BAB II

TINJAUAN PUSTAKA

2.1 *Scrum*

Scrum (kb): Sebuah kerangka kerja yang dapat menyelesaikan permasalahan kompleks yang sering berubah-ubah, dengan tujuan untuk menghasilkan produk (*output*) dengan nilai setinggi mungkin secara kreatif dan produktif. *Scrum* bersifat ringan, mudah dipahami, tetapi sulit untuk dikuasai

Scrum adalah kerangka kerja yang telah digunakan untuk mengelola pengembangan produk kompleks sejak awal tahun 1990-an. *Scrum* bukanlah sebuah proses ataupun teknik untuk mengembangkan produk, tetapi dengan kerangka kerja ini Tim *Scrum* dapat memasukkan beragam proses dan teknik. *Scrum* akan mengekspos pergerakan efektifitas manajemen produk (*output*) dan praktik pengembangan yang sedang Tim *Scrum* jalani, dengan begitu Tim Scrum dapat melakukan peningkatan atau perbaikan.

Berikut beberapa keunggulan kerangka kerja *Scrum*, yaitu :

a. Mudah menyesuaikan diri terhadap perubahan target yang akan dicapai.
b. Dengan menerapkan *Scrum*, tim besar dapat dipecah menjadi tim-tim kecil yang disebut Tim *Scrum*, sehingga memperlancar komunikasi, mengurangi biaya, dan saling memberdayakan satu sama lain.

c. Dokumentasi selama proses pelaksanaan tugas (*task*) dilakukan secara terus menerus pada setiap iterasi.
d. *Scrum* dapat mengidentifikasi status *task* pada setiap iterasi.

Karena kerangka kerja *Scrum* ini sangat fleksibel terhadap perubahan, Tim Penjualan harus bersedia dan selalu siap untuk menerima perubahan yang terjadi di dalam proses pengembangan kapan saja. Dan ini menjadi kelemahan dari kerangka kerja *Scrum*.

Scrum bukanlah sebuah metodologi, tetapi sebuah *framework* atau kerangka kerja dimana kurang bersifat preskriptif. Berbeda dengan metodologi yang lebih bersifat detail dalam mengatur personil di dalamnya. Kerangka kerja kebalikannya, hanya menjadi kerangka yang memberikan banyak ruang untuk improvisasi. Oleh karena itu penerapannya pun bisa dilakukan terhadap berbagai bidang pekerjaan.

2.1.1 Teori *Scrum*

Scrum didasari oleh teori kontrol proses empiris, atau dengan kata lain, empirisme. Empirisme menekankan bahwa pengetahuan berasal dari pengalaman dan pembuatan keputusan didasari oleh pengetahuan yang telah dimiliki hingga saat ini. *Scrum* menggunakan pendekatan berkala (*iterative*) dan bertahap (*incremental*) untuk meningkatkan prediktabilitas dan mengendalikan resiko.

Ada tiga pilar dari setiap implementasi kontrol proses empiris yakni: transparansi, inspeksi dan adaptasi.

a. Transparansi

Aspek-aspek penting dari proses yang berjalan harus dapat ditinjau oleh pihak-pihak yang bertanggungjawab terhadap hasilnya. Transparansi mengharuskan aspek-aspek tersebut didefinisikan dengan standar yang sama, sehingga semua peninjau memiliki pemahaman yang sama mengenai apa yang sedang ditinjau.

Sebagai contoh,

1) Istilah-istilah pada proses yang sedang dijalankan harus dapat dimengerti oleh semua pihak; dan,
2) Setiap pihak yang bekerja dan pihak yang menerima hasil pekerjaan harus memiliki pemahaman yang sama mengenai arti kata “Selesai”.

b. Inspeksi

Pengguna *Scrum* harus secara rutin meninjau artefak *Scrum* (*Scrum Board*) beserta perkembangannya agar perubahan dapat terdeteksi. Peninjauan sebaiknya tidak terlalu sering sehingga dapat menyebabkan terhambatnya pekerjaan. Peninjauan paling bermanfaat jika dilakukan secara rutin, oleh peninjau yang kompeten, pada saat pekerjaan berjalan.

c. Adaptasi

Apabila peninjau mendapatkan satu atau lebih aspek dari proses mengalami deviasi di luar batasan yang dapat diterima, hingga hasil akhirnya menjadi tidak dapat diterima, maka proses atau materi yang diolah harus diatur

ulang. Pengaturan ulang harus dibuat sesegera mungkin untuk meminimalisir deviasi yang lebih jauh.

Scrum menyediakan empat acara formal, yang dibungkus di dalam *Sprint*, untuk inspeksi dan adaptasi, sebagaimana dijelaskan di bagian Acara Scrum di dalam dokumen ini:

1) *Sprint Planning*
2) *Daily Scrum*
3) *Sprint Review*
4) *Sprint Retrospective*

2.1.2 Tim *Scrum*

Tim *Scrum* terdiri dari *Product Owner*, Tim Pengembang, dan *Scrum Master*. Tim *Scrum* mengatur diri mereka sendiri dan berfungsi antar-lintas. Tim yang mengatur dirinya sendiri menentukan cara terbaik untuk menyelesaikan pekerjaannya. Tim yang berfungsi antar-lintas memiliki semua kompetensi yang dibutuhkan untuk menyelesaikan pekerjaan, tanpa mengandalkan pihak lain yang berada di luar anggota tim. Model tim di dalam *Scrum* dirancang sedemikian rupa untuk menjaga fleksibilitas, kreatifitas dan produktifitas, demi mencapai hasil terbaik.

Tim *Scrum* menyampaikan hasil pekerjaan yang sudah dianggap selesai secara berkala dan bertahap, ini dilakukan agar Tim *Scrum* bisa mendapatkan masukan sejak dini dari setiap hasil pekerjaan yang disampaikan.

