Sistem Informasi Manajemen Kerja Tim Scrum dalam Proyek Pengembangan Software

Aprilia Ruh Sufiati
Jurusan Teknik Informatika, Fakutlas Teknologi Industri, Universitas Islam Indonesia
Yogyakarta, Indonesia
13523204@students.uii.ac.id

***Abstrakt*—Upaya peningkatan kinerja tim yang mengerjakan suatu proyek khususnya pengembangan Software dengan kerangka kerja Scrum perlu dilakukan. Hal ini berkaitan dengan kebutuhan Tim yang semakin meningkat untuk memudahkan dan mempercepat pekerjaan yang dilakukan. Kerangka kerja Scum memiliki banyak komponen di dalamnya seperti aktor yang berperan, acara atau aktivitas yang perlu dilakukan dan artefak yang harus dipenuhi. Sistem Informasi Manajemen Kerja Tim Scrum dibuat untuk memenuhi kebutuhan tersebut dan mengaplikasikan kerangka kerja Scrum dalam sebuah sistem yang terkomputerisasi. Sistem dibuat menggunakan metode prototyping, serta melakukan pengujian fungsionalitas dan pengujian terhadap beberapa pengguna yang sudah menggunakan kerangka kerja Scrum pada tempat kerjanya.**

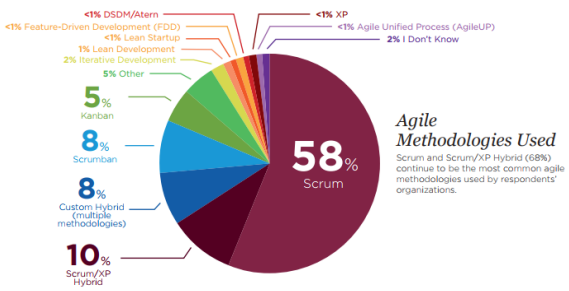
Kata Kunci—Manajemen, Proyek, Software, Scrum, Prototyping.

I. PENDAHULUAN

Pada abad 21 ini keterikatan manusia terhadap teknologi informasi semakin meningkat. Teknologi informasi memiliki peran penting dan menjadi kebutuhan untuk mendukung aktivitas sehari-hari. Peningkatan aktivitas dan kebutuhan akan teknologi informasi ini mendorong manusia untuk memenuhi kebutuhan maupun menyelesaikan masalah yang timbul. Salah satunya adalah kebutuhan akan sistem maupun aplikasi yang dapat mempercepat dan memudahkan dalam aktivitas-aktivitas yang biasanya memakan waktu dan tenaga yang tidak sedikit. Tidak terkecuali aktivitas untuk memanajemen kinerja tim dalam mengerjakan pekerjaannya.

Manajemen Kinerja berfokus pada tiga hal. Pertama, bagaimana para manajer dan pemimpin kelompok bekerja secara efektif dengan orang-orang yang ada disekitar mereka. Kedua, bagaimana para individu bekerja sama dengan para manajer dan kelompok. Ketiga, bagaimana individu dapat dikembangkan untuk meningkatkan pengetahuan, keahlian, kepiawaian dan tingkat kompetensi serta kinerja mereka [1].

Saat ini telah dikembangkan sebuah kerangka kerja Scrum untuk manajemen kerja Tim dalam menangani proyek. Menurut State of Agile Survey, 58% Organisasi yang menjadi responden menggunakan Scrum sebagai kerangka kerja untuk mengembangkan perangkat lunak [2] seperti yang terlihat pada diagram Gambar 1. Scrum dibuat sangat sederhana agar organisasi dapat berkolaborasi dan kreatif dalam menyederhanakan proses pengembangan *Software* [3]. Dengan menggunakan kerangka kerja Scrum, Tim Scrum dapat leluasa menentukan dan mengatur cara kerja menggunakan kreativitasnya sendiri untuk menjalankan proyek pengembangan perangkat lunak. Namun tetap dibutuhkan sebuah wadah untuk ‘menampung’ perkembangan proyek tersebut, dalam hal ini diperlukan peran seorang *Scrum Master* untuk memfasilitasi Tim Scrum dalam menjalankan proyeknya.



