

BAB II

DASAR TEORI

2.1 Perangkat Lunak

Perangkat lunak adalah sebuah program komputer yang menyediakan layanan interaksi antara pengguna dan perangkat keras (Pfleeger, 2006). Perangkat lunak dibangun melalui proses rekayasa yang terstruktur dengan berbagai metode tergantung kebutuhan fungsionalitas dan hasil akhir yang diharapkan.

Perangkat lunak memiliki dua peran: sebagai sebuah produk dan sebagai pengontrol pembuatan sebuah produk (Pressman, 2012). Sebagai sebuah produk, perangkat lunak mengantarkan potensi perhitungan yang dibangun serta mentransformasikan informasi yang diproduksi, diatur, diperoleh, dimodifikasi, ditampilkan atau dipancarkan. Sedangkan peran sebagai pengontrol yang dipakai untuk mengantarkan produk, perangkat lunak berlaku sebagai dasar untuk kontrol komputer (sistem operasi), komunikasi informasi (jaringan), dan penciptaan serta kontrol dari program-program lain (peranti dan lingkungan perangkat lunak).

2.2 Agile Development

Agile development adalah salah satu metode pengembangan perangkat lunak yang mengedepankan interaksi anggota tim dan kolaborasi dengan klien daripada proses dan jenis perangkat yang digunakan (Martin & Martin, 2006). Metode *agile* cocok diterapkan pada perangkat lunak yang diharuskan responsif serta toleran terhadap perubahan kebutuhan yang cepat.

Prioritas utama penerapan *agile* pada sebuah perangkat lunak adalah kepuasan pelanggan melalui produk perangkat lunak secara berkelanjutan. Salah satu prinsip dari *agile* untuk mendukung hal ini salah satunya adalah proses pemaparan hasil yang dilakukan dalam jangka waktu dua sampai empat minggu, dengan preferensi pada skala waktu yang lebih pendek (Martin & Martin, 2006).

2.3 Scrum

Scrum dikembangkan Jeff Sutherland pada 1993 yang dengan tujuan sebagai metode pengembangan dan pengelolaan yang mengikuti prinsip *agile* (Pham, 2011). Pengembangan *scrum* selanjutnya dilakukan oleh Schwaber dan Beedle. *Scrum* sendiri mempunyai proses yang kompleks karena adanya banyak faktor yang memengaruhi hasil akhir. (Majeed, 2012)

Scrum yang terdiri dari *scrum team* dan peran-peran yang diperlukan, acara (*event*), artefak (*artifact*), dan aturan main (Pressman, 2012). Jantung dari pelaksanaan *scrum* adalah *sprint*. *Sprint* merupakan batasan waktu yang diberikan untuk menyelesaikan masalah. Pada praktiknya, Paperlust menetapkan waktu selama dua minggu untuk setiap satu *sprint* yang berjalan. Tahapan *sprint* yang berjalan di antaranya: *Sprint Planning*, *Daily Scrum*, *Sprint Review*, dan *Sprint Retrospective*. Proses implementasi kerangka kerja ini memiliki *key practice* yaitu: (1) *Scrum* memungkinkan pengerjaan dan pengumpulan kebutuhan, perancangan arsitektur dan antarmuka secara bersamaan, (2) Fokus pada *sprint*, pengkajian hasil, dan jadwal pengerjaan, (3) Fokus pada jadwal yang telah disepakati, (4) Bekerja sesuai dengan *sprint* secara konsisten dan terstruktur, (5) Semua pekerjaan ditandai sebagai *product backlog*, (6) *Product backlog* dasar melakukan *sprint* dan tim harus dapat memutuskan skala prioritas terhadap daftar *product backlog* yang telah disusun sebelumnya, (7) Melakukan pertemuan setiap hari, (8) *Scrum master* bertanggung jawab menerima dan mengevaluasi hasil *sprint* (Martin & Martin, 2006).