2.1.2.1 *Product Owner*

Product Owner bertanggungjawab untuk meninjau produk yang disampaikan, dan memastikan untuk mendapatkan nilai produk yang maksimal dari Tim Pengembang. *Product Owner* merupakan satu-satunya orang yang bertanggungjawab untuk mengelola *Product Backlog*. Pengelolaan *Product Backlog* mencakup:

a. Mendeskripsikan dengan jelas *item Product Backlog*;
b. Mengurutkan item di dalam *Product Backlog* untuk mencapai tujuan dan misi dengan cara terbaik;
c. Melakukan *review* terhadap hasil pekerjaan Tim Pengembang untuk mendapatkan hasil yang optimal;
d. Memastikan *Product Backlog* transparan, jelas, dan dapat dilihat semua pihak, dan mendefinisikan apa yang akan dikerjakan oleh Tim *Scrum* selanjutnya;
e. Memastikan Tim Pengembang dapat memahami *task* yang didaftarkan dalam *Product Backlog*, termasuk batasan yang diperlukan.

Product Owner dapat saja mengerjakan pekerjaan-pekerjaan di atas, atau menyerahkan pengerjaannya kepada Tim Pengembang, namun satu-satunya pihak yang bertanggungjawab tetaplah *Product Owner*.

Product Owner adalah satu orang dan bukan berupa sebuah komite. *Product Owner* dapat menerima aspirasi dari komite ke dalam *Product Backlog*, namun perubahan prioritas pada *Product Backlog* tetap menjadi wewenang *Product Owner*.

Agar *Product Owner* berhasil menjalankan tugasnya, seluruh organisasi harus menghormati setiap keputusan yang dibuat. Keputusan dari *Product Owner* ini dapat dilihat dari isi dan urutan *Product Backlog*. Tidak ada seorang pun yang dapat memerintah Tim Pengembang untuk mengerjakan kebutuhan lain selain *Product Owner*. Dan Tim Pengembang pun tidak diperbolehkan untuk melakukan apa yang diperintahkan oleh pihak lain selain *Product Owner*.

2.1.2.2 Tim Pengembang

Tim Pengembang terdiri dari para profesional yang bekerja untuk menyelesaikan *task* atau menghasilkan potongan produk, atau output (selanjutnya disebut Inkremen), yang akan dirilis pada setiap akhir *Sprint*. Hanya anggota Tim Pengembang yang mengembangkan Inkremen ini.

Tim Pengembang dibentuk dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang ada di dalam tim akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan.

Tim Pengembang memiliki karakteristik sebagai berikut:

a. Mengatur dirinya sendiri, sehingga tidak ada satu orang pun (bahkan *Scrum Master*) yang diperbolehkan memerintah Tim Pengembang, perihal bagaimana cara mengubah *Product Backlog* menjadi Inkremen yang berpotensi untuk dirilis;
b. Tim Pengembang berfungsi antar-lintas, dan tim ini sebaiknya memiliki semua keahlian yang dibutuhkan untuk menghasilkan produk;
c. *Scrum* tidak mengenal adanya jabatan tertentu untuk anggota Tim Pengembang, apapun pekerjaan yang dikerjakan oleh masing-masing anggota tim tersebut, tetap statusnya sebagai Pengembang.
d. Tim Pengembang tidak mengenal adanya sub-tim yang dikhususkan untuk bidang tertentu seperti pengujian atau analisa bisnis.
e. Anggota Tim Pengembang boleh memiliki spesialisasi keahlian dan fokus di satu area tertentu, namun akuntabilitas dari hasil pekerjaan secara keseluruhan adalah milik Tim Pengembang.

Ukuran Tim Pengembang

Agar dapat berkoordinasi dengan cepat, Tim Pengembang sebaiknya dibentuk dalam jumlah anggota kecil, tetapi jumlah anggota tersebut sudah cukup besar untuk dapat menyelesaikan pekerjaan dalam satu *Sprint*. Jumlah anggota tim yang kurang dari tiga orang akan mengurangi interaksi dan akan menyebabkan produktifitas yang rendah. Tim Pengembang yang kecil kemungkinan akan

mengalami kekurangan keahlian tertentu pada saat *Sprint* berjalan, yang pada akhirnya menyebabkan Tim Pengembang tidak dapat menghasilkan Inkremen yang berpotensi untuk dirilis. Tim Pengembang dengan jumlah anggota lebih dari sembilan orang membutuhan terlalu banyak koordinasi. Tim Pengembang dengan jumlah anggota tim yang banyak, akan menimbulkan terlalu banyak kompleksitas bagi proses yang berbasiskan empirisme. *Product Owner* dan *Scrum Master* tidak termasuk dalam hitungan, kecuali mereka juga turut ikut mengerjakan pekerjaan yang ada di *Sprint Backlog*.

2.1.2.3 *Scrum Master*

Scrum Master bertanggung jawab untuk memastikan *Scrum* telah dipahami dan dilaksanakan. *Scrum Master* melakukannya dengan memastikan Tim *Scrum* mengikuti teori, praktik, dan aturan main *Scrum*.

Scrum Master adalah seorang pemimpin yang melayani Tim Scrum. *Scrum Master* membantu pihak di luar Tim *Scrum*, untuk memahami apakah interaksi mereka dengan Tim *Scrum* berjalan dengan baik atau tidak. *Scrum Master* membantu setiap pihak untuk merubah interaksi-interaksi yang tidak bermanfaat sehingga bisa memaksimalkan nilai yang dihasilkan oleh Tim *Scrum*.

Layanan Scrum Master kepada Product Owner

Scrum Master melayani *Product Owner* dengan berbagai cara yang mencakup:

a. Mencari teknik yang paling efektif untuk mengelola *Product Backlog*;

b. Membantu Tim *Scrum* untuk memahami pentingnya *Product Backlog* dengan memberikan informasi yang singkat dan jelas;

c. Memahami bagaimana perencanaan produk pada lingkungan yang didasarkan empirisme;

d. Memastikan *Product Owner* tahu bagaimana mengelola *Product Backlog* guna memaksimalkan nilai dari produk;

e. Memahami dan mempraktikkan *agility*; dan,

f. Memfasilitasi acara-acara dalam Scrum bila dipanggil dan dibutuhkan.

