

BAB II. TINJAUAN PUSTAKA

2.1. Studi Sebelumnya

Berdasarkan penelitian yang dilakukan Hema dkk. [7] pada tahun 2020, memiliki tujuan menganalisis penggunaan metode *Agile* untuk pengembangan *software* yang efektif. Hasil penelitian menunjukkan bahwa pengembangan *software* menggunakan metode *Agile* memiliki tingkat keberhasilan lebih tinggi dibandingkan dengan pengembangan *software* menggunakan metode *waterfall*. Hal ini disebabkan karena *Agile* menawarkan kelebihan berupa koordinasi tim yang efektif, partisipasi aktif dari pelanggan atau pemilik produk untuk umpan balik yang cepat, adaptabilitas terhadap perubahan kebutuhan bisnis dan pelanggan, membangun kepuasan dan kepercayaan di antara tim, perencanaan yang terstruktur, dan pengiriman produk dalam rilis kecil dan sering.

Penelitian yang dilakukan oleh Ardiansyah dkk. [8] bertujuan untuk mengembangkan sistem informasi terpadu Universitas Lampung di Fakultas MIPA. Pendekatan pengelolaan proyek yang digunakan adalah metode *Scrum*. Tahapan penelitian yang dilakukan perumusan masalah, studi literatur, Metode *Scrum* dan penulisan laporan. Penelitian dilakukan selama 7 *sprint* dengan total *product backlog* item sebanyak 23 item. Hasil akhir dari penelitian ini adalah sebuah *website* sistem informasi layanan administrasi yang ada di fakultas MIPA di Universitas Lampung.

Penelitian lainnya yang dilakukan oleh Almeida & Carneiro [9] pada tahun 2021, memiliki tujuan untuk menganalisis metrik yang dapat digunakan dalam lingkungan *Scrum* untuk memonitor dan mengevaluasi perkembangan tim dan dukungan pelaksanaan perbaikan. Hasil penelitian ini menunjukkan terdapat metrik penting yang dapat diukur dan dilacak untuk meningkatkan kualitas tiap *sprint* seperti *release burnout*, *standard violation*, *number of stories*, *work in progress*, *business value delivered* dan *sprint goal success*.

Kemudian penelitian yang dilakukan oleh Prabowo & Wiguna [10] pada tahun 2021 dengan tujuan untuk membangun sistem informasi bengkel berbasis web. Pendekatan yang dipakai untuk mengembangkan sistem informasi tersebut adalah pendekatan *Scrum*. Tahapan penelitian yang dilakukan meliputi *Product Log*, *Sprint Backlog*, *Sprint*, dan *Working increment of software*. Pada tahap implementasi, digunakan bahasa pemrograman PHP untuk *Graphic User Interface* dan *database MySQL* sebagai penyimpanan data. Hasil penelitian ini adalah sistem informasi UMKM bengkel berbasis web.

Penelitian yang dilakukan oleh Dixit & Bhushan [11] pada tahun 2019 memiliki tujuan untuk menganalisis *Scrum Metrics* yang penting dan berguna dalam kesuksesan dalam pengembangan produk. Metrik yang digunakan dalam penelitian ini meliputi pengukuran *deliverable*, efektivitas, *monitoring* tim dan *reporting*. Hasil dari penelitian ini adalah hasil survei yang menunjukkan pentingnya *Scrum Metrics* sebagai alat *monitoring*, manajemen dan membantu dalam pengambilan keputusan dalam sebuah organisasi.

Tabel 1.1 Studi Sebelumnya

No	Penulis	Tahun	Tujuan	Metode	Hasil
1	Hema dkk. [7]	2020	Menganalisis penggunaan metode <i>Agile</i> untuk pengembangan <i>software</i> yang efektif.	Literature <i>Review</i>	Pengembangan <i>software</i> menggunakan metode <i>Agile</i> memiliki tingkat keberhasilan lebih tinggi dibandingkan dengan pengembangan <i>software</i> menggunakan metode <i>waterfall</i>
2	Ardiansyah dkk. [8]	2021	Mengembangkan sistem informasi terpadu Universitas Lampung di Fakultas MIPA	<i>Scrum</i>	Sebuah <i>website</i> sistem informasi layanan administrasi yang ada di fakultas MIPA di Universitas Lampung
3	Almeida & Carneiro [9]	2021	Menganalisis metrik yang dapat digunakan dalam lingkungan <i>Scrum</i> untuk memonitor dan mengevaluasi perkembangan tim dan dukungan pelaksanaan perbaikan.	Penelitian kuantitatif menggunakan survei terhadap profesional <i>Scrum</i> .	Terdapat metrik penting yang dapat diukur dan dilacak untuk meningkatkan kualitas tiap <i>sprint</i> seperti <i>release burnout</i> , <i>standard violation</i> , <i>number of stories</i> , <i>work in progress</i> , <i>business value delivered</i> dan <i>sprint goal success</i>