Gambar A Hasil Survey Responden Organisasi

Dalam kerangka kerja Scrum terdapat beberapa *Sprint Backlog* yang setiap *Sprint* harus diselesaikan sebuah tim pengembang dalam waktu maksimal 30 hari atau kurang dari itu sesuai kesepakatan yang telah dibuat, kemudian pada *Sprint* berikutnya akan dikerjakan oleh tim pengembang lainnya. Akan ada beberapa pertemuan Tim Scrum dalam pengembangan proyek, yaitu antara *Product Owner, Scrum Master* dan Tim Pengembang seperti *Sprint Planning Meeting, Daily Scrum Meeting, Sprint Review* dan *Sprint Retrospective*. Pada *Daily Scrum Meeting*, pertemuan tidak boleh lebih dari 15 menit. Pertemuan harus singkat, padat dan tepat sasaran. Jika ada yang perlu dibahas lebih lanjut, pokok bahasannya tinggal dicatat dan dibicarakan sesudah pertemuan harian. Aturan ini dirancang supaya informasi yang paling bernilai dan paling perlu ditindak lanjuti dapat diedarkan dalam waktu sesingkat-singkatnya [4].

Selain itu dengan adanya banyak pertemuan dalam kerangka kerja Scrum, maka Scrum harus bersifat fleksibel terhadap perubahan yang mungkin terjadi dari setiap hasil pertemuan yang dilakukan dalam pengerjaan proyek, masalah dokumentasi progres kerja Tim Scrum perlu dikelola secara *real time*. Namun hal ini akan menjadi sulit dilakukan apabila terdapat banyak proyek dan Tim Scrum yang terlibat dalam

proyek pengembangan *Software* yang berbeda. Satu Tim Scrum dapat terlibat dalam beberapa proyek sekaligus yang tetap harus memperhatikan target dan jangka waktu yang telah ditentukan dalam masing-masing proyek. Dengan banyaknya aktivitas, pihak yang terlibat dan keterbatasan waktu dalam kerangka kerja Scrum, maka *Scrum Master* harus dapat melakukan kontrol kinerja dan memastikan seluruh proyek berjalan sebagaimana mestinya.

II. LANDASAN TEORI

A. Software Development

Permasalahan terbesar dalam software development adalah interaksi manusia yang terlibat di dalamnya. Dalam dunia software development sebuah software tidak dikembangkan oleh sebuah teknologi lainnya, melainkan dikembangkan oleh software developer yang merupakan manusia. Software developer perlu berkolaborasi dengan software developer lainnya baik dari internal maupun eksternal perusahaan untuk menghasilkan software berkualitas tinggi. Selain itu software developer harus berhadapan dengan politik dan birokrasi yang diciptakan oleh stakeholder, namun terkadang stakeholder yang memiliki kepentingan terhadap software yang diminta tidak tahu bagaimana bentuk software yang sebenranya mereka mau. Hal ini secara tidak langsung mengakibatkan mayoritas permasalahan dalam software development adalah people problem.

Banyak orang berpikir kalau permasalahan dalam proses software development dapat dipecahkan dengan meningkatkan kapasitas maupun kualitas teknologi yang digunakan. Namun selama satu dekade yang telah berlalu, kita dapat melihat jika peningkatan kapasitas teknologi tidak banyak meningkatkan efisiensi proses software development. Sudah banyak teknologi baru yang dibuat, sudah banyak bahasa pemrograman yang kita gunakan 30 tahun lalu yang ternyata tidak membuat proyek software development kita lebih cepat dan lebih murah [3]..

B. Agile Software Development

Agile Development adalah kemampuan untuk pengembangan *Software* dengan cepat dalam menghadapi perubahan kebutuhan yang cepat. Untuk mencapainya, diperlukan penerapan yang memiliki keterampilan dan umpan balik di dalamnya. Selain itu diperlukan penerapan prinsip-prinsip desain yang menjaga fleksibelitas dan pemeliharaan *Software*. Perlu diketahui pola desain yang telah ditunjukkan untuk menyeimbangkan prinsip-prinsip tersebut untuk masalah tertentu [5].

