

SISTEM INFORMASI MANAJEMEN KERJA TIM SCRUM DALAM PROYEK PENGEMBANGAN *SOFTWARE*



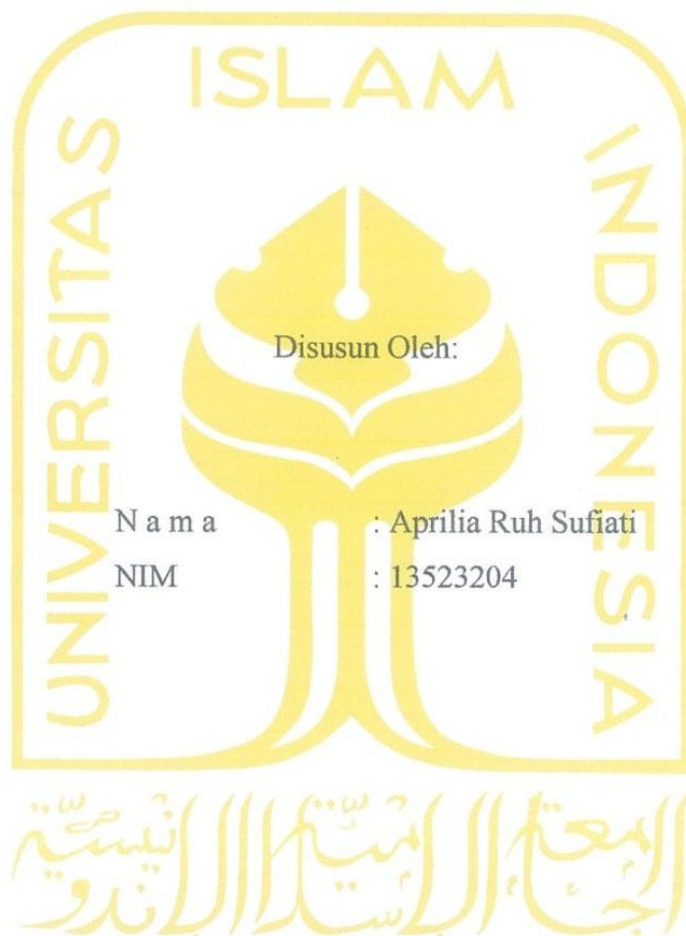
Disusun Oleh:

N a m a : Aprilia Ruh Sufiati

NIM : 13523204

**PROGRAM STUDI TEKNIK INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2018

HALAMAN PENGESAHAN DOSEN PEMBIMBING**SISTEM INFORMASI MANAJEMEN KERJA TIM SCRUM
DALAM PROYEK PENGEMBANGAN *SOFTWARE*****TUGAS AKHIR**

Disusun Oleh:

N a m a : Aprilia Ruh Sufiati
NIM : 13523204

Yogyakarta, 15 Agustus 2018

Pembimbing,

(Kholid Haryono, S.T., M.Kom)

HALAMAN PENGESAHAN DOSEN PENGUJI

**SISTEM INFORMASI MANAJEMEN KERJA TIM SCRUM
DALAM PROYEK PENGEMBANGAN SOFTWARE****TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 15 Agustus 2018

Tim Penguji

Kholid Haryono, S.T., M.Kom.

Anggota 1

Almed Hamzah, ST., M.Eng.

Anggota 2

Beni Suranto, S.T., M.Soft.Eng.



Mengetahui,

Ketua Program Studi Teknik Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Aprilia Ruh Sufiati

NIM : 13523204

Tugas akhir dengan judul:

**SISTEM INFORMASI MANAJEMEN KERJA TIM SCRUM
DALAM PROYEK PENGEMBANGAN SOFTWARE**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 15 Agustus 2018



(Aprilia Ruh Sufiati)

HALAMAN PERSEMBAHAN

Seluruh proses dan hasil pencapaian tugas akhir ini penulis persembahkan kepada kedua orang tua penulis, papa H. Achmad Sulastoro dan mama Samawati Tripujiastuti yang menjadi semangat bagi penulis untuk menyelesaikan jenjang kuliah dan segera memasuki jenjang kehidupan selanjutnya. Tanpa orang tua yang kuat, tabah dan selalu menginginkan yang terbaik untuk anak semata wayangnya, penulis tidak akan sampai pada titik ini.

HALAMAN MOTO

"Banyak kegagalan dalam hidup ini dikarenakan orang-orang tidak menyadari betapa dekatnya mereka dengan keberhasilan saat mereka menyerah."

(Thomas Alva Edison)

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah, penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat, hidayah, dan karunia-Nya, sehingga laporan tugas akhir ini dapat penulis selesaikan. Tak lupa shalawat dan salam penulis haturkan kepada junjungan kita Nabi Muhammad SAW, yang telah membawa kita dari zaman jahiliyah menuju jaman terang benderang.

Tugas Akhir ini dibuat sebagai salah satu syarat yang harus dipenuhi untuk memperoleh gelar sarjana di Jurusan Teknik Informatika Universitas Islam Indonesia. Adapun Tugas Akhir yang dilakukan penulis merupakan Sistem Informasi Manajemen Kinerja Tim *Scrum* dalam Proyek Pengembangan *Software*.

Pelaksanaan Tugas Akhir ini merupakan salah satu mata kuliah wajib jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia. Selain itu merupakan sarana bagi penulis untuk menambah wawasan serta pengalaman dalam menerapkan keilmuan, sesuai dengan yang diambil di bangku perkuliahan.

Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih kepada :

1. Kedua orang tua atas segala doa dan dukungan selama penulis menyelesaikan Tugas Akhir.
2. Fathul Wahid, selaku Rektor Universitas Islam Indonesia
3. Prof. Dr. Ir. Hari Purnomo, M.T., selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Hendrik, S.T., M.Eng., selaku Ketua Jurusan Teknik Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Kholid Haryono, S.T., M.Kom., selaku Dosen Pembimbing Tugas Akhir di Jurusan Teknik Informatika.
6. Mas Wisnu Adityaselaku mentor yang telah membimbing dan menjadi teman diskusi penulis mengenai materi Tugas Akhir.
7. Burhanuddin Aufa yang selalu memberi semangat dan menjadi teman berbagi keluh kesah selama penulis mengerjakan Tugas Akhir.
8. Teman-teman ETERNITY 2013 khususnya sahabat-sahabat penulis yang menjadi teman seperjuangan di bangku kuliah, sedikit banyak memberi dukungan dan mempengaruhi penulis untuk segera menyelesaikan Tugas Akhir.

9. Semua pihak yang telah banyak membantu penulis dalam pelaksanaan Tugas Akhir yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih belum sempurna karena keterbatasan kemampuan dan pengalaman penulis di lapangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun demi kesempurnaan Laporan Tugas Akhir ini. Akhir kata, penulis berharap agar laporan ini dapat bermanfaat bagi semua pihak.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 15 Agustus 2018

(Aprilia Ruh Sufiati)

SARI

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 *Scrum* 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 terhadap beberapa pengguna yang sudah menggunakan kerangka kerja *Scrum* pada tempat kerjanya. Hasil pengujian sistem menunjukkan rata-rata presentase skor yang diberikan pengguna adalah 82,67%.

Kata kunci: *Manajemen, Proyek, Software, Scrum, Prototyping.*

GLOSARIUM

Scrum	Proses untuk mengubah berkas kode program dengan berkas lain yang terkait menjadi berkas yang siap untuk dieksekusi oleh sistem operasi secara langsung.
Backlog	Langkah untuk menelusuri kesalahan kode program.
Sprint	Batasan dan pembagian waktu dalam proyek yang menggunakan kerangka kerja <i>Scrum</i> .
Issue	Tugas yang dimiliki sebuah proyek untuk dikerjakan oleh Tim <i>Scrum</i> berupa <i>user story</i> .
Story Point	Estimasi waktu yang diperlukan untuk mengerjakan <i>issue</i> .
Epic	Pengelompokan <i>issue</i> berdasarkan kebutuhan sistem yang dikerjakan dalam sebuah proyek.
User Story	Tugas yang dimiliki sebuah proyek yang di deskripsikan secara singkat.
Product Owner	Pihak yang memiliki wewenang untuk mengambil keputusan dalam proyek yang dimiliki Tim <i>Scrum</i> .
Scrum Master	Pihak yang mengawasi dan melayani kebutuhan Tim <i>Scrum</i> dalam menjalankan Proyek. Memastikan Proyek berjalan sebagaimana mestinya.
Development Team	Pihak yang menjalankan keseharian proyek dan mengerjakan proyek.

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI.....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI	ix
GLOSARIUM.....	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiii
DAFTAR GAMBAR	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat Penelitian.....	3
1.6 Metode Penelitian.....	3
1.7 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI.....	6
2.1 <i>Software Development</i>	6
2.2 <i>Agile Software Development</i>	6
2.2.1 Pengertian Agile Development	6
2.2.2 Prinsip <i>Agile</i>	7
2.3 <i>Scrum</i>	7
2.3.1 Pengertian <i>Scrum</i>	7
2.3.2 Tim <i>Scrum</i>	8
2.3.3 Acara-acara <i>Scrum</i>	9
2.3.4 Artefak <i>Scrum</i>	12
2.4 <i>Unified Modelling Language (UML)</i>	13
2.4.1 <i>Use Case Diagram</i>	13

2.4.2	<i>Activity Diagram</i>	15
2.5	<i>Skala Likert</i>	17
BAB III METODOLOGI PENELITIAN		19
3.1	Identifikasi Masalah	19
3.1.1	Tujuan Penyelesaian Masalah	19
3.1.2	Pengumpulan Data	19
3.2	Pembuatan <i>Prototyping</i>	20
3.2.1	<i>Initial Requirement</i>	20
3.2.2	<i>Design</i>	23
3.2.3	<i>Prototyping</i>	38
BAB IV IMPLEMENTASI DAN PENGUJIAN.....		46
4.1	Implementasi	46
4.1.1	Skenario Proyek	46
4.1.2	Implementasi Sistem	48
4.1.3	Fitur Pendukung	52
4.2	Pengujian Sistem	54
4.2.1	Objek Pengujian.....	54
4.2.2	Subjek Pengujian.....	54
4.2.3	Prosedur Pengujian	55
4.2.4	Hasil Pengujian	55
BAB V KESIMPULAN DAN SARAN		57
5.1	Kesimpulan.....	57
5.2	Saran	57
DAFTAR PUSTAKA		58
LAMPIRAN.....		59

DAFTAR TABEL

Tabel 2.1 Skala Likert bentuk <i>checklist</i>	17
Tabel 3.1 Prosedur Pengambilan Data.....	19
Tabel 3.2 <i>User Requirement</i>	20
Tabel 3.3 Tabel user.....	32
Tabel 3.4 Tabel project	33
Tabel 3.5 Tabel team.....	33
Tabel 3.6 Tabel epic.....	34
Tabel 3.7 Tabel issue	34
Tabel 3.8 Tabel sprint	35
Tabel 3.9 Tabel user_team.....	36
Tabel 3.10 Tabel sprint_issue	36
Tabel 3.11 Tabel activity	36
Tabel 3.12 Tabel burn_down	37
Tabel 4.1 Skenario Sistem	46
Tabel 4.2 Hasil Kuesioner Pengujian Sistem.....	55

DAFTAR GAMBAR

Gambar 1.1 Hasil Survey Responden Organisasi	1
Gambar 2.1 <i>Use Case Diagram</i>	14
Gambar 2.2 Simbol <i>Use Case</i>	14
Gambar 2.3 Simbol <i>Actor</i>	15
Gambar 2.4 <i>Uses or Include Relationship</i>	15
Gambar 2.5 <i>Activity Diagram</i>	16
Gambar 3.1 Proses Bisnis Sistem	22
Gambar 3.2 <i>Use Case Diagram</i> Sistem	23
Gambar 3.3 <i>Activity Diagram</i> Login	24
Gambar 3.4 <i>Activity Diagram</i> Membuat Proyek	25
Gambar 3.5 <i>Activity Diagram</i> Edit Proyek	26
Gambar 3.6 <i>Activity Diagram</i> Membuat Tim	27
Gambar 3.7 <i>Activity Diagram</i> Membuat <i>Product Backlog Item</i>	28
Gambar 3.8 <i>Activity Diagram</i> Membuat <i>Sprint</i>	29
Gambar 3.9 <i>Activity Diagram</i> Edit <i>Sprint</i>	30
Gambar 3.10 <i>Activity Diagram</i> Mencatat Review dan Retrospective	30
Gambar 3.11 <i>Activity Diagram</i> Memindahkan <i>Task</i>	31
Gambar 3.12 Relasi Antar Tabel	32
Gambar 3.13 Desain Tampilan Halaman <i>Login</i>	38
Gambar 3.14 Desain Tampilan Halaman <i>Register</i>	39
Gambar 3.15 Desain Tampilan Halaman <i>Dashboard</i>	39
Gambar 3.16 Desain Tampilan Halaman <i>Projects</i>	40
Gambar 3.17 Desain Tampilan Halaman <i>Create Project</i>	40
Gambar 3.18 Desain Tampilan Halaman <i>Backlog</i>	41
Gambar 3.19 Desain Tampilan Halaman <i>ActiveSprint</i>	41
Gambar 3.20 Desain Tampilan Halaman <i>Reports</i>	42
Gambar 3.21 Desain Tampilan Halaman <i>Teams</i>	42
Gambar 3.22 Desain Tampilan Halaman <i>Create Team</i>	43
Gambar 3.23 Desain Tampilan Halaman <i>Create Epic</i>	43
Gambar 3.24 Desain Tampilan Halaman <i>Add Issues</i>	44
Gambar 3.25 Desain Tampilan Halaman <i>Create Sprint</i>	44
Gambar 4.1 Formulir Pembuatan Proyek Baru.....	48

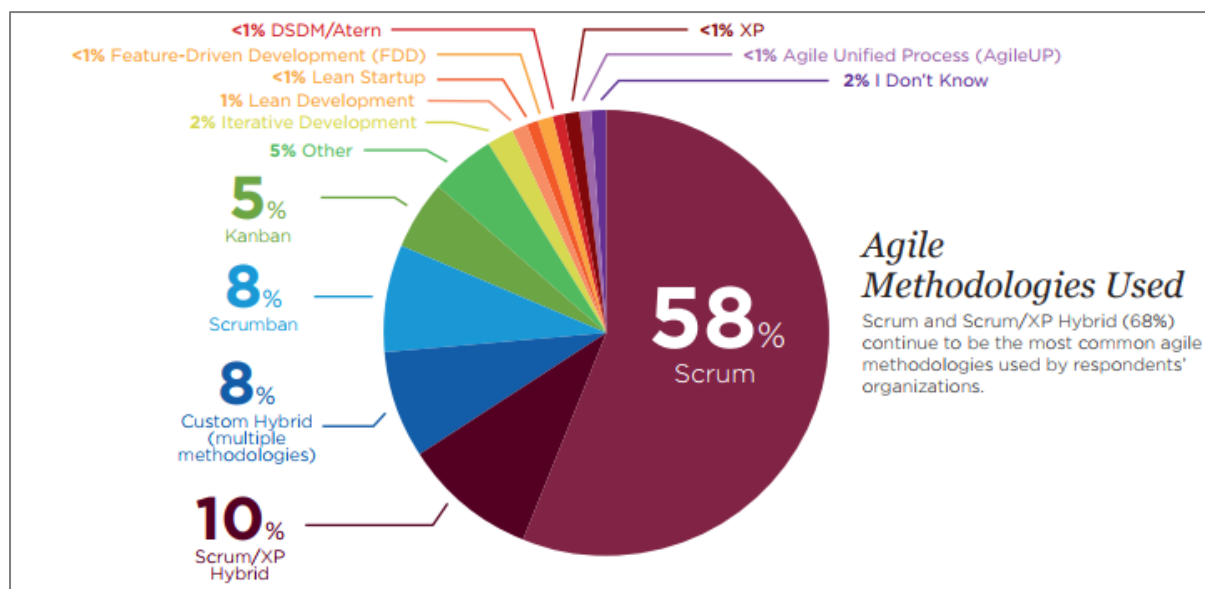
Gambar 4.2 Formulir Pembuatan Epic	49
Gambar 4.3 Formulir Pembuatan Issue	49
Gambar 4.4 Formulir Pembuatan Tim Baru	50
Gambar 4.5 Formulir Pembuatan <i>Sprint</i> Baru	50
Gambar 4.6 Menambahkan Issue ke <i>Sprint</i>	51
Gambar 4.7 <i>Drag and Drop</i> pada <i>Sprint</i> Aktif	51
Gambar 4.8 Memasukkan catatan Review dan Retrospective	52
Gambar 4.9 Menutup <i>Sprint</i>	52
Gambar 4.10 Grafik Burndown	53
Gambar 4.11 <i>Recent Activity</i>	54

BAB I PENDAHULUAN

1.1 Latar Belakang

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 manajemen 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 (Dharma, 2009).



Gambar 1.1 Hasil Survey Responden Organisasi

Sumber: Versionone 11th Annual State of Agile Report (2017)

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 (Versionone, 2017) seperti yang terlihat pada diagram Gambar 1.1 diatas. *Scrum* dibuat sangat sederhana agar organisasi dapat berkolaborasi dan kreatif dalam menyederhanakan proses pengembangan *Software* (Partogi, 2015). 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.

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 (Sutherland, 2017).

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.

Dalam implementasinya *Scrum* bukan sekedar kerangka bekerja, *Scrum* merupakan kerangka berfikir sehingga dalam adopsinya *Scrum* tidak dapat digeneralisasi. Penelitian ini

mengambil studi kasus pada unit pengembangan sistem disalah satu perguruan tinggi. Permasalahan yang timbul pada unit ini adalah memungkinkan seseorang berperan ganda. Seorang *Scrum Master* dapat berperan sebagai Tim Pengembang seperti dalam hal menyelesaikan *task* tertentu.

Dengan melihat permasalahan-permasalahan tersebut, maka penulis mengajukan tugas akhir dengan judul “Sistem Informasi Manajemen Kerja Tim *Scrum* dalam Proyek Pengembangan *Software*”.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, rumusan masalah dari penelitian ini adalah bagaimana mengimplementasikan pengelolaan Tim *Scrum* yang sedang melakukan proyek pengembangan *software* melalui aplikasi komputer?

1.3 Batasan Masalah

- a. Sistem mampu mengelola kerja tim yang sedang melakukan pengembangan *Software*.
- b. Sistem mampu mengelola kerja tim yang menggunakan kerangka kerja *Scrum* dalam pengembangan *software* yang akan dikerjakan.

1.4 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah mengimplementasikan aplikasi komputer untuk meningkatkan pengelolaan Tim *Scrum* yang sedang melakukan proyek pengembangan *software*.

1.5 Manfaat Penelitian

Dengan melakukan penelitian ini, diharapkan penggunaan aplikasi komputer untuk mengelola Tim *Scrum*, dapat membantu memangkas waktu dan tenaga yang diperlukan Tim *Scrum* sehingga dapat dialokasikan untuk pengembangan proyek lainnya.

