



**Memperkenalkan *Scrum* Sebagai Kerangka Kerja Perusahaan
Demi Memenuhi Tantangan Perubahan Pasar
Di Era Revolusi Industri 4.0 Yang Dinamis**

Oleh :
Ni Putu Diah Untari Ningsih
STKIP Agama Hindu Amlapura
niputudiahun@gmail.com

Keywords:

*Scrum,
Framework,
Industrial
Revolution 4.0,
Agile*

Abstract

The digital era of the dynamic industrial revolution 4.0, requires companies to quickly innovate to make products that can meet the needs of consumers and released in a short time. This method is to answer the company's efforts to stay ahead of competitors. Building software requires a framework to facilitate the planning, development and maintenance. The Scrum method can be used as a framework for developing software interoperability with a faster time (Agile). Scrum was made by Jeff Sutherland and Ken Schwaber, who was a developer in the mid-1990s in America. Thanks to this finding, it can lead technology companies like Amazon, Google, and Facebook to become giants until now. The Scrum method has a work pattern of breaking jobs into smaller tasks (tasks) and is distributed among the Development Team on a regular basis in a short cycle of time so that system development can be completed quickly. Scrum is recommended for project types that are not deadline oriented, with creative and high-quality product targets. Because Scrum's focus is on the process, with iteratives and targets per Sprint, it is hoped that each piece of product will be better than the previous Sprint. The advantages of the Scrum method in this study such as the work pattern of breaking work into tasks that are evenly distributed to all development teams to be completed in a fast work cycle requires excessive accuracy in determining the Backlog.

Kata Kunci:

*Scrum, Kerangka
Kerja, Revolusi
Industri 4.0, Agile*

Abstrak

*Era digital revolusi industri 4.0 yang dinamis, menuntut perusahaan untuk cepat berinovasi membuat produk yang dapat menjawab kebutuhan konsumen dan dirilis dalam waktu singkat. Cara ini untuk menjawab upaya perusahaan agar tetap terdepan dibandingkan kompetitor. Membangun Perangkat lunak diperlukan adanya kerangka kerja untuk dapat memudahkan dalam proses perencanaan, pengembangan, hingga pemeliharannya. Metode *Scrum* dapat digunakan*

sebagai kerangka kerja untuk mengembangkan interoperabilitas perangkat lunak dengan waktu yang lebih cepat (*Agile*). Scrum dibuat oleh Jeff Sutherland dan Ken Schwaber yang merupakan seorang pengembang pada medio tahun 1990-an di Amerika. Berkat temuannya ini, bisa menggiring perusahaan teknologi seperti *Amazon*, *Google*, dan *Facebook* menjadi raksasa sampai sekarang. Metode *Scrum* memiliki pola kerja memecah pekerjaan menjadi tugas yang lebih kecil (*task*) dan terdistribusi di antara Tim Pengembang secara rutin dalam siklus waktu yang singkat sehingga pengembangan sistem dapat diselesaikan dengan cepat. *Scrum* direkomendasikan untuk tipe proyek yang tidak berorientasi pada *deadline*, dengan target produk yang kreatif dan berkualitas tinggi. Karena fokus *Scrum* adalah pada proses, dengan iteratif dan target per *Sprint*, diharapkan setiap potongan produk, akan menjadi lebih baik dari *Sprint* sebelumnya. Keunggulan dari metode *Scrum* pada penelitian ini seperti pola kerja memecah pekerjaan menjadi *task* yang terdistribusi merata ke semua tim pengembang untuk diselesaikan dalam siklus kerja yang cepat diperlukan adanya ketelitian yang berlebih dalam menentukan *Backlog*.

Pendahuluan

Era digital revolusi industri 4.0 yang dinamis, menuntut perusahaan untuk cepat berinovasi membuat produk yang dapat menjawab kebutuhan konsumen dan dirilis dalam waktu singkat. Cara ini untuk menjawab upaya perusahaan agar tetap terdepan dibandingkan kompetitor. Pengembangan perangkat lunak diakui oleh banyak pengembang merupakan suatu proses yang rumit, tentang bagaimana melakukan pengembangan perangkat lunak secara benar, efisien dan tepat. Seluruh aspek tersebut bukanlah merupakan skill yang secara lahiriah diperoleh masing-masing individu sejak lahir, bukan dimiliki secara langsung dari proses belajar programming, melainkan dengan terjun langsung ke dalam proses ini.

Scrum merupakan sebuah kerangka kerja dimana pihak-pihak dapat mencari jalan keluar dari permasalahan yang kompleks dan pada saat yang bersamaan membuat produk yang memiliki nilai setinggi mungkin secara produktif dan kreatif (Schwaber, et al, 2011). Scrum dibuat oleh Jeff Sutherland dan Ken Schwaber yang merupakan seorang pengembang pada medio tahun 1990-an di Amerika. Berkat temuannya ini, bisa menggiring perusahaan teknologi seperti *Amazon*, *Google*, dan *Facebook* menjadi raksasa sampai sekarang. *Scrum* sendiri merupakan sebuah kerangka kerja sederhana

untuk mengembangkan produk yang dikembangkan oleh Jeff Sutherland dan Ken Schwaber semenjak tahun 1995.

