

Analisis dan Desain Pengembangan Program Aplikasi DMS dengan Metodologi Scrum Framework

O.Ginjar*, Asep Sholahudin, Erick Paulus
Program Studi S-1 Teknik Informatika, Fakultas MIPA
Universitas Padjadjaran
Jl. Raya Bandung-Sumedang Km 21 Jatinangor
ginanjaroo13@gmail.com

Abstrak— *System Development Life Cycle (SDLC)* dalam proses pengembangan program aplikasi penting untuk diperhatikan. Metode SDLC saat ini harus bisa menyelesaikan permasalahan tim pengembang dalam mengembangkan program aplikasi dengan kebutuhan pengguna yang dinamis dan menuntut pengembangan program aplikasi yang cepat. Pada penelitian ini pengembangan aplikasi menggunakan *Agile Software Development* atau *Agile Methodology*. *Agile Methodology* menerapkan *incremental* dan *iterative development* sehingga pengembang bisa terus melakukan evaluasi bersama pengguna terhadap produk program aplikasi yang sedang dikembangkan. Proses pengembangan menggunakan salah satu *Agile Methodology* yaitu *Scrum Framework*. Pada penelitian ini, *Scrum Framework* diimplementasikan dalam pengembangan program aplikasi web untuk mengelola data kegiatan organisasi yang disebut dengan *Integrated Database Management System (IDMS)*. Hasil dari penerapan *Scrum Framework* pada pengembangan aplikasi IDMS berjalan dengan baik dan dapat mengatasi perubahan kebutuhan yang terjadi di tengah proses pengembangan. Pelaksanaan pengembangan aplikasi dibagi dalam 3 kali sprint dengan nilai *focus factor* masing-masing sprint adalah 27%, 36% dan 47%. Dengan adanya peningkatan pada setiap sprint hal ini menandakan tim *developer* mengalami peningkatan kemampuan dan fokus dalam pengembangan aplikasi.

Kata kunci— *System Development Life Cycle (SDLC)*; *Agile Methodology*; *Scrum Framework*; *Focus Factor*.

I. PENDAHULUAN

System Development Life Cycle (SDLC) atau Siklus Hidup Pengembangan Sistem adalah pendekatan bertahap untuk melakukan analisa dan membangun rancangan sistem dengan menggunakan siklus yang spesifik terhadap kegiatan pengguna[1]. *System Development Life Cycle (SDLC)* dalam proses pengembangan program aplikasi penting untuk diperhatikan, karena hal ini sebagai salah satu penunjang keberhasilan program aplikasi yang akan dikembangkan. Metode SDLC saat ini harus bisa menyelesaikan permasalahan tim pengembang dalam mengembangkan program aplikasi di mana kebutuhan pengguna yang selalu berubah-ubah atau dinamis dan menuntut pengembangan produk program aplikasi yang cepat. Saat ini ada dua metode yang banyak digunakan oleh kebanyakan pengembang sistem, yaitu *Traditional Software Development* dan *Agile Software Development*[2]. Pada penelitian ini peneliti memilih untuk mengembangkan

aplikasi menggunakan *Agile Software Development* atau *Agile Methodology*. *Agile Methodology* menerapkan *incremental* dan *iterative development* sehingga pengembang bisa terus melakukan evaluasi bersama pengguna atau pemilik produk terhadap produk program aplikasi yang sedang dikembangkan. *Agile Methodology* memiliki beberapa turunan di antaranya adalah *Extreme Programming* dan *Framework Scrum*[2]. *Scrum* dinilai dapat menghasilkan kualitas program aplikasi yang baik atau sesuai dengan keinginan pengguna karena mendapatkan *feedback* secara berkelanjutan, dengan kondisi tim pengembang yang terbatas[3]. *Scrum* bisa menghitung kecepatan kerja sebuah tim dengan penilaian Rumus "*Focus Factor*". Perhitungan ini sebuah tim dapat diukur seberapa cepat tim mengerjakan pekerjaannya dan bisa menjadi estimasi dalam mengerjakan di setiap sprint berikutnya. Pada implementasinya peneliti mengembangkan sistem aplikasi yang dimiliki sebuah organisasi masyarakat yang mempunyai nama *Integrated Database Management System (IDMS)*. *Integrated Database Management System (IDMS)* adalah sebuah aplikasi untuk mengelola data proyek dan kegiatan organisasi.