1.6 Metode Penelitian

Peneliti akan menggunakan beberapa metode dalam melakukan penelitian sebagai berikut:

a. Identifikasi Masalah

Peneliti mengumpulkan data yang dibutuhkan terkait dengan masalah yang melatarbelakangi penelitian. Data diperoleh melalui kajian pustaka, observasi dan

wawancara sebagai referensi peneliti untuk menganalisa kebutuhan sistem yang akan dirancang oleh peneliti.

Dalam penelitian ini, kajian pustaka dilakukan dengan mengumpulkan dan mempelajari buku-buku, jurnal penelitian dan artikel yang membahas mengenai Manajemen Kerja Tim, *Scrum* dan Pengembangan *Software*. Peneliti menggunakan dua buku yang menjadi referensi utama dalam penelitian ini, yaitu buku terjemahan berjudul “*Scrum: Meningkatkan Produktivitas Dua Kali Lipat dalam Waktu Setengahnya Saja*” yang ditulis Jeff Sutherland dan buku “*Manajemen Modern Scrum*” yang ditulis Joshua Partogi. Sedangkan jurnal penelitian dan artikel digunakan sebagai informasi penunjang yang didapatkan melalui pencarian di perpustakaan maupun melalui aplikasi pencarian di internet.

Observasi dilakukan dengan mengamati aktivitas, aktor yang terlibat dan permasalahan yang terjadi dalam lingkungan objek penelitian ini. Dalam penelitian ini lingkungan yang dimaksud adalah lingkungan tim yang sedang melakukan pengembangan *Software* dengan menerapkan kerangka kerja *Scrum* didalamnya. Selanjutnya dilakukan wawancara mendalam terhadap Tim *Scrum* khususnya *Scrum Master* untuk dapat mengidentifikasi permasalahan dan penyebabnya secara lebih spesifik.

b. Pembuatan *Prototyping*

Hasil dari tahapan identifikasi masalah akan dijadikan masukan untuk pembuatan *prototyping* artefak yang akan dikembangkan.

c. Implementasi Artefak

Pada tahap ini peneliti membangun dan menerapkan sistem berdasarkan rancangan dari tahap sebelumnya (pembuatan *prototyping*) yang telah dibuat. Hasil artefak yang dikembangkan pada tahap ini akan di implementasikan menggunakan perangkat laptop yang telah disediakan peneliti.

d. Pengujian Artefak

Untuk mengetahui seberapa jauh hasil implementasi dapat menjawab rumusan masalah, maka dilakukan pengujian terhadap sistem yang telah dibuat peneliti. Selain itu pengujian sistem dilakukan untuk menemukan kekurangan sistem yang kemudian dapat diperbaiki sebelum sistem siap di rilis untuk pengguna.

Hasil implementasi akan diujikan kepada 5 pengguna. Pengguna pada pengujian ini merupakan Mahasiswa yang sedang melakukan Kerja Praktek dan Karyawan Badan Sistem Informasi Universitas Islam Indonesia (BSI UII). Kemudian pengguna akan diminta untuk mengisi kuesioner meliputi hasil pengujian sistem.

1.7 Sistematika Penulisan

Sistematika penulisan penelitian ini disusun sebagai gambaran umum untuk memudahkan dalam memahami penelitian yang dijalankan. Secara garis besar, sistematika penulisan penelitian ini adalah sebagai berikut:

a. BAB I : PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

b. BAB II : LANDASAN TEORI

Bab ini berisi penjelasan teori-teori yang digunakan sebagai landasan untuk menyelesaikan permasalahan yang dibahas pada penelitian ini. Bahasan dalam penelitian ini yaitu *Software Development*, *Agile Software Development* dan *Scrum*.

c. BAB III : METODOLOGI PENELITIAN

Bab ini berisi penjelasan metode yang akan digunakan peneliti selama melakukan penelitian. Selain itu bab ini juga berisi penjelasan mengenai rancangan dan kebutuhan dalam penelitian ini.

Metodologi yang akan dibahas meliputi Identifikasi masalah dan pembuatan *prototyping*. Perancangan yang akan dibahas antara lain meliputi *initial requirement*, *design* dan *prototyping*.

d. BAB IV : IMPLEMENTASI DAN PENGUJIAN

Bab ini berisi hasil implementasi penelitian dan pembuatan sistem yang telah dilakukan, berdasarkan perancangan yang telah dibuat sebelumnya. Bab ini juga membahas pengujian hasil penelitian untuk mengetahui bagaimana hasil dan tanggapan mengenai penelitian yang telah dilakukan.

e. BAB V : KESIMPULAN DAN SARAN

Pada bab ini akan diberikan kesimpulan dari keseluruhan penelitian yang telah dilakukan. Kemudian dari kesimpulan yang didapatkan, peneliti akan memberikan saran guna mengembangkan penelitian yang telah dilakukan peneliti.

BAB II

LANDASAN TEORI

2.1 *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 sebenarnya 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 (Partogi, 2015).

2.2 *Agile Software Development*

2.2.1 *Pengertian Agile Development*

Menurut Robert C. Martin (2003), *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 fleksibilitas dan pemeliharaan *Software*. Perlu diketahui pola desain yang telah ditunjukkan untuk menyeimbangkan prinsip-prinsip tersebut untuk masalah tertentu.

2.2.2 Prinsip Agile

Terdapat beberapa prinsip-prinsip *Agile* yang merupakan karakteristik yang membedakannya dengan praktik proses kelas berat (Martin, 2003), diantaranya sebagai berikut:

- a. Prioritas utamanya adalah memuaskan pelanggan melalui pengiriman *software* bernilai yang cepat dan berkesinambungan.
- b. Menerima perubahan. Partisipan dalam proses *agile* tidak takut akan perubahan, karena dengan adanya perubahan berarti tim telah mempelajari lebih baik mengenai apa yang dibutuhkan untuk memuaskan pasar.
- c. Rajin mengirim pekerjaan, dalam beberapa minggu sampai beberapa bulan dengan preferensi untuk skala waktu yang lebih pendek.
- d. Orang bisnis dan pengembang harus bekerjasama setiap harinya sepanjang pengerjaan proyek.
- e. Membangun proyek disekitar individu yang termotivasi. Termotivasi yang dimaksud adalah lingkungan, dukungan yang diperlukan dan kepercayaan untuk menyelesaikan pekerjaannya.
- f. Metode penyampaian informasi yang paling efisien dan efektif dalam tim pengembang adalah percakapan *face-to-face*.
- g. Pengerjaan *software* adalah ukuran utama kemajuan.
- h. Proses *agile* mempromosikan pembangunan berkelanjutan.
- i. Perhatian secara berkelanjutan untuk keunggulan teknis dan desain yang baik.
- j. Kesederhanaan. Seni untuk memaksimalkan jumlah pekerjaan yang dapat terselesaikan adalah penting.
- k. Memiliki arsitektur, persyaratan dan desain terbaik yang terbentuk melalui *self-organizing team*.
- l. Interval, tim merefleksikan bagaimana untuk menjadi lebih efektif, kemudian menyesuaikan kebiasaan.

2.3 Scrum

2.3.1 Pengertian 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 (Schwaber & Sutherland, 2017). *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.

2.3.2 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 (Schwaber & Sutherland, 2017).

Product Owner

Product Owner berperan dalam memaksimalkan nilai bisnis untuk perusahaan melalui produk yang dikembangkan oleh tim pengembang. Selain itu *Product Owner* berperan untuk memastikan produk yang dikembangkan oleh Tim Pengembang menghasilkan nilai tambah untuk pelanggan dan perusahaan.

Seluruh anggota Tim *Scrum* dan *stakeholder* proyek pengembangan produk harus menghormati setiap keputusan yang dibuat *Product Owner*. Keputusan dari *Product Owner* dapat dilihat dari isi dan urutan *Product Backlog*. Namun *Product Owner* harus tetap membuka diri terhadap saran dan masukan yang diberikan oleh anggota Tim *Scrum* dan *stakeholder* lainnya.

Tim Pengembang

Tim Pengembang terdiri dari para profesional yang bekerja bersama-sama untuk menghasilkan potongan produk yang dirilis setiap akhir *Sprint*. Tim pengembang berperan sebagai perancang, pengembang dan penguji yang diatur oleh dirinya sendiri untuk menyelesaikan pekerjaannya dalam mengembangkan produk.

Scrum Master

Scrum Master adalah seorang *Agile Manager* dan seorang *Servant Leader*. *Servant Leader* adalah seorang pemimpin yang melayani dalam memimpin sebuah organisasi. *Servant Leader* menggunakan otoritas yang ia miliki bukan untuk memerintah orang lain, namun untuk membuat kualitas hidup orang-orang di dalam perusahaan meningkat. Bagi *Servant Leader* segala kesuksesan di dalam perusahaan bukan miliknya, melainkan milik tim yang ia awasi. Hal ini sesuai dengan konsep yang digunakan oleh *Scrum Master* dalam mengawasi dan memfasilitasi Tim *Scrum*. *Scrum Master* bertanggung jawab untuk melayani *Product Owner*, Tim Pengembang, pelanggan dan *stakeholder* lainnya dalam organisasi.

Selain peran secara umum yang telah dideskripsikan diatas, terdapat beberapa peran yang saling berkaitan antara peran *Scrum Master* dengan *Product Owner* dan Tim Pengembang dalam sebuah Tim *Scrum*, di antaranya sebagai berikut:

Scrum Master untuk Product Owner

- a. Membantu memahami perancangan produk secara empirisme, yaitu berdasarkan data yang nyata contohnya data respon pengguna di lapangan.
- b. Senantiasa membantu mencari teknik terbaik untuk mengelola *Backlog*.

ScrumMaster untuk Tim Pengembang

- a. Memfasilitasi termasuk mengedukasi mengenai kerangka kerja *Scrum* untuk melaksanakan pekerjaannya, terutama untuk Tim Pengembang yang baru mengenal *Scrum*.
- b. Membantu untuk dapat mengatur diri sendiri. Seperti mengawasi tingkat interaksi antara anggota Tim Pengembang.
- c. Mendorong untuk selalu memperbaiki cara kerja dan kualitas produk yang dikembangkan.
- d. Melindungi bahkan menghilangkan hal-hal yang menghambat pekerjaan diluar kendali.

2.3.3 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 (Schwaber & Sutherland, 2017).

Sprint

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. Bila jangka waktu *Sprint* terlalu panjang, maka definisi mengenai apa yang dibangun dapat berubah, kompleksitas dapat meningkat dan resiko dapat bertambah. Setiap *Sprint* memiliki definisi mengenai apa yang akan dikembangkan, sebuah desain dan perancangan fleksibel yang akan membimbing pengembangan, pekerjaan yang akan dilakukan dan hasil dari produk.

Sprint dapat dibatalkan sebelum batasan waktu *Sprint* selesai. *Sprint* harus dibatalkan apabila *Sprint* menjadi tidak masuk akal lagi apabila dilanjutkan. Hanya *Product Owner* yang dapat membatalkan *Sprint*, walaupun keputusan yang dibuat dapat dipengaruhi oleh para *stakeholder*, Tim Pengembang dan *Scrum Master*. *Sprint* terdiri dari *Sprint Planning*, *Daily Scrum*, pengembangan, *Sprint Review* dan *Sprint Retrospective*.

Sprint Planning

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.

Ada beberapa pertanyaan yang harus dapat dijawab dalam melakukan *Sprint Planning*, yaitu:

- a. Apa tujuan atau *goal* dari *Sprint*?
- b. Apa yang dapat dikerjakan dalam *Sprint* ini?
- c. Apa yang harus dilakukan untuk menyelesaikan pekerjaan yang telah dipilih?

Daily Scrum

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.

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

- a. Apa yang sudah saya lakukan kemarin yang telah membantu Tim Pengembang mencapai *Sprint Goal*?
- b. Apa yang akan saya lakukan hari ini untuk membantu Tim Pengembang mencapai *Sprint Goal*?
- c. Apakah ada hambatan yang dapat menghalangi saya atau Tim Pengembang untuk mencapai *Sprint Goal*?

Sprint Review

Acara ini 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.

Hasil dari *Sprint Review* adalah revisi dari *Product Backlog* yang mendefinisikan kemungkinan item *Product Backlog* untuk *Sprint* berikutnya. *Product Backlog* dapat dirubah secara keseluruhan sebagai tanggapan atas peluang-peluang baru.

Sprint Retrospective

Pada acara ini 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.

Adapun tujuan dari *Sprint Retrospective* adalah sebagai berikut (Schwaber & Sutherland, 2017):

- a. Meninjau bagaimana *Sprint* telah selesai, termasuk hal-hal yang berkaitan dengan orang-orangnya, hubungan antara orang-orang, proses dan perangkat kerja;
- b. Mengidentifikasi dan mengurutkan hal-hal prioritas yang berjalan baik dan berpotensi untuk ditingkatkan;

- c. Membuat rencana implementasi dengan tujuan untuk meningkatkan cara kerja Tim *Scrum*.

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. Pada saat *Sprint Retrospective*, Tim *Scrum* merencanakan cara untuk meningkatkan kualitas dari produk, dengan merubah definisi dari “Selesai” sebagaimana dibutuhkan.

Di akhir *Sprint Retrospective*, Tim *Scrum* harus dapat mengidentifikasi peningkatan-peningkatan yang akan diimplementasikan di *Sprint* berikutnya. Mengimplementasikan peningkatan ini di *Sprint* berikutnya, merupakan salah satu bentuk adaptasi dari hasil peninjauan Tim *Scrum* itu sendiri. Walaupun peningkatan-peningkatan dapat diimplementasikan kapanpun juga, *Sprint Retrospective* memberikan kesempatan formal untuk fokus pada peninjauan dan adaptasi.

2.3.4 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(Schwaber & Sutherland, 2017).

Product Backlog

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.

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.

Sprint Backlog

Sprint Backlog adalah sekumpulan item *Product Backlog* yang telah dipilih untuk dikerjakan dalam sebuah *Sprint*, beserta rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai

fungsi apa yang akan tersedia di Inkremen berikutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsi tersebut menjadi potongan tambahan produk yang “Selesai”.

Increment

Increment adalah gabungan dari semua *Product Item* yang diselesaikan maupun nilai inkremen dari *Sprint* sebelumnya. Pada akhir *Sprint*, inkremen terbaru harus 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.

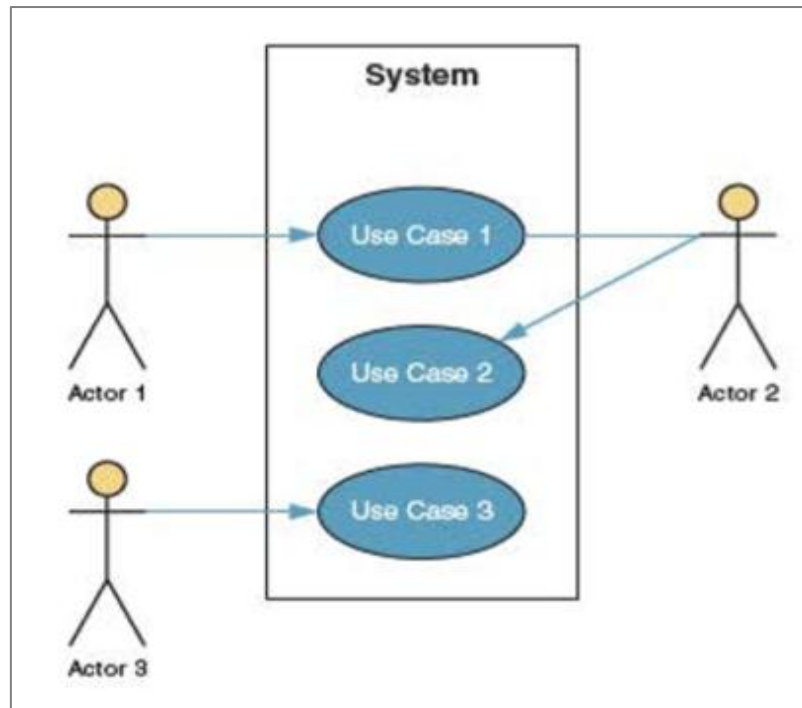
2.4 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah sekumpulan kaidah pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dari segi obyek (Whitten & Bentley, 2007).

Perkembangan UML berawal pada tahun 1994 ketika Grady Booch dan James Rumbaugh menggabungkan masing-masing metode pengembangan berorientasi objek milik mereka dengan tujuan menciptakan sebuah proses standar untuk sistem pengembangan berorientasi objek. Kemudian pada tahun 1995 Ivar Jacobson bergabung dan mengubah fokus mereka untuk membuat sebuah standar Bahasa pemodelan objek. UML versi 1.0 rilis pada 1997.

2.4.1 Use Case Diagram

Use Case merupakan sebuah diagram yang menggambarkan interaksi antara sistem dan pengguna, yaitu bagaimana pengguna akan berinteraksi dengan sistem dan siapa yang akan menggunakan sistem dengan cara mengidentifikasi dan menjelaskan fungsi sistem dari perspektif pengguna. *Use Case* mewakili sebuah tujuan dari sistem serta menjelaskan urutan kegiatan dan interaksi pengguna untuk mencapai tujuan pembuatan sistem. (Whitten & Bentley, 2007). Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.1.



Gambar 2.1 *Use Case Diagram*

Sumber: System Analysis & Design Method (2017)

Terdapat beberapa komponen yang dapat digambarkan dalam sebuah *use case* (Whitten & Bentley, 2007), diantaranya yang digunakan pada sistem ini sebagai berikut :

a. *Use Case*

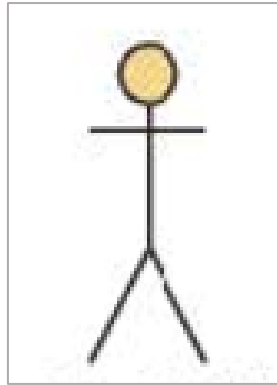
Use case digambarkan dalam bentuk elips yang merupakan aktivitas sistem yang dilakukan oleh *actor* seperti pada Gambar 2.2.