Scrum banyak digunakan oleh perusahaan-perusahaan software seperti *Microsoft*, *Adobe*, *Salesforce.com*, *Google*, *Netflix*, *Spotify*, dan masih banyak lagi (Partogi, 2014). *Scrum* sendiri bukan satu-satunya metode yang menggunakan pendekatan AGILE. Mungkin kita juga pernah mendengar metode Extreme Programming (XP) yang juga menggunakan pendekatan AGILE dalam rekayasa perangkat lunak. Masing-masing metode memiliki fokus atau penekanan yang berbeda yang tentu saja dapat dikombinasikan untuk menghasilkan proses yang optimal.

Teknik *Scrum* dapat dilakukan disebuah kepanitiaan ataupun project lain diluar bisnis teknologi informasi. Dalam teknik *Scrum* terbagi dalam tiga *roles*, yang pertama adalah *Product Owner*, *Scrum Master* dan *Development/Scrum Team*. *Product owner* bertugas mengatur urusan dengan Stakeholder sedangkan *Scrum Master* mengurus bagian internal, di bagian *Development Team* mengatur urusan teknik pengerjaan project dan pembahasan yang lebih rinci. Banyak perusahaan multinasional menginternalisasi atau mengadopsi teknik ini sebagai standar bekerja mereka, karena secara umum teknik *Scrum* berhasil membuat beberapa perusahaan menaikkan omset karena system yang telah teruji ini. Teknik *Scrum* membuat pekerjaan anda menjadi lebih tertata dan lebih detail.

Perusahaan-perusahaan multinational di Indonesia yang mengadopsi metode *Scrum* masih sangat sedikit, walaupun ada itu hanya segelintir perusahaan yang memakai metode ini, karena pada umumnya tidak akan menjadi sebuah standar dan tidak akan berkembang sebagaimana di negara-negara barat bahkan tidak akan berkembang pesat seperti di negara-negara Asia Tenggara lainnya seperti Malaysia, Singapura dan Filipina, permasalahan ini yang telah diungkapkan oleh Joshua Partogi yang merupakan *Professional Scrum Trainer* pertama dari Asia yang ditraining secara langsung oleh *co-founder Scrum*, Ken Schwaber.

Orang Indonesia masih kesulitan melihat dari sudut pandang produk yang dapat meningkatkan *revenue* perusahaan dan selalu melihat dari sudut pandang proyek yang memiliki batasan *budget*. Tidak peduli berapa kali dikatakan kalau *Scrum* hanyalah sebuah kerangka kerja untuk mengembangkan produk, orang Indonesia akan selalu kembali melihatnya dari sudut pandang manajemen proyek. Sehingga sampai sekarang *Scrum* tidak populer atau berkembang di Indonesia, karena pola pikir masyarakatnya.

Namun seiring pesatnya perkembangan dunia *start-up* berbasis teknologi informasi, *Scrum* mulai banyak diterapkan sebagai kerangka kerja dalam *software development*. Karena, *Scrum* dapat membuat proyek yang akan atau sedang dikerjakan menjadi lebih rapi, dan segala sesuatunya dapat dipastikan selesai apabila sudah memiliki persiapan yang matang dari awalnya.

Scrum dikembangkan oleh Jeff Sutherland dan Ken Schwaber di awal tahun 90-an. Mereka mengkodifikasikan *Scrum* pada tahun 1995 pada konferensi OOPSLA di Austin, Texas (AS) dan menerbitkan kertas “*Scrum Software Development Process*”. Uniknya, nama *Scrum* diambil Ken dan Jeff berdasarkan sebuah kertas “*The New Product Development Game*” yang dibuat oleh Takeuchi dan Nonaka. Istilah *Scrum* diambil dari permainan rugby yang menekankan pentingnya kerjasama tim dalam bermain olahraga tersebut dan menjadi sukses dalam hal permainan pengembangan produk baru (Anonim, 2016).

Scrum merupakan sebuah kerangka kerja untuk mengembangkan sebuah produk yang kompleks, dimana visi dari *Scrum* adalah produk yang bernilai tinggi secara kreativitas maupun produktivitas. *Scrum* didasari oleh teori kontrol empiris yang disebut empirisme, yang menekankan bahwa pengetahuan didapatkan dari pengalaman, sehingga pembuatan keputusan sebaiknya didasari atas pengetahuan tersebut. *Scrum* menggunakan pendekatan secara *iterative* dan *incremental* untuk mengantisipasi ketidakpastian dalam proyek dan pengendalian resiko.