C. Scrum

Scrum adalah sebuah kerangka kerja di mana orang-orang dapat menyelesaikan permasalahan kompleks yang senantiasa berubah, di mana pada saat bersamaan menghasilkan produk dengan nilai setinggi mungkin secara kreatif dan produktif [6]. Scrum bukanlah sebuah proses ataupun metode, melainkan hanya sebuah kerangka kerja dalam mengembangkan produk kompleks seperti perangkat lunak. Tidak seperti manajemen proyek pada umumnya, pada kerangka kerja Scrum tidak akan ada deskripsi rinci yang diberikan oleh manajer proyek tentang hal-hal yang harus dikerjakan tim pengembang.

Tim Scrum

Dalam setiap Tim Scrum terdapat tiga peran, yaitu *Product Owner*, Tim Pengembang dan *Scrum Master*. Dimana tim harus bersifat fleksibel terhadap perubahan yang sewaktu-waktu terjadi, bersifat kreatif untuk mengatur dan menentukan sendiri cara terbaik untuk menyelesaikan pekerjaannya dan bersifat produktif sehingga pekerjaannya dapat memberikan hasil sebagaimana mestinya. Dengan demikian berbeda dengan kerangka kerja dan metode tradisional, di dalam Tim Scrum tidak mengenal tingkatan peranan seperti *Project Manager* atau *Technical Leader*. Tim Scrum mengerjakan pekerjaannya secara berkala dan bertahap disertai dengan dokumentasi pekerjaan yang telah diselesaikan dalam setiap tahapannya [6].

Acara-acara Scrum

Acara-acara wajib dalam Scrum dihadiri 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 untuk memastikan waktu digunakan secukupnya tanpa ada yang terbuang sia-sia di sepanjang proses [6].

Jantung dari Scrum adalah *Sprint*, sebuah batasan waktu yang konsisten selama satu bulan kalender atau kurang, sepanjang proses pengembangan produk yang digunakan untuk menyelesaikan sesuatu. *Sprint* baru akan langsung dimulai setelah *Sprint* yang sebelumnya berakhir.

Anggota Tim Scrum secara kolaboratif membuat perencanaan pekerjaan yang akan dilaksanakan dalam *Sprint*. Waktunya dibatasi selama 8 jam untuk *Sprint* dengan durasi 1 bulan. 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.

Daily Scrum merupakan kegiatan dengan batasan waktu 15 menit setiap harinya agar Tim Pengembang dapat mendiskusikan pekerjaan yang telah dikerjakan maupun perencanaan untuk pekerjaan yang akan dilakukan 24 jam ke depan.

Sprint Review dilakukan pada akhir Sprint untuk meninjau dan merubah *Product Backlog* jika diperlukan. Tim Scrum dan *stakeholder* akan berkolaborasi untuk membahas apa yang telah dikerjakan pada *Sprint* yang baru selesai. Kemudian para hadirin berkolaborasi menentukan apa yang dapat dikerjakan di *Sprint* berikutnya untuk mengoptimalkan nilai produk.

Pada Retrospective Tim Scrum mendapat kesempatan untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan pada Sprint berikutnya. Sprint Retrospective dilakukan setelah Sprint

Review selesai dan sebelum Sprint Planning berikutnya. Batasan waktunya adalah 3 jam untuk *Sprint* dengan durasi 1 bulan. Scrum Master mengedukasi Tim Scrum untuk membuat peningkatan akan kerangka kerja proses Scrum, juga proses dan praktik pengembangannya, sehingga lebih efektif dan menyenangkan di Sprint berikutnya.

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 [6].

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

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".