Layanan Scrum Master kepada Tim Pengembang

Scrum Master melayani Tim Pengembang lewat berbagai cara yang mencakup:

a. Membimbing Tim Pengembang untuk dapat mengatur dirinya sendiri dan berfungsi antar-lintas;

b. Membantu Tim Pengembang untuk membuat produk bernilai tinggi;

c. Menghilangkan hambatan-hambatan yang dialami oleh Tim Pengembang;

d. Memfasilitasi acara-acara dalam *Scrum* bila dipanggil dan dibutuhkan; dan,

e. Membimbing Tim Pengembang dalam suasana organisasi di mana *Scrum* belum sepenuhnya diterapkan dan dipahami.

Layanan Scrum Master Service kepada Organisasi

Scrum Master melayani organisasi tempat dia berada lewat berbagai cara yang mencakup:

a. Memimpin dan membimbing organisasi dalam penerapan *Scrum*;

b. Merencanakan implementasi *Scrum* di dalam organisasi;

c. Membantu setiap pegawai dan *stakeholder* dalam memahami dan menggunakan *Scrum* dan pengembangan produk dengan metoda empiris;

d. Membuat perubahan yang dapat meningkatkan produktifitas di dalam Tim *Scrum*; dan,

e. Bekerja bersama dengan *Scrum Master* lainnya guna meningkatkan efektifitas dari pengaplikasian *Scrum* di dalam organisasi.

2.1.3 Acara-acara Scrum

Acara-acara wajib dalam *Scrum* dibuat untuk menciptakan sebuah kesinambungan dan mengurangi adanya acara-acara lain yang tidak tercantum di dalam *Scrum*. Setiap acara di dalam *Scrum* memiliki batasan waktu, yang artinya selalu memiliki durasi maksimum. Pada saat *Sprint* dimulai, durasinya tetap dan

tidak dapat diperpendek maupun diperpanjang. Acara-acara lainnya dapat diakhiri saat tujuan dari acara tersebut telah tercapai, hal ini dilakukan untuk efektifitas dan efisiensi waktu sepanjang proses pengerjaan *sprint*.

Setiap acara dalam *Scrum* merupakan kesempatan formal untuk meninjau dan merubah sesuatu. Acara-acara ini dirancang secara khusus untuk menciptakan transparansi dan peninjauan sampai ke tingkatan kritis. Tidak adanya pelaksanaan salah satu acara ini akan mengurangi transparansi dan menghilangkan kesempatan untuk meninjau dan membuat perubahan.

2.1.3.1 *Sprint*

Jantung dari *Scrum* adalah *Sprint*. *Sprint* memiliki batasan waktu maksimal selama satu bulan atau kurang. *Sprint* biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk. *Sprint* yang baru, langsung dimulai setelah *Sprint* yang sebelumnya berakhir.

Aktifitas pertemuan *Sprint* terdiri dari *Sprint Planning*, *Daily Scrum*, pengembangan, *Sprint Review*, dan *Sprint Retrospective*.

Berikut hal-hal yang perlu diperhatikan pada saat *Sprint* dijalankan, diantaranya adalah:

a. Tidak boleh ada perubahan yang dapat membahayakan tercapainya *Sprint Goal*;
b. Kualitas dari *Sprint Goal* tidak boleh menurun;

c. *Scope* dapat diklarifikasikan dan dinegosiasikan ulang diantara *Product Owner* dan Tim Pengembang seiring dengan bertambahnya pengetahuan.

Setiap *Sprint* dapat dikatakan sebagai sebuah proyek dengan batasan waktu tidak lebih dari satu bulan. Sama halnya dengan proyek, *Sprint* digunakan untuk menyelesaikan sesuatu. Setiap *Sprint* memiliki definisi mengenai apa yang akan dikembangkan, sebuah desain dan perencanaan yang fleksibel yang akan membimbing pengembangan, pekerjaan yang akan dilakukan dan hasil dari produk.

Sprint dibatasi maksimal pada satu bulan kalender. Bila jangka waktu *Sprint* terlalu panjang, maka definisi mengenai apa yang akan dibangun dapat berubah, kompleksitas dapat meningkat, dan resiko dapat bertambah. *Sprint* meningkatkan prediktabilitas karena adanya peninjauan dan pengadaptasian terhadap perkembangan, setidaknya setiap satu bulan sekali. *Sprint* juga membatasi resiko biaya hingga satu bulan saja.

2.1.3.2 Membatalkan *Sprint*

Sprint dapat dibatalkan sebelum batasan waktu *Sprint* selesai. Hanya *Product Owner* yang dapat membatalkan *Sprint*, walaupun keputusan yang dia buat mungkin saja dipengaruhi oleh para *stakeholder*, Tim Pengembang, ataupun *Scrum Master*.

Sprint dibatalkan apabila *Sprint Goal* sudah tidak sesuai harapan semula. Hal ini dapat terjadi apabila arahan perusahaan berubah, atau bila kondisi pasar atau teknologi berubah. Pada umumnya, *Sprint* harus dibatalkan apabila *Sprint* menjadi

tidak masuk akal lagi apabila dilanjutkan. Namun karena batasan waktu *Sprint* yang begitu singkat, pembatalan biasanya jarang terjadi.

Ketika *Sprint* dibatalkan, item *Product Backlog* yang progresnya sudah "Selesai" ditinjau kembali. Apabila hasil pekerjaan dari *Product Backlog* tersebut berpotensi untuk dirilis, biasanya *Product Owner* akan menerima hasil pekerjaan tersebut. Semua item *Product Backlog* yang tidak selesai, diestimasi dan dimasukkan kembali ke dalam *Product Backlog*. Pekerjaan dalam *Product Backlog* yang sudah selesai tersebut akan mengalami depresiasi nilai dengan cepat dan harus diestimasi ulang sesering mungkin.