2.3.1 Peran-peran dalam tim *scrum*

Scrum mempunyai tatanan nilai seperti: komitmen, keberanian, fokus, keterbukaan, dan respek yang apabila diwujudkan dan hidup di dalam *scrum team*, maka pilar-pilar *scrum* seperti: transparansi, inspeksi dan adaptasi akan menjadi hidup dan membangun rasa percaya satu sama lain. Peran-peran yang mendukung tercapainya keberhasilan penggunaan *scrum* dalam penerapan tatanan nilai tersebut adalah sebagai berikut:

a. *Product Owner*

Product owner bertanggung jawab untuk menentukan spesifikasi dan memaksimalkan nilai bisnis perangkat lunak yang akan dibangun. *Product owner* juga bertanggung jawab dalam pengelolaan *product backlog* atau komponen minimal yang harus dilakukan oleh *development team* yang termasuk diantaranya sebagai berikut (S. & S., 2017):

- Mengurutkan *product backlog item* guna mencapai tujuan dengan cara terbaik.
- Mengoptimalkan nilai bisnis dari pekerjaan yang dilakukan *development team*.
- Memastikan agar *product backlog* dapat dilihat, transparan, dan jelas untuk semua pihak, dan menampilkan apa yang akan dikerjakan selanjutnya oleh *scrum team*.
- Memastikan *development team* memahami *product backlog* hingga batas tertentu.

b. *Scrum Master*

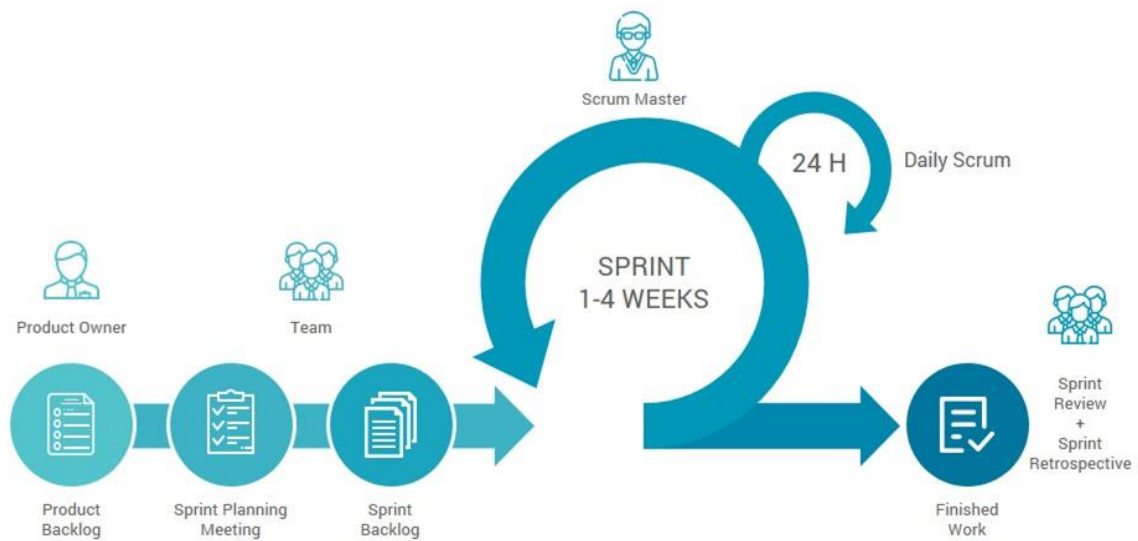
Scrum master adalah pemimpin yang mengatur keberlangsungan kerangka kerja *scrum*. *Scrum master* bertanggung jawab dalam memastikan semua orang di proyek mengimplementasikan *scrum* dengan baik dan benar. Selain itu, peran ini juga membantu *product owner* untuk menemukan teknik yang paling efektif untuk mengelola *product backlog* yang telah ditentukan sebelumnya serta memfasilitasi acara-acara *scrum* bila diminta atau dibutuhkan (S. & S., 2017).

c. *Development Team*

Development team terdiri dari para ahli profesional seperti: analis bisnis, analis sistem, pengembang, penguji, dan lainnya (S. & S., 2017). Mereka bekerja untuk menghantarkan hasil pengembangan yang memiliki potensi untuk dirilis di setiap akhir *sprint*. tim ini juga bertanggung jawab untuk menyelesaikan *product backlog* yang telah didefinisikan oleh *product owner*, dimana anggota tim bertanggung jawab untuk setiap *backlog* yang telah dibagi serta mampu mengetahui apa yang harus dilakukan selanjutnya.