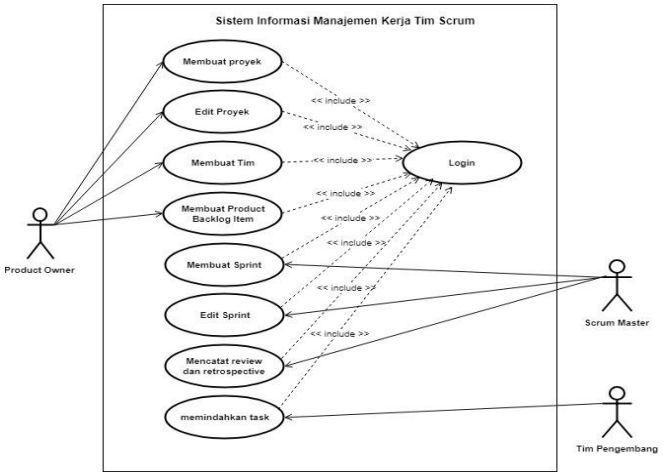
Increment adalah gabungan dari semua *Product Item* yang diselesaikan maupun nilai inkremen dari *Sprint* sebelumnya. Pada akhir Sprint, inkremen terbaru harus "Selesai", yang artinya berada dalam kondisi yang berfungsi penuh dan memenuhi definisi "Selesai" yang dibuat oleh Tim Scrum. Terlepas apakah *Product Owner* akan merilis produknya, produk harus selalu berada dalam kondisi yang berfungsi penuh

III. METODOLOGI PENELITIAN

Metode Penelitian yang digunakan dalam penelitian ini adalah metode *prototyping*, hal ini bertujuan untuk mengetahui lebih lanjut mengenai masalah yang akan dijadikan obyek pada penelitian.

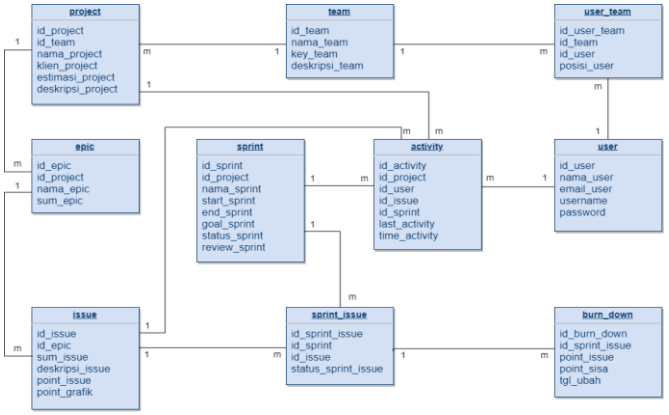
A. Desain Use Case

Use Case Diagram pada Gambar 2 menjelaskan fungsionalitas sistem yang akan diimplementasikan pada bab berikutnya. Terdapat 3 aktor yang berperan dalam sistem ini yaitu Product Owner, Scrum Master dan Tim Pengembang. Selain Login terdapat 8 proses yang ada pada sistem, yaitu Product Owner melakukan proses membuat proyek, edit proyek, membuat tim dan membuat product backlog item. Aktor Scrum Master dapat melakukan proses membuat Sprint, edit Sprint serta mencatat *review* dan *retrospective*. Aktor Tim Pengembang dapat melakukan proses memindahkan *task*.



Gambar 2 *Use Case Diagram*

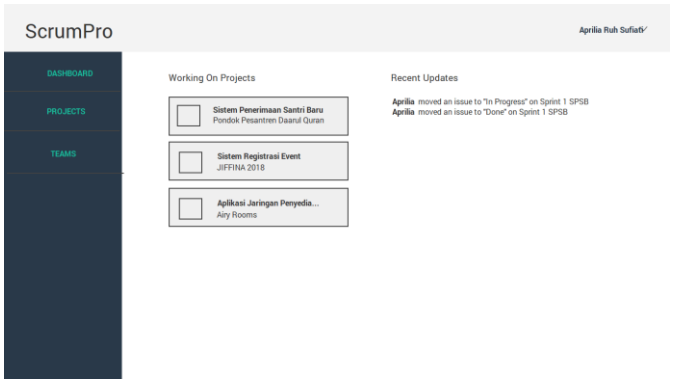
B. Desain Database



Gambar 3 Relasi Tabel

Terdapat 10 tabel yang memiliki relasi, yaitu tabel user, team, project, sprint, issue, epic, user_team, sprint_issue, activity dan burn_down. Relasi tabel tersebut dapat digambarkan seperti pada Gambar 3.