Gambar 1. Komponen *Scrum*

Scrum terdiri dari beberapa komponen, yaitu:

1. Roles

a. Pemilik Produk (Owner)

Pemilik Produk bertanggung jawab untuk memaksimalkan nilai dari produk dan hasil kerja dari Tim Pengembang. Cara untuk melakukan ini akan beragam di setiap organisasi, Tim *Scrum*, dan masing-masing individu. Pemilik Produk adalah orang yang bertanggung-jawab untuk mengelola *Product Backlog*. Pengelolaan *Product Backlog* mencakup:

- 1) Menjabarkan item *Product Backlog* secara jelas;
- 2) Mengurutkan item-item di dalam *Product Backlog* untuk dapat mencapai misi dan tujuan dengan cara terbaik;
- 3) Menilai hasil pekerjaan dari Tim Pengembang;
- 4) Memastikan bahwa *Product Backlog* kelihatan, transparan, dan jelas bagi semua pihak, dan menentukan *Product Backlog* mana yang harus dikerjakan selanjutnya oleh Tim *Scrum*; dan,
- 5) Memastikan bahwa Tim Pengembang dapat memahami item di dalam *Product Backlog* hingga batasan yang mampu diberikan oleh Pemilik Produk.

b. Tim Pengembang

Tim Pengembang terdiri dari para ahli yang bekerja untuk menghasilkan potongan produk “Selesai” yang berpotensi untuk dirilis di setiap akhir Sprint. Hanya anggota tim yang menyelesaikan potongan produk ini. Tim Pengembang distrukturisasi dan didukung oleh organisasi untuk mengatur dan mengelola pekerjaannya secara mandiri. Sinergi yang dihasilkan akan meningkatkan efisiensi dan efektifitas dari Tim Pengembang secara keseluruhan. Tim Pengembang memiliki karakteristik sebagai berikut:

- 1) Mereka mengatur dirinya sendiri. Tidak ada satu orang pun (bahkan *Scrum Master* sekalipun) yang memerintah Tim Pengembang bagaimana caranya menghasilkan potongan fungsionalitas dari *Product Backlog* yang berpotensi untuk dirilis;
- 2) Tim Pengembang bersifat antar lintas, yang artinya semua keahlian yang dibutuhkan untuk menghasilkan potongan dari produk harus ada di dalam tim;
- 3) *Scrum* tidak mengenal adanya jabatan tertentu untuk anggota tim selain Pengembang, apapun itu pekerjaan yang dikerjakan oleh masing-masing anggota tim; tidak ada pengecualian untuk aturan yang satu ini;

- 4) Individu Tim Pengembang boleh memiliki spesialisasi keahlian dan fokus di satu area tertentu, namun hasil dari pekerjaan secara keseluruhan adalah milik Tim Pengembang; dan,
- 5) Tim Pengembang tidak memiliki sub-tim yang dikhususkan untuk bidang tertentu seperti pengujian atau analisa bisnis.

c. *Scrum Master*.

Scrum Master bertanggungjawab untuk memastikan *Scrum* telah dipahami dan dilaksanakan. *Scrum Master* melakukan ini dengan berpedoman pada teori, praktik, dan aturan main *Scrum*. *Scrum Master* adalah seorang pemimpin yang melayani Tim *Scrum*. *Scrum Master* membantu pihak yang berada di luar Tim *Scrum* memahami interaksi dengan Tim *Scrum* mana yang bermanfaat dan mana yang tidak. *Scrum Master* membantu setiap pihak untuk merubah interaksi ini untuk memaksimalkan nilai yang dihasilkan oleh Tim *Scrum*.

2. *Ceremonies*

- a. *Sprint Planning*: Pemilihan produk yang akan dikembangkan dan membuat *High-Level Design*
- b. *Sprint Review*: Rapat informal pada akhir sprint. Dimana tim mendemokan apa yang telah dikerjakan pada *management, customer dan product owner*.
- c. *Sprint Retrospective*: Rapat yang terdiri dari *Scrum Master* dan *Scrum Team* yang dilaksanakan pada akhir dari masing-masing *Sprint*. Membahas apa yang akan dilakukan dan dikembangkan pada *Sprint* selanjutnya
- d. *Daily Scrum Meeting*: Rapat Harian (setiap hari pada hari kerja) yang dilakukan untuk memonitor pekerjaan yang telah diselesaikan pada hari kemarin, pekerjaan apa yang akan dilakukan, serta permasalahan yang mungkin ada pada saat pengerjaan produk

3. *Artifacts*

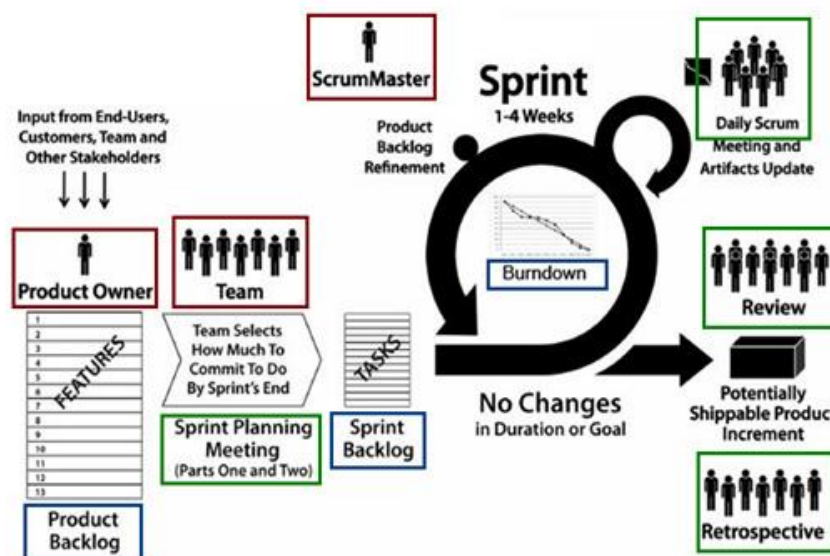
- a. *Product Backlog*: Daftar fitur yang akan dikembangkan beserta prioritasnya.
- b. *Sprint Backlog*: Fitur yang diprioritaskan untuk dikembangkan.
- c. *Burndown Charts*: Perkiraan harian yang digunakan untuk plot sprint burndown, dan mengetahui pekerjaan yang tersisa.