Berdasarkan latar belakang, peneliti melakukan implementasi *Agile Methodology* dengan *Framework Scrum* pada pengembangan program aplikasi IDMS.

II. METODE

A. Scrum Framework

Scrum merupakan bagian dari *Methodology Agile Software Development*. *Scrum* diperkenalkan oleh Jeff Sutherland tahun awal tahun 1990-an. Pengembangan berikutnya dilakukan oleh Schwaber dan Beedle. *Scrum* adalah sebuah *framework* 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[4].

B. Kegiatan Scrum

Kegiatan dalam *scrum* yang biasanya disebut sprint. *Sprint* adalah sebuah batasan waktu untuk menyelesaikan sebuah potongan produk. Tahapan selama satu sprint terdiri dari *Sprint Planning*, *Daily Scrum*, *Development*, *Sprint Review*, *Sprint Retrospective*[4].

1) *Sprint Planning*

Sprint planning adalah kegiatan di mana tim *scrum* merencanakan apa yang akan dikerjakan dalam satu *sprint* ke depan. Pada tahapan *sprint planning* setiap *developer* harus paham dengan pencapaian apa yang harus selesai selama *sprint* berlangsung.

2) *Daily Scrum*

Daily scrum adalah kegiatan dengan batasan waktu 10 menit setiap harinya agar tim bisa melakukan sinkronisasi pekerjaan dan merencanakan apa yang akan dikerjakan di hari selanjutnya.

3) *Development*

Pada tahapan ini Tim mulai melakukan pekerjaan untuk menghasilkan sebuah produk sesuai dengan hasil *sprint planning*.

4) *Sprint Review*

Sprint review adalah kegiatan yang dilakukan di setiap akhir *sprint* untuk meninjau potongan produk yang sudah dikerjakan dan belum selesai dikerjakan.

5) *Sprint Retrospective*

Sprint retrospective adalah sebuah kesempatan bagi tim *Scrum* untuk meninjau dirinya sendiri dan membuat perencanaan mengenai peningkatan yang akan dilakukan di *Sprint* berikutnya.

C. *Tim Scrum*

Tim dalam *scrum* terdiri dari 3 bagian, yaitu *product owner*, tim pengembang, dan *scrum master*[4]. Setiap tim harus bisa mengatur dirinya sendiri sehingga tim bisa menyelesaikan kerjanya sesuai dengan kebutuhan masing-masing.

1) *Product Owner*

Product owner adalah pemimpin utama sebuah produk. Dia adalah otoritas tunggal yang bertanggung jawab untuk menentukan fitur dan fungsionalitas mana yang akan dibangun dan perintah untuk membangunnya[6].

2) *Tim Pengembang*

Tim Pengembang terdiri dari para profesional yang bekerja untuk menghasilkan produk[4].

Dalam Panduan *scrum* yang ditulis oleh Schwaber dan Sutherland pada tahun 2013, Tim Pengembang memiliki karakteristik sebagai berikut :

- Mengatur dirinya sendiri untuk menyelesaikan potongan *product backlog*,
- Tim pengembang memiliki keahlian yang dibutuhkan selama pengembangan produk,
- Tidak ada jabatan tertentu dalam tim pengembang,
- Tim pengembang tidak mengenal adanya *sub-tim* yang dikhususkan untuk bidang tertentu seperti pengajuan atau analisis bisnis,
- Anggota tim boleh memiliki keahlian tertentu dan fokus dalam satu area tertentu, namun akuntabilitas dari hasil pekerjaan secara keseluruhan adalah milik tim pengembang.