No	Penulis	Tahun	Tujuan	Metode	Hasil
4	Prabowo & Wiguna [10]	2021	Membangun sistem informasi bengkel berbasis web.	<i>Scrum</i>	Sistem informasi UMKM bengkel berbasis web.
5	Dixit & Bhushan [11]	2019	Menganalisis <i>Scrum Metrics</i> yang penting dan berguna dalam kesuksesan dalam pengembangan produk	Analisis komparatif dan survei	Hasil survei yang menunjukkan pentingnya <i>Scrum Metrics</i> sebagai alat <i>monitoring</i> , manajemen dan membantu dalam pengambilan keputusan dalam sebuah organisasi

2.2. Dasar Teori

2.2.1. Website

Website adalah kumpulan berbagai halaman web yang ditulis menggunakan HTML (*HyperText Markup Language*). *Website* harus disimpan atau di-*hosting* di komputer yang terhubung ke Internet selama 24 jam setiap hari agar orang di seluruh dunia dapat mengaksesnya. Selain itu, *website* tersebut mungkin mengandung dokumen dan *file* tambahan, seperti gambar, video, atau aset digital lainnya [12][13].

2.2.2. Front End

Front End adalah bagian dari sistem yang bertanggung jawab untuk menampilkan tampilan kepada pengguna. Di bagian *Front End*, seorang pengembang akan membangun *user interface* dengan yang menarik dan tidak membosankan untuk dilihat [14]. Aplikasi web dibangun dengan menggunakan teknologi web yang populer seperti HTML, CSS, dan *JavaScript*. HTML membangun struktur halaman web, CSS mengatur tampilan halaman web, dan *JavaScript* menambah logika dan interaksi pada halaman web [15].

2.2.3. Multi-tenant

Aplikasi *multi-tenant* memungkinkan beberapa penyewa(pengguna) menggunakan sumber daya perangkat keras yang sama, dengan memberikan satu aplikasi dan basis data yang digunakan bersama, sambil tetap memungkinkan mereka menyesuaikan aplikasi tersebut sesuai kebutuhan masing-masing seakan-akan berjalan di lingkungan yang didedikasikan khusus untuk mereka [16][17].

2.2.4. Pengelolaan Proyek

Pengelolaan Proyek merupakan penerapan yang digunakan untuk merencanakan, menjadwalkan dan mengelola sumber daya untuk menyelesaikan tujuan dan sasaran proyek. Pengelolaan proyek bertujuan untuk menghilangkan rintangan selama masa pengembangan proyek [18]. Ada

beberapa jenis pengelolaan proyek yang dapat digunakan, seperti Waterfall, yang melibatkan pendekatan linear dan berurutan, pengelolaan proyek Kanban yang menggunakan papan visual, serta pengelolaan proyek *Rapid Applications Development* (RAD) yang berfokus pada prototyping dan iterasi cepat. Selain itu terdapat juga pengelolaan proyek Agile, yang adaptif dan iteratif yang berfokus pada fleksibilitas [19].