Product owner akan memulainya dengan mengkonversi segala keinginan dari pelanggan menjadi *product backlog* dimana *product backlog* adalah segala list terkait dengan keberhasilan proyek, dalam pengembangan software bisa dalam bentuk fitur

ataupun bagaimana arsitektur sistem yang akan dibuat. Lalu *product backlog* itu akan dibawa kepada *sprint planning meeting* yang akan didatangi oleh semua team agar dapat memberikan pengetahuan kepada semua anggota team bagaimana sistem yang akan dibuat sehingga dari situ akan dibuatlah *sprint backlog* yaitu apa saja yang akan dilakukan agar pengembangan dapat berhasil.

Setelah *sprint backlog* dibuat maka akan dilakukan lah *sprint* yaitu proses pengembangan atau pembuatan sistem yang biasanya berlangsung selama 1-4 minggu, saat proses *sprint* tidak boleh ada gangguan dari luar, yang dilakukan oleh team adalah tetap mengikuti dari *sprint backlog* yang sudah dibuat. Selama pelaksanaan *sprint* diadakan *daily Scrum* yaitu pertemuan yang dihadiri oleh semua anggota team untuk sharing masalah apa saja yang mereka hadapi saat pengembangan kemarin. Saat jangka waktu pelaksanaan suatu *sprint* selesai maka akan dilaksanakan *review* yang dimana menunjukkan fitur apa saja yang sudah diselesaikan pada proses *sprint* yang sudah dilakukan.

Lalu akan dilakukan lagi proses *sprint* dengan *sprint backlog* yang berbeda hingga semua *product backlog* dapat dipenuhi semuanya, saat team merasa *product backlog* dapat di *refinemenet* atau dievolusi maka team dapat melakukannya saat proses *sprint*. Saat semua *product backlog* terpenuhi maka akan dilaksanakan *retrospective* yang berguna untuk mereview secara keseluruhan kinerja dari *project owner*, *Scrum master* dan juga team yang nantinya akan digunakan sebagai *feedback* agar proyek selanjutnya dapat lebih baik lagi. Untuk lebih jelasnya dapat dilihat pada gambar 2.



Gambar 2. Proses Dari Bagaimana *Framework Scrum* Akan Dilakukan

Scrum direkomendasikan untuk tipe proyek yang tidak berorientasi pada deadline, dengan target produk yang kreatif dan berkualitas tinggi. Karena fokus *Scrum* adalah pada proses, dengan iteratif dan target per Sprint, diharapkan setiap potongan produk, akan menjadi lebih baik dari Sprint sebelumnya. Salah satu implementasi dari *Scrum* adalah Proyek Pengembangan Perangkat bergerak Radio XYZ. Proyek Radio XYZ menggunakan *Scrum* dalam pengembangannya. Penggunaan *Scrum* harus mencocokkan apakah proyek yang kita jalani cocok atau tidak untuk mengimplementasikan *Scrum* dikarenakan *Scrum* juga memiliki beberapa kelemahan yaitu *Scrum* sangat kurang dalam hal dokumentasi sehingga kita harus memastikan apakah dokumentasi proyek sangat berpengaruh pada proyek anda atau tidak (Mahalaksmi & Sundararajan, 2013).

Selain itu kerjasama tim sangat dibutuhkan dalam menggunakan *Scrum* sehingga bila kerjasama tim kurang dan dedikasi untuk mengerjakan proyek kurang maka proyek akan gagal, juga harus bersedia dan selalu siap untuk menerima perubahan yang terjadi di dalam proses pengembangan. Tim terdistribusi yang menggunakan *Scrum* secara global menghadapi tantangan pada proses komunikasi, kontrol, dan koordinasinya. Hal ini dapat dipengaruhi oleh faktor distribusi yang memiliki jarak baik secara geografis maupun sosial-budaya. Perbedaan budaya ini nantinya dapat mempengaruhi proses kolaborasi (Hossain, et al, 2009).

