BAB III
DASAR TEORI

3.1Manajemen Risiko

Risiko mengacu pada kondisi di masa depan atau keadaan yang terjadi diluar kendali tim proyek yang akan memberikan dampak yang merugikan proyek (Dey, et al., 2007). Berdasarkan pengertian tersebut, risiko biasanya merupakan hal yang belum terjadi. Namun, risiko dapat diprediksikan oleh tim proyek. Risiko buruk yang terjadi akan mengakibatkan proyek gagal atau hasil dari proyek tidak maksimal.

Manajemen risiko proyek adalah seni dan ilmu untuk mengidentifikasi, menganalisis, dan menanggapi risiko melalui siklus hidup proyek dan dalam keinginan untuk mencapai tujuan proyek. Manajemen risiko konvensional dalam sebuah proyek memiliki beberapa tahapan yakni, perencanaan manajemen risiko, identifikasi risiko, melakukan analisis kualitatif, melakukan analisis kuantitatif, merencanakan tanggapan terhadap risiko serta melakukan pemantauan dan pengendalian risiko(Schwalbe, 2011). Perencanaan manajemen risiko merupakan fase awal yang mencoba untuk menentukan bagaimana cara pendekatan dan aktivitas apa saja yang perlu dilakukan sebagai rencana manajemen risiko didalam sebuah proyek.

Setelah dilakukan perencanaan manajemen risiko, fase berikutnya dari manajemen risiko adalah identifikasi risiko. Identifikasi risiko merupakan cara menentukan risiko apa saja yang mungkin berpengaruh terhadap proyek kemudian dilakukan proses dokumentasi

terhadap risiko-risiko tersebut. Identifikasi risiko merupakan proses untuk menentukan apa, bagaimana dan mengapa hal-hal dapat terjadi (Cooper, et al., 2005). Identifikasi risiko ini nantinya akan menghasilkan Risk Register yang merupakan daftar dari risiko-risiko yang berhasil di identifikasi. Risk Register ini merupakan alat untuk mendokumentasikan risiko-risiko yang berpotensial.

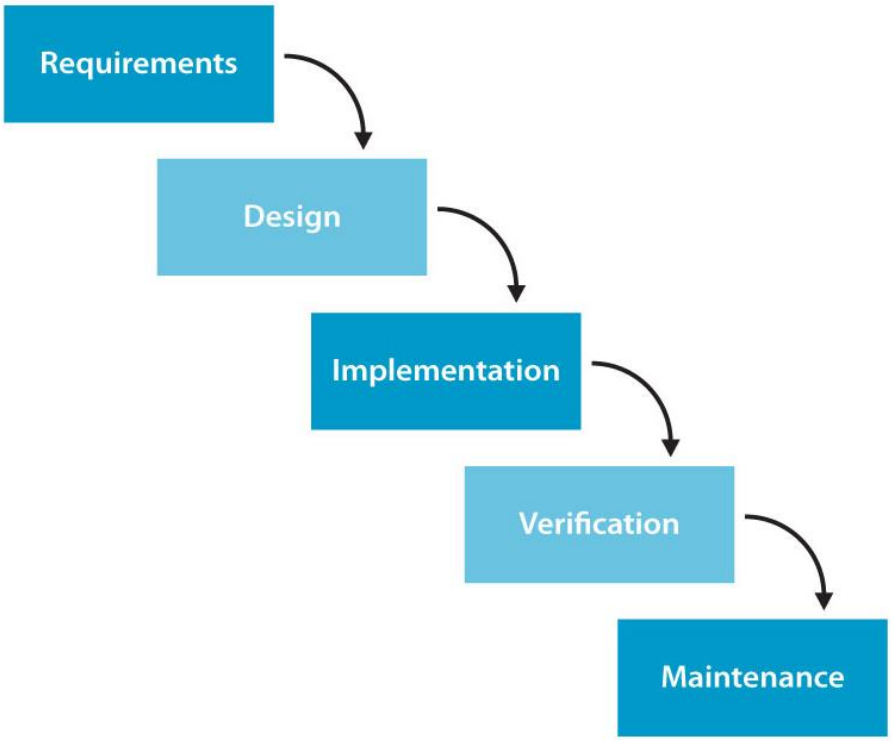
Kemudian risiko-risiko yang sudah terdaftar akan dianalisis dengan metode analisis kualitatif dan kuantitatif. Analisis kualitatif merupakan proses memprioritaskan risiko berdasarkan kemunkinan terjadi dan dampaknya jika terjadi. Analisis kuantitatif mencoba memperkirakan efek dari risiko pada tujuan dari proyek secara numerik.