Gambar 2.2 Simbol *Use Case*

b. *Actor*

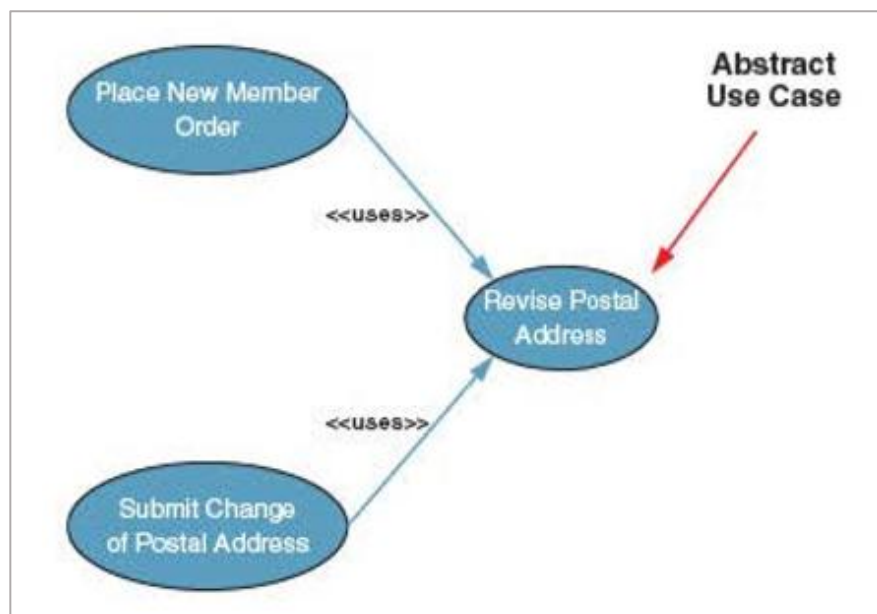
Actor merupakan pelaku yang melakukan interaksi dengan sistem. *Actor* disimbolkan seperti pada Gambar 2.3.



Gambar 2.3 Simbol Actor

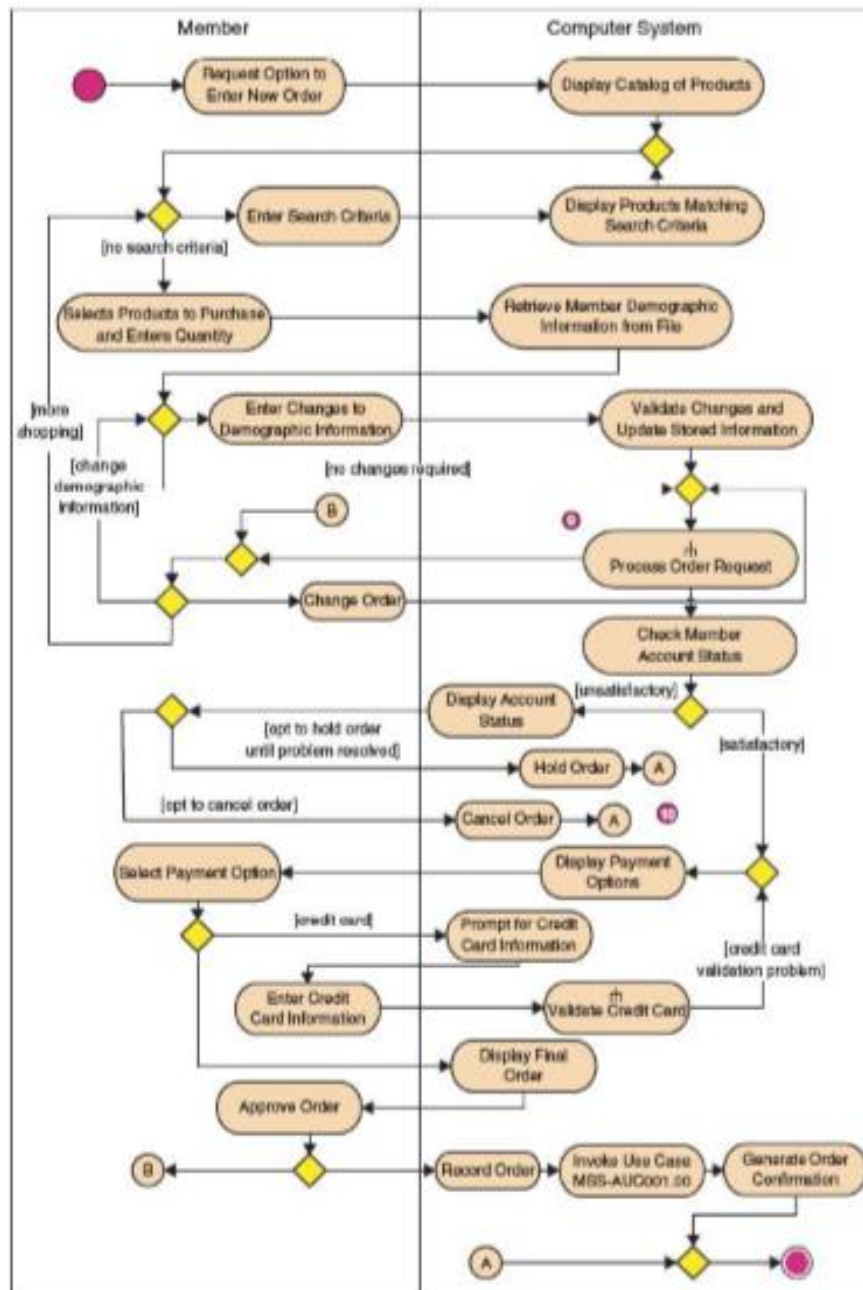
c. *Relationship*

Relationship digambarkan dalam bentuk tanda panah yang menghubungkan antara *actor* dengan *use case*. Pada sistem ini menggunakan *relationship* dengan bentuk *uses* atau *include*. *Uses* atau *include* berfungsi untuk menjelaskan hubungan ketergantungan yang terjadi pada *use case*, seperti pada contoh Gambar 2.4.

Gambar 2.4 *Uses or Include Relationship*

2.4.2 Activity Diagram

Activity Diagram digunakan sebagai model tahapan proses atau aktivitas yang ada pada sistem. Menggambarkan aliran aktivitas yang berurutan dari proses bisnis maupun *use case* seperti *flowchart*, namun mekanismenya pada *Activity Diagram* dibuat secara paralel. (Whitten & Bentley, 2007).



Gambar 2.5 Activity Diagram

Gambar 2.5 merupakan salah satu contoh *Activity Diagram* yang notasi sebagai berikut (Whitten & Bentley, 2007). :

a. *Initial Node*

Berbentuk lingkaran penuh, menggambarkan dimulainya sebuah proses.

b. *Action*

Berbentuk panjang dengan sisi melengkung, menggambarkan setiap tahapan. Urutan tindakannya dibuat sebanyak total aktivitas yang ditampilkan oleh diagram.

c. *Flow*

Tanda panah pada diagram menunjukkan perkembangan dengan tindakan. Kebanyakan tidak disertai kata-kata untuk mengidentifikasinya.

d. *Decision*

Berbentuk belah ketupat dengan 1 aliran masuk dan 2 atau lebih aliran keluar. Aliran yang dibuat untuk mengidentifikasi kondisi.

e. *Merge*

Berbentuk belah ketupat dengan 2 atau lebih aliran masuk dan 1 aliran keluar, digunakan untuk menyatukan yang dipisahkan oleh *decision*.

f. *Fork*

Berbentuk balok hitam panjang yang memiliki 1 aliran masuk dan 2 atau lebih aliran keluar, yang berperan untuk memecah alur paralel dalam sebuah aktivitas.

g. *Join*

Berbentuk balok hitam panjang yang memiliki 2 atau lebih aliran masuk dan 1 aliran keluar. Seluruh tindakan yang ada harus selesai sebelum melanjutkan proses selanjutnya.

h. *Activity Final*

Berbentuk lingkaran penuh didalam sebuah lingkaran lain yang menandakan berakhirnya suatu *Activity Diagram*.

2.5 Skala Likert

Skala likert merupakan salah satu metode penghitungan terhadap penilaian, sikap, pendapat dan persepsi seseorang dalam sebuah variabel penelitian. Variabel yang diukur menjadi sebuah indikator yang kemudian dijadikan sebagai titik tolak untuk menyusun item instrument berupa pernyataan atau pertanyaan (Sugiyono, 2010).

Jawabannya mempunyai gradasi dari sangat positif sampai sangat negatif kemudian dapat diberi skor, misalnya Sangat Setuju dengan skor 5, Setuju dengan skor 4, Ragu-ragu dengan Skor 3, Tidak Setuju dengan skor 2 dan Sangat Tidak Setuju dengan skor 1 (Sugiyono, 2010). Tabel 2.1 merupakan contoh perhitungan kuesioner dengan skala likert dalam bentuk *checklist*.

Tabel 2.1 Skala Likert bentuk *checklist*

No	Pertanyaan	Jawaban				
		SS	S	R	TS	STS
1	Sekolah ini akan menggunakan teknologi informasi dalam		√			

	pelayanan administrasi dan akademik					
2					

Kemudian dari kuesioner tersebut diberikan kepada 100 orang karyawan dengan hasil 25 orang menjawab SS, 40 orang menjawab S, 5 orang menjawab R, 20 orang menjawab TS dan 10 orang menjawab STS. Selanjutnya dianalisis dengan menghitung rata-rata pada setiap jawaban responden, dengan perhitungan sebagai berikut:

$$\text{Jumlah skor SS} = 25 \times 5 = 125$$

$$\text{Jumlah skor S} = 40 \times 4 = 160$$

$$\text{Jumlah skor R} = 5 \times 3 = 15$$

$$\text{Jumlah skor TS} = 20 \times 2 = 20$$

$$\text{Jumlah skor STS} = 10 \times 1 = 10$$

$$\text{Total} = 350 \quad (2.1)$$

Idealnya jika seluruh responden menjawab SS maka skor bernilai $5 \times 100 = 500$, sedangkan total skor yang diperoleh pada penelitian adalah 350. Berdasarkan data tersebut maka tingkat persetujuan responden terhadap pertanyaan dari kuesioner diatas adalah $(350 : 500) \times 100\% = 70\%$ dari yang diharapkan.

BAB III

METODOLOGI PENELITIAN

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

3.1 Identifikasi Masalah

3.1.1 Tujuan Penyelesaian Masalah

Tujuan penyelesaian masalah dalam penelitian ini adalah untuk mengidentifikasi prinsip-prinsip yang dibutuhkan oleh *Scrum Master* dalam pengelolaan Tim *Scrum*, serta melakukan implementasi aplikasi komputer untuk meningkatkan pengelolaan Tim *Scrum*.

3.1.2 Pengumpulan Data

Untuk dapat mencapai tujuan penyelesaian masalah, maka diperlukan pengumpulan data pendukung yang didapatkan dari berbagai sumber. Prosedur pengambilan data pendukung yang dilakukan pada tugas akhir ini dapat dilihat pada tabel 3.1.

Tabel 3.1 Prosedur Pengambilan Data

Teknik	Data	Sumber	Waktu
Kajian Pustaka	<ul style="list-style-type: none"> - Teori kerangka kerja <i>Scrum</i> secara umum - Teori kerangka kerja <i>Scrum</i> dalam tim <i>Scrum</i> dalam proyek pengembangan <i>software</i> 	Buku, jurnal dan internet	Januari s.d. Maret 2018
Observasi	<ul style="list-style-type: none"> - Penerapan manajemen kinerja tim <i>Scrum</i> dalam proyek pengembangan <i>software</i> - Kinerja Tim <i>Scrum</i> dalam proyek pengembangan <i>software</i>. 	Lingkungan kerja Tim <i>Scrum</i>	Februari s.d. Maret 2018
Wawancara	<ul style="list-style-type: none"> - Penerapan manajemen kinerja tim <i>Scrum</i> dalam proyek pengembangan <i>software</i>. - Kinerja Tim <i>Scrum</i> dalam proyek pengembangan <i>software</i>. 	Tim <i>Scrum</i>	Februari 2018

3.2 Pembuatan *Prototyping*

3.2.1 *Initial Requirement*

a. *User Requirement*

Untuk dapat mengetahui kebutuhan sistem dari sudut pandang pengguna, maka diperlukan *user requirement* yang di dapatkan dari hasil diskusi penulis dengan Dosen Pembimbing Tugas Akhir, seperti pada Tabel 3.1 berikut:

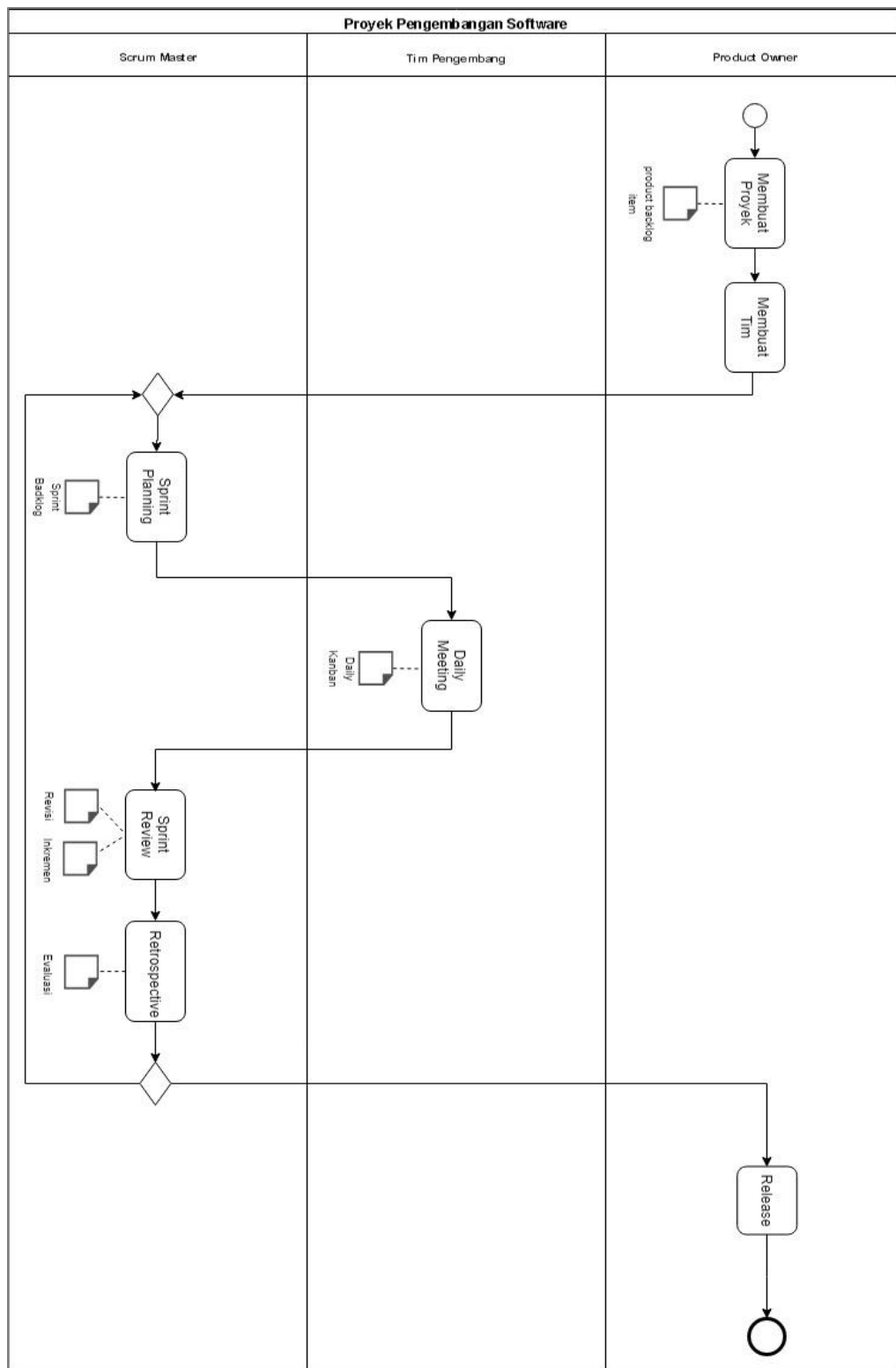
Tabel 3.2 *User Requirement*

No	Kebutuhan	Skenario
1	Sebagai <i>user</i> saya ingin dapat mendaftarkan diri ke sistem sehingga saya dapat melakukan <i>login</i> .	<ul style="list-style-type: none"> - Memasukkan nama, <i>username</i>, <i>e-mail</i> dan <i>password</i> - Menggunakan tombol <i>Register</i> untuk menambahkan data ke <i>database</i>
2	Sebagai <i>user</i> saya ingin dapat <i>login</i> sehingga saya dapat masuk ke sistem	<ul style="list-style-type: none"> - Memasukkan <i>username</i> dan <i>password</i> - Menggunakan tombol <i>Login</i> untuk masuk ke sistem
3	Sebagai <i>user</i> saya ingin memiliki halaman utama yang menampilkan aktivitas proyek yang saya kerjakan sehingga saya dapat mengetahui perkembangan proyek yang saya kerjakan	<ul style="list-style-type: none"> - Setelah <i>login</i> akan diarahkan ke halaman <i>dashboard</i> - Halaman <i>dashboard</i> dapat menampilkan <i>working on project</i> dan <i>recent activity</i> dari proyek yang dikerjakan <i>user</i>
4	Sebagai <i>user</i> saya ingin memiliki halaman proyek sehingga saya dapat melihat seluruh daftar proyek yang pernah dan sedang saya kerjakan	<ul style="list-style-type: none"> - Sistem memiliki halaman <i>projects</i> - Halaman <i>projects</i> dapat menampilkan seluruh proyek yang melibatkan <i>user</i>
5	Sebagai <i>Product Owner</i> saya ingin membuat proyek baru sehingga saya dapat mengelola proyek yang telah saya buat.	<ul style="list-style-type: none"> - Disediakan tombol untuk membuat proyek baru - Masuk ke halaman berupa formulir pembuatan proyek baru - Mengisi formulir - Menggunakan tombol <i>submit</i> untuk menambahkan data ke <i>database</i>
6	Sebagai <i>Product Owner</i> saya ingin mengelola tim sehingga saya dapat memilih <i>user</i> yang akan terlibat dalam proyek yang saya buat	<ul style="list-style-type: none"> - Sistem memiliki halaman <i>Team</i> - Terdapat tombol <i>Create Team</i> untuk dapat membuat tim baru - Masuk ke halaman berupa formulir - Mengisi formulir untuk menentukan nama tim dan daftar <i>user</i> yang ingin dilibatkan dalam proyek - Menggunakan tombol <i>submit</i> untuk menambahkan data ke <i>database</i>
7	Sebagai <i>Product Owner</i>	- Sistem memiliki halaman <i>Backlog</i>

	saya ingin mengelola <i>product Backlog item</i> berdasarkan permintaan <i>client</i> yang nantinya akan menjadi pengguna, sehingga saya dapat memberikan <i>task</i> kepada anggota Tim <i>Scrum</i>	untuk mengelola <i>product Backlog item</i> . <ul style="list-style-type: none"> - Terdapat tombol untuk membuat <i>product Backlog</i> beserta <i>item</i>. - Mengisi formulir untuk membuat <i>product Backlog item</i> - <i>Product Backlog Item</i> di deskripsikan dengan bentuk <i>user story</i> - Setiap anggota tim mendapatkan <i>task</i> berdasarkan kesepakatan pengambilan tugas pada saat <i>Sprint Planning</i> - Menggunakan tombol submit untuk menambahkan data formulir ke <i>database</i>
8	Sebagai <i>Scrum Master</i> saya ingin dapat membuat <i>Sprint</i> baru sehingga saya dapat mengelola <i>Sprint</i> dan Tim dapat mengerjakan <i>task</i>	<ul style="list-style-type: none"> - Terdapat tombol <i>create Sprint</i> untuk membuat <i>Sprint</i> baru - Masuk ke halaman berupa formulir - Mengisi formulir untuk menentukan nama <i>Sprint</i>, waktu <i>Sprint</i> dan <i>Sprint goal</i>. - Saat <i>Sprint</i> selesai, <i>Scrum Master</i> dapat mengisi formulir <i>Sprint review</i>. - Menggunakan tombol <i>submit</i> untuk menambahkan data ke <i>database</i> - Menggunakan tombol <i>close Sprint</i> untuk menyatakan sebuah <i>Sprint</i> telah selesai dilaksanakan.
9	Sebagai Tim Pengembang saya ingin memiliki <i>task board</i> sehingga saya dapat mengetahui daftar pekerjaan yang akan, sedang dan sudah Tim saya lakukan	<ul style="list-style-type: none"> - Sistem memiliki halaman berisi tabel <i>product Backlog item</i> dalam sebuah <i>Sprint</i> - Tabel berupa <i>Kanban Board</i> terdiri dari kolom <i>To Do</i>, <i>In Progress</i>, dan <i>Done</i>. - Setiap <i>task</i> yang sudah dikerjakan dapat dipindahkan dengan <i>drag and drop</i> dari satu kolom ke kolom lainnya.

b. Kebutuhan Proses Bisnis

Untuk menggambarkan logika dari langkah-langkah proses bisnis pada sistem ini, digunakan BPMN (*Business Process Modeling Notation*) seperti pada Gambar 3.1. Terdapat 1 *start event*, 1 *end event*, 6 *task*, 2 *gateway* dan 6 dokumen dalam proses bisnis sistem ini.