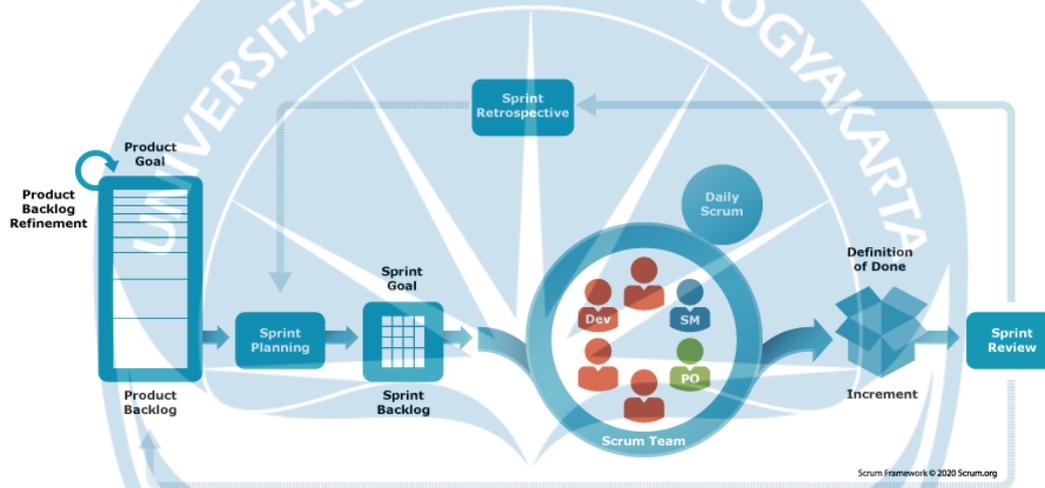
2.2.5. Agile

Metode pengembangan perangkat lunak *Agile* adalah metode pengembangan perangkat lunak yang ringan. Tujuan utama metode ini adalah untuk mempercepat pengembangan dan secara efektif menanggapi perubahan yang diminta. Banyak proyek yang menggunakan teknik pengembangan perangkat lunak tradisional menghadapi masalah signifikan, terutama terkait dengan pemeliharaan dan modifikasi berdasarkan permintaan pengguna. Perubahan signifikan yang dianggap sebagai masalah besar dalam pengembangan perangkat lunak dapat dihasilkan oleh beberapa perubahan ini. Metode pengembangan perangkat *Agile* dibutuhkan untuk mengatasi perubahan tersebut [20].

Pengembangan Agile menggunakan siklus *delivery* singkat yang disebut sprint, disertai sesi umpan balik rutin. Proses Agile memerlukan lingkungan kerja yang mendukung kolaborasi intensif dengan rekan kerja dan klien. Komunikasi yang kuat memungkinkan penyesuaian produk selama setiap sprint. Pemangku kepentingan meninjau setiap tahap dan memberikan perbaikan. Tim bertanggung jawab secara kolektif atas pengembangan dan keberhasilan produk, dengan setiap anggota mengambil peran dalam perencanaan, pengembangan, dan pelaksanaan proyek. Tim Agile tidak mengikuti perencanaan yang kaku karena perencanaan bersifat iteratif dan fleksibel, memungkinkan penyesuaian hasil akhir sesuai kebutuhan [19].

2.2.6. Scrum

Scrum adalah metode *Agile* dan kerangka kerja pengelolaan proyek dengan variasi yang signifikan dalam praktik-praktik seperti perincian kerja, estimasi, penentuan prioritas, penugasan, serta peran tim dan artefak [21]. *Scrum* menekankan pada fitur *incremental delivery* dan dirancang agar fleksibel sehingga pelanggan dapat berubah pikiran selama pengembangan tanpa mengganggu upaya pengembangan [22]. *Scrum* dalam praktiknya sangat bervariasi yang dapat berbeda dari buku *Scrum Guide*. Variasi *Scrum* dalam praktiknya, memungkinkan tim perangkat lunak untuk menyesuaikannya dengan kebutuhan mereka [21].



Gambar 2.1 Tahapan *Scrum*

Sumber gambar: <https://www.scrum.org/resources/what-scrum-module>

Seperti yang dapat dilihat pada gambar 2.1 tentang tahapan *Scrum* terdapat beberapa tahapan yang terdiri dari:

a. *Backlog*

Dalam praktik Pengembangan Perangkat Lunak *Agile*, *backlog* merupakan sebuah artefak yang menentukan cakupan dari pekerjaan yang akan dilakukan, mirip dengan daftar persyaratan fungsional. *Backlog* adalah daftar komprehensif dari semua fitur yang membentuk proyek, dan harus diuraikan dan dijelaskan secara rinci sejak awal proyek. Daftar *backlog* dalam pengembangan perangkat lunak diperoleh melalui proses pengumpulan

kebutuhan dari pemangku kepentingan, analisis prioritas fitur, estimasi usaha pengembangan, serta penyesuaian berkelanjutan berdasarkan umpan balik pengguna dan perubahan kebutuhan bisnis. Setiap fitur harus diidentifikasi dan diurutkan berdasarkan prioritas eksekusi. Umumnya, pengembang perangkat lunak akan menambahkan fitur baru, memodifikasi fitur yang ada, dan memperbaiki *bug* berdasarkan item yang ada di bagian atas *backlog*. *Product backlog* berperan penting dalam mengarahkan aktivitas pengembangan, memungkinkan tim untuk mengarahkan seluruh proses pengembangan proyek. Dalam konteks kerangka kerja *Scrum*, *product backlog* dibagi menjadi dua artefak: *product backlog* dan *sprint backlog*.