3) *Scrum Master*

Scrum master adalah seseorang yang bertanggung jawab untuk memastikan *scrum* telah dipahami dan dilaksanakan. *Scrum master* adalah seorang pemimpin yang melayani Tim *scrum*[4].

D. *Focus Factor*

Focus factor adalah perhitungan *velocity* dibagi *work capacity*[5]. *Velocity* adalah jumlah perkiraan pekerjaan yang selesai pada *sprint*. *Velocity* disebut juga dengan jumlah *story point* yang berhasil dikerjakan dalam *sprint*. Perhitungan *focus factor* dilakukan setelah *sprint* selesai, dengan menggunakan rumus ini sebuah tim bisa mengetahui estimasi waktu kerja untuk bisa menyelesaikan *sprint* yang akan dimulai. Sedangkan untuk tim yang sudah berjalan, *focus factor* dapat menjadi acuan pada pengembangan perangkat lunak selanjutnya. Sehingga tim dapat mengetahui perkiraan waktu pengembangan perangkat lunak akan selesai.

E. *Burndown Chart*

Burndown chart merupakan grafik sederhana yang digunakan untuk memantau dan menganalisis kinerja tim dan mengetahui perkembangan dari proses *development* yang sedang dijalankan. *Burndown chart* adalah cara terbaik untuk menggambarkan hubungan antara jumlah pekerjaan yang tersisa pada setiap titik waktu dan kemajuan Tim pengembangan[7]. *Burndown chart* terdiri dari garis *ideal task remaining* dan *actual task day remaining* yang merupakan perubahan jumlah *story point* pada *work day*. *Ideal task remaining* adalah garis pengingat perkiraan *story point* yang harus selesai dikerjakan pada titik hari kerja. *Actual task day remaining* adalah garis pengingat *story point* yang tersisa. *Work day* adalah hari kerja tim *developer* dalam satu *sprint*.

F. *Pengumpulan Data dan Analisis Proses Bisnis*

Pengumpulan data didapatkan dari dokumen *Functional Requirement System* (FRS) yang disusun oleh *product owner* berdasarkan kebutuhan organisasi. Berdasarkan dokumen FRS Aplikasi IDMS terbagi dalam beberapa modul yaitu modul Manajemen Administrasi, modul *Beneficiaries*, modul Manajemen *Project*, modul Manajemen *Go No Go Evaluation & Tracker* (GET) dan modul Manajemen *Reference*.

G. *Membuat 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[4]. *Product owner* sebagai penanggung jawab terhadap *product backlog*. Daftar terurut dalam *product backlog* disebut dengan *product backlog Item*. *Product backlog item* terdiri dari :

- Id, adalah nomor ataupun identitas unik dari masing-masing *product backlog item*.
- *Product backlog item*, adalah deskripsi singkat dari tugas.
- *Value*, adalah nilai yang menunjukkan tingkat kepentingan dari sebuah *product backlog*. Semakin tinggi nilai *Value* berarti menunjukan kepentingan sebuah *product backlog item* untuk dikerjakan.

- Estimasi atau *story point*, Estimasi atau *story point* adalah perkiraan waktu pengerjaan untuk setiap *Product backlog item* yang biasanya dihitung dalam satuan hari atau waktu.

Product backlog disusun berdasarkan dokumen FRS. *Product backlog* pada pengembangan program aplikasi IDMS terdiri dari 11 *product backlog item* dan 66 *story point*. Nilai *value* tertinggi 20 dan terendah 10. Jumlah *product backlog item* dan *story point* bisa berubah tergantung dengan saat *sprint planning*. *Product backlog item* dengan nilai *story point* tinggi akan dilakukan *refinement* pada saat *sprint planning*. *Refinement product backlog item* adalah menambahkan detail, memperbaiki urutan prioritas dan estimasi di setiap *Product backlog item*. *Product backlog* dapat dilihat pada Tabel 1.