Gambar 3.1 Proses Bisnis Sistem

Proses dimulai dengan *Product Owner* membuat proyek baru di sistem beserta dengan dokumen *Product BacklogItem*, kemudian membentuk Tim Scrum untuk terlibat pada proyek tersebut. Proses selanjutnya adalah melakukan *Sprint Planning* untuk membahas

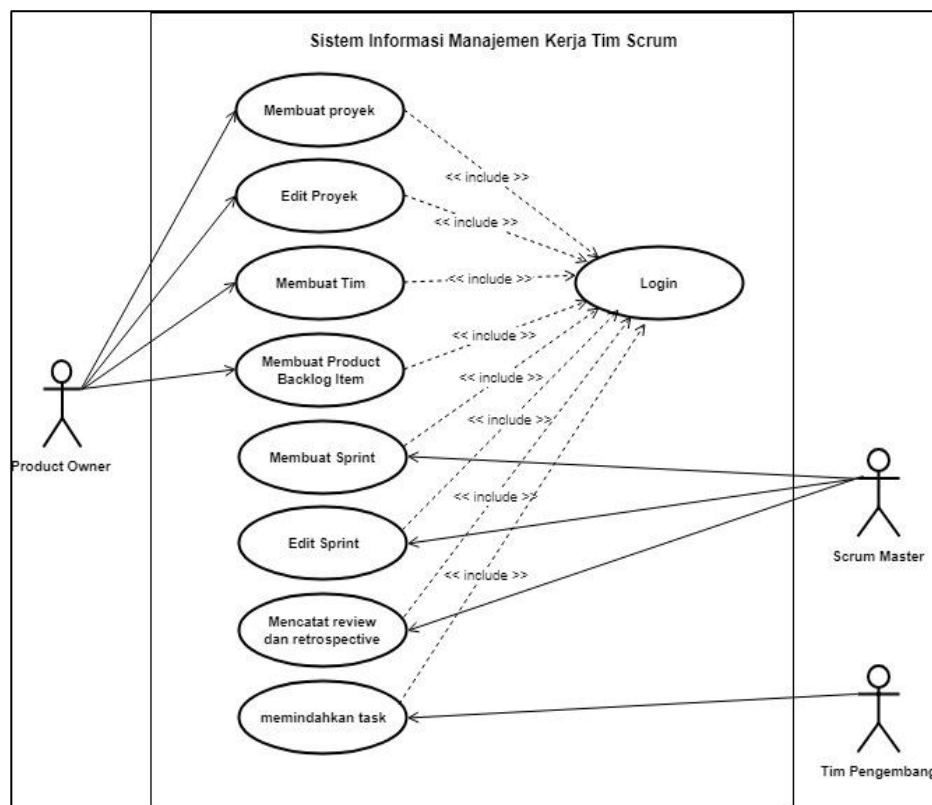
Product Backlog Item yang akan dikerjakan dalam satu *Sprint*, proses ini menghasilkan *Sprint Backlog* yang didokumentasikan oleh *Scrum Master*.

Tim Pengembang akan melakukan *daily meeting* untuk membahas perkembangan proyek yang akan didokumentasikan menjadi *Daily Kanban*. Setelah itu akan ada proses *Sprint Review* yang menghasilkan catatan revisi *Sprint* dan *Increment*, serta *Retrospective* yang menghasilkan catatan evaluasi di akhir *Sprint*. Setelah *Sprint* tersebut selesai, jika masih ada *Sprint* yang harus dijalankan maka akan dilakukan *Sprint Planning* yang baru, jika sudah tidak ada *Sprint* yang harus dijalankan maka hasil dari pengembangan proyek siap untuk di *Release*.

3.2.2 Design

a. Use Case

Untuk mengetahui fungsi yang ada pada sebuah sistem beserta interaksinya, dibuat desain *use case diagram* sebagai berikut:



Gambar 3.2 Use Case Diagram Sistem

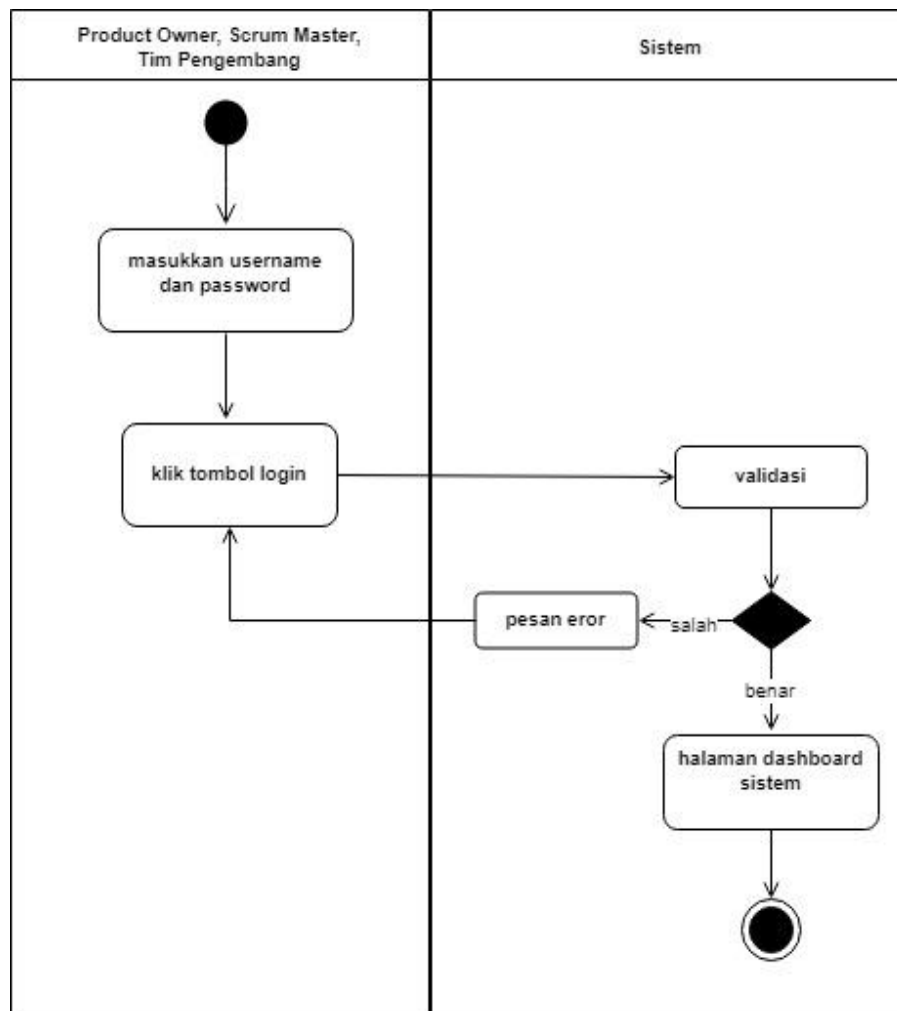
Use Case Diagram pada Gambar 3.1 menjelaskan fungsionalitas sistem yang akan diimplementasikan pada bab berikutnya. Terdapat 3 aktor yang berperan dalam sistem

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*.

b. *Activity Diagram*

Selanjutnya untuk mengetahui *detail* interaksi yang terjadi pada masing-masing fungsi yang sudah dibuat pada *use case*, dibuat desain *Activity Diagram* sebagai berikut:

1. Fungsi *Login*



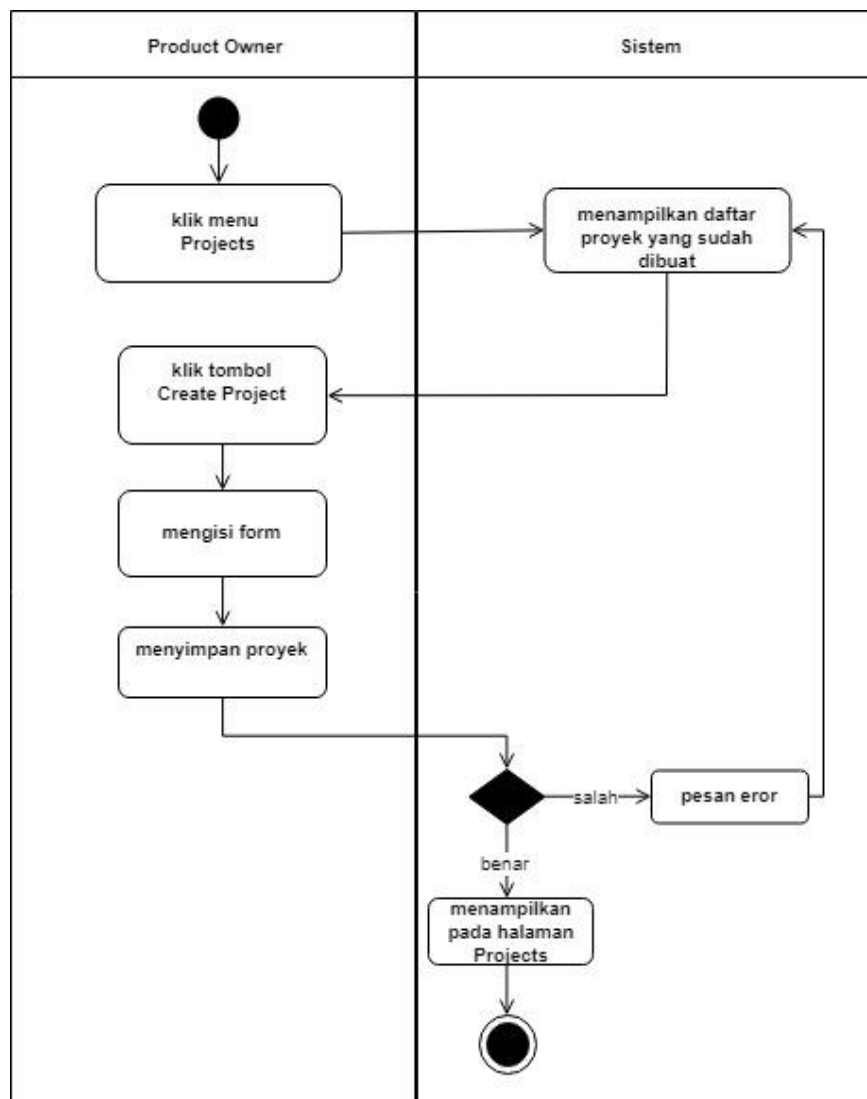
Gambar 3.3 *Activity Diagram* Login

Seluruh aktor atau pengguna sistem dapat melakukan fungsi login, dimulai dengan memasukkan *username* dan *password* pada form login, kemudian klik tombol login.

Selanjutnya sistem akan melakukan validasi terhadap *username* dan *password* tersebut, apabila benar sistem akan menampilkan halaman Dashboard, namun jika salah maka akan muncul pesan *error* dan pengguna dapat kembali memasukkan *username* dan *password* seperti yang digambarkan pada Gambar 3.3.

2. Fungsi Membuat Proyek

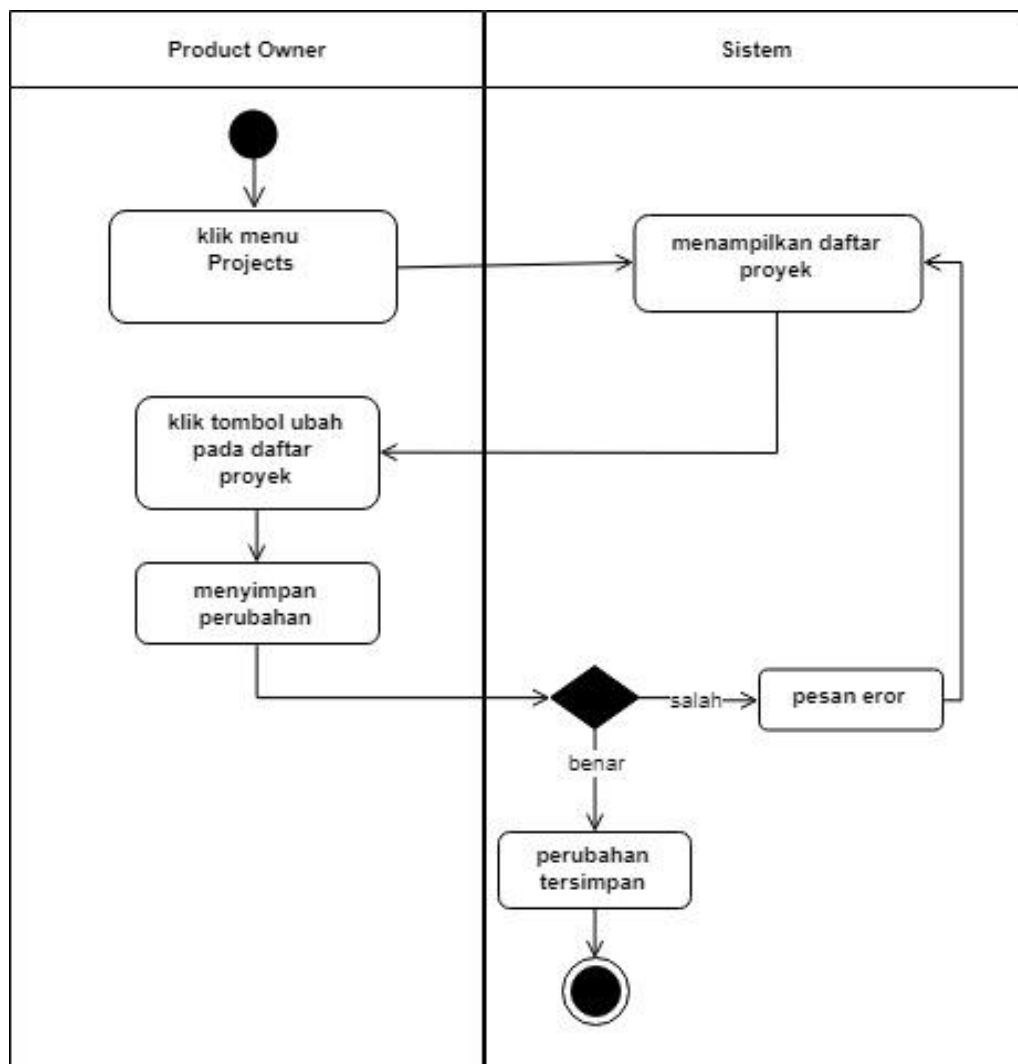
Fungsi membuat proyek baru ditunjukkan pada Gambar 3.4, dimana ketika Product Owner memilih menu Projects akan muncul halaman berisikan daftar proyek yang sudah pernah dibuat. Selanjutnya klik tombol Create Project untuk mengisi form pembuatan proyek baru, lalu simpan proyek. Apabila proyek berhasil disimpan maka akan muncul pada daftar proyek, jika salah akan muncul pesan error dan sistem akan menampilkan kembali halaman Projects.



Gambar 3.4 Activity Diagram Membuat Proyek

3. Fungsi Mengubah Rincian Proyek

Gambar 3.5 menggambarkan fungsi mengubah proyek, dimana ketika Product Owner memilih menu Projects akan muncul halaman berisikan daftar proyek yang sudah pernah dibuat. Selanjutnya klik tombol *Edit Project* untuk mengisi form perubahan proyek, lalu simpan proyek. Apabila proyek berhasil disimpan maka akan muncul pada daftar proyek, jika salah akan muncul pesan error dan sistem akan menampilkan kembali halaman Projects.