Pembatalan *Sprint* membuang banyak tenaga, karena semua orang harus menyusun kembali kelompoknya dalam *Sprint Planning* baru untuk memulai *Sprint* baru. Pembatalan *Sprint* sering menyebabkan trauma bagi Tim *Scrum*, dan sangat jarang terjadi.

2.1.3.3 *Sprint Planning*

Pekerjaan yang akan dilaksanakan di dalam *Sprint* direncanakan pada saat *Sprint Planning*. Perencanaan ini dibuat secara kolaboratif oleh seluruh anggota Tim *Scrum*.

Sprint Planning dibatasi maksimum delapan jam untuk *Sprint* yang berdurasi maksimal satu bulan. Untuk *Sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa acara ini dilaksanakan

dan setiap hadirin memahami tujuannya. *Scrum Master* mengedukasi Tim *Scrum* untuk melaksanakannya dalam batasan waktu yang telah ditentukan.

Sprint Planning harus dapat menjawab pertanyaan-pertanyaan berikut:

a. Apa *goal* dari *Sprint*?
b. Apa yang dapat dihasilkan di dalam Inkremen sebagai hasil dari *Sprint* yang sedang berjalan?
c. Apa yang perlu dilakukan untuk dapat menghantarkan Inkremen tersebut?

Topik Pertama: Apa yang dapat dilakukan di dalam *Sprint* ini?

Tim Pengembang bekerja untuk memperkirakan fungsionalitas yang akan dikembangkan pada saat *Sprint*. *Product Owner* menjabarkan secara objektif apa saja yang harus dicapai di *Sprint* ini, dan item *Product Backlog* mana yang dapat diselesaikan di *Sprint* ini, sehingga dapat tercapai *Sprint Goal* seperti yang diharapkan. Seluruh anggota Tim *Scrum* berkolaborasi untuk memahami pekerjaan di dalam *Sprint*.

Masukan dari acara ini adalah *Product Backlog*, sedangkan outputnya adalah Inkremen. Dalam merencakan *Sprint* perlu diketahui juga proyeksi kapasitas Tim Pengembang dalam satu *Sprint*, dan histori performa Tim Pengembang. Jumlah item yang dipilih dari *Product Backlog* untuk *Sprint* sepenuhnya diserahkan kepada Tim Pengembang. Hanya Tim Pengembang yang dapat menilai seberapa banyak item yang dapat diselesaikan di *Sprint* ini.

Setelah Tim Pengembang memperkirakan item *Product Backlog* yang akan selesai dan dihantarkan dalam *Sprint* ini, Tim Scrum mulai membuat *Sprint Goal*. *Sprint Goal* dapat menciptakan sebuah keselarasan di dalam pekerjaan Tim Pengembang, yang mungkin tidak akan ada bila masing-masing anggota tim memiliki inisiatif sendiri-sendiri tanpa adanya tujuan yang sama.

Topik Kedua: Bagaimana pekerjaan yang telah dipilih dapat diselesaikan?

Setelah *Sprint Goal* dibuat dan item *Product Backlog* dipilih, Tim Pengembang menentukan bagaimana mengembangkan fungsionalitas ini menjadi sebuah Inkremen yang dapat selesai tepat waktu. Item *Product Backlog* yang telah dipilih dan dimasukkan kedalam *Sprint,* beserta target waktu pekerjaan hingga status task menjadi "selesai" dan disampaikan kepada *Product Owner* disebut sebagai *Sprint Backlog*.

Tim Pengembang biasanya memulai dengan merancang sistem dan pekerjaan yang perlu dilakukan, lalu memasukkannya kedalam daftar *Product Backlog,* hingga menjadi Inkremen yang berfungsi penuh. Pekerjaan yang dirancang mungkin akan memiliki ukuran atau estimasi yang berbeda-beda. Walaupun demikian, jumlah pekerjaan yang direncanakan pada saat *Sprint Planning* cukup banyak untuk dikerjakan selama satu *Sprint*. Pekerjaan yang direncanakan untuk hari-hari pertama dari *Sprint* dibagi menjadi bagian-bagian kecil, biasanya dalam satuan satu hari atau kurang. Tim Pengembang mengatur

dirinya sendiri untuk mengambil pekerjaan di dalam *Sprint Backlog*, baik pada saat *Sprint Planning* maupun sepanjang *Sprint*.

Pada saat Tim Pengembang membuat perencanaan, rencana tersebut selalu mengacu pada *Sprint Goal*. Pada saat *Sprint* berjalan, pekerjaan yang harus dilakukan terkadang berbeda dengan apa yang telah direncanakan oleh Tim Pengembang pada saat *Sprint Planning*. Tim Pengembang akan berkolaborasi dengan *Product Owner* untuk menentukan cara terbaik untuk merevisi perencanaan dengan tetap mencapai *Sprint Goal*. *Sprint Goal* menyediakan fleksibilitas mengenai bagaimana fungsionalitas dapat diimplementasikan sebelum *Sprint* berakhir.

Product Owner dapat membantu mengklarifikasi item *Product Backlo*g yang dipilih dan membuat pengecualian. Apabila Tim Pengembang mengatakan mereka memiliki terlalu banyak atau terlalu sedikit pekerjaan, mereka dapat menegosiasikan ulang item *Product Backlog* yang telah dipilih dengan *Product Owner*. Tim Pengembang juga dapat mengundang pihak lain untuk menghadiri acara ini guna memberikan masukan yang berhubungan dengan hal teknis ataupun domain permasalahan.

Di akhir *Sprint Planing*, Tim Pengembang sudah harus dapat menjelaskan kepada *Product Owner* ataupun *Scrum Master*, bagaimana mereka berencana untuk bekerja sebagai tim yang mengatur dirinya sendiri untuk menyelesaikan *Sprint Goal*, dan membuat Inkremen yang telah diantisipasi.