TABEL 1. PRODUCT BACKLOG

Id	Product Backlog Item	Value	Story Point
1	Migrasi IDMS v1 ke v2	20	22
2	Fitur <i>upload</i> dan <i>download Project</i>	19	6
3	Fitur <i>import</i> dan <i>export beneficiaries</i>	19	2
4	Fitur <i>export Activities</i>	19	2
5	<i>Mock Up</i> Modul GET	18	4
6	<i>Opportunity Management</i>	18	13
7	<i>Go no Go Assessment</i>	16	9
8	<i>Go no Go proposal</i>	17	2
9	<i>Approval Show and Judgement</i>	15	2
10	<i>Notification</i>	10	2
11	<i>Question Category dan Question</i>	17	2

H. Merancang Sprint

Pada pengembangan aplikasi IDMS ini peneliti membagi *sprint* ke dalam 3 pengulangan *sprint* berdasarkan nilai pada *value*. Pada *sprint* 1 peneliti mengambil *product backlog item* dengan nilai *value* 20. *Sprint* 1 adalah mengganti versi *framework* laravel 4.0 ke laravel 4.2. *Sprint* 2 peneliti mengambil *priority* dengan nilai 19 – 17 di mana pada tahap *sprint* 2 *developer* akan menambah fitur dari modul yang sudah ada dan mempersiapkan untuk pengerjaan modul baru. *Sprint* 3 peneliti mengambil *product backlog item* dengan nilai *value* 18 sampai dengan 10. *Sprint* 3 di sini *developer* mulai mengembangkan modul baru yaitu modul GET.

I. Desain Sistem

Tahapan Selanjutnya dalam merancang sistem aplikasi adalah desain sistem. Desain sistem yang digunakan peneliti adalah *Unified Modeling Language* (UML). Diagram yang akan digunakan peneliti adalah *use case diagram*, *activity diagram* dan *class diagram*.

III. HASIL DAN DISKUSI

Pelaksanaan *sprint* dimulai pada tanggal 11 Oktober 2016. Pengembangan program aplikasi IDMS dilaksanakan dalam tiga kali *sprint*. *Sprint* dibagi berdasarkan nilai *value* yang sudah ditetapkan.

A. Pelaksanaan Sprint 1

Sprint 1 dilaksanakan mulai tanggal 11 Oktober 2016. *Sprint* 1 terdiri dari 4 *developer* yang akan mengerjakan pengembangan produk. *Sprint backlog* pada *sprint* 1 mengalami *refinement* dari perencanaan awal. *Product backlog* yang dikerjakan memiliki nilai *value* 20. *Product backlog item* yang sudah dilakukan *refinement* dapat dilihat di Table 2.

Estimasi kerja di awal *sprint* memakai nilai *focus factor default* 0,7, dikarenakan belum ada acuan sebelumnya. Dengan mengetahui nilai *story point* dan nilai *focus factor* pertama maka nilai *work capacity ideal* adalah 31,4. Untuk mengetahui estimasi waktu pengerjaan maka nilai *work capacity ideal* dibagi dengan tim *developer* yang berjumlah 4 maka di dapatkan nilai 7,85. Estimasi waktu pelaksanaan *sprint* 1 dibulatkan dari 7,85 menjadi 8 hari. Catatan hasil *daily scrum* disajikan pada Tabel 2.