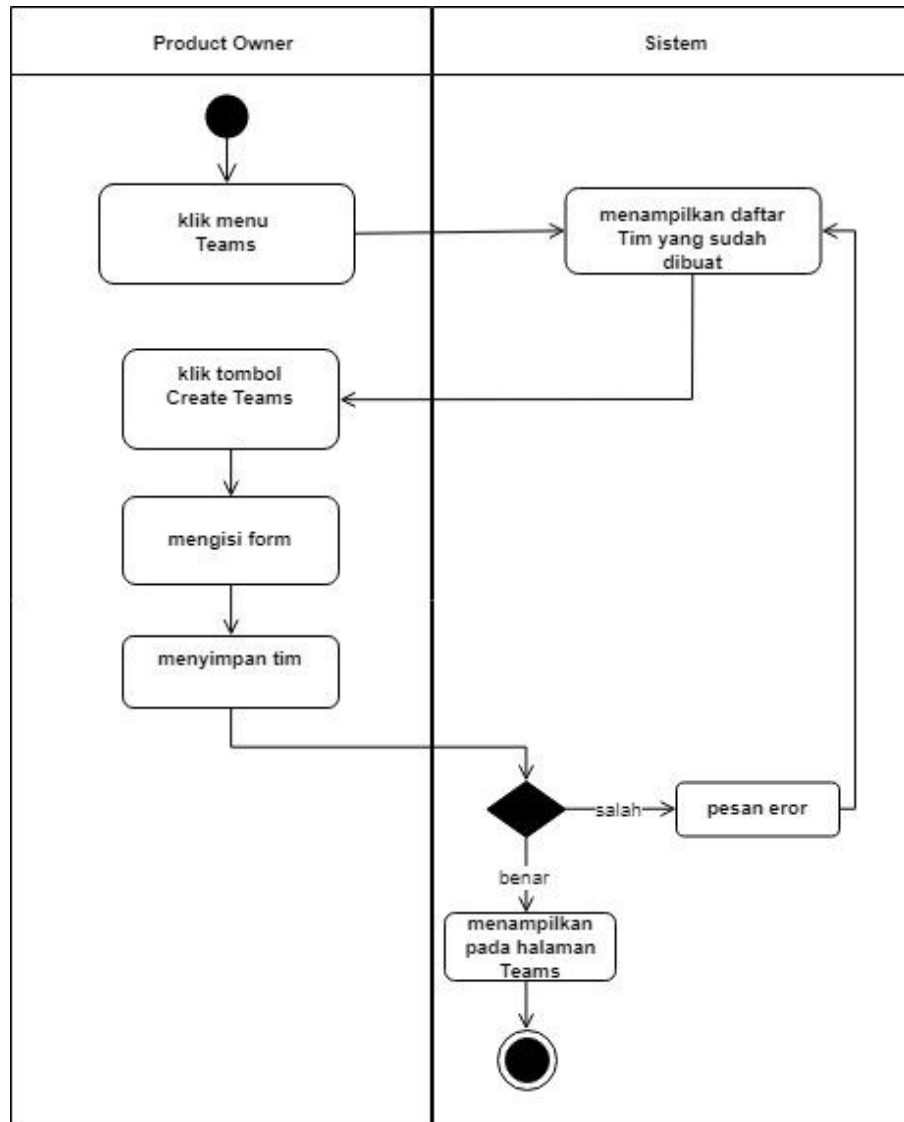


Gambar 3.5 Activity Diagram Edit Proyek

4. Fungsi Membuat Tim

Fungsi membuat tim baru ditunjukkan pada Gambar 3.6, dimana ketika *Product Owner* memilih menu *Teams* akan muncul halaman berisikan daftar tim yang sudah pernah dibuat. Selanjutnya klik tombol *Create Project* untuk mengisi form

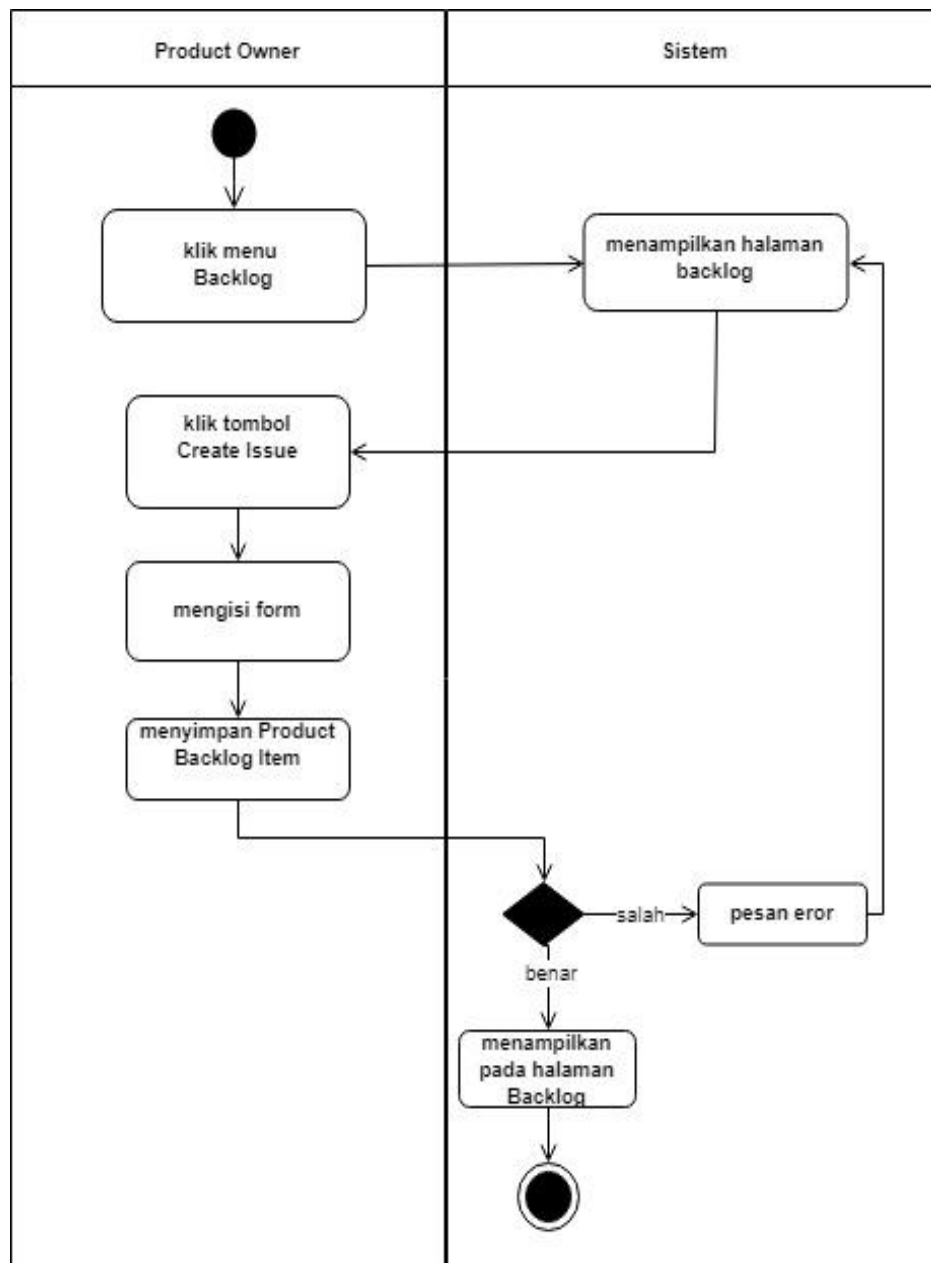
pembuatan proyek baru, lalu simpan proyek. Apabila proyek berhasil disimpan maka akan muncul pada daftar proyek, jika salah akan muncul pesan error dan sistem akan menampilkan kembali halaman Projects.



Gambar 3.6 Activity Diagram Membuat Tim

5. Fungsi Membuat Product Backlog Item

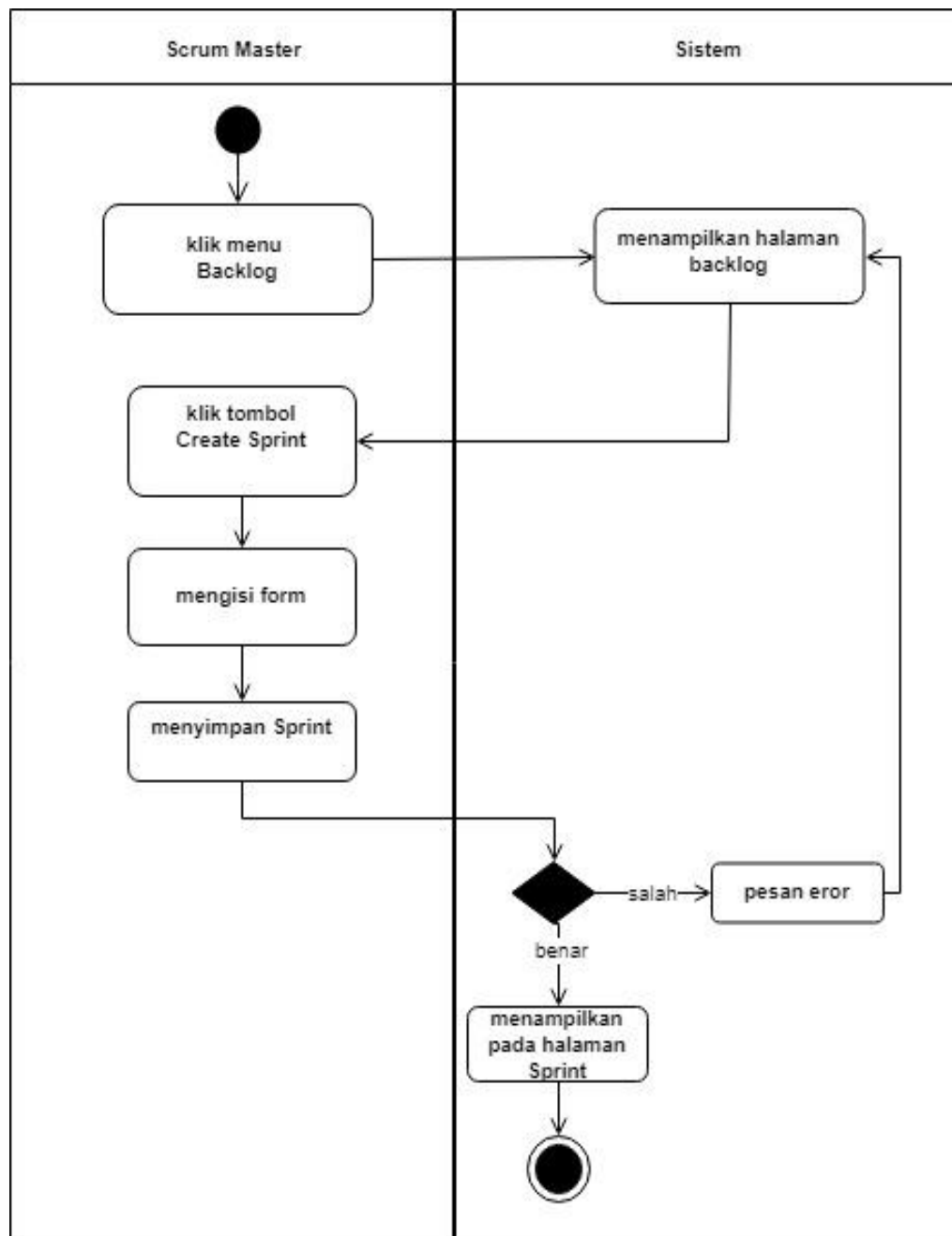
Gambar 3.7 menggambarkan fungsi membuat *task* baru, dimana ketika *Product Owner* memilih menu *Backlog* akan muncul halaman berisikan daftar *issue* yang sudah pernah dibuat. Selanjutnya klik tombol *Create Issue* untuk mengisi form pembuatan *issue* baru, lalu simpan. Apabila *issue* berhasil disimpan maka akan muncul pada daftar *issue*, jika salah akan muncul pesan *error* dan sistem akan menampilkan kembali halaman *Backlog*.



Gambar 3.7 Activity Diagram Membuat Product Backlog Item

6. Fungsi Membuat *Sprint*

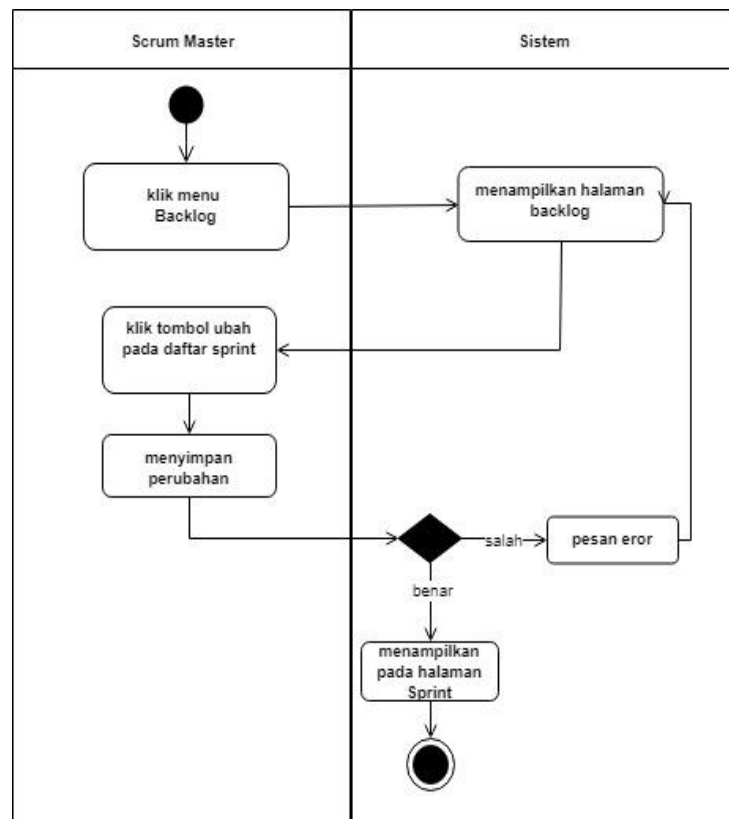
Gambar 3.8 menggambarkan fungsi membuat *Sprint* baru, dimana ketika *Scrum* Master memilih menu *Backlog* akan muncul halaman berisikan daftar *Sprint* yang sudah pernah dibuat. Selanjutnya klik tombol *Create Sprint* untuk mengisi form pembuatan *Sprint* baru, lalu simpan. Apabila *Sprint* berhasil disimpan maka akan muncul pada daftar *Sprint*, jika salah akan muncul pesan *error* dan sistem akan menampilkan kembali halaman *Backlog*.



Gambar 3.8 Activity Diagram Membuat Sprint

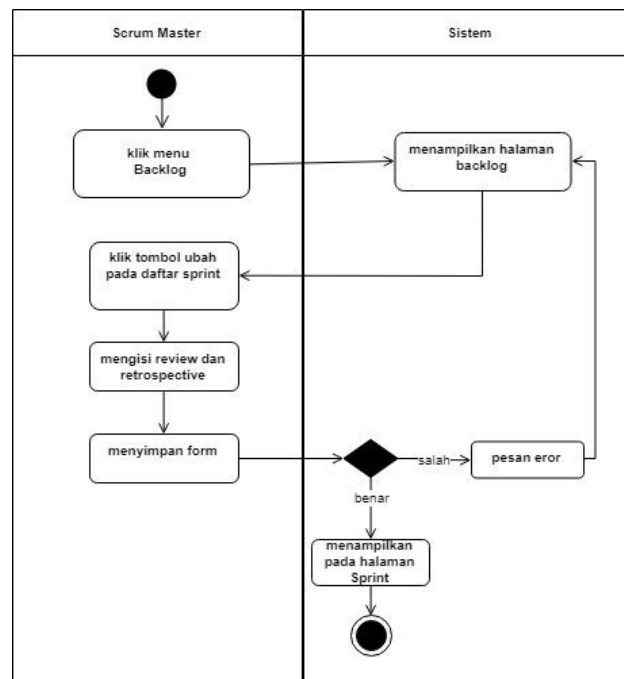
7. Fungsi Mengubah Rincian Sprint

Fungsi mengubah *Sprint* ditunjukkan pada Gambar 3.9, dimana ketika *Scrum Master* memilih menu *Backlog* akan muncul halaman berisikan daftar *Sprint* yang sudah pernah dibuat. Selanjutnya klik tombol *EditSprint* untuk mengisi form perubahan *Sprint*, lalu simpan *Sprint*. Apabila *Sprint* berhasil disimpan maka akan muncul pada daftar *Sprint*, jika salah akan muncul pesan error dan sistem akan menampilkan kembali halaman *Backlog*.



Gambar 3.9 Activity Diagram Edit Sprint

8. Fungsi Review & Retrospective

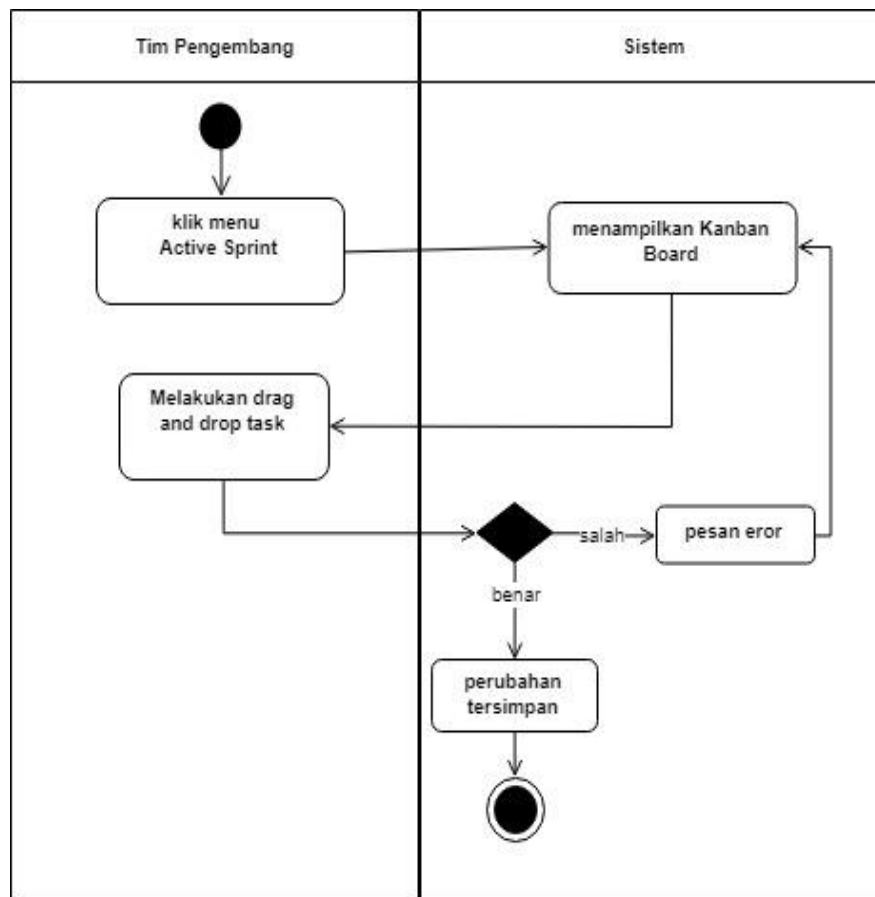


Gambar 3.10 Activity Diagram Mencatat Review dan Retrospective

Fungsi untuk menyimpan catatan *Review* dan *Retrospective* yang dimiliki *Sprint* ditunjukkan pada Gambar 3.10. Pada halaman *Backlog* akan muncul daftar *Sprint* yang memiliki tombol *Edit Sprint*, selanjutnya klik tombol untuk masuk ke formulir perubahan *Sprint*. Isikan *Review* dan *Retrospective* pada formulir yang tersedia, lalu simpan.