Setelah risiko berhasil dianalisis, tim dan manajer proyek sudah dapat membuat perencanaan mengenai bagaimana cara menangani risiko-risiko tersebut. Perencanaan tanggapan risiko merupakan tahapan untuk meningkatkan peluang terjadinya risiko positif atau peluang dan mengurangi kemungkinan risiko negatif atau ancaman yang dapat membahayakan proyek.

Tahap terakhir dari manajemen risiko konvensional adalah pemantauan dan pengendalian. Pemantauan dan pengendalian merupakan pemantauan terhadap risiko yang telah teridentifikasi atau risiko yang muncul sebagai dampak dari penanganan risiko, pengidentifikasian risiko baru, membuat perencanaan tanggapan risiko, dan pengevaluasian keefektifan dari penggunaan strategi tertentu untuk menangani risiko.

3.2Software Development Life Cycle (SDLC)

Software Development Life Cycle atau SDLC merupakan metodologi untuk mendesain, membangun, dan mempertahankan informasi di sistem industri(Bassil, 2012). Berbagai macam model SDLC juga dikembangkan seperti Waterfall, Spiral, V-Model dan lain-lain. Waterfall merupakan salah satu SDLC yang paling banyak digunakan dalam pengembangan software (Mahalakshmi & Sundararajan, 2013).Ini dikarenakan metode Waterfall sudah diperkenalkan oleh Winston Royce pada tahun 1970.SDLC mempunyai tahapan-tahapan untuk membangun sebuah perangkat lunak seperti perencanaan, analisis, desain, implementasi, pengujian, dan pemeliharaan. Gambar proses yang terjadi pada metodologi pengembangan perangkat lunak Waterfall dapat dilihagt pada gambar 3.1.



Sumber: http://www.sitssolutions.com/IT/testing_services.php

Gambar 3.1 Tahapan dalam Metode Waterfall.

Tahapan yang ada pada metodologi Waterfall dimulai dengan fase *requirement analysis*atau analisis kebutuhan. Pada fase ini, pengembang perangkat lunak akan mengumpulkan kebutuhan yang diperlukan untuk mengetahui produk perangkat lunak yang dibangun. Kebutuhan didapatkan dari *client* yang ingin dibuatkan perangkat lunak.

Tahapan kedua adalah proses desain. Proses desain merupakan proses untuk mengembangkan *mock up* desain produk yang akan dibuat. Selain itu, tahap desain juga akan menghasilkan hal-hal teknis seperti desain basis data, diagram kelas dan lain-lain.

Tahapan ketiga adalah proses implementasi. Proses implementasi merupakan proses bagi pengembang untuk melakukan pengkodean sesuai dengan desain yang telah direncanakan sebelumnya. Hasil dari proses ini adalah perangkat lunak yang sesuai dengan perencanaan.

Tahapan keempat adalah proses *verification* atau proses *testing*. Proses ini merupakan proses untuk menguji apakah perangkat lunak yang dibangun telah sesuai dengan apa yang direncanakan. Proses ini biasanya dilakukan oleh Quality Assurance untuk menguji perangkat lunak dan mencari *bugs* yang anda.

Tahapan terakhir adalah proses pemeliharaan. Proses pemeliharaan ini adalah proses untuk menjaga kehandalan dari perangkat lunak. Selain itu, di proses ini juga dilakukan untuk mencegah jika adanya *bugs* atau *error* yang tidak teridentifikasi sebelumnya pada perangkat lunak yang dibangun.

