

BAB III

LANDASAN TEORI

3.1 Risiko

Risiko adalah suatu peristiwa atau kondisi yang mungkin terjadi, yang apabila terjadi berdampak pada tujuan proyek. Risiko dinilai berdampak negatif pada tujuan perusahaan (The MITRE Corporation, 2014). Oleh karena itu, dibutuhkan respon yang tepat untuk menangani risiko yang terjadi pada penggunaan kerangka kerja Scrum. Ada beberapa jenis respon risiko yang dapat dilakukan untuk mengurangi ancaman atau risiko negatif terhadap proyek, antara lain penghindaran risiko (risk avoidance), pemindahan risiko (risk transfer), penerimaan risiko (risk acceptance), pemantauan risiko (risk monitoring), pengurangan risiko (risk mitigation), rencana cadangan (contingency plan), dan rencana strategis (strategic plan) (Pandian, 2006). Mitigasi risiko dianggap lebih efektif daripada mencoba untuk memperbaiki kerusakan risiko yang telah terjadi (Project Management Institute, 2008).

3.2 Mitigasi Risiko

Mitigasi risiko adalah salah satu respon pada manajemen risiko untuk menangani risiko dengan cara mengambil tindakan dini untuk mengurangi probabilitas dan atau dampak risiko. Mitigasi risiko hanya menyentuh permukaan risiko. Hal tersebut merupakan perluasan kesadaran risiko. Orang-orang yang memulai rencana mitigasi tahu lebih banyak tentang risiko daripada

mereka yang berhenti di analisis (Pandian, 2006). Mitigasi risiko termasuk dalam proses manajemen risiko.

3.3 Manajemen Risiko

Manajemen risiko adalah bagian utama dari setiap manajemen strategis organisasi. Manajemen risiko adalah proses dimana organisasi mengatasi risiko yang melekat pada kegiatan mereka dengan tujuan mencapai keuntungan yang berkelanjutan dalam setiap aktivitas dan seluruh kegiatan mereka (The Institute of Risk Management, 2002). Manajemen risiko memiliki 4 tahapan yang disebut IAMT, yaitu identifikasi (identification), analisis (analysis), mitigasi (mitigation), dan pelacakan (tracking). Siklus IAMT menunjukkan bahwa manajemen risiko adalah proses yang berkesinambungan dan tak berujung (Pandian, 2006). Manajemen risiko pada perusahaan IT biasanya terintegrasi pada Software Development Life Cycle (SDLC) (Unuakhalu, Sigdel, & Garikapati, 2014). Metode Agile adalah jenis pengembangan sistem jangka pendek yang fleksibel. Sehingga metode ini dapat beradaptasi terhadap perubahan requirement yang cepat. Beberapa metode yang termasuk Agile, antara lain Extreme Programming (XP), Scrum, Kanban, Feature Driven Development (FDD), Dynamic System Development Methodology (DSDM) dan Lean Software Development.

3.4 SDLC

Software Development Life Cycle (SDLC) adalah proses yang menggambarkan metode dan strategi untuk mengembangkan desain dan memelihara proyek perangkat

lunak untuk memastikan bahwa semua tujuan, sasaran, fungsional, dan kebutuhan pengguna terpenuhi (Arora & Arora, 2016). SDLC memiliki tahap-tahap analisis kebutuhan, desain, pengkodean (**Coding**), uji coba dan pengelolaan (Kumar, Zadgaonkar, & Shukla, 2013). Beberapa contoh model SDLC, antara lain Waterfall, V-Model, Agile, dll.

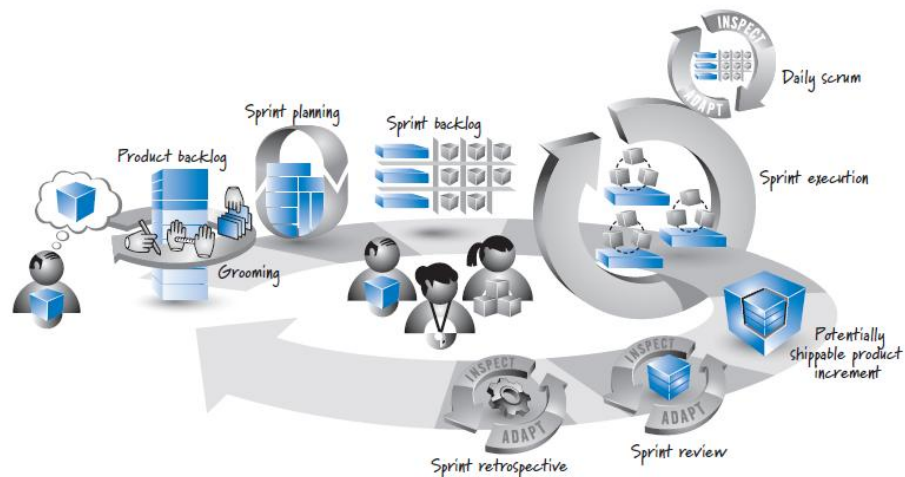
3.5 Agile

Agile adalah salah satu model Software Development Life Cycle. Agile merupakan metodologi pengembangan perangkat lunak untuk membangun perangkat lunak secara bertahap menggunakan iterasi singkat yaitu 1 sampai 4 minggu sehingga proses pembangunan selaras dengan perubahan kebutuhan bisnis (tutorialspoint, 2014). Beberapa metode yang termasuk Agile, antara lain Extreme Programming (XP), Scrum, Kanban, Feature Driven Development (FDD), Dynamic System Development Methodology (DSDM) dan Lean Software Development.

3.6 Scrum

Scrum pertama kali dikenalkan pada awal tahun 1990 oleh Ken Schwaber dan Jeff Sutherland. Scrum merupakan salah satu kerangka kerja yang mengadopsi prinsip Agile. Scrum bukanlah sebuah proses atau teknik untuk mengembangkan produk. Scrum disebut sebagai sebuah kerangka kerja yang didalamnya kita dapat memasukkan beragam proses dan teknik. Scrum merupakan sebuah kerangka kerja yang membuat orang bisa mengarahkan permasalahan adaptif yang kompleks, sembari dengan produktif dan kreatif menghasilkan produk dengan

kemungkinan nilai tertinggi (Sutherland & Schwaber, 2016).



Gambar 3.1 Framework Scrum oleh Kenneth S. Rubin
(Rubin, 2013)

Penjelasan hal-hal yang ada pada praktik Scrum, sebagai berikut:

3.6.1 Peran Scrum

Framework Scrum memiliki 3 peran, yaitu:

1. Product Owner

Product Owner adalah seseorang yang bertanggung jawab untuk memutuskan urutan fitur dan fungsi yang akan dibangun. Product Owner juga menjaga dan menyampaikan kepada semua anggota Scrum visi yang jelas mengenai apa yang tim Scrum coba untuk selesaikan (Rubin, 2013).

2. Scrum Master

Scrum Master adalah seseorang yang membantu semua orang yang terlibat di dalam Scrum untuk memahami dan menerapkan nilai-nilai, prinsip, dan praktik Scrum. Sebagai fasilitator, Scrum Master juga bertanggung jawab untuk melindungi

tim dari luar gangguan dan mengambil peran kepemimpinan dalam menghilangkan hambatan yang menghambat produktivitas tim (Rubin, 2013).

3. Tim Pengembang (Development Team)

Tim pengembang di dalam Scrum berisi orang-orang yang memiliki keahlian dan tanggung jawab yang berbeda-beda, misalnya programmer, UI designer, UX designer, Quality Assurance, dan lain-lain. Tim pengembang berhak mengatur dan menentukan cara terbaik untuk mencapai tujuan yang ditetapkan oleh Product Owner. Tim pengembang biasanya terdiri dari 5-9 orang atau ± 7 orang dimana anggotanya harus memiliki semua kemampuan yang dibutuhkan untuk menghasilkan perangkat lunak dengan kualitas yang baik (Rubin, 2013).

3.6.2 Artefak Scrum

1. Product Backlog

Didalam Scrum pekerjaan yang paling penting atau bernilai dilakukan pertama kali. Untuk pengembangan produk yang sedang berlangsung, Product Backlog bisa berisi fitur baru, perubahan fitur, perbaikan fitur, perbaikan teknis, dan lain-lain (Rubin, 2013).

2. Sprint Backlog

Sprint Backlog terdiri dari daftar pekerjaan yang akan dipilih oleh tim pengembang untuk dikerjakan selama sebuah Sprint dilakukan. Hanya tim pengembang yang berhak untuk melakukan modifikasi dalam Sprint Backlog. Setiap item dalam Sprint Backlog dirancang dan

diturunkan menjadi tugas individu (Rahman & Das, 2015).

3.6.3 Aktifitas Scrum

1. Sprint

Di dalam Scrum, pekerjaan dilakukan di dalam iterasi atau siklus maksimal satu bulan. Pekerjaan yang selesai di setiap sprint harus menciptakan sesuatu yang memiliki nilai nyata untuk pelanggan atau pengguna (Rubin, 2013).

2. Sprint Planning

Pertemuan Sprint Planning dilakukan di awal Sprint. Pekerjaan yang harus dilakukan di sprint direncanakan pada Sprint Planning. Sprint Planning maksimal 8 jam untuk satu bulan sprint. Untuk sprint yang lebih pendek, batasan waktunya biasanya lebih singkat (Sutherland & Schwaber, 2016).

3. Daily Scrum

Daily Scrum adalah kegiatan dengan batasan waktu 15 menit agar tim pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam ke depan. Hal ini dilakukan dengan meninjau pekerjaan mulai dari Daily Scrum terakhir dan memperkirakan pekerjaan yang dapat dilakukan sebelum Daily Scrum berikutnya (Sutherland & Schwaber, 2016).

4. Sprint Review

Sprint Review diadakan di akhir Sprint untuk meninjau kemajuan dan merubah Product Backlog bila diperlukan. Pada saat Sprint Review, tim Scrum dan stakeholder berkolaborasi untuk membahas apa yang

telah dikerjakan dalam Sprint yang baru saja selesai dan menentukan apa yang akan dikerjakan di Sprint selanjutnya (Sutherland & Schwaber, 2016).

5. Sprint Retrospective

Setelah Sprint Review selesai tim Scrum melakukan Sprint Retrospective untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di sprint berikutnya. Acara ini memiliki batasan waktu maksimum selama tiga jam untuk Sprint yang berdurasi satu bulan (Sutherland & Schwaber, 2016).

6. Sprint Grooming

Sprint Grooming biasanya dilakukan setelah Daily Scrum. Sprint Grooming tidak mencakup semua dari anggota tim. Misalnya, setelah Daily Scrum Product Owner mungkin meminta bantuan untuk menyempurnakan sebuah story yang besar kepada anggota tim yang berpengaruh dan berkepentingan (Rubin, 2013).