9. Fungsi Memindahkan *Task*

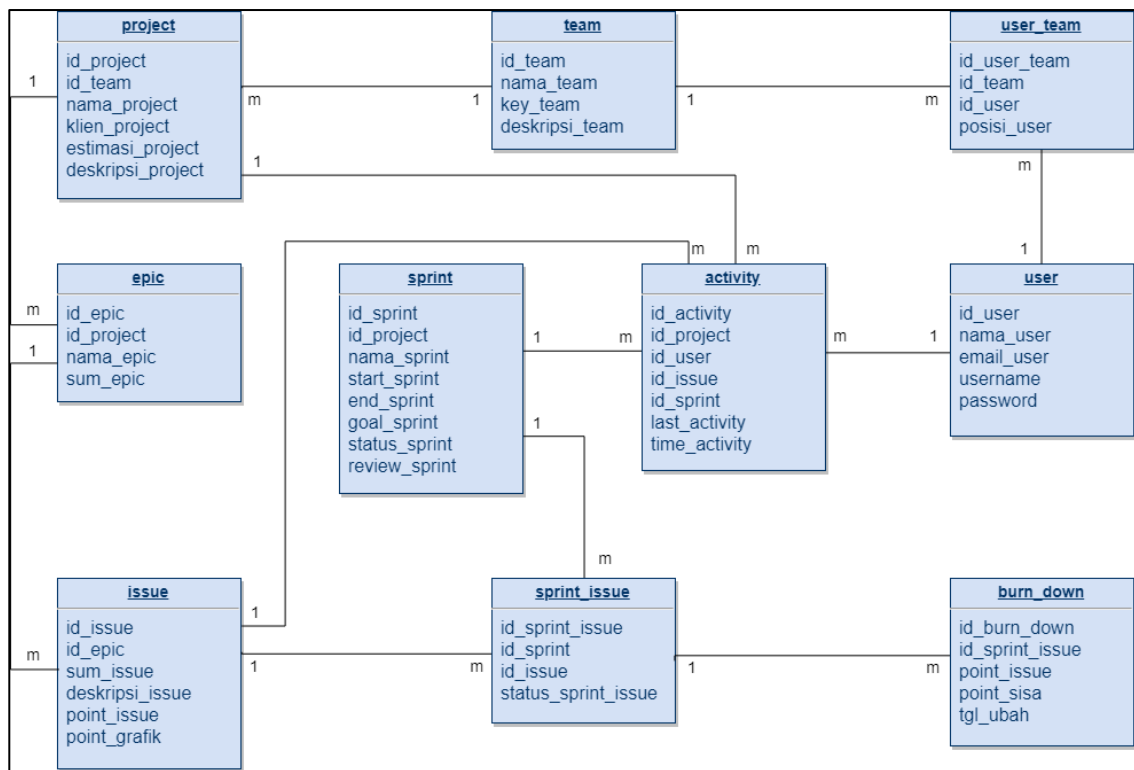
Gambar 3.11 menggambarkan fungsi untuk melakukan pemindahan *task*. Tim Pengembang melakukan klik menu *Active Sprint*, kemudian sistem akan memunculkan halaman berisikan Kanban Board. Pada halaman tersebut, Tim Pengembang dapat melakukan *drag and drop*.



Gambar 3.11 Activity Diagram Memindahkan Task

c. Relasi Antar Tabel

Pada perancangan *database* sistem ini, 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 3.12.



Gambar 3.12 Relasi Antar Tabel

d. Struktur Tabel

Untuk mengetahui lebih rinci mengenai atribut yang dimiliki setiap tabel, dibuat struktur tabel sebagai berikut:

Tabel 3.3 Tabel user

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_user	Integer(5)	Primary_key	Not Null
2	nama_user	Varchar (50)		Not Null
3	email_user	Varchar(25)		Not Null
4	username	Varchar(15)		Not Null
5	password	Varchar(255)		Not Null

Tabel user digunakan untuk menyimpan data pengguna sistem seperti pada Tabel 3.3, terdiri dari 5 kolom atribut. Berikut masing-masing penjelasan atribut :

id_user : Tipe data integer, batasan karakter 5, *Primary Key*, *AutoIncrement*
 nama_user : Tipe data varchar, batasan karakter 50
 email_user : Tipe data varchar, batasan karakter 25
 username : Tipe data varchar, batasan karakter 15

password : Tipe data varchar, batasan karakter 255

Tabel 3.4 Tabel project

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_project	Integer(11)	Primary_key	Not Null
2	id_team	Integer(5)	Foreign_key	Tabel team
3	nama_project	Varchar(100)		Not Null
4	klien_project	Varchar(100)		Not Null
5	estimasi_project	Varchar(50)		Not Null
6	deskripsi_project	Text		Not Null

Tabelproject digunakan untuk menyimpan data proyek seperti pada Tabel 3.4, terdiri dari 6 kolom atribut. Berikut masing-masing penjelasan atribut :

id_project :Tipe data integer, batasan karakter 11, *Primary Key*, Auto Increment.

id_team :Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *team*, digunakan untuk menyimpan tim yang terlibat dalam pengembangan proyek.

nama_project : Tipe data varchar, batasan karakter 100.

klien_project : Tipe data varchar, batasan karakter 100.

estimasi_project : Tipe data varchar, batasan karakter 50.

deskripsi_project : Tipe data text.

Tabel 3.5 Tabel team

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_team	Integer(5)	Primary_key	Not Null
2	nama_team	Varchar(50)		Not Null
3	key_team	Varchar(15)		Not Null
4	deskripsi_team	Text		Not Null

TabelTeam digunakan untuk menyimpan data Tim *Scrum* seperti pada Tabel 3.5, terdiri dari 4 kolom atribut. Berikut masing-masing penjelasan atribut :

id_team : Tipe data integer, batasan karakter 5, *Primary Key*, Auto Increment.

nama_team : Tipe data varchar, Batasan karakter 50

id_user : Tipe data integer, batasan karakter 5, *Foreign Key*, digunakan untuk menyimpan tim yang terlibat dalam pengembangan proyek.

nama_team : Tipe data varchar, batasan karakter 50

key_team : Tipe data varchar, batasan karakter 15, digunakan untuk menyimpan nama bersifat unik dari sebuah Tim *Scrum*.

deskripsi_team : Tipe data text.

Tabel 3.6 Tabel epic

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_epic	Integer(5)	Primary_key	Not Null
2	id_project	Integer(5)	Foreign_key	Tabel project
3	nama_epic	Varchar(75)		Not Null
4	sum_epic	Text		Not Null

TabelEpic digunakan untuk menyimpan *highlight task* pada sebuah proyek seperti pada Tabel 3.6, terdiri dari 4 kolom atribut. Berikut masing-masing penjelasan atribut :

id_epic : Tipe data integer, batasan karakter 5, *Primary Key*, *AutoIncrement*

id_project : Tipe data integer, batasan karakter 5, *Foreign Key* dari table project digunakan untuk menyimpan proyek yang diberikan *epic* tertentu

nama_epic : Tipe data varchar, batasan karakter 75

sum_epic : Tipe data *Text*, digunakan untuk menyimpan ringkasan deskripsi epic

Tabel 3.7 Tabel issue

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_issue	Integer(5)	Primary_key	Not Null
2	id_epic	Integer(5)	Foreign_key	Tabel Epic
3	sum_issue	Varchar(255)		Not Null
4	deskripsi_issue	Text		Not Null
5	point_issue	Integer(5)		Not Null
6	point_grafik	Integer(10)		Not Null

TabelIssue digunakan untuk menyimpan *task* berupa *product Backlog item* seperti pada Tabel 3.7, terdiri dari 6 kolom atribut. Berikut masing-masing penjelasan atribut :

id_issue : Tipe data integer, batasan karakter 5, *Primary Key*, *Auto Increment*.

id_epic : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel epic, digunakan untuk menyimpan *issue* yang dimiliki *epic* tertentu.

sum_issue	:Tipe data varchar, batasan karakter 255, digunakan untuk menyimpan ringkasan <i>issue</i> .
deskripsi_issue	:Tipe data text, digunakan untuk menyimpan deskripsi <i>issue</i> .
point_issue	:Tipe data integer, batasan karakter 5, digunakan untuk menyimpan poin estimasi yang dimiliki sebuah <i>task</i> .
point_grafik	:Tipe data integer, batasan karakter 10, digunakan untuk menyimpan poin estimasi yang mempengaruhi grafik burndown.

Tabel 3.8 Tabel sprint

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_sprint	Integer(5)	<i>Primary_key</i>	<i>Not Null</i>
2	id_project	Integer (5)	<i>Foreign_key</i>	Tabel project
3	nama_sprint	Varchar(25)		<i>Not Null</i>
4	start_sprint	Date		<i>Not Null</i>
5	end_sprint	Date		<i>Not Null</i>
6	goal_sprint	Varchar(255)		<i>Not Null</i>
7	status_sprint	Varchar(20)		<i>Not Null</i>
8	review_sprint	Text		<i>Not Null</i>

Tabel *Sprint* digunakan untuk menyimpan data mengenai tugas pada proyek yang dikerjakan dalam batasan waktu tertentu seperti pada tabel 3.8, terdiri dari 8 kolom atribut. Berikut masing-masing penjelasan atribut :

id_sprint	: Tipe data integer, batasan karakter 5, <i>Primary Key</i> , Auto Increment.
id_project	: Tipe data integer, batasan karakter 5, <i>Foreign Key</i> dari tabel project.
nama_sprint	: Tipe data Varchar, batasan karakter 25.
start_sprint	:Tipe data date, digunakan untuk menyimpan tanggal dimulainya <i>Sprint</i> .
end_sprint	:Tipe data date, digunakan untuk menyimpan tanggal <i>Sprint</i> berakhir.
goal_sprint	:Tipe data varchar, batasan karakter 255, digunakan untuk menyimpan tujuan dari sebuah <i>Sprint</i> yang dibuat.
status_sprint	:Tipe data varchar, batasan karakter 20.
review_sprint	:Tipe data text, digunakan untuk menyimpan catatan hasil sebuah <i>Sprint</i> .

Tabel 3.9 Tabel user_team

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_user_team	Integer(5)	<i>Primary_key</i>	<i>Not Null</i>
2	id_team	Integer(5)	<i>Foreign_key</i>	Tabel team
3	id_user	Integer(5)	<i>Foreign_key</i>	Tabel user
4	posisi_user	Varchar(25)		<i>Not Null</i>

Tabel *user_team* digunakan untuk menyimpan *user* yang terlibat dalam sebuah tim seperti pada Tabel 3.9, terdiri dari 4 kolom atribut. Berikut masing-masing penjelasan atribut :

id_user_team : Tipe data integer, batasan karakter 5, *Primary Key*, Auto Increment.

id_team : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *team*.

id_user : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *user*.

posisi_user : Tipe data varchar, Batasan karakter 25, digunakan untuk menyimpan posisi *user* dalam tim.

Tabel 3.10 Tabel sprint_issue

No	Atribut	Tipe Data	Constrain	Keterangan
1	id_sprint_issue	Integer(5)	<i>Primary_key</i>	<i>Not Null</i>
2	id_sprint	Integer(5)	<i>Foreign_key</i>	Tabel <i>Sprint</i>
3	id_issue	Integer(5)	<i>Foreign_key</i>	Tabel <i>issue</i>
4	Status_sprint_issue	Varchar(10)		<i>Not Null</i>

Tabel *Sprint_issue* digunakan untuk menyimpan *task issue* berupa *user story* seperti pada Tabel 3.10, terdiri dari 4 kolom atribut. Berikut masing-masing penjelasan atribut :

id_sprint_issue : Tipe data integer, batasan karakter 5, *Primary Key*, Auto Increment.

id_sprint : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *Sprint*.

id_issue : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *issue*.

status_sprint_issue : Tipe data varchar, batasan karakter 10.

Tabel 3.11 Tabel activity

No	Atribut	Tipe Data	Constrain	Keterangan
----	---------	-----------	-----------	------------

1	id_activity	Integer(11)	<i>Primary_key</i>	<i>Not Null</i>
2	id_project	Integer(11)	<i>Foreign_key</i>	Tabel project
3	id_user	Integer(11)	<i>Foreign_key</i>	Tabel user
4	id_issue	Integer(10)	<i>Foreign_key</i>	Tabel issue
5	id_Sprint	Integer(5)	<i>Foreign_key</i>	Tabel <i>Sprint</i>
6	last_activity	Text		<i>Not Null</i>
7	time_activity	Datetime		<i>Not Null</i>

Tabelactivity digunakan untuk menyimpan bentuk aktivitas dari Kanban Board untuk mencatat perpindahan *task* saat sudah selesai dikerjakanseperti pada tabel 3.11, terdiri dari 7 kolom atribut. Berikut masing-masing penjelasan atribut :

- id_activity : Tipe data integer, batasan karakter 11, *Primary Key*, Auto Increment.
- id_project : Tipe data integer, batasan karakter 11, *Foreign Key* dari tabel project.
- id_user : Tipe data integer, batasan karakter 11, *Foreign Key* dari tabel user.
- id_issue : Tipe data integer, batasan karakter 10, *Foreign Key* dari tabel issue.
- id_sprint : Tipe data integer, batasan karakter 5, *Foreign Key* dari tabel *Sprint*.
- last_activity : Tipe data text, digunakan untuk menyimpan aktivitas terakhir dalam sebuah *Sprint* yang sedang berjalan.
- time_activity :Tipe data datetime, digunakan untuk menyimpan waktu terjadinya aktivitas terakhir dalam sebuah *Sprint* yang sedang berjalan.

Tabel 3.12 Tabel burn_down

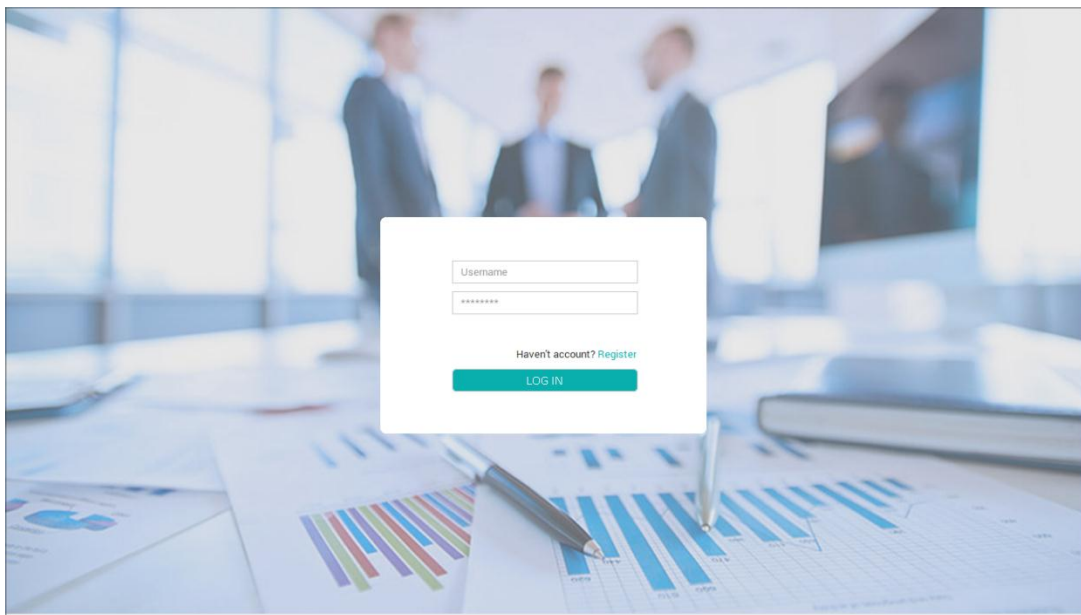
No	Atribut	Tipe Data	Constrain	Keterangan
1	id_burn_down	Integer(10)	<i>Primary_key</i>	<i>Not Null</i>
2	id_sprint_issue	Integer(10)	<i>Foreign_key</i>	Tabel sprint_issue
3	point_issue	Integer(10)		<i>Not Null</i>
4	point_sisa	Integer(10)		<i>Not Null</i>
5	tgl_ubah	Date		<i>Not Null</i>

Tabelburn_down digunakan untuk menyimpan data yang diperlukan untuk menghasilkan grafik burndowndalam sebuah *Sprint*seperti pada Tabel 3.12, terdiri dari 5 kolom atribut. Berikut masing-masing penjelasan atribut :

<code>id_burn_down</code>	:Tipe data integer, batasan karakter 10, <i>Primary Key</i> , <i>AutoIncrement</i> .
<code>id_sprint_issue</code>	:Tipe data integer, batasan karakter 10, <i>Foreign Key</i> dari tabel <i>Sprint_issue</i> .
<code>point_issue</code>	:Tipe data integer, Batasan karakter 10, digunakan untuk menyimpan poin estimasi pengerjaan sebuah <i>issue</i> .
<code>point_sisa</code>	:Tipe data integer, Batasan karakter 10, digunakan untuk menyimpan jumlah sisa poin estimasi yang dimiliki <i>proyek</i> .
<code>tgl_ubah</code>	:Tipe data date, digunakan untuk menyimpan tanggal saat terjadinya perubahan yang mempengaruhi grafik burndown.

3.2.3 Prototyping

a. Desain Tampilan Halaman *Login*



Gambar 3.13 Desain Tampilan Halaman *Login*

Halaman ini merupakan tampilan awal sistem, seperti pada gambar 3.13. *User* harus memasukkan *username* dan *password* untuk dapat melakukan peminjaman. Terdapat tombol *Login* untuk masuk ke dalam sistem.

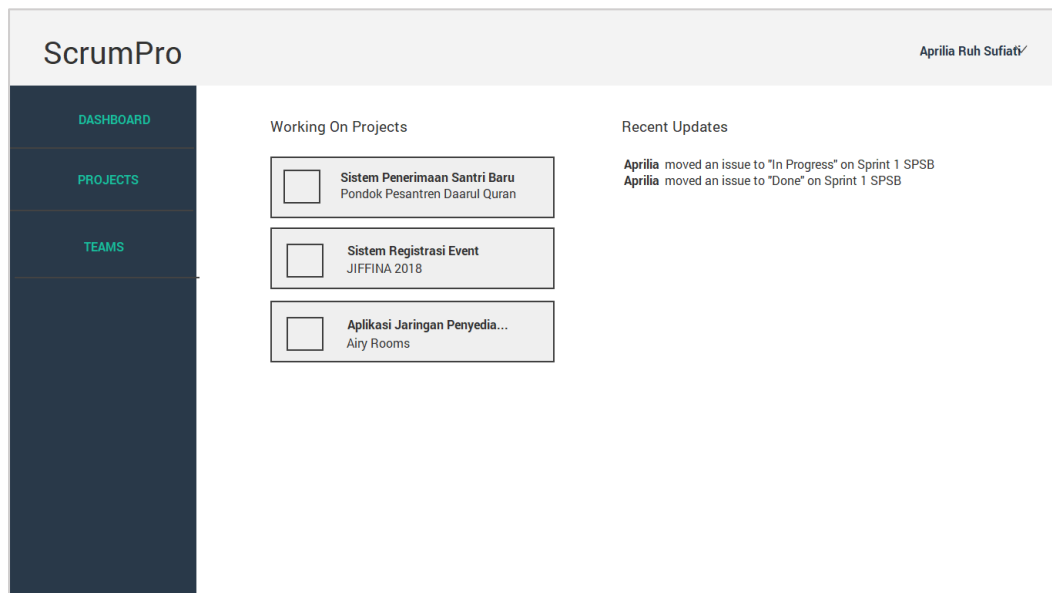
b. Desain Tampilan Halaman *Register*

Gambar 3.14 merupakan tampilan *form* sederhana untuk registrasi pengguna baru yang sebelumnya belum memiliki akses untuk masuk ke sistem. *Form* yang harus diisi berupa nama, *email*, *username*, dan *password*.