2.1.3.4 *Sprint Goal*

Sprint Goal adalah sekumpulan tujuan yang akan dicapai dalam satu *Sprint* sepanjang pengimplementasian *Product Backlog*. *Sprint Goal* memberikan arahan bagi Tim Pengembang mengapa mereka mengembangkan Inkremen dalam *Sprint* tersebut. *Sprint Goal* dibuat pada saat *Sprint Planning*. *Sprint Goal* memberikan Tim Pengembang fleksibilitas terkait bagaimana implementasi fungsionalitas pada saat mengerjakan *Sprint*. Item-item *Product Backlog* yang terpilih menghantarkan pada satu fungsionalitas yang selaras. Di mana bisa berupa *Sprint Goal* itu sendiri. *Sprint goal* bisa juga berupa fungsionalitas yang selaras apapun, yang pada akhirnya membuat Tim Pengembang berkerja bersama, bukan inisiatif sendiri-sendiri.

Tim Pengembang berkerja dengan dipandu oleh *Sprint Goal*. Untuk memenuhi *Sprint Goal*, mereka mengimplementasikan fungsionalitas & teknologi. Jika hasil kerja mereka ternyata berbeda dengan yang mereka duga sebelumnya, mereka berkolaborasi dengan *Product Owner* untuk menegosiasikan ruang lingkup dari *Sprint Backlog* pada suatu *Sprint*.

2.1.3.5 *Daily Scrum*

Daily Scrum adalah kegiatan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam ke depan. Hal ini dilakukan dengan meninjau

pekerjaan semenjak acara *Daily Scrum* terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum melakukan *Daily Scrum* berikutnya.

Daily Scrum dilaksanakan pada waktu dan tempat yang sama setiap hari untuk mengurangi kompleksitas. Pada saat pertemuan, Tim Pengembang menjelaskan hal-hal berikut :

a. Apa yang telah dilakukan sehingga *Sprint Goal* tercapai?
b. Apa yang akan dilakukan hari ini untuk mencapai *Sprint Goal*?
c. Apakah ada hambatan yang dapat menghalangi untuk mencapai *Sprint Goal*?

Tim Pengembang menggunakan *Daily Scrum* untuk meninjau perkembangan menuju *Sprint Goal* dan meninjau tren perkembangan menuju selesainya pekerjaan yang ada di dalam *Sprint Backlog*. *Daily Scrum* mengoptimalisasi kemungkinan Tim Pengembang akan mencapai *Sprint Goal*. Setiap hari, Tim Pengembang harus memahami bagaimana caranya agar mereka dapat bekerja bersama sebagai tim yang mengatur dirinya sendiri untuk menyelesaikan *Sprint Goal*, dan menghasilkan Inkremen pada akhir *Sprint*. Tim Pengembang atau beberapa anggota tim seringkali langsung bertemu setelah *Daily Scrum* dilakukan untuk diskusi lebih detail, atau untuk adaptasi tentang perubahan rencana pada sisa pekerjaan dalam *Sprint*.

Scrum Master yang bertanggungjawab mengadakan dan memimpin *Daily Scrum Meeting*, dan anggotanya adalah Tim Pengembang. *Scrum Master* memimpin Tim Pengembang untuk melangsungkan *Daily Scrum* tidak lebih dari

15 menit. *Scrum Master* memastikan bahwa hanya anggota Tim Pengembang yang berpartisipasi pada saat *Daily Scrum*.

Daily Scrum dilakukan untuk efektifitas komunikasi, sehingga tidak perlu lagi ada pertemuan lain selain *Daily Scrum* untuk membahas inkremen yang akan dihasilkan, dan identifikasi hambatan. Kegiatan ini mendukung pembuatan keputusan secara cepat dan dapat meningkatkan pengetahuan tim. Pertemuan ini adalah kunci dari proses peninjauan dan adaptasi.

2.1.3.6 *Sprint Review*

Sprint Review diadakan di akhir *Sprint* untuk meninjau Inkremen dan merubah *Product Backlog* bila diperlukan. Pada *Sprint Review*, Tim Scrum dan *stakeholder* berkolaborasi untuk membahas apa yang telah dikerjakan dalam *Sprint* yang baru saja diselesaikan. Dari pembahasan tersebut, peserta *Sprint Review* berkolaborasi menentukan apa yang dapat dikerjakan di *Sprint* berikutnya, untuk optimalisasi nilai produk. Pertemuan ini bersifat informal, dan presentasi dari Inkremen diharapkan dapat mengumpulkan masukan dan menumbuhkan semangat kolaborasi.

Ini adalah acara dengan batasan waktu maksimum selama 4 (empat) jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan pelaksanaan acara ini, dan setiap peserta yang hadir pada pertemeuan tersebut memahami tujuannya.

Scrum Master mengedukasi Tim *Scrum* untuk melaksanakannya dalam batasan waktu yang telah ditentukan.

Sprint Review mencakup elemen-elemen berikut:

a. Peserta yang hadir adalah *Product Owner*, Tim Scrum, dan *stakeholder* yang diundang oleh *Product Owner*;

b. *Product Owner* menjelaskan progres *Product Backlog*, memberikan informasi mengenai *task* apa saja yang sudah "Selesai" dan apa saja yang belum "Selesai";

c. Tim Pengembang menjelaskan hal-hal apa saja yang berjalan dengan baik sepanjang *Sprint*, masalah apa yang dihadapi, dan bagaimana menyelesaikan masalah tersebut;

d. Tim Pengembang mendemonstrasikan pekerjaan yang sudah dianggap "selesai" dan menjawab pertanyaan-pertanyaan yang muncul.

e. *Product Owner* menjelaskan keadaan terakhir *Product Backlog*. Ia dapat memproyeksikan tanggal perkiraan selesai produk (bila dibutuhkan);

f. Seluruh peserta pertemuan berkolaborasi membahas pekerjaan selanjutnya, dengan begitu *Sprint Review* mendapatkan masukan yang berarti bagi *Sprint Planning* berikutnya;

g. Ulasan mengenai keadaan pasar (*market review*), kemungkinan perubahan potensi penggunaan produk, dan hal-hal penting lainnya;

h. *Review timeline*, *budget*, potensi kapabilitas dan *marketplace* untuk antisipasi pada saat rilis produk.