2.3.2 Alur kerja *scrum*

Jantung dari pelaksanaan *scrum* adalah *sprint*, yang mana adalah sebuah batasan waktu dengan durasi satu hingga empat minggu. Durasi setiap daur hidup *sprint* yang berjalan selalu konsisten dan selalu dimulai kembali setelah *sprint* sebelumnya telah dinyatakan selesai. *Sprint* dianggap selesai apabila telah menghasilkan produk (*deliverable product*) yang sesuai dengan *acceptance criteria* yang telah ditetapkan dan disepakati sebelumnya. Alur kerja *sprint* dapat dilihat pada Gambar 2.1 dengan tahapan sebagai berikut :



Gambar 2.1 Alur kerangka kerja *scrum*

1. *Product Backlog*

Product Backlog adalah daftar prioritas dari kebutuhan proyek berupa: peningkatan, perbaikan dari fitur yang akan dikerjakan serta menjadi sumber kebutuhan untuk semua perubahan yang perlu diberlakukan terhadap produk. Proses pendeskripsian item *product backlog* dilakukan *product owner*, termasuk isi, ketersediaan dan urutannya. *Product backlog* bersifat dinamis dan berubah secara konstan agar produk menjadi layak, kompetitif dan bermanfaat. *Product backlog* berevolusi seiring dengan perkembangan produk dan lingkungan dimana produk tersebut digunakan. Selama produk masih ada, *product backlog* juga akan selalu ada.

2. *Sprint Planning*

Sebelum *sprint* berjalan, pekerjaan direncanakan terlebih dahulu pada tahap *sprint planning*. Perencanaan ini dilakukan secara kolaboratif oleh seluruh anggota tim *scrum*. *Sprint planning* dapat digambarkan sebagai tahap analisa yang berfokus pada masalah yang akan diselesaikan. Durasi *sprint planning* dapat berbeda-beda berdasarkan seberapa kompleks dan besar masalah yang akan diselesaikan. Pada tahap ini, *scrum master* harus memastikan semua anggota dapat memahami masalah serta metode penyelesaiannya (S. & S., 2017). *Sprint planning* menjawab pertanyaan-pertanyaan berikut:

- Apa yang dapat dihantarkan ke dalam *increment* dari *sprint* ini hingga mencapai *sprint goal*?
- Bagaimana *development team* bisa menyelesaikan pekerjaan yang dibutuhkan untuk menghantarkan *increment*?

3. *Sprint Backlog*

Sprint backlog merupakan daftar *product backlog item* yang terpilih untuk ditambah pada *sprint* perencanaan untuk mencapai *sprint goal* (S. & S., 2017). *Sprint backlog* adalah prakiraan dari *development team* mengenai fungsionalitas yang akan masuk ke dalam daftar pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitasnya menjadi *increment* yang “selesai”. Untuk memastikan adanya peningkatan berkelanjutan, *sprint backlog* berisi setidaknya satu peningkatan proses dengan prioritas tertinggi dari hasil pertemuan *Retrospective Sprint* sebelumnya (Permana, 2015).

4. *Daily Meeting*

Daily meeting umumnya berdurasi selama kurang lebih 15 menit untuk setiap pertemuannya, namun pada praktiknya, durasi dapat disesuaikan tergantung jumlah anggota dan jenis pembahasan. Struktur dari pertemuan ini fleksibel dan dapat diadakan lewat berbagai macam cara selama pertemuan ini fokus terhadap kemajuan menuju *sprint goal* (S. & S., 2017). *Daily Meeting* berfungsi untuk mengoptimalkan kolaborasi dan performa dari tim dengan melakukan inspeksi pada pekerjaan yang dilakukan semenjak *daily meeting* sebelumnya (S. & S., 2017). Struktur dari pertemuan ini bisa berbeda-beda dan bisa diadakan lewat berbagai macam cara selama pertemuan ini fokus terhadap kemajuan menuju *sprint goal*.