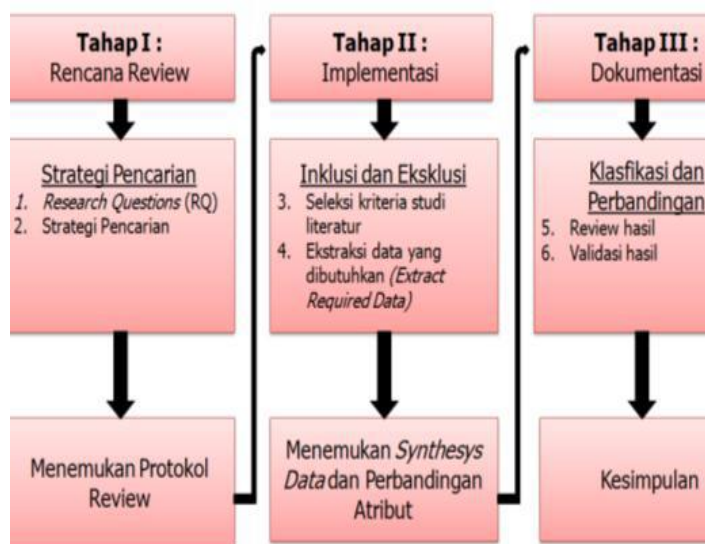
Walaupun memiliki beberapa kelemahan, *Scrum* juga memiliki beberapa kelebihan. Berikut ini merupakan kelebihan *Scrum*, yaitu:

1. Mudah menyesuaikan diri dengan perubahan yang terjadi pada relitas bisnis dengan cepat
2. Tim yang besar dipecah menjadi team-team kecil yang disebut Team *Scrum*. Sehingga memperlancar komunikasi, mengurangi biaya, dan saling memberdayakan satu sama lain.
3. Dokumentasi dan pengujian selama proses pengembangan dilakukan secara terus-menerus
4. Proses *Scrum* dapat menyatakan bahwa suatu proses pengembangan “sudah selesai” kapan saja.

Metode

Metode penelitian yang akan digunakan adalah kajian pustaka/literatur yakni dengan mengkaji literatur-literatur yang berkaitan dengan topik penelitian. Teori dan pemikiran yang tertuang dalam sumber pustaka digunakan sebagai dasar bagi peneliti untuk mendeskripsikan, menganalisis dan memunculkan ide-ide baru yang berkaitan dengan *Scrum*. Metode penelitian ini didasarkan pada penerapan metode SLR (*Systematic Literature Review*) (Qadir dan Usman, 2011) (Jamshidi, dkk., 2013) sebagai suatu cara untuk melakukan ulasan penelitian dengan melakukan klasifikasi dan perbandingan terhadap beberapa penelitian yang telah ada baik berupa artikel, jurnal, atau *conference*.

Penelitian ini memiliki tiga tahapan analisis data, antara lain: (i) Rencana review yang mendefinisikan *protocol review* untuk menentukan pertanyaan penelitian. (ii) Implementasi dibutuhkan kriteria inklusi dan eksklusi untuk mendapatkan kriteria kata kunci utama yang berhubungan dengan pertanyaan penelitian. (iii) Dokumentasi yang bertujuan untuk memberikan informasi yang diperoleh dari setiap ulasan berupa ringkasan. SLR menjelaskan proses pencarian dokumen secara relevan dari pertanyaan penelitian yang diajukan. Gambar 3 menunjukkan metode penelitian berbasis SLR.



Gambar 3 Metode Penelitian Kajian Literatur

A. Tahap I

1. *Research question* digunakan untuk mendefinisikan kriteria atau atribut pertanyaan yang menjadi acuan kerangka pertanyaan.

2. Strategi Pencarian yaitu sebuah langkah dalam proses pencarian pertanyaan.

B. Tahap II

1. Seleksi Kriteria Inklusi dan Eksklusi,

Diperoleh hasil pencarian yang sangat banyak, oleh sebab itu untuk membatasi ruang lingkup pencarian maka dibutuhkan penyaringan kata kunci dengan menggunakan penyaringan berdasarkan kriteria inklusi sebagai berikut:

- a) Fokus pada analisa Scrum *framework*.
- b) Sumber dalam bahasa Inggris dan memberikan jawaban atas pertanyaan penelitian dan kata kunci penelitian.
- c) Publikasi literatur dalam bentuk jurnal atau *conference* mulai tahun 2010 hingga 2020.

Untuk mempercepat pencarian, maka diperlukan suatu cara agar dapat mempersempit ruang lingkup pencarian yang mengabaikan literatur hasil penyaringan eksklusi dengan kondisi sebagai berikut:

- a) Literatur yang menggunakan bahasa selain bahasa Inggris.
- b) Literatur yang tidak terkait dengan pertanyaan dan kata kunci penelitian.
- c) Sumber yang hanya memberikan informasi dalam bentuk abstrak.
- d) Literatur ganda dari sumber pencarian.

2. Ekstraksi Data dan Sintesis

Selanjutnya adalah ekstraksi jurnal, *conference* secara manual yang ada kaitannya dengan judul, abstrak, kata kunci dan referensi yang sesuai dengan pertanyaan penelitian dan kata kunci penelitian. Penelitian yang dilakukan ini tidak memberikan sebuah usulan penyelesaian dengan pendekatan tertentu, karena *review* studi pustaka diperoleh dari sumber literatur yang sudah ada.

C. Tahap III

Dokumentasi dan Validasi Hasil adalah kumpulan jurnal dan *conference* hasil ekstraksi kemudian divalidasi dengan tujuan untuk mengetahui gambaran analisa penerapan *Scrum framework* yang dituangkan dalam bentuk ringkasan dan ulasan pada bagian pembahasan.