Hasil dari *Sprint Review* adalah revisi dari *Product Backlog* dan didefinisikan sebagai daftar *task* pada *Product Backlog* untuk *Sprint* berikutnya. *Product Backlog* dapat diubah secara keseluruhan atau sebagian.

2.1.3.7 *Sprint Retrospective*

Sprint Retrospective adalah sebuah kesempatan bagi Tim Scrum untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

Sprint Retrospective dilangsungkan setelah *Sprint Review* selesai dan sebelum *Sprint Planning* berikutnya. Ini adalah acara dengan batasan waktu maksimum selama (3) tiga jam untuk *Sprint* yang berdurasi satu bulan. Untuk *Sprint* yang lebih pendek, batasan waktunya biasanya lebih singkat. *Scrum Master* memastikan bahwa acara ini dilaksanakan dan setiap peserta pertemuan memahami tujuannya. *Scrum Master* mengedukasi Tim *Scrum* untuk melaksanakannya dalam batasan waktu yang telah ditentukan. *Scrum Master* berpartisipasi sebagai rekan yang bertanggungjawab terhadap proses *Scrum*.

Tujuan dari *Sprint Retrospective* adalah:

a. Melakukan *review* terhadap personil, hubungan antar personil, proses, dan peralatan pada *Sprint* yang telah selesai.

b. Mengidentifikasi dan mengurutkan hal-hal utama yang berjalan baik, dan hal-hal yang berpotensi untuk ditingkatkan; dan,

c. Membuat rencana implementasi, dengan tujuan peningkatan cara-cara kerja Tim *Scrum*.

Scrum Master mengedukasi Tim *Scrum* untuk membuat peningkatan kinerja dengan kerangka kerja Scrum, sehingga *Sprint* berikutnya dapat dilakukan dengan lebih efektif dan menyenangkan. Pada tahap *Sprint Retrospective*, Tim Scrum merencanakan cara untuk meningkatkan kualitas dari produk, dengan merubah definisi "Selesai" sebagaimana dibutuhkan.

Di akhir *Sprint Retrospective*, Tim *Scrum* harus dapat mengidentifikasi peningkatan-peningkatan yang akan diimplementasikan di *Sprint* berikutnya. *Sprint Retrospective* memberikan kesempatan formal untuk fokus pada kegiatan *review* dan adaptasi.

2.1.4 Artefak *Scrum*

Artefak Scrum merepresentasikan pekerjaan atau nilai, bertujuan untuk menyediakan transparansi, dan kesempatan-kesempatan untuk peninjauan dan adaptasi. Artefak yang didefinisikan oleh Scrum secara khusus dirancang untuk meningkatkan transparansi dari informasi kunci, dengan begitu semua pihak dapat memiliki pemahaman yang sama terhadap artefak.

2.1.4.1 *Product Backlog*

Product Backlog adalah daftar terurut, dari setiap hal yang berkemungkinan dibutuhkan di dalam produk, dan juga merupakan sumber utama, dari daftar kebutuhan mengenai semua hal yang perlu dilakukan terhadap produk. *Product Owner* bertanggungjawab terhadap *Product Backlog*, termasuk isinya, ketersediaannya, dan urutannya.

Product Backlog akan selalu di-update sesuai dengan kebutuhan hingga target akhir pekerjaan, bisa jadi *product backlog* ini tidak pernah selesai karena produk selalu dikembangkan. Pada awal pembuatannya hanya terjabar daftar kebutuhan yang hanya diketahui dan dipahami pada saat itu. *Product Backlog* berkembang seiring dengan berkembangnya produk dan lingkungan dimana produk tersebut digunakan. *Product Backlog* bersifat dinamis, senantiasa berubah agar produk dapat menjadi layak, kompetitif di pasar, dan bermanfaat bagi penggunanya. Selama produk masih eksis maka *Product Backlog* juga eksis.

Product Backlog menjabarkan semua fitur, fungsi, kebutuhan, penyempurnaan dan perbaikan terhadap produk yang akan dirilis. Item *Product Backlog* memiliki atribut deskripsi, urutan, estimasi dan nilai bisnis.

Seiring dengan digunakannya produk dan semakin bertambahnya nilai dari produk, dan bertambahnya masukan dari pasar, *Product Backlog* semakin berkembang menjadi lebih besar. Daftar kebutuhan tidak pernah berhenti berubah, sehingga *Product Backlog* dapat dikatakan sebagai artefak yang hidup. Perubahan

dalam kebutuhan bisnis, keadaan pasar, ataupun teknologi dapat menyebabkan perubahan pada *Product Backlog*.

Tidak jarang ditemukan lebih dari satu Tim *Scrum* mengerjakan satu produk yang sama. Satu *Product Backlog* digunakan untuk menggambarkan pekerjaan selanjutnya terhadap sebuah produk. Bisa ditambahkan sebuah atribut, untuk mengelompokkan item *Product Backlog*.

Product Backlog refinement adalah kegiatan menambahkan detail, mengestimasi dan mengurutkan item di dalam *Product Backlog*. Kegiatan ini berkesinambungan, di mana *Product Owner* dan Tim Pengembang berkolaborasi untuk merinci item *Product Backlog*. Pada saat *Product Backlog refinement*, item ditinjau-ulang dan direvisi. Tim *Scrum* sendiri yang menentukan bagaimana dan kapan proses *refinement* diadakan. *Refinement* biasanya memakan tidak lebih dari 10% kapasitas Tim Pengembang. Walaupun demikian, item *Product Backlog* dapat diperbarui kapanpun juga oleh *Product Owner* atau siapapun atas arahan *Product Owner* kapanpun diinginkan.