Sprint Backlog merupakan subbagian dari *product backlog* yang akan diprioritaskan untuk pengembangan dalam iterasi berikutnya. Subbagian ini perlu dijelaskan dengan jelas agar tim pengembang memahami tujuan *Sprint* tersebut. Pada tahap ini, cerita pengguna sedang disiapkan, dengan individu yang bertanggung jawab atas pengembangan cerita tersebut, estimasi waktu penyelesaian, serta langkah-langkah dan tugas yang diperlukan untuk menyelesaikannya [23].

b. *Sprint Planning*

Pada tahapan *Sprint Planning*, tim *scrum* melakukan pertemuan untuk mengevaluasi daftar item yang terdapat dalam *product backlog*. Tim *scrum* membahas setiap fitur yang akan dikembangkan oleh masing-masing anggota tim beserta perkiraan waktu yang diperlukan untuk menyelesaikannya [24].

c. *Daily Scrum*

Semua sesi *sprint* dijalankan dengan transparansi penuh. Tim menyelenggarakan pertemuan harian untuk meninjau kemajuan, menerima umpan balik, dan menyoroiti potensi masalah yang muncul. Komunikasi ini dilakukan baik di dalam tim inti maupun secara terpisah dengan *Product Owner* untuk memastikan pencapaian tujuan secara menyeluruh [25].

d. *Sprint Review*

Sprint Review terjadi pada akhir *sprint* dan bertujuan untuk meninjau semua tugas yang telah dilakukan selama periode *sprint*, serta yang telah diselesaikan. Hasil dari peninjauan ini kemudian disajikan kepada *stakeholder* yang bertanggung jawab untuk mendapatkan persetujuan. Jika tugas-tugas tersebut disetujui, maka akan dihitung selesai. Namun, jika tidak disetujui, tugas-tugas akan kembali dipindahkan ke *backlog*. Hal ini disebabkan karena tim *developer* perlu melakukan peninjauan ulang, namun kali ini mereka memiliki masukan dari pelanggan serta informasi tambahan yang akan membantu memastikan bahwa hanya satu iterasi lagi yang diperlukan [26].

e. *Sprint Retrospective*

Sprint Retrospective merupakan kegiatan yang dilakukan pada akhir setiap *Sprint*, setelah *Sprint Review*. Semua anggota tim diwajibkan untuk hadir dalam sesi retrospektif dan berperan aktif dalam diskusi. Retrospektif ini menjadi kesempatan bagi tim untuk merefleksikan sebuah iterasi, mengidentifikasi pencapaian dan kegagalan, serta untuk mengaitkan pengalaman terkait dengan individu, proses pengembangan, praktik teknis, dan peralatan yang digunakan. Tujuan utamanya adalah untuk merencanakan perbaikan yang akan diterapkan pada iterasi berikutnya, sehingga tim dapat terus meningkatkan kinerjanya secara berkelanjutan [27].

2.2.7. *Scrum Metric*

Scrum Metric adalah berbagai poin data dan statistik yang dicatat, dilacak dan dianalisis oleh tim *scrum* untuk membantu meningkatkan efektivitas, kinerja dan efisiensi tim. Sebuah tim *scrum* untuk menginformasikan pengambilan keputusan mereka. Hal ini membantu menciptakan tujuan yang lebih jelas dan memfasilitasi manajemen *sprint* yang lebih baik. Beberapa metrik yang dapat membantu tim *scrum* yaitu *team velocity*, *effort estimation*, *variance*, *sprint burndown*, *epic and release burndown*, *retrospective process improvement*, *defect leakage* dan *scope change* [28].

2.2.8. *Story Point*

Story Point adalah satuan relatif yang menggambarkan kompleksitas dan usaha yang dibutuhkan untuk menyelesaikan sebuah *user story*. Setiap *user story* dalam *backlog* diberi nilai *story point*. Tidak ada formula tetap untuk menentukan ukuran sebuah *user story*. Sebaliknya, estimasi *story point* merupakan gabungan dari jumlah usaha yang terlibat dalam mengembangkan fitur, kompleksitasnya, dan risiko yang melekat dalam proses pengembangannya. Pendekatan umum untuk menentukan nilai *story point* dari sebuah *user story* adalah dengan memilih salah satu *user story* terkecil dalam *backlog* dan memberinya satu *story point*. Kemudian, *user story* yang lebih kompleks dan lebih luas mendapatkan lebih banyak poin relatif terhadap ukurannya [29].