Hasil dan Pembahasan

A. Memulai *Scrum*

Teknik *Scrum* dimulai dari *Product Owner* yang membahas proses pengembangan perangkat lunak dengan membuat list pekerjaan yang disebut juga dengan *Product backlog*, yaitu gambaran bagaimana pengguna melakukan sesuatu dan berinteraksi dengan produk, biasanya berupa fitur, perbaikan kesalahan, ataupun peningkatan fitur. *Product Backlog* biasanya dibuat dalam bentuk *user story* yang kemudian ditempel pada *Task Board*. Setiap *backlog* diberikan estimasi dan standar tingkat kesulitannya dengan angka 1,2,3 dan seterusnya. *User Story* biasanya di definisikan dengan urutan sebagai berikut:

“As a ..., I want..., so that...”

Berikut adalah contoh *user story* dari pengembangan perangkat bergerak radio xyz.

Tabel 1. *User Stories*

No	<i>User Story</i>	<i>Story Point</i>
1.	Sebagai pendengar radio, saya ingin mendengarkan radio kesukaan saya tanpa menggunakan tape, sehingga saya dapat menikmatinya dimana saja.	5
2.	Sebagai pendengar radio, saya ingin melihat Top Chart dari radio yang sedang saya dengarkan, sehingga saya dapat mengetahui lagu apa saja lagu baru yang masuk di radio tersebut.	2
3.	Sebagai pendengar radio, saya ingin aplikasi radio dapat berjalan bersamaan dengan aplikasi lain, sehingga saya tetap dapat mendengarkan siaran radio.	2
4.	Sebagai pendengar radio, saya ingin melihat informasi lagu yang sedang didengarkan, sehingga saya mengetahui siapa penyanyi dan judul lagunya.	8
5	Sebagai pendengar radio, saya ingin melihat informasi penyiar yang sedang siaran, sehingga saya mengetahui nama penyiar tersebut.	3
6	Sebagai pendengar radio, saya ingin melihat daftar acara off air, sehingga saya dapat menghadiri acara tersebut.	4
7	Sebagai pendengar radio, saya ingin melihat kontak radio yang sedang saya dengar, sehingga saya dapat menghubungi radio tersebut.	6
8	Sebagai pendengar radio, saya ingin mengetahui versi aplikasi, sehingga saya dapat memastikan aplikasi yang terpasang adalah versi terbaru.	3

User Story diperbaharui secara berkala oleh *Product Owner* untuk menyesuaikan kebutuhan pelanggan, teknologi yang diterapkan, dan sebagainya.

B. Menentukan Sprint

Disetiap awal sprint dilakukan adanya *Sprint Planning Meeting*. *Product Owner*, *Scrum Master*, dan Tim Pengembang meninjau kembali Product Backlog, dan memberikan kesempatan secara langsung kepada Product Owner untuk menyampaikan idenya kepada Tim Pengembang. Selanjutnya Tim Pengembang memilih item dari Product Backlog kemudian berkomitmen untuk menyelesaikan item tersebut sampai dengan akhir sprint, pemilihan Product Backlog biasanya dimulai dari skala prioritas tertinggi bagi Product Owner. Namun, tidak menutup kemungkinan tim Pengembang memilih prioritas yang sedikit lebih rendah, yang mereka anggap bagian tersebut merupakan bagian yang harus diselesaikan terlebih dahulu dibandingkan dengan bagian yang skala prioritasnya yang lebih tinggi.

Tabel 2. Estimasi Jam Pengerjaan

<i>Sprint Length</i>			2 Minggu
<i>Workdays During Sprint</i>			10 hari
<i>Team Member</i>	<i>Available Days During Sprint</i>	<i>Available Hours Per Day</i>	<i>Total Available Hours During Sprint</i>
Diah	10 hari	5	50 jam
Untari	9 hari	4	45 jam
Ningsih	8 hari	5	40 jam

Ditahap ini *Product Owner* masih turut mendampingi memperjelas poin, memverifikasi pekerjaan, memecah item *Backlog* yang lebih besar ke dalam potongan yang lebih kecil, dan umumnya memastikan bahwa Tim Pengembang sepenuhnya memahami apa yang dimintanya. Sampai akhirnya menghasilkan daftar semua tugas untuk diselesaikan dan berapa banyak perkiraan waktu yang akan dibutuhkan (biasanya dalam hitungan jam atau fraksi dalam sehari). Untuk memudahkan Tim *Scrum* dalam memonitoring *Sprint* yang berjalan, biasanya membuat *Scrum Board* untuk melakukan pelacakan tugas secara visual, task yang dikerjakan ditulis di *Post-It Notes* dan akan dipindahkan ke dalam kolom berlabel sesuai dengan prosesnya "*Backlog*", "*In Progress*", "*Test*", dan "*Complete*".