Item *Product Backlog* pada urutan yang lebih atas biasanya lebih jelas dan lebih detail dibandingkan item di bawahnya. Estimasi dengan presisi tinggi diberikan berdasarkan tingkat kejelasan dan detail yang tinggi. Semakin bawah urutan dari item *Product Backlog*, maka semakin rendah pula tingkat kedetailannya. Item *Product Backlog* yang akan dikerjakan oleh Tim Pengembang untuk *Sprint* yang mendatang di-*refine* supaya setiap item yang dikerjakan dapat di-"Selesai"-kan dalam satu *Sprint*. Item *Product Backlog* yang dianggap dapat di-"Selesai"-kan oleh Tim Pengembang dalam satu *Sprint* dikatakan "Siap" untuk diseleksi pada saat

Sprint Planning. Item *Product Backlog* biasanya akan memiliki tingkat transparansi yang tinggi karena adanya aktifitas *refinement* ini.

Tim Pengembang bertanggungjawab terhadap seluruh estimasi. *Product Owner* dapat mempengaruhi Tim Pengembang dengan cara membantu mereka memahami *Product Backlog* dan membuat pengecualian terhadap *Product Backlog*, namun orang-orang yang akan mengerjakan item *Product Backlog*-lah yang akan membuat estimasi final.

Memantau perkembangan menuju *Sprint Goal*

Di titik manapun, jumlah pekerjaan yang tersisa hingga akhir tujuan pengembangan dapat dijumlahkan. *Product Owner* memantau total sisa pekerjaan ini setidaknya di setiap *Sprint Review*. *Product Owner* membandingkan kondisi saat ini dengan jumlah sisa pekerjaan di *Sprint Review* sebelumnya guna meninjau perkembangan menuju tujuan akhir dengan waktu yang diharapkan. Informasi ini transparan untuk setiap *stakeholder*.

Berbagai macam praktik proyeksi terhadap *trending* telah digunakan untuk memperkirakan kemajuan, seperti *burn-down graphic* atau *burn-up graphic*. Hal ini telah terbukti berguna. Namun hal ini tidak menggantikan pentingnya peran empirisme. Di dalam lingkungan yang kompleks, apa yang akan terjadi di masa mendatang tidak dapat diketahui sebelumnya. Hanya apa yang telah terjadi yang dapat digunakan untuk membuat keputusan di masa mendatang.

2.1.4.2 *Sprint Backlog*

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di Inkremen selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan tambahan produk yang "Selesai".

Sprint Backlog menampilkan semua pekerjaan yang dibutuhkan untuk mencapai *Sprint Goal* yang dibuat oleh Tim Pengembang.

Sprint Backlog adalah sebuah rencana yang cukup detail, di mana perubahan-perubahannya di tengah *Sprint* bisa dipahami saat *Daily Scrum Meeting*. Tim Pengembang memodifikasi *Sprint Backlog* sepanjang *Sprint* berlangsung, dan *Sprint Backlog* dapat berubah kapanpun juga sepanjang *Sprint*. Perubahan ini terjadi seiring dengan berkerjanya Tim Pengembang sesuai rencana pada saat itu, dan semakin meningkatnya wawasan tim untuk mencapai tujuan *Sprint*.

Dengan bertambahnya pekerjaan baru, Tim Pengembang menambahkannya ke dalam *Sprint Backlog*. Dengan dikerjakannya atau diselesaikannya pekerjaan, estimasi sisa pekerjaan juga diperbaharui. Ketika ada elemen dari perencanaan tidak dibutuhkan lagi, maka elemen tersebut dikeluarkan dari *Sprint Backlog*. Hanya Tim Pengembang yang dapat merubah *Sprint Backlog* pada saat *Sprint* sedang berjalan. *Sprint Backlog* sangat transparan, menggambarkan secara *real-time* pekerjaan yang

akan diselesaikan oleh Tim Pengembang pada saat *Sprint*, dan ia sepenuhnya menjadi milik Tim Pengembang.

Memantau perkembangan *Sprint*

Di titik manapun dalam Sprint, jumlah sisa pekerjaan dalam *Sprint Backlog* dapat dijumlahkan. Tim Pengembang memantau sisa pekerjaan ini, setidaknya di setiap *Daily Scrum*, untuk memproyeksikan kemungkinan mereka akan mencapai *Sprint Goal*. Dengan memantau sisa pekerjaan ini sepanjang *Sprint*, Tim Pengembang dapat mengelola perkembangan pekerjaan.

2.1.4.3 Inkremen

Inkremen (tambahan potongan produk) adalah gabungan dari semua item *Product Backlog* yang diselesaikan pada *Sprint* berjalan dan nilai-nilai dari Inkremen *sprint-sprint* sebelumnya. Pada akhir *Sprint*, inkremen terbaru harus "Selesai", yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi "Selesai" yang dibuat oleh Tim Scrum. Terlepas apakah *Product Owner* akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh.

2.1.5 Transparansi Artefak

Scrum berlandaskan transparansi. Keputusan-keputusan untuk mengoptimalisasi nilai produk dan mengendalikan resiko dibuat berdasarkan

keadaan artefak hingga saat ini. Pada titik di mana transparansi berada pada tingkat tinggi, keputusan yang dibuat semakin dapat dipercaya. Pada titik di mana transparansi berada pada tingkat rendah, keputusan dapat dimanipulasi, nilai produk akan menurun dan resiko akan meningkat.

Scrum Master harus bekerja dengan *Product Owner*, Tim Pengembang dan pihak-pihak lain untuk bersama-sama memahami apakah semua artefak sudah sepenuhnya transparan. Ada banyak praktik untuk menangani belum penuhnya transparansi; dalam keadaan ini, *Scrum Master* harus membantu semua pihak untuk menerapkan praktik yang sesuai terhadap hilangnya transparansi. *Scrum Master* dapat mendeteksi hilangnya transparansi dengan meninjau semua artefak, mengamati pola-pola yang terjadi, menyimak apa yang dikatakan, dan melihat perbedaan antara apa yang diharapan dengan hasil yang sebenarnya.

Tugas *Scrum Master* adalah bekerja bersama Tim Scrum dan organisasi dimana ia berada untuk meningkatkan tingkat transparansi dari artefak-artefak yang digunakan. Pekerjaan ini biasanya membutuhkan pembelajaran, pendekatan persuasif, serta perubahan. Transparansi tidak terjadi dalam semalam, melainkan sebuah perjalanan jangka panjang.