3.3Agile

Agile merupakan *mindset* yang dibangun untuk membuat suatu sistem pembangunan perangkat lunak yang fleksibel dan dapat menanggapi perubahan. Agile sendiri muncul ketika ada sekelompok orang dari ahli industri bertemu untuk membangun gagasan berupa nilai dan prinsip yang akan mengizinkan tim pengembang perangkat lunak agar dapat bekerja dengan cepat untuk menanggapi perubahan yang terjadi (Martin, 2012). Dari pertemuan tersebut dihasilkan nilai-nilai Agile beserta dengan prinsip-prinsipnya.

Pada zaman dimana kebutuhan dapat berubah dengan cepat, orang-orang semakin tertarik dengan prinsip Agile. Jumlah kerangka kerja pengembangan perangkat lunak yang mengikuti prinsip Agile semakin meningkat. Kerangka kerja Agile yang banyak digunakan dalam pengembangan perangkat lunak antara lain Scrum, Extreme Programming, Crystal, Feature Driven Development, Lean Development dan lain-lain. Menurut survei yang dilakukan oleh Versionone, dari 6.042 responden, Scrum yang memegang 52 persen suara menjadi yang paling banyak digunakan dibandingkan dengan kerangka kerja Agile lainnya.

3.4Scrum

Scrum merupakan sebuah *framework* yang membuat orang bisa mengatasi permasalahan adaptif yang kompleks, sembari dengan produktif dan kreatif menghasilkan produk dengan kemungkinan nilai tertinggi.

Scrum sendiri sudah digunakan sejak awal tahun 1990-an untuk mengatur pengembangan produk yang kompleks (Schwaber &Sutherland, 2013). Scrum sendiri disebut sebagai sebuah *framework* atau kerangka kerja dan bukan disebut sebagai metodologi. Dalam buku Joshua Partogi yang berjudul Manajemen Modern dengan Scrum, ia menjelaskan bahwa Scrum tidak menjelaskan dengan detail mengenai langkah-langkah yang harus dilakukan tim Scrum seperti yang ada pada metodologi pengembangan perangkat lunak lainnya. Scrum hanya menjelaskan apa yang perlu ada dalam Scrum (Partogi, 2015).

3.4.1 Tim Scrum

Tim yang bekerja pada kerangka kerja Scrum terdiri dari Product Owner, Tim Pengembang, dan seorang Scrum Master(Schwaber & Sutherland, 2013). Scrum dapat dijalankan dengan baik apabila tim Scrum merupakan tim yang dapat *self-organizing*. Didalam kerangka kerja Scrum, tim Scrum dituntut untuk dapat menentukan sendiri bagaimana cara terbaik yang dapat mereka lakukan untuk dapat menyelesaikan pekerjaan mereka dari pada harus diarahkan oleh orang lain diluar tim.

a. Product Owner

Dalam kerangka kerja Scrum, Product Owner atau biasa disebut Product Manager adalah orang yang bertanggung jawab untuk menentukan nilai dari produk dan apa saja yang harus dibuat oleh Development Team. Product owner berkewajiban untuk mengelola Product Backlog Item atau disingkat PBI(Partogi, 2015). Dalam mengelola Product Backlog, Product Owner harus dapat

membuat Product Backlog Item yang jelas. Product Owner juga harus mengurutkan Product Backlog sesuai dengan prioritas agar dapat mencapai tujuan yang ingin dicapai organisasi atau perusahaan. Product Owner juga harus mengoptimalkan nilai dari pekerjaan yang dikerjakan oleh tim pengembang. Product Owner berusaha untuk membuat Product Backlog transparan dan jelas agar tim Scrum dapat mengetahui apa yang akan dikerjakan selanjutnya. Selain itu, Product Backlog yang dibuat oleh Product Owner juga harus dimengerti oleh tim pengembang.

b. Tim Pengembang

Tim Pengembang merupakan orang-orang yang bekerja untuk mengembangkan *incremental product*. Produk yang dihasilkan oleh tim pengembang disebut sebagai *incremental product* karena tim pengembang akan menghasilkan produk secara iteratif pada setiap akhir Sprint. Tim pengembang merupakan tim yang *cross-functional*(Schwaber & Sutherland, 2013). Tim yang *cross-functional* berarti tim yang terdiri dari orang-orang yang memiliki kemampuan berbeda yang diperlukan dalam sebuah tim. Tim bekerja sama untuk dapat menghasilkan produk secara iteratif. Didalam Scrum Guides, tim pengembang disarankan terdiri dari 3 hingga 9 orang.

