan melalui pendekatan yang lebih spesifik terhadap peran klien (Arifin dkk., 2020). Setiap *level* dalam SMM memiliki tujuan yang ditetapkan berdasarkan prinsip *Agile*, dimana setiap tujuan memiliki objektif dan memiliki praktik yang harus diimplementasikan (Abimaaulana dkk., 2021).

SMM tidak menjelaskan secara detail bagaimana cara menganalisis data, sehingga banyak penelitian menggunakan pendekatan *Key Process Area* (KPA) *Rating* dari AMM untuk menilai dan menghitung implementasi *Scrum* SMM (Panjaitan dan Legowo, 2022). SMM digunakan untuk mengevaluasi implementasi praktik-praktik *Scrum* secara spesifik, Sementara itu, AMM digunakan untuk mengkalkulasi nilai melalui KPA *Rating*. Dengan mengombinasikan kedua model tersebut, hasil evaluasi tidak hanya menunjukkan kondisi kematangan praktik *Scrum* secara operasional, tetapi juga memetakan tingkat adopsi *Agile* secara menyeluruh (Patara dan Raharjo, 2023).

Jika analisis dilakukan tanpa menggunakan AMM, maka kesimpulan hanya akan diambil berdasarkan hasil kuesioner secara langsung. Hal ini dapat menyebabkan hasil evaluasi menjadi kurang mendalam karena tidak ada sistem penilaian yang jelas untuk menentukan tingkat kematangan *Scrum* secara objektif.

Pada Sistem Informasi Manajemen Aset Terpadu Universitas Diponegoro, proses pengembangan perangkat lunak telah mengadopsi *Scrum*, tetapi sejauh mana praktik tersebut dijalankan secara konsisten dan sesuai standar belum pernah dipetakan. Minimnya gambaran mengenai tingkat kematangan ini dapat menyebabkan ketidakefisienan proses, ketidakselarasan peran, serta potensi hambatan pada kolaborasi tim. Oleh karena itu, diperlukan pemahaman yang jelas mengenai kondisi kematangan praktik *Scrum* saat ini agar organisasi memiliki dasar yang kuat untuk melakukan peningkatan pada proses pengembangan perangkat lunak.

1.2. Tujuan Penelitian

Penelitian ini bertujuan untuk mengevaluasi tingkat kematangan manajemen proyek pengembangan perangkat lunak berbasis *Scrum* pada Sistem Informasi Manajemen Aset Terpadu Universitas Diponegoro untuk memberikan rekomendasi perbaikan dengan menganalisis tingkat kematangan menggunakan *Scrum Maturity Model* dengan kombinasi *Agile Maturity Model*.

1.3. Manfaat Penelitian

Manfaat dari penelitian ini adalah secara akademis dapat menambah wawasan dan literatur mengenai penerapan metode *Scrum* dalam pengembangan perangkat lunak. Sedangkan manfaat dari praktisi adalah dapat memberikan panduan praktis bagi pengembang perangkat lunak dan manajer proyek di Universitas Diponegoro dalam mengimplementasikan *Scrum* untuk meningkatkan efektivitas dan efisiensi proyek.