Tabel 3. *Sprint Backlog*

<i>User Story</i>	<i>Task</i>	<i>Owner</i>	<i>Initial Time Estimate</i>
Sebagai pendengar radio, saya ingin melihat Top Chart dari radio yang sedang saya dengarkan, sehingga saya	Menyimpan data lagu yang sering diputar ke dalam tabel Top Chart	Diah	4 jam

dapat mengetahui lagu apa saja lagu baru yang masuk di radio tersebut.	Mengambil data dari tabel Top Chart untuk ditampilkan	Untari	2 jam
	Membuat fungsi perbaharui Top Chart	Ningsih	5 jam

Tabel 4. *Scrum Board*

<i>Backlog</i>	<i>In Progress</i>	<i>Test</i>	<i>Complete</i>
Menyimpan data lagu yang sering diputar ke dalam tabel Top Chart			
Mengambil data dari tabel Top Chart untuk ditampilkan			
Membuat fungsi perbaharui Top Chart			

Product Owner tidak dapat menambahkan permintaan baru selama Sprint berlangsung. Bahkan jika setengah jalan saat Sprint berlangsung *Product Owner* memutuskan bahwa mereka ingin menambahkan sesuatu yang baru, dia tidak bisa melakukan perubahan sampai dengan dimulainya Sprint berikutnya. Apabila keadaan eksternal muncul, ini berarti secara signifikan akan mengubah prioritas, dan tim akan membuang-buang waktunya jika terus berlanjut berkerja. Terpaksanya *Product Owner* menghentikan Sprint, ini berarti Tim Pengembang menghentikan semua pekerjaan mereka kemudian memulai kembali *Sprint Planning Meeting*.

- 1) Di sisi lain terdapat pengaruh positif yang didapatkan, diantaranya:
- 2) Tim Pengembang berhasil mengetahui dengan pasti kepastiannya, dan tetap berkomitmen untuk menyelesaikannya.
- 3) Mendisiplinkan *Product Owner* untuk lebih teliti dalam menentukan skala prioritas *Product Backlog*.

C. *Daily Standup Meeting*

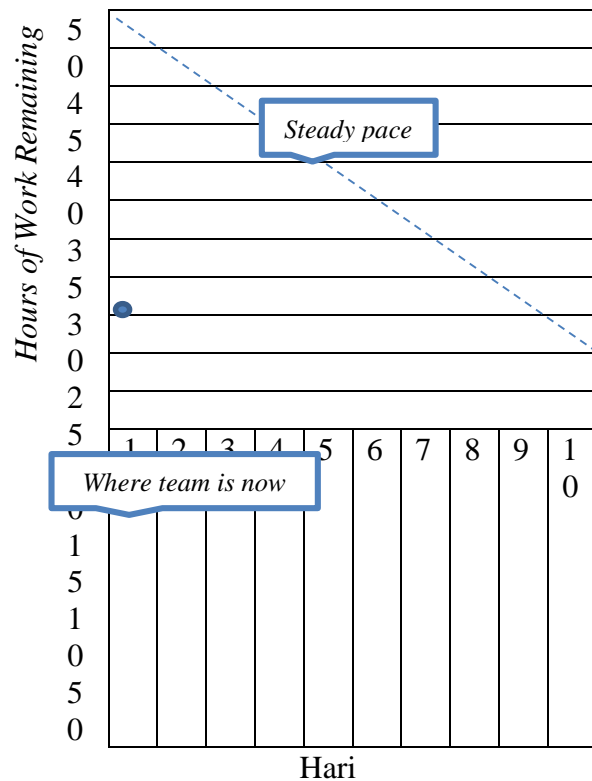
Saat Sprint sudah dimulai, semua Tim *Scrum* wajib melakukan pertemuan singkat harian pada hari kerja yang telah ditentukan, pertemuan ini berlangsung kurang lebih 15 menit. Masing-masing tim wajib melaporkan tiga hal yaitu Pekerjaan yang telah dilakukan sejak *Sprint Meeting* atau *Standup Meeting* terakhir, pekerjaan yang akan di lakukan, dan kendala yang dialami selama pengerjaan.

Scrum Master akan membuat catatan *Blocks*, yang kemudian membantu anggota tim untuk mengatasi setelah pertemuan singkat tersebut. Setelah selesai *Standup Meeting*, masing-masing anggota tim memperbaharui jumlah waktu yang tersisa untuk menyelesaikan masing-masing *task* (Gambar *Boundown chart*) sekaligus memindahkan masing-masing *task* pada *Scrum board*. Saat sebuah *task* dimulai, sebuah *notes* akan dipindahkan ke dalam kolom berlabel “*In Progress*”, apabila *task* sedang dilakukan pengecekan maka *notes* akan dipindahkan ke dalam kolom berlabel “*Test*” dan apabila *task* sudah selesai dilakukan pengecekan maka *notes* akan dipindahkan ke dalam kolom berlabel “*Completed*”

Tabel 5. *Daily Estimates*

<i>Task</i>	<i>Owner</i>	<i>Hours of Work Remaining on Each day of the Sprint</i>									
		1	2	3	4	5	6	7	8	9	10
Menyimpan data lagu yang sering diputar ke dalam tabel Top Chart	Diah	4									
Mengambil data dari tabel Top Chart untuk ditampilkan	Untari	2									
Membuat fungsi perbaharui Top Chart	Ningsih	5									
Total		11									

Pada kasus lainnya, dengan jumlah tim yang sangat besar dimungkinkan terjadinya *multiple Daily Scrum*, yaitu masing-masing tim melakukan *Daily Scrum* untuk berbagi 3 jawaban dengan anggota lainnya sesama tim, kemudian diwakili oleh Scrumaster yang akan melakukan *Daily Scrum* dengan *Scrum Master* lainnya yang merupakan perwakilan dari tim-tim lainnya. Tantangan terbesar dari *Daily Scrum* jenis ini adalah bagaimana menjaga aliran informasi tetap sama hingga ke atas dan turun lagi ke tim pengembang (Smits, et al, 2007).