C. Prototyping



Gambar 4 Desain Halaman Dashboard

Dari hasil desain *Use Case* dan Tabel diatas, peneliti merancang 13 bagian antarmuka, yaitu Halaman Login, Halaman Register, Halaman Dashboard, Halaman Projects, Halaman Create Projects, Halaman Backlog, Halaman Active

Sprint, Halaman Reports, Halaman Teams, Halaman Create Team, Halaman Create Epic, Halaman Create Issue, Halaman Create Sprint, Setelah berhasil *login* ke sistem, maka pengguna akan dihadapkan dengan tampilan halaman *dashboard* dengan desain yang ditunjukkan pada Gambar 4.

IV. HASIL DAN PEMBAHASAN

Untuk melakukan penerapan terhadap rancangan sistem yang telah dibuat, dilakukan Implementasi pengembangan Sistem Informasi Manajemen Kerja Tim Scrum. Selanjutnya akan dilakukan pengujian sistem untuk memastikan seluruh fungsiosnalitas sistem berjalan sebagaimana mestinya dan mengetahui seberapa jauh *user* dapat menerima sistem yang telah dibuat.

A. Implementasi

Guna memudahkan dalam pemahaman implementasi sistem, digunakan dua pendekatan yaitu skenario dan implementasi Sistem Informasi Manajemen Kerja Tim Scrum dalam Proyek Pengembangan *Software*. Pada skenario akan dijelaskan bagaimana proses bisnis sistem dapat menangani sebuah proyek pengembangan *software* dari awal hingga akhir pelaksanaan proyek tersebut dinyatakan selesai. Skenario sistem menggunakan studi kasus proyek pengembangan Sistem Informasi Penerimaan Mahasiswa Baru Universitas Islam Indonesia (PMB UII). Selanjutnya skenario tersebut akan divisualisasikan pada bagian implementasi sistem

1. Skenario Proyek

Dalam skenario ini terdapat tiga aktor yang berperan, yaitu Product Owner, Scrum Master dan Tim Pengembang.

Contoh skenario pada penelitian ini adalah proyek pengembangan Sistem Informasi PMB UII. Tahapan yang akan dilakukan ditunjukkan pada Tabel 4.1.

Tabel 1 Skenario Sistem

Aktor	Tahapan	Skenario
Product Owner	Menentukan Proyek	- Product Owner memutuskan untuk mengerjakan proyek pengembangan Sistem PMB UII. - Product Owner bertemu dengan seluruh *stakeholder* PMB UII untuk menentukan *requirement* dalam proyek PMB UII. Pertemuan tersebut akan menghasilkan daftar fitur beserta product backlog item yang akan dimiliki sistem PMB UII.
Product Owner	Membentuk Tim	Berdasarkan *requirement* yang telah didiskusikan dengan *stakeholder*, Product Owner membentuk sebuah Tim Scrum yang komposisinya ditentukan sesuai dengan kebutuhan proyek PMB UII. Tim yang dibentuk terdiri dari dirinya sendiri sebagai Product Owner, Scrum Master dan beberapa anggota untuk Tim Pengembang yang akan terlibat dalam proyek PMB UII.
Scrum Master	Sprint Planning	Product Owner bertemu dengan Master Scrum dan Tim Pengembang untuk membahas rencana kerja dalam setiap Sprint.
Scrum Master	Mengelola Sprint	Product Owner membagi Product Backlog Item ke dalam setiap Sprint yang telah ditentukan dalam Sprint Planning.
Tim Pengembang	Daily Meeting	- Tim Pengembang didampingi oleh Master Scrum melakukan Daily Meeting maksimal selama 15 menit sebelum melanjutkan pekerjaan pada hari tersebut. Hal ini terus berulang dan berlanjut hingga akhir Sprint.
Scrum Master	Review & Retrospective	Pada akhir Sprint, Tim Scrum akan kembali melakukan pertemuan untuk membahas pelaksanaan proyek dalam satu Sprint terakhir. Dari pertemuan ini akan menghasilkan catatan mengenai inkremen dan evaluasi yang berkaitan dalam pengerjaan proyek PMB UII.
Product Owner	Menutup Sprint	- Hasil pekerjaan dari sebuah Sprint siap dirilis. - Product Owner menyatakan Sprint selesai dan ditutup.