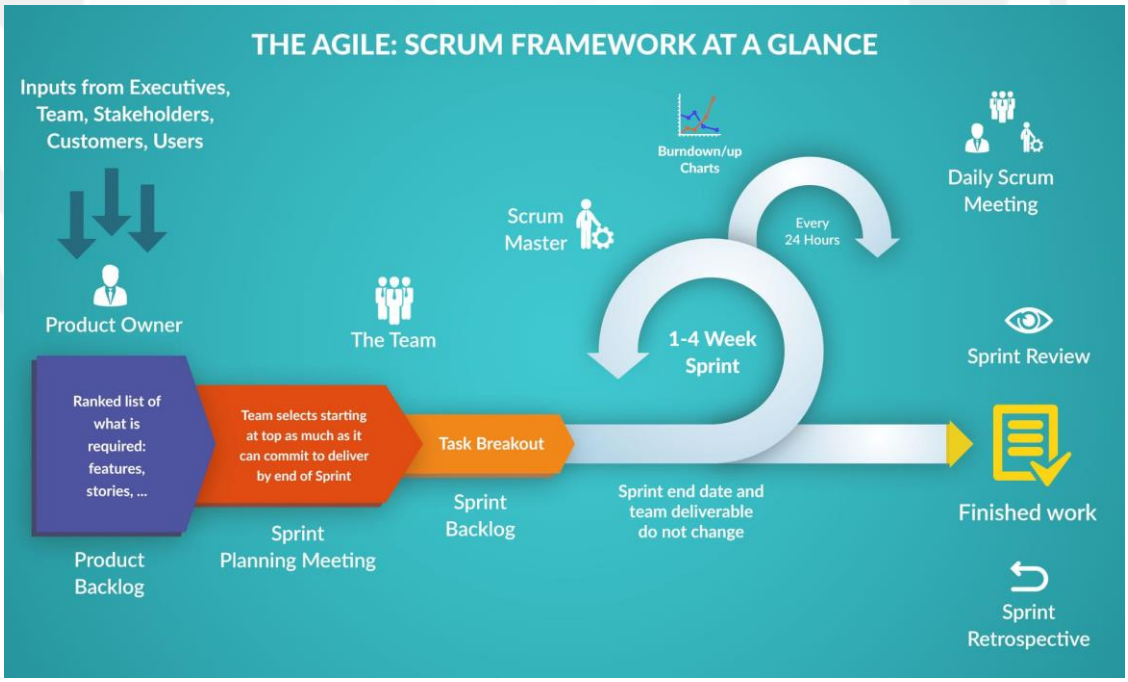
c. Scrum Master

Scrum Master adalah orang yang bertugas untuk memastikan bahwa Scrum dipahami dan dijalankan dengan benar(Schwaber & Sutherland, 2013). Scrum Master

haruslah orang yang sangat memahami prinsip Agile dan kerangka kerja Scrum untuk dapat mengedukasi tim Scrum agar dapat bekerja dengan baik dalam kerangka kerja Scrum. Scrum Master harus mencegah adanya kesalahan *mindset* Agile didalam proses pengembangan perangkat lunak yang dilakukan tim pengembang. Untuk itu, Scrum Master harus menjadi fasilitator dalam acara Scrum.

3.4.2 Proses Scrum

Kerangka kerja Scrum ini memiliki beberapa acara yang perlu diikuti oleh pengguna Scrum. Tahapan-tahapan pada kerangka kerja Scrum antara lain, Sprint Planning, Sprint, Daily Scrum, Sprint Review dan Sprint Retrospective. Gambar proses Scrum secara keseluruhan dapat dilihat pada gambar 3.2.