Gambar 4. *Burndown Chart*

D. *Sprint Review*

Setelah Sprint berakhir, terdapat Sprint Review, di mana tim demo apa yang telah di bangun selama Sprint. Lamanya Sprint dapat berbeda untuk onshore dengan offshore-nya. Pada penelitiannya Paasivaara et al (2009) menemukan terdapat tim onshore yang memiliki panjang *Sprint* selama 4 minggu dan untuk tim *offshore*-nya hanya memiliki panjang 2 minggu. Hal ini ditujukan untuk memudahkan pemantauan tim *offshore* oleh tim *onshore*.

Dalam pertemuan ini dihadiri oleh *Product Owner*, Tim Pengembang, dan *Scrum Master*, pelanggan, *stake holder*, dan siapapun yang tertarik. Ini bukan "presentasi" yang diberikan Tim Pengembang, dalam pertemuan ini tidak menggunakan PowerPoint, dan biasanya tidak lebih dari 30 menit dihabiskan untuk mempersiapkannya - ini benar-benar hanya demo tentang apa yang telah dibangun, dan siapa saja yang hadir dalam pertemuan tersebut bebas bertanya dan memberi masukan. Pertemuan ini sangat diperlukan karena untuk menunjukkan apa yang telah dibangun dan untuk mendapatkan umpan balik.

Kesimpulan

Setelah dilakukan pengkajian proses dan cara kerja *Scrum* didapatkan beberapa kesimpulan bahwa Metode *Scrum* dapat digunakan sebagai kerangka kerja untuk mengembangkan interoperabilitas perangkat lunak dengan waktu yang lebih cepat. Selain itu metode *Scrum* menggunakan dokumen teknis secara minimal dan memiliki pola kerja memecah pekerjaan menjadi tugas yang lebih kecil (*task*) dan terdistribusi didiantara Tim Pengembang secara rutin dalam siklus waktu yang singkat sehingga pengembangan sistem dapat diselesaikan dengan cepat.

Kegiatan *Daily Standup Meeting* baik untuk pemantauan dan kontrol perkembangan kerja karena siklus waktunya pendek dan berulang dengan cepat. Pada situasi Tim Pengembang yang terpisah jarak cukup jauh dapat dikembangkan sistem komunikasi dan kontrol secara online sehingga tim pengembang tidak perlu harus selalu bertatap muka secara fisik. Keunggulan dari metode *Scrum* pada penelitian ini seperti pola kerja memecah pekerjaan menjadi *task* yang terdistribusi merata ke semua tim pengembang untuk diselesaikan dalam siklus kerja yang cepat diperlukan adanya ketelitian yang berlebih dalam menentukan *Backlog*.

Daftar Pustaka

- Anonim. 2016. *Mengembangkan Produk Software dengan Teknik Scrum*. Diakses dari <http://humancapitaljournal.com/mengembangkan-produk-software-dengan-teknik-Scrum/> pada tanggal 16 Juli 2020
- Hossain, E., Babar, M.A., Paik, H. 2009. *Using Scrum in Global Software Development: A Systematic Literature Review*. IEEE Computer Science.
- Jamshidi, Pooyan., Ghafari, Mohammad., Ahmad, Aakash. dan Pahl, Claus. 2013. *A framework for classifying and comparing architecture-centric software evolution research*. In 17th European Conference on Software Maintenance and Reengineering (CSMR), (hal. 305-314). IEEE.
- Mahalakshmi, M. & Sundararajan, M., 2013. *Tradisional SDLC Vs Scrum Methodology – A Comparative Study*. *International Journal of Emerging Technology and Advanced Engineering*, [online] Diakses dari http://www.ijetae.com/files/Volume3Issue6/IJETAE_0613_32.pdf pada tanggal 16 Juli 2020

- Paasivaara, M., Durasiewicz, S., Lassenius, C. 2009. *Using Scrum in Distributed Agile Development: A Multiple Case Study*. IEEE Computer Science.
- Partogi, Joshua. 2014. *Kenapa Scrum Tidak Akan Berkembang di Indonesia*. Diakses dari <https://startupbisnis.com/kenapa-Scrum-tidakakan-berkembang-di-indonesia/> pada tanggal 15 Juli 2020
- Qadir, Muhammad Manan. dan Usman, Muhammad. 2011. *Software engineering curriculum: A systematic mapping study*. In 5 th Malaysian Conference on Software Engineering (MySEC), (hal. 269-274). IEEE.
- Schwaber, K., Sutherland, J., 2011, *The Definitive Guide to Scrum: The Rules of The Game*. Scrum Alliance, Scrum: The Basic. Diakses dari <http://www.scrumalliance.org> pada tanggal 21 Juli 2020.
- Smits, H., Pshigoda, G. 2007. *Implementing Scrum in a Distributed Software Development Organization*. IEEE Computer Society.
- Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. 2007. *Distributed Scrum: Agile Project Management with Outsourced Development Teams*. IEEE Computer Society.