TABEL 2. PRODUCT BACKLOG SPRINT 1

Id	Product Backlog Item	Story point	Selesai
1	Migrasi <i>Projects</i>	2	14-10-2016
2	Migrasi <i>Beneficiaries</i>	2	20-10-2016
3	Migrasi <i>Themes</i> dan <i>Subthemes</i>	4	17-10-2016
4	Migrasi <i>Activities</i>	4	7-11-2016
5	Migrasi <i>Users</i>	2	4-11-2016
6	Migrasi <i>User Profile</i>	2	4-11-2016
7	Migrasi <i>Total Reach</i>	2	3-11-2016
8	Migrasi <i>Project Coverage</i>	2	2-11-2016
9	Migrasi <i>e-Library</i>	2	8-11-2016

Proses *development* pada *sprint* 1 mengalami perpanjangan waktu dari estimasi awal 8 hari kerja menjadi 20 hari kerja. Jumlah hari kerja tim sebenarnya adalah 80 hari kerja. Perpanjangan waktu *sprint* pada pelaksanaan *sprint* 1 dikarenakan *story point* tidak habis atau selesai sesuai dengan estimasi waktu perencanaan. Karena *sprint* 1 merupakan awal dari tim melaksanakan *sprint* jadi waktu pelaksanaan *sprint* ditambah sampai *story point* habis untuk mengetahui nilai *focus factor* tim yang akan digunakan untuk menghitung estimasi waktu *sprint* berikutnya. Nilai *focus factor* dengan nilai *velocity* 22 *story point* dan *work capacity* 80 hari, maka nilai *focus factor* pada *sprint* 1 adalah 0,27 atau 27%.

B. Pelaksanaan Sprint 2

Sprint 2 dilaksanakan mulai tanggal 10 November 2016. *Sprint* 2 terdiri dari 4 *developer* yang akan mengerjakan pengembangan produk. *Sprint* 2 akan mengerjakan *sprint backlog* dengan nilai *value* 19. Hasil *refinement* jumlah *product backlog Item* menjadi 10 dan *story point* menjadi 20. *Sprint backlog* pada *sprint* 2 dijelaskan pada Tabel 3. Pada *sprint* 2 target pencapaian *sprint* secara keseluruhan adalah menambahkan fitur *download* dan *upload* data di beberapa modul dan membuat *mock-up* untuk modul baru GET. Modul yang ditambahkan fitur *download* dan *upload* data yaitu modul *Beneficiaries*, *Project* dan *Activities*.

Estimasi kerja pada *sprint 2* memakai nilai *focus factor* yang didapat di *sprint 1* yaitu 0,27. Dengan mengetahui nilai *story point* dan nilai *focus factor* maka nilai *work capacity ideal* menggunakan rumus 2, hasilnya adalah 74,07 dibulatkan menjadi 74 hari. Untuk mengetahui estimasi waktu pengerjaan maka nilai *work capacity ideal* dibagi dengan tim *developer* yang berjumlah 4 maka di dapatkan nilai 18.5. Estimasi waktu pelaksanaan *sprint 2* dibulatkan dari 18.5 menjadi 19 hari. Catatan hasil *daily scrum sprint 2* disajikan pada Tabel 3.

TABEL 3. CATATAN DAILY SCRUM SPRINT 2

Id	Product Backlog Item	Story Point	Selesai
10	Fitur upload document project	2	29-11-2016
11	Fitur download data Project	2	14-11-2016
12	Fitur download data beneficiaries	2	14-11-2016
13	Mock Up Evaluation GET	2	24-11-2016
14	fitur import beneficiaries activities	2	19-11-2016
15	Download template import beneficiaries	2	18-11-2016
16	fitur import beneficiaries	2	26-11-2016
17	Mock Up Opportunity Proposal	2	29-11-2016
18	Import Beneficiaries modul Activity	2	23-11-2016
19	Fitur Export Activities	2	23-11-2016

Proses *development* pada *sprint 2* lebih cepat dari estimasi awal 19 hari kerja menjadi selesai dengan 13 hari kerja. Jumlah hari kerja tim sebenarnya adalah 56 hari kerja. Sehingga nilai *focus factor* dengan menggunakan rumus 1 dengan nilai *velocity* 20 *