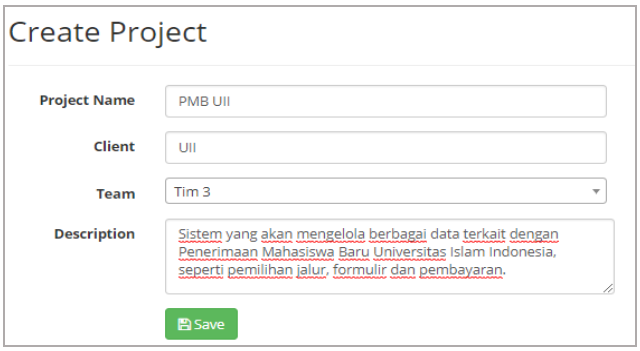
2. Implementasi Sistem

Masing-masing skenario divisualisasikan untuk memudahkan pembaca dalam memahami alur proses bisnis sistem dalam penelitian ini.

Product Owner menentukan Proyek

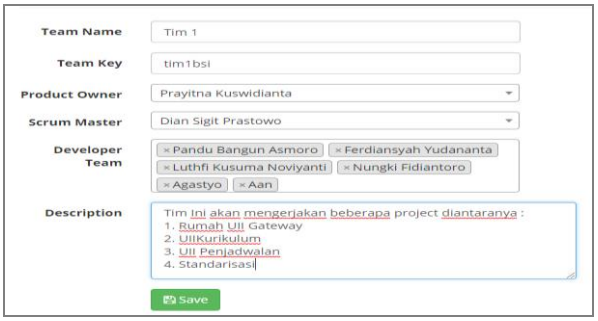
Dalam implementasi sistem mengenai Product Owner menentukan proyek, terdapat beberapa proses, yaitu membuat proyek baru, membuat daftar fitur yang disebut Epic beserta Product Backlog Item yang disebut Issue.

Untuk membuat proyek baru, gunakan tombol “Create Project” pada halaman Backlog. Kemudian akan muncul sebuah halaman berisi form untuk membuat proyek baru seperti pada Gambar 5.



Gambar 5 Formulir Pembuatan Proyek Baru

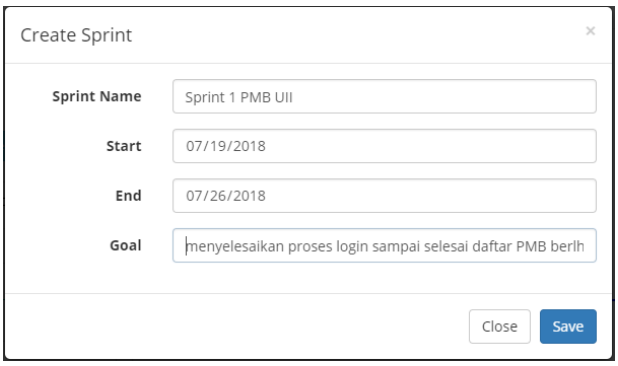
Product Owner membentuk Tim



Gambar 6 Formulir Pembuatan Tim Baru

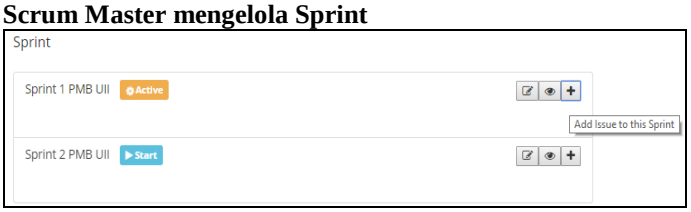
Scrum Master mencatat Sprint Planning

Jika user menekan tombol Create Sprint pada halaman backlog, akan muncul sebuah modal berupa form untuk membuat Sprint baru seperti pada Gambar 7. Data yang perlu diisi adalah Sprint Name, Start yaitu tanggal dimulainya sebuah Sprint, End yaitu tanggal berakhirnya sebuah Sprint dan Goal yaitu penjelasan mengenai tujuan atau hasil yang diharapkan dari sebuah Sprint.