Definisi Selesai

Ketika sebuah item *Product Backlog* atau Inkremen dikatakan “Selesai”, setiap pihak harus mengerti dengan apa yang dimaksud dengan “Selesai”. Walaupun definisi ini berbeda-beda antar tim Scrum, sesama anggota tim harus

memiliki pemahaman yang sama mengenai pekerjaan yang harus mereka selesaikan guna memastikan adanya transparansi. Ini adalah definisi selesai untuk Tim Scrum dan ini digunakan untuk memeriksa apakah pekerjaan untuk mengembangkan Inkremen dianggap selesai.

Definisi yang sama akan membimbing Tim Pengembang dalam mengetahui berapa banyak item *Product Backlog* yang mereka bisa ambil pada saat *Sprint Planning*. Tujuan dari setiap *Sprint* adalah untuk menghantarkan Inkremen, yang berpotensi untuk dirilis, yang memenuhi definisi "Selesai" terkini yang dibuat oleh Tim Scrum.

Tim Pengembang menghantarkan Inkremen yang berfungsi setiap *Sprint*. Inkremen ini dapat digunakan, supaya *Product Owner* dapat merilis produk tersebut sesegera mungkin jika ia mau. Apabila definisi "Selesai" untuk sebuah Inkremen adalah bagian dari konvensi, standar atau panduan pengembangan dari organisasi, setiap Tim Scrum harus mengikuti seluruhnya sebagai *minimum requirement*. Apabila "Selesai" untuk sebuah Inkremen bukan merupakan bagian dari konvensi, standar atau panduan pengembangan dari organisasi, Tim Pengembang harus membuat definisi "Selesai" yang pantas untuk produk yang dikembangkan. Apabila ada beberapa Tim Scrum yang mengembangkan sistem atau produk yang sama, seluruh Tim Pengembang di setiap Tim Scrum harus menentukan definisi selesai yang sama bersama-sama.

Setiap Inkremen merupakan gabungan dari Inkremen *Sprint-sprint* sebelumnya dan diuji secara teliti, untuk memastikan setiap Inkremen dapat berfungsi secara penuh.

Seiring dengan bertambah dewasanya Tim Scrum, mereka diharapkan untuk membuat definisi selesai yang lebih baik dan ketat lagi demi peningkatan kualitas. Produk atau sistem manapun harus memiliki definisi “Selesai” yang merupakan sebuah standar untuk pekerjaan yang akan dilakukan.

2.1.6 Catatan Akhir

Scrum bersifat gratis dan disediakan di panduan ini. Peran, artefak, acara dan aturan dalam *Scrum* tidak dapat diubah walaupun penggunaan Scrum secara sebagian sangat memungkinkan, hasilnya bukanlah Scrum. Scrum ada karena seluruh komponen ini dan berfungsi dengan baik sebagai pembungkus untuk teknik, metodologi maupun praktik lain.

2.2 *Sales Management*

Berikut gambaran singkat tentang proses penjualan yang sering dilakukan oleh Sales atau Tenaga Penjualan (Solingen, 2011).

a. Tenaga penjualan terlalu fokus pada progres pekerjaan individu.

   Tenaga Penjualan biasanya bertanggung jawab atas tugas dan hasil pekerjaan mereka sendiri. Tidak ada komunikasi antara sesama Tenaga Penjualan tentang rencana, strategi, dan metode penjualan yang akan digunakan. Bahkan sistem CRM yang digunakan untuk menyimpan informasi rinci tentang pelanggan pun jarang diakses. Hal ini

mengakibatkan kegiatan Tenaga Penjualan menjadi sangat individual bahkan mereka memiliki *database* sendiri untuk daftar pelanggan yang di-followup. Pada akhirnya Tenaga Penjualan tersebut hanya memberikan informasi jumlah penjualan yang terjadi. Setiap awal minggu, Tenaga Penjualan memberikan target penjualan mereka kepada atasan, dan pada akhir minggu mereka melaporkan angka penjualan yang telah direalisasikan.

b. Tenaga penjualan bekerja secara acak

   Tenaga penjualan seringkali tidak merencakan apa yang akan dilakukan untuk keesokan hari, dan seringkali tidak mendokumentasikan progres pekerjaannya.

c. Tenaga penjualan hanya fokus terhadap hasil akhir.

   Satu-satunya ukuran keberhasilan dalam tim penjualan adalah jumlah penjualan. Tenaga penjualan meyakini bahwa tidak ada sistem yang dapat mempengaruhi hasil akhir.

d. Tenaga penjualan tidak pernah tau kapan penjualan (*deal*) akan terjadi.

   Tenaga penjualan meyakini bahwa tidak ada sesuatu yang pasti, yang menentukan hasil akhir penjualan. Sehingga seorang tenaga penjualan dituntuk harus bekerja lebih keras dalam melakukan kegiatannya, seperti melakukan *Sales Call*, dan membuat janji pertemuan dengan pelanggan atau calon pelanggan sebanyak mungkin. Hal ini akan meningkatkan peluang bagi tenaga penjualan untuk meningkatkan penjualannya.

e. Tidak ada pengembangan atau perbaikan pada kegiatan penjualan

   Tenaga penjualan seringkali hanya fokus pada rutinitas mereka sehari-hari dan tidak menyadari kebutuhan untuk berimprovisasi, berinovasi, dan belajar hal-hal baru pada kegiatan penjualan.

2.3 Penelitian Terkait

Dalam penyusunan skripsi ini, penulis sedikit banyak terinspirasi dan mereferensi dari penelitian - penelitian sebelumnya yang berkaitan dengan latar belakang masalah pada penelitian ini. Adapun penelitian yang berhubungan dengan skripsi ini dilakukan oleh R. van Solingen, D. de Waard, & J. Sutherland, pada tahun 2011 dengan judul *Scrum in Sales: How to Improve Account Management and Sales Processes*. *Conference Paper*. DOI: 10.1109/AGILE.2011.12.