Gambar 3.14 Desain Tampilan Halaman *Register*

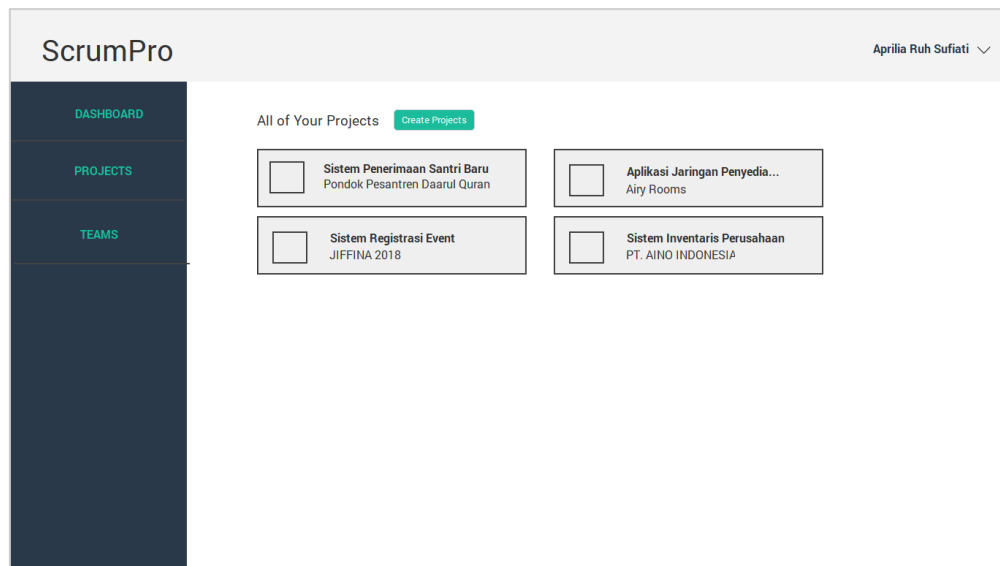
c. Desain Tampilan Halaman *Dashboard*



Gambar 3.15 Desain Tampilan Halaman *Dashboard*

Pada gambar 3.15 terdapat tampilan halaman *dashboard*, halaman ini merupakan tampilan yang pertama kali ditampilkan setelah *user* berhasil melakukan *login*. Pada halaman ini terdapat tampilan *Workingon Projects* dan *Recent Updates*.

d. Desain Tampilan Halaman *Projects*



Gambar 3.16 Desain Tampilan Halaman *Projects*

Pada Gambar 3.16 merupakan tampilan halaman *Projects*. Halaman ini menampilkan seluruh proyek yang dimiliki *user*. Terdapat tombol *Create Project* untuk dapat membuat proyek baru.

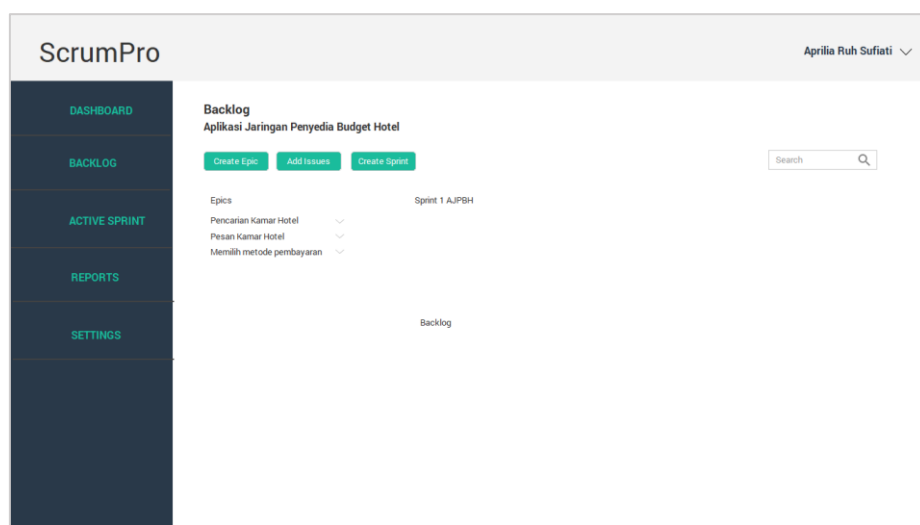
e. Desain Tampilan Halaman *Create Project*

Gambar 3.17 Desain Tampilan Halaman *Create Project*

Pada Gambar 3.17 merupakan tampilan halaman *Create Project*. Pada halaman ini terdapat *form* yang perlu diisi untuk membuat proyek baru. *Form* yang perlu diisi adalah *Name*, *Key*, *Description* dan *Team*.

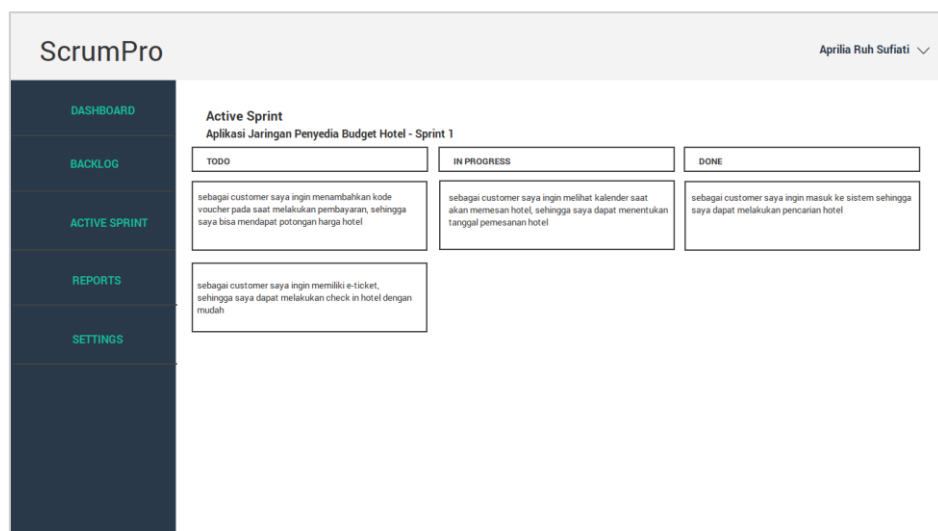
f. Desain Tampilan Halaman *Backlog*

Pada Gambar 3.18 merupakan tampilan halaman *Backlog*. Halaman ini menampilkan daftar pekerjaan yang dimiliki sebuah proyek. Pada halaman ini terdapat tombol “Create Epic” untuk menambahkan *highlight* pekerjaan, tombol “Add Issue” untuk menambahkan pekerjaan berupa *user story* dan tombol “Create Sprint” untuk membuat *Sprint Backlog*.



Gambar 3.18 Desain Tampilan Halaman *Backlog*

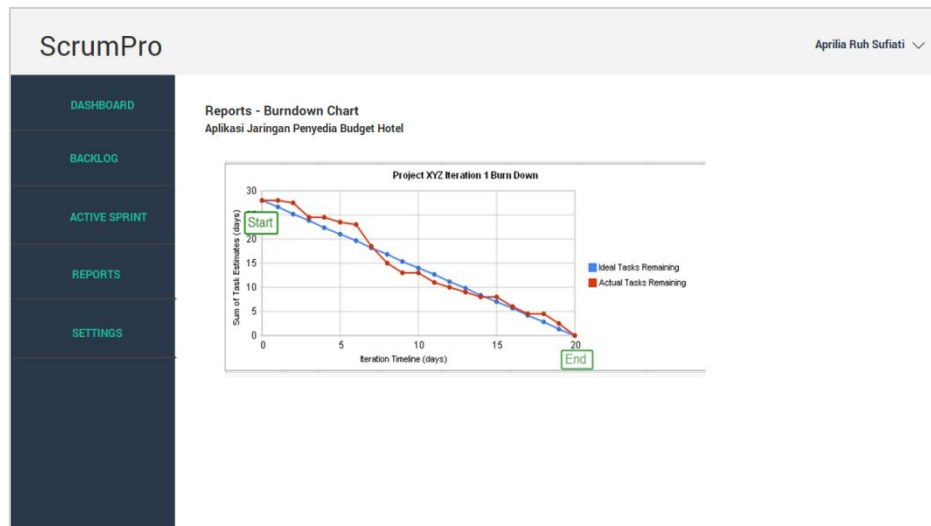
g. Desain Tampilan Halaman *Active Sprint*



Gambar 3.19 Desain Tampilan Halaman *ActiveSprint*

Halaman *Active Sprint* ditunjukkan pada Gambar 3.19, terdapat sebuah Kanban Board untuk memindahkan status issue dari satu kolom ke kolom lainnya.

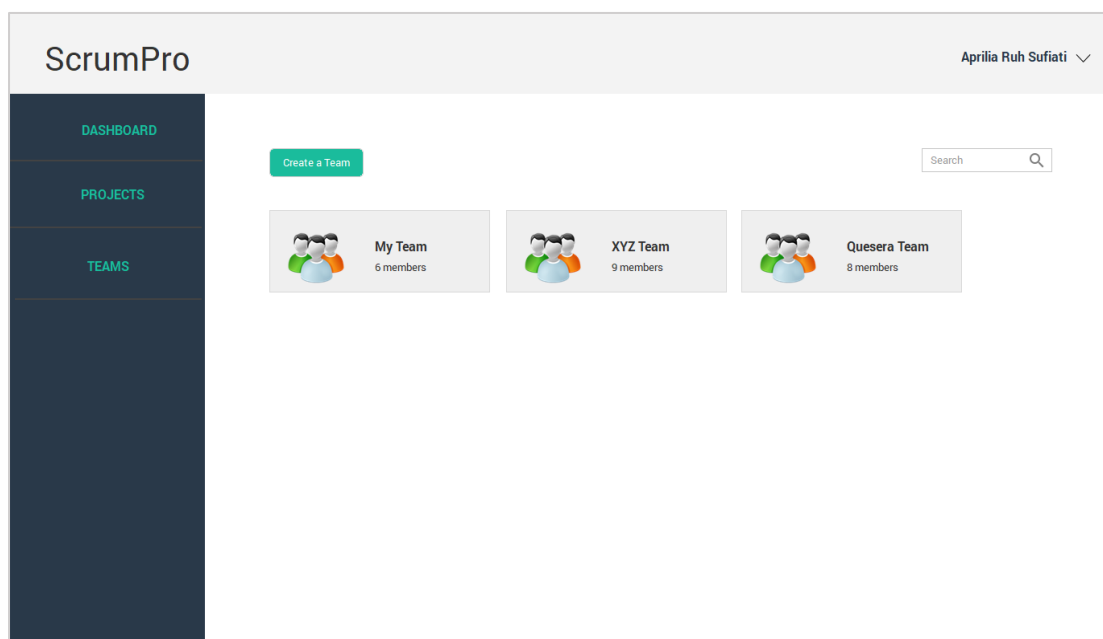
h. Desain Tampilan Halaman Reports



Gambar 3.20 Desain Tampilan Halaman *Reports*

Pada Gambar 3.20 merupakan tampilan halaman Reports. Halaman ini akan menampilkan sebuah *Burndown Chart* realisasi pekerjaan dalam satu *Sprint*. Jumlah grafik akan sama dengan jumlah *Sprint* yang dimiliki suatu proyek.

i. Desain Tampilan Halaman Teams



Gambar 3.21 Desain Tampilan Halaman *Teams*

Pada Gambar 3.21 merupakan tampilan halaman Teams. Halaman ini menampilkan daftar Tim yang telah dibuat Product Owner yang akan dipilih untuk mengerjakan proyek tertentu. Terdapat tombol *Create Team* untuk membuat Tim baru.

j. Desain Tampilan Halaman *Create Team*

The screenshot shows the 'Create a Team' form in the ScrumPro application. The form is located on the 'TEAMS' page, as indicated by the sidebar. The form fields are: Team Name, Team Key, Product Owner, Master Scrum, Developers (a grid for 5 developers), and Description. A Submit button is at the bottom right.

Gambar 3.22 Desain Tampilan Halaman *Create Team*

Halaman ini merupakan tampilan *form* untuk membuat sebuah Tim *Scrum* baru, seperti pada Gambar 3.22. *Form* yang harus diisi berupa *Team Name*, *Team Key*, *Product Owner*, *Master Scrum*, *Developers* dan *Description*.

k. Desain Tampilan Halaman *Create Epic*

The screenshot shows the 'Create Epic' form in the ScrumPro application. The form is a modal dialog that appears over the 'Backlog' page. The form fields are: Epic Name and Summary. There are 'Create' and 'Cancel' buttons at the bottom right.

Gambar 3.23 Desain Tampilan Halaman *Create Epic*

Pada Gambar 3.23 merupakan tampilan halaman *Create Epic*. Halaman ini menampilkan *form* yang perlu diisi untuk membuat *highlight* pekerjaan baru. *Form* yang perlu diisi adalah *Epic Name* dan *Summary*.

1. Desain Tampilan Halaman *Add Issues*

Gambar 3.24 Desain Tampilan Halaman *Add Issues*

Pada Gambar 3.24 merupakan tampilan halaman *Add Issue*. Halaman ini menampilkan *form* yang perlu diisi untuk membuat *Product Backlog Item* baru. *Form* yang perlu diisi adalah *Summary*, *Description*, *Story Point* dan *Sprint*. *Issue* yang dibuat dalam bentuk *user story*, sehingga dapat mengetahui kebutuhan sistem berdasarkan sudut pandang pengguna sistem.

m. Desain Tampilan *Create Sprint*

Gambar 3.25 Desain Tampilan Halaman *Create Sprint*

Pada Gambar 3.25 merupakan tampilan halaman *CreateSprint*. Halaman ini menampilkan *form* yang perlu diisi untuk membuat *Sprint* baru. *Form* yang perlu diisi adalah *Sprint Name*, *Start Date*, *End Date* dan *Sprint Goal*.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas mengenai implementasi dan pengujian, untuk melakukan penerapan terhadap rancangan sistem yang telah dibuat pada bab sebelumnya. Selanjutnya akan dilakukan pengujian sistem untuk memastikan seluruh fungsionalitas sistem berjalan sebagaimana mestinya dan mengetahui seberapa jauh *user* dapat menerima sistem yang telah dibuat.

4.1 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.

4.1.1 Skenario Proyek

Dalam skenario ini terdapat tiga aktor yang berperan, yaitu Product Owner, *Scrum* Master dan Tim Pengembang. Product Owner merupakan pihak yang bertanggung jawab atas keseluruhan proyek sekaligus menjembatani antara stakeholder dengan Tim *Scrum*. *Scrum* Master bertanggung jawab untuk mengawasi dan memfasilitasi Tim *Scrum* agar proyek berjalan sebagaimana mestinya. Sedangkan Tim Pengembang menjadi pelaksana dalam pengembangan *software* dari proyek yang sedang dijalankan.

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

Tabel 4.1 Skenario Sistem

No	Aktor	Tahapan	Skenario
1	Product Owner	Menentukan Proyek	- Product Owner memutuskan untuk mengerjakan proyek

			<p>pengembangan Sistem PMB UII.</p> <ul style="list-style-type: none"> - Product Owner bertemu dengan seluruh <i>stakeholder</i> PMB UII untuk menentukan <i>requirement</i> dalam proyek PMB UII. Pertemuan tersebut akan menghasilkan daftar fitur beserta product <i>Backlog</i> item yang akan dimiliki sistem PMB UII.
2	Product Owner	Membentuk Tim	Berdasarkan <i>requirement</i> yang telah didiskusikan dengan <i>stakeholder</i> , Product Owner membentuk sebuah Tim <i>Scrum</i> yang komposisinya ditentukan sesuai dengan kebutuhan proyek PMB UII. Tim yang dibentuk terdiri dari dirinya sendiri sebagai Product Owner, <i>Scrum</i> Master dan beberapa anggota untuk Tim Pengembang yang akan terlibat dalam proyek PMB UII.
3	<i>Scrum</i> Master	<i>Sprint Planning</i>	Product Owner bertemu dengan Master <i>Scrum</i> dan Tim Pengembang untuk membahas rencana kerja dalam setiap <i>Sprint</i> .
4	<i>Scrum</i> Master	Mengelola <i>Sprint</i>	Product Owner membagi Product <i>Backlog</i> Item ke dalam setiap <i>Sprint</i> yang telah ditentukan dalam <i>Sprint Planning</i> .
5	Tim Pengembang	<i>Daily Meeting</i>	Tim Pengembang didampingi oleh Master <i>Scrum</i> melakukan Daily Meeting maksimal selama 15 menit sebelum melanjutkan pekerjaan pada hari tersebut. Hal ini terus berulang dan berlanjut hingga akhir <i>Sprint</i> .
6	<i>Scrum</i> Master	<i>Review dan Retrospective</i>	Pada akhir <i>Sprint</i> , Tim <i>Scrum</i> akan kembali melakukan pertemuan untuk membahas pelaksanaan proyek dalam satu <i>Sprint</i> terakhir. Dari pertemuan ini akan menghasilkan catatan mengenai inkremen dan evaluasi yang berkaitan dalam pengerjaan proyek PMB UII.
7	Product Owner	Menutup <i>Sprint</i>	<ul style="list-style-type: none"> - Hasil pekerjaan dari sebuah <i>Sprint</i> siap dirilis. - Product Owner menyatakan <i>Sprint</i> selesai dan ditutup.

4.1.2 Implementasi Sistem

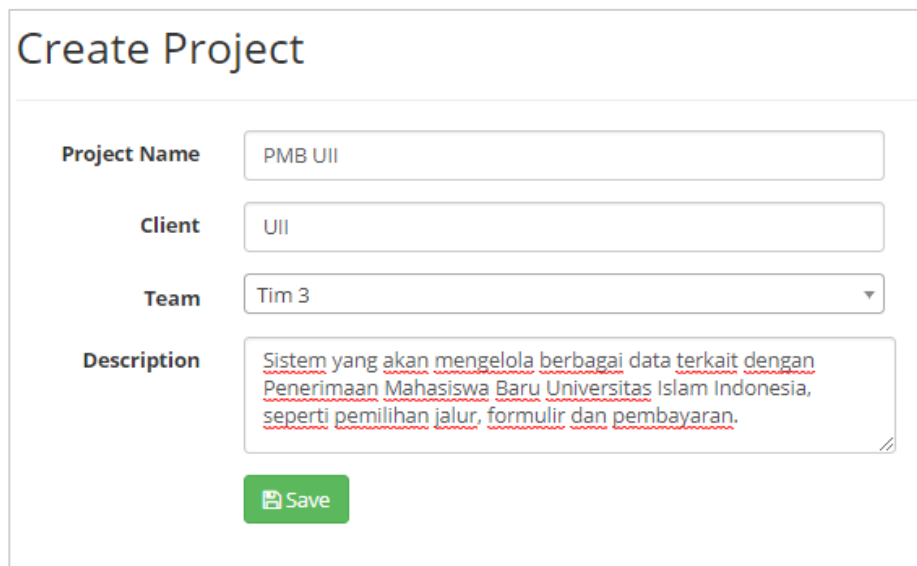
Implementasi akan memvisualisasikan sistem berdasarkan skenario dengan tahapan sebagai berikut:

- a. *Product Owner* menentukan Proyek
- b. *Product Owner* membentuk Tim
- c. *Scrum Master* mencatat *Sprint Planning*
- d. *Scrum Master* mengelola *Sprint*
- e. Tim Pengembang melakukan *Daily Meeting*
- f. *Scrum Master* mencatat *Review* dan *Retrospective*
- g. *Product Owner* menutup *Sprint*

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

- a. *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.