Gambar 7 Formulir Pembuatan Sprint Baru

Scrum Master mengelola Sprint

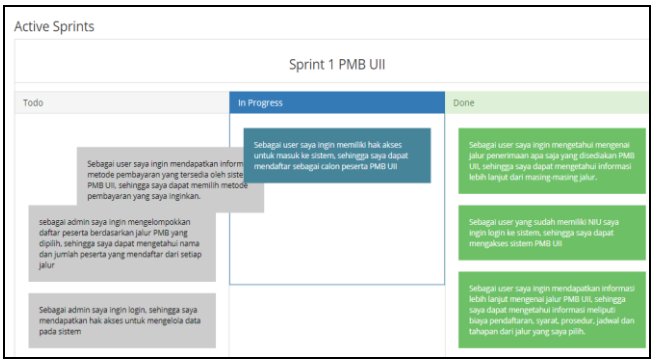


Gambar 8 Menambahkan Issue ke Sprint

Untuk melakukan pembagian issue ke dalam beberapa Sprint, gunakan tombol dengan simbol "+" pada daftar nama Sprint yang tersedia seperti pada Gambar 8.

Tim Pengembang melakukan Daily Meeting

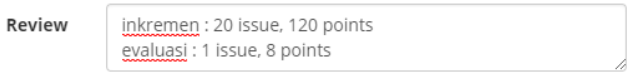
Seluruh *issue* baru yang sudah dibuat pada halaman *Backlog* berstatus "Todo", *user* dapat mengubah status *issue* dengan melakukan *drag and drop* dari kolom Todo ke kolom In Progress atau Done seperti pada Gambar 9.



Gambar 9 *Drag and Drop* pada Sprint Aktif

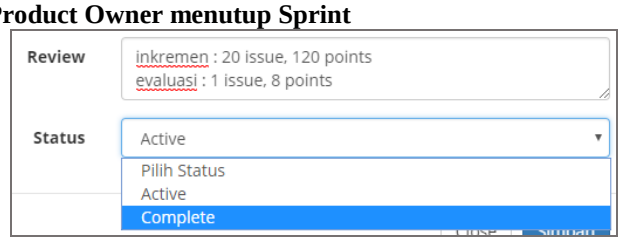
Scrum Master mencatat Review dan Retrospective

Gambar 10 merupakan tampilan formulir *Edit Sprint* untuk mengisi Review dengan catatan inkremen dan evaluasi hasil dari pertemuan Review dan Retrospective. Isi dari formulir tersebut dapat di deskripsikan lebih luas, sesuai dengan keinginan dan kebutuhan Tim.



Gambar 10 Memasukkan catatan Review dan Retrospective

Product Owner menutup Sprint



Gambar 10 Menutup Sprint

Jika review sudah diisi maka akan muncul pilihan untuk mengubah status Sprint seperti pada Gambar 13.

B. Pengujian

Untuk memastikan sistem berjalan sebagaimana mestinya maka dilakukan pengujian terhadap beberapa objek seperti fungsi, alur dan *error handling* yang dimiliki sistem. Sistem diujikan kepada 5 pengguna sebagai *tester* yang terdiri dari Mahasiswa Kerja Praktek dan Karyawan Badan Sistem Informasi Universitas Islam Indonesia (BSI UII). Mahasiswa yang melakukan pengujian merupakan mahasiswa jurusan Teknik Informatika yang sedang melaksanakan Kerja Praktek di BSI UII.

Pengujian dimulai dengan menyampaikan latar belakang pembuatan sistem, penjelasan secara garis besar dan proses bisnis dari sistem yang telah dibuat. Selanjutnya pengguna diminta untuk menjalankan sistem berdasarkan skenario implementasi yang akan diberikan oleh penulis. Setelah selesai melakukan implementasi, pengguna diminta untuk mengisi kuesioner yang telah disediakan.