Komponen pembahasan pada *daily meeting developer team* di tahap ini di antaranya: hal yang telah dilakukan sejak pertemuan terakhir tim, kendala yang ditemui selama pengerjaan, serta rencana yang telah disusun untuk mencapai sesuatu sebelum rapat tim berikutnya. Tujuan dari tahap ini adalah dapat mengetahui sisa pekerjaan yang perlu dikejar selama sisa waktu *sprint*.

5. *Sprint Review*

Sprint review dilaksanakan untuk menginspeksi *increment* dan meninjau apa saja pekerjaan yang telah dan belum diselesaikan. Inspeksi dan peninjauan tidak hanya

dilakukan pada hasil, proses bekerja tim juga dinilai sehingga dapat ditentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *sprint* berikutnya (S. & S., 2017). *Sprint* dianggap selesai apabila telah menghasilkan produk (*deliverable product*) yang sesuai dengan *acceptance criteria* yang telah ditetapkan dan disepakati sebelumnya.

Sprint review dilaksanakan setiap akhir masa *sprint*. Pertemuan ini paling lama diselenggarakan selama empat jam untuk *sprint* berdurasi satu bulan. Untuk *sprint* yang lebih singkat, durasi acara ini biasanya lebih singkat. Hasil dari *sprint review* adalah *product backlog* yang sudah direvisi berdasarkan kebutuhan *product backlog item* yang akan diimplementasikan di *sprint* berikutnya.

6. *Sprint Retrospective*

Sprint retrospective adalah sebuah kesempatan bagi tim *scrum* untuk menginspeksi dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *sprint* berikutnya (S. & S., 2017). *Sprint retrospective* terselenggara sebelum *sprint planning* berikutnya. *Scrum master* memastikan acara ini terselenggara dan setiap peserta memahami tujuannya.

Sebelum tahap ini berakhir, tim *scrum* harus menyepakati peningkatan yang akan mereka implementasikan di *sprint* berikutnya (S. & S., 2017). Mengimplementasikan peningkatan di *sprint* berikutnya adalah bentuk adaptasi terhadap tim *scrum* itu sendiri. Meskipun peningkatan dapat dilakukan kapanpun di sepanjang *sprint*, *sprint retrospective* adalah sebuah kesempatan formal yang fokus pada inspeksi dan adaptasi (Permana, 2015).

2.3.3 Tata nilai *scrum*

Scrum memiliki beberapa nilai yang apabila diwujudkan dan di hidup dalam *scrum team*, maka pilar-pilar *scrum* seperti: transparansi, inspeksi, dan adaptasi akan menjadi hidup dan membangun rasa percaya satu sama lain. Selain itu nilai yang ada juga menjadi dasar pelaksanaan *scrum* itu sendiri. Nilai-nilai tersebut di antaranya sebagai berikut: (S. & S., 2017)

a. Komitmen

Setiap tim dalam *scrum* mengarahkan dan mengawasi dirinya sendiri. Setiap anggota tim harus mengerti tanggung jawabnya dan kewajiban masing-masing yang telah disepakati sebelumnya.

b. Fokus

Kerangka kerja *scrum* membatasi distraksi sehingga setiap anggota tim akan lebih berkonsentrasi pada pekerjaan yang harus diselesaikan atau target masing-masing.

c. Keterbukaan

Setiap anggota harus terbuka terhadap segala sesuatu yang dikerjakannya kepada anggota tim maupun orang yang berada diluar tim. Keterbukaan ini termasuk apa yang sedang dan akan dikerjakan serta permasalahan yang dialami. Setiap anggota juga diharapkan untuk dengan sukarela meminta bantuan orang lain, begitu sebaliknya.

d. Respek

Setiap anggota tim harus saling menghormati dan menghargai pekerjaan anggota tim lain. Setiap individu memiliki tingkat kemampuan dan kemandirian berbeda-beda.

e. *Courage*

Setiap anggota memiliki keinginan untuk berubah, menerima bahwa pendapat atau ide yang diajukan kurang sesuai dengan tujuan atau target tim.