Create Project

Project Name

Client

Team

Description

Gambar 4.1 Formulir Pembuatan Proyek Baru

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 4.1. Data yang harus diisi adalah *Project Name*, *Client*, *Team* dan *Description*.

Gambar 4.2 Formulir Pembuatan Epic

Ketika user menekan tombol *Create Epic* pada halaman *Backlog*, akan muncul sebuah modal berupa form untuk membuat Epic baru seperti pada Gambar 4.2. Data yang harus diisi adalah *Epic Name* dan *Summary* yaitu penjelasan singkat mengenai sebuah Epic.

Gambar 4.3 Formulir Pembuatan Issue

Jika *user* menekan tombol *Create Issue* pada halaman *Backlog*, akan muncul sebuah modal berupa form untuk membuat *Issue* baru seperti pada Gambar 4.3. *Issue* yang dibuat berupa sebuah *user story* dari sebuah *task* yang akan dikerjakan oleh *user* dari *Tim Scrum*.

- b. *Product Owner* membentuk *Tim*

Gambar 4.4 Formulir Pembuatan Tim Baru

Jika *user* menekan tombol *Create Team* pada halaman *Teams*, akan ditampilkan halaman berisikan form untuk membuat *Tim* baru seperti pada Gambar 4.4. Data yang diisikan berupa *Team Name*, *Team Key*, *Product Owner*, *Scrum Master*, *Developer Team* dan *Description*.

- c. *Scrum Master* mencatat *Sprint Planning*

Gambar 4.5 Formulir Pembuatan *Sprint* Baru

Jika user menekan tombol *Create Sprint* pada halaman *Backlog*, akan muncul sebuah modal berupa form untuk membuat *Sprint* baru seperti pada Gambar 4.5. 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*.

d. *ScrumMaster* mengelola *Sprint*

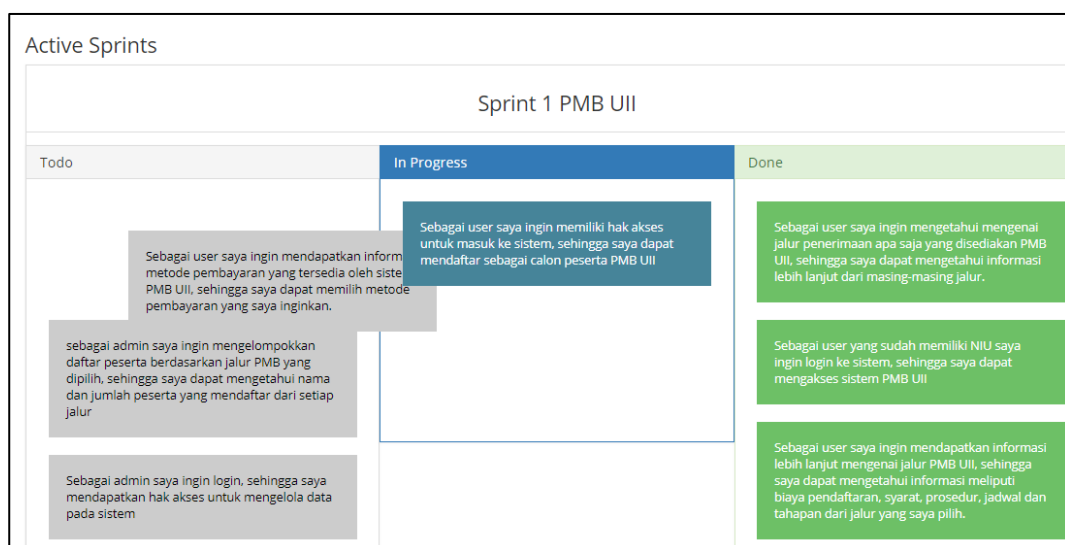


Gambar 4.6 Menambahkan Issue ke *Sprint*

Untuk melakukan pembagian *issue* ke dalam beberapa *Sprint*, gunakan tombol dengan simbol “+” pada daftar nama *Sprint* yang tersedia seperti pada Gambar 4.6.

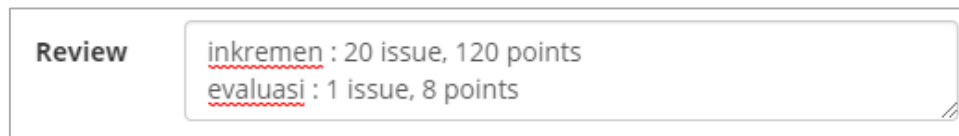
e. Tim Pengembang melakukan *Daily Meeting*

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 4.7.



Gambar 4.7 *Drag and Drop* pada *Sprint* Aktif

- f. *Scrum* Master mencatat *Review* dan *Retrospective*

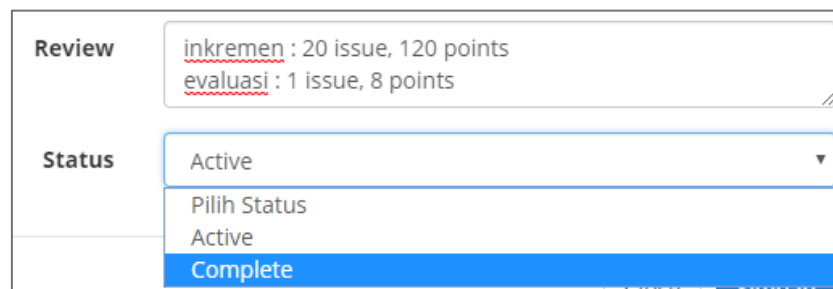


The screenshot shows a form with a label 'Review' on the left. To its right is a text input area containing two lines of text: 'inkremen : 20 issue, 120 points' and 'evaluasi : 1 issue, 8 points'. Both lines are underlined with a red wavy line. There is a small icon in the bottom right corner of the input area.

Gambar 4.8 Memasukkan catatan *Review* dan *Retrospective*

Tampilan formulir *Edit Sprint* untuk mengisi *Review* dengan catatan inkremen dan evaluasi hasil dari pertemuan *Review* dan *Retrospective* ditunjukkan pada Gambar 4.8. Isi dari formulir tersebut dapat di deskripsikan lebih luas, sesuai dengan keinginan dan kebutuhan Tim.

- g. Product Owner menutup *Sprint*



The screenshot shows a form with two labels: 'Review' and 'Status'. The 'Review' label is next to a text input area containing 'inkremen : 20 issue, 120 points' and 'evaluasi : 1 issue, 8 points'. The 'Status' label is next to a dropdown menu. The dropdown menu is open, showing a list of options: 'Active', 'Pilih Status', 'Active', and 'Complete'. The 'Complete' option is highlighted in blue. There are also 'Close' and 'Simpan' buttons at the bottom right of the form.

Gambar 4.9 Menutup *Sprint*

Jika review sudah diisi maka akan muncul pilihan untuk mengubah status *Sprint* seperti pada Gambar 4.9.

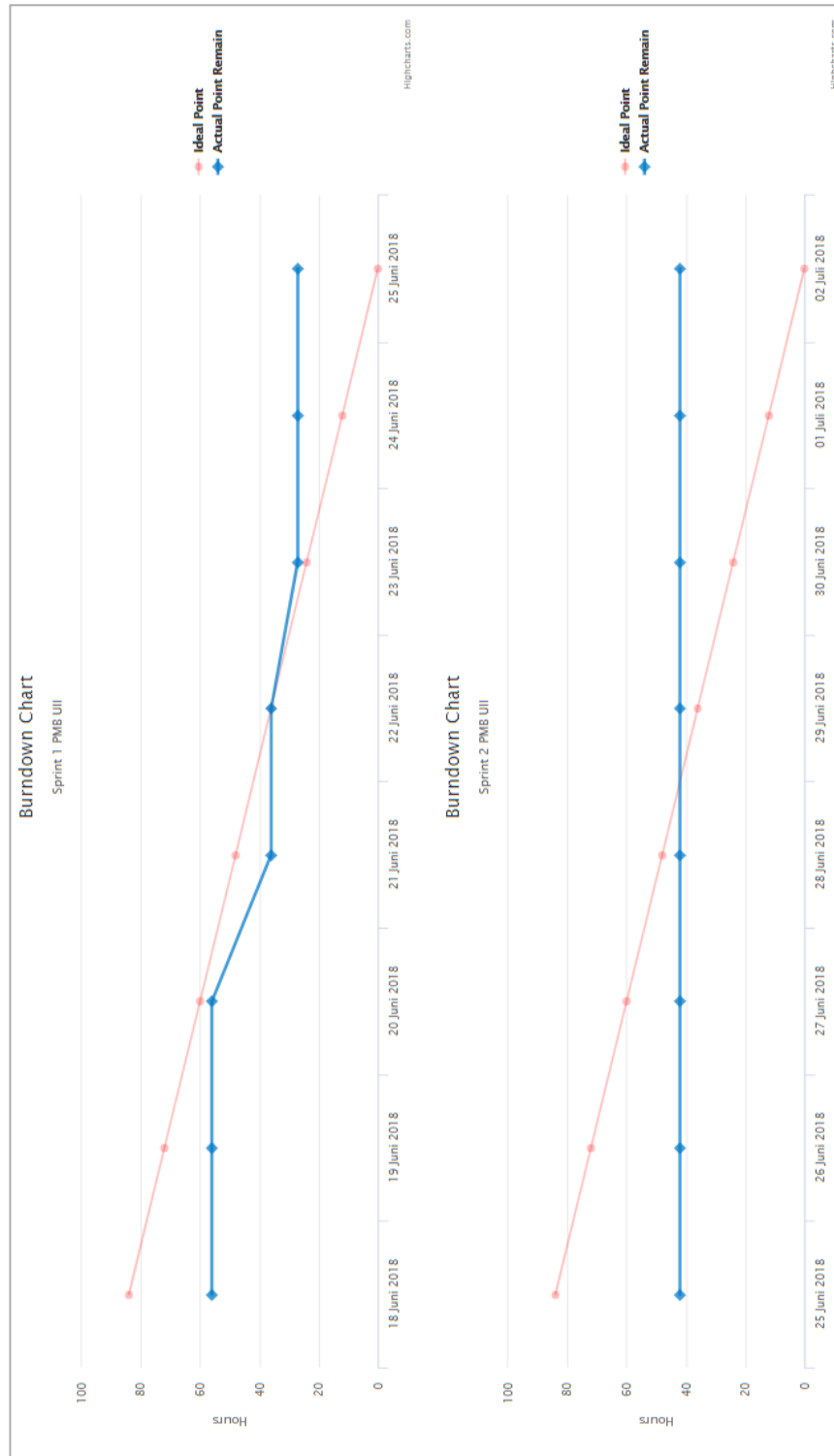
4.1.3 Fitur Pendukung

Selain fitur-fitur yang ditampilkan pada tahap implementasi, terdapat beberapa fitur pendukung yang tidak masuk dalam skenario proyek diatas. Dalam penelitian ini penulis membuat fitur Grafik *Burndown* yang dapat digunakan untuk melihat perkembangan pada *Sprint* yang dimiliki suatu proyek. Selain itu terdapat fitur Recent Activity untuk menampilkan beberapa aktivitas terakhir dalam perkembangan proyek khususnya mengenai perubahan status issue pada halaman *Active Sprint*.

- a. Fitur Grafik *Burndown*

Issue yang sudah berstatus *Done* akan dihitung dan disajikan dalam sebuah grafik *Burndown* seperti pada Gambar 4.10. Jumlah grafik pada halaman *Reports* akan

menyesuaikan dengan jumlah *Sprint* yang sudah dibuat dalam suatu proyek. Setiap grafik memiliki 2 sumbu. Sumbu X berisikan tanggal terjadinya perpindahan status *issue* menjadi *Done* dan sumbu Y berisikan jumlah *story point* yang dalam sebuah *Sprint*.

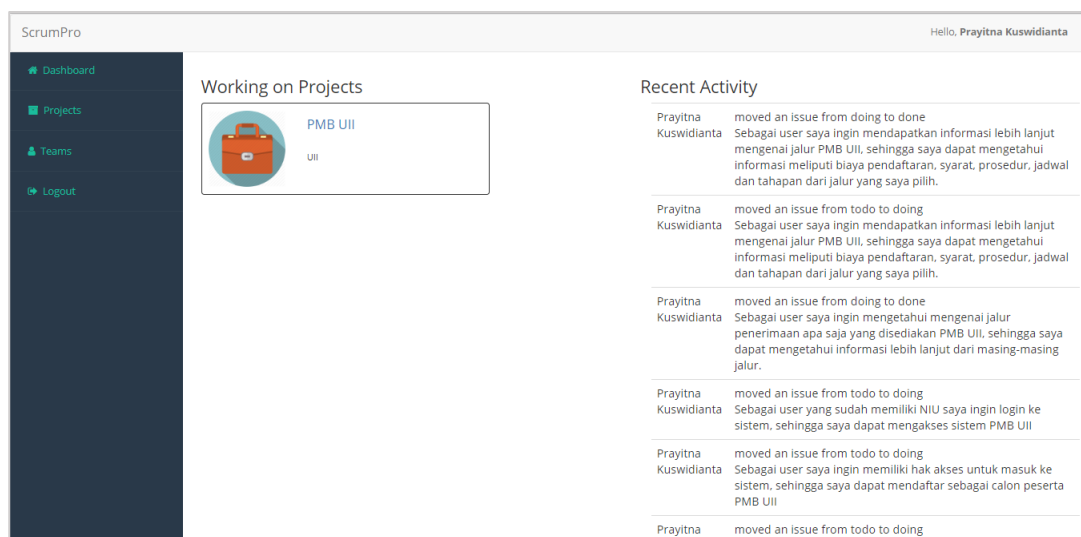


Gambar 4.10 Grafik Burndown

Setiap *Sprint* memiliki tanggal yang berbeda, dimulai dari tanggal *Sprint* dimulai sampai dengan tanggal *Sprint* berakhir, seperti pada tanggal yang telah dimasukkan pada saat pembuatan *Sprint* di Sistem. Idealnya *story point* yang dimiliki *Sprint* akan berkurang setiap harinya yang digambarkan oleh garis merah, sehingga garis tersebut selalu mengalami penurunan dari hari ke hari. Sedangkan garis biru menggambarkan progres *story point* dalam keseharian pelaksanaan *Sprint*, jika garis biru masih berada sama dengan atau lebih rendah dari garis merah, maka pelaksanaan *Sprint* dianggap masih memiliki progres yang baik.

b. Fitur *Recent Activity*

Gambar 4.11 merupakan tampilan *Recent Activity* pada halaman *dashboard*. *Recent Activity* hanya menampilkan aktivitas berdasarkan proyek dan tim yang melibatkan pengguna yang sedang *login* pada sistem. Sehingga pengguna yang tidak terlibat tidak dapat melihat proyek milik pengguna lain diluar proyek yang melibatkannya.



Gambar 4.11 *Recent Activity*

4.2 Pengujian Sistem

4.2.1 Objek Pengujian

Untuk memastikan sistem berjalan sebagaimana mestinya maka dilakukan pengujian terhadap beberapa objek seperti fungsi, alur dan *error handling* yang dimiliki sistem.

4.2.2 Subjek Pengujian

Pada pengujian ini 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. Mahasiswa Informatika dianggap sudah mampu menggunakan sistem terkomputerisasi terutama berbasis web, selain itu BSI UII sudah menerapkan kerangka kerja *Scrum* sehingga dianggap sudah akrab dengan implementasi *Scrum* dalam pekerjaan sehari-hari.

4.2.3 Prosedur Pengujian

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.

4.2.4 Hasil Pengujian

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

Tabel 4.2 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 <i>error</i> .				3	2	88%
4	Sistem sudah sesuai dengan kebutuhan Manajemen Kerja Tim <i>Scrum</i> berdasarkan teori kerangka kerja <i>Scrum</i> .				3	2	88%
5	Sistem mampu membantu kinerja Tim <i>Scrum</i> .			1	3	1	80%
6	Sistem mampu mendukung pengelolaan <i>database</i> proyek pengembangan <i>software</i> dengan menggunakan kerangka kerja <i>Scrum</i> .			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% dari yang diharapkan. Maka hasil pengujian sistem terhadap pengguna berada pada kategori Sangat setuju.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

- a. Implementasi sistem yang dibuat sudah berjalan dengan baik tanpa adanya *error*, serta memiliki alur dan fungsionalitas yang mudah dipahami saat di implementasikan.
- b. Sistem dianggap sesuai dengan kebutuhan manajemen kerja Tim *Scrum* di BSI UII, berdasarkan kerangka kerja *Scrum* dan membantu Tim *Scrum* dalam meningkatkan kinerja Tim. Secara keseluruhan, sistem memiliki penilaian Sangat Setuju atau Sangat Baik.

5.2 Saran

Untuk pengembangan sistem lebih lanjut, diharapkan Sistem Informasi Manajemen Kerja Tim *Scrum* kedepannya dapat dikembangkan sebagai berikut:

- a. Menambahkan fitur yang lebih fleksibel, seperti pengguna dapat mengubah *sprint backlog* sehingga lebih sesuai dengan prinsip manajemen proyek yang menggunakan kerangka kerja *Scrum*.
- a. Menambahkan fitur yang dapat mendukung berlangsungnya sistem manajemen kerja Tim *Scrum*, seperti notifikasi melalui *e-mail* untuk mempermudah perputaran informasi dan komunikasi, *import* dan *export* data menggunakan Microsoft Excel untuk keperluan dokumentasi proyek.

DAFTAR PUSTAKA

- Dharma, S. (2009). *Manajemen Kinerja Falsafah, Teori dan Penerapannya*. Yogyakarta: Pustaka Pelajar.
- Martin, R. C. (2003). *Agile Software Development: Principles, Patterns and Practices*. New Jersey: Prentice Hall.
- Partogi, Joshua. (2015). *Manajemen Modern dengan Scrum*. Yogyakarta: Andi.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*.
- Sutherland, Jeff. (2017). *Scrum Meningkatkan Produktifitas Dua Kali Lipat Dalam Waktu Setengahnya Saja*. Yogyakarta: Bentang Pustaka.
- Sugiyono. 2010. *Metode Penelitian Pendidikan Pendekatan Kuantitatif, kualitatif, dan R&D*. Bandung: Alfabeta.
- Versionone(2017). *Versionone 11th Annual State of Agile Report*. Atlanta: Versionone.com.
- Whitten, J.L., & Bentley, L. D. (2007). *System Analysis and Design Method - Seventh Edition*. New York: McGraw-Hill

LAMPIRAN