Dengan jawaban dari kuesioner pengujian, selanjutnya dilakukan penghitungan nilai menggunakan skala Likert untuk mengukur seberapa jauh pengguna dapat menerima sistem yang telah dibuat.

Tabel 2 Hasil Kuesioner Pengujian Sistem

No	Pertanyaan	Jawaban					Presentase Skor
		STS	TS	R	S	SS	
1	Sistem memiliki tampilan yang menarik.				4	1	84%
2	Sistem memiliki alur dan fungsionalitas yang mudah dipahami.				5		80%
3	Secara keseluruhan sistem berjalan dengan baik tanpa adanya *error*.				3	2	88%
4	Sistem sudah sesuai dengan kebutuhan Manajemen Kerja Tim Scrum berdasarkan teori kerangka kerja Scrum.				3	2	88%
5	Sistem mampu membantu kinerja Tim Scrum.			1	3	1	80%
6	Sistem mampu mendukung pengelolaan *database* proyek pengembangan *software* dengan menggunakan kerangka kerja Scrum.			1	4		76%
Rata-rata Presentase Skor							**82,67%**

Berdasarkan hasil kuesioner pengujian terhadap pengguna pada Tabel 4.2 diperoleh rata-rata skor dengan presentase 82,67%. Maka hasil pengujian sistem terhadap pengguna berada pada kategori Sangat setuju.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan penelitian dan pembahasan tugas akhir mengenai Sistem Informasi Manajemen Kinerja Scrum dalam Proyek Pengembangan *Software*, disimpulkan bahwa :

1. Terdapat beberapa prinsip yang diperlukan Scrum Master untuk pengelolaan Tim Scrum, diantaranya:
   - Pembentukan Tim yang memiliki komitmen.
   - Planning yang matang dan memiliki tujuan.
   - Pelaksanaan proyek yang jelas, transparan, teratur dan fleksibel namun memiliki batas waktu.
   - Pengelompokan task secara jelas dan tertata.
   - Hasil dan evaluasi proyek terdokumentasi dengan baik dan jelas.
2. Sistem ini sudah sesuai dengan kebutuhan manajemen kerja Tim Scrum berdasarkan kerangka kerja Scrum dan membantu Tim Scrum dalam meningkatkan kinerja Tim. Secara keseluruhan, sistem memiliki penilaian Sangat Setuju atau Sangat Baik.

B. Saran

Untuk pengembangan sistem lebih lanjut, diharapkan Sistem Informasi Manajemen Kerja Tim Scrum kedepannya dapat membuat fitur yang lebih fleksibel sehingga dapat lebih mudah dan sesuai untuk penggunaan manajemen proyek yang menggunakan kerangka kerja Scrum. Selain itu perlu ditambahkan beberapa fitur yang dapat mendukung berlangsungnya sistem manajemen kerja Tim Scrum seperti notifikasi melalui *e-mail*, *import* dan *export* data menggunakan Microsoft Excel.

DAFTAR PUSTAKA

[1] Dharma, S. (2009). Manajemen Kinerja Falsafah, Teori dan Penerapannya. Yogyakarta: Pustaka Pelajar.

[2] Versionone (2017). Versionone 11th Annual State of Agile Report. Atlanta: Versionone.com..

[3] Partogi, Joshua. (2015). Manajemen Modern dengan Scrum. Yogyakarta: Andi.

[4] Sutherland, Jeff. (2017). Scrum Meningkatkan Produktifitas Dua Kali Lipat Dalam Waktu Setengahnya Saja. Yogyakarta: Bentang Pustaka.

[5] Martin, R. C. (2003). Agile Software Development: Principles, Patterns and Practices. New Jersey: Prentice Hall.

[6] Schwaber, K., & Sutherland, J. (2017). The Scrum Guide.

[7] Whitten, J.L., & Bentley, L. D. (2007). System Analysis and Design Method - Seventh Edition. New York: McGraw-Hill.