Sumber: http://ddi-dev.com/blog/it-news/scrum-and-agile-when-you-should-adopt-them/

Gambar 3.2 Proses Scrum

Berdasarkan pada gambar, Sprint dimulai dari *input* kebutuhan oleh pihak eksekutif, tim Scrum, *stakeholder*, pelanggan atau pengguna produk yang ditampung oleh Product Owner. Product Owner menyusun kebutuhan tersebut menjadi Product Backlog. Product Backlog merupakan daftar segala sesuatu yang dibutuhkan dalam produk yang terurut berdasarkan prioritas (Schwaber & Sutherland, 2013). Product Backlog ini akan terus diisi ketika ada kebutuhan baru yang muncul.

Kebutuhan dalam bentuk Product Backlog ini akan digunakan pada Sprint Planning. Sprint Planning merupakan acara bagi tim Scrum untuk merencanakan apa saja yang akan dikerjakan pada Sprint berdasarkan Product Backlog yang ada (Schwaber & Sutherland, 2013). Sprint Planning diadakan dalam waktu yang terbatas yakni paling lama 8 jam untuk Sprint yang berdurasi 1 bulan. Scrum Master harus menjaga agar tim tidak melakukan Sprint Planning melebihi waktu yang ditetapkan. Hal yang dibahas dalam Sprint Planning biasanya adalah apa yang akan dikerjakan pada Sprint ini dan bagaimana pekerjaan yang telah dipilih dapat diselesaikan agar dapat mencapai tujuan Sprint (Schwaber & Sutherland, 2013).

Setelah dilakukan Sprint Planning, tim akan memulai mengerjakan proses pengembangan. Proses pengembangan ini berdurasi satu hingga empat minggu. Dalam melakukan proses pengembangan, setiap harinya tim pengembang akan melakukan Daily Scrum. Daily Scrum merupakan acara berdurasi 15 menit untuk tim pengembang agar dapat mensinkronisasi pekerjaan (Schwaber &

Sutherland, 2013). Hal yang dibahas didalam Daily Scrum antara lain:

- Apa yang saya lakukan kemarin yang dapat membantu tim pengembang mencapai tujuan Sprint?
- Apa yang akan saya lakukan hari ini untuk membantu tim pengembang mencapai tujuan Sprint?
- Apakah ada halangan atau hambatan yang mencegah saya atau tim pengembang dalam mencapai tujuan Sprint?

Ketiga pertanyaan tersebut harus dijawab oleh masing – masing anggota di dalam Daily Scrum.

Diakhir Sprint, ada acara Sprint Review yang merupakan acara untuk meninjau kembali hasil pekerjaan tim pengembang selama Sprint berlangsung (Partogi, 2015). Didalam acara ini, tim Scrum akan mendemokan hasil pekerjaan selama satu Sprint kepada *stakeholder* terkait. Dengan demikian, hasil pekerjaan tim Scrum akan menjadi transparan untuk pihak-pihak terkait. Sama seperti acara Scrum lainnya, acara ini juga dibatasi waktunya dengan waktu paling lama 4 jam untuk Sprint yang berdurasi 1 bulan. Untuk Sprint yang berdurasi kurang dari 1 bulan, waktu Sprint Review juga akan lebih cepat.

Selain membahas hasil pekerjaan yang telah dilakukan tim Scrum selama satu Sprint pada Sprint Review, tim Scrum juga memiliki acara untuk membahas proses yang berjalan selama mengembangkan produk perangkat lunak. Pada Scrumguides.org, dijelaskan bahwa tujuan adanya Sprint Retrospective adalah untuk meninjau kembali bagaimana Sprint yang telah berjalan berlangsung dengan melihat dari orang, hubungan, proses dan *tools*. Selain itu, Sprint Retrospective juga

menjadi ajang untuk mengidentifikasi hal-hal apa saja yang telah berjalan dengan baik pada Sprint sebelumnya dan apa saja yang dapat dikembangkan lagi kedepannya. Didalam Sprint Retrospective juga dibuat rencana untuk mengimplementasikan perbaikan pada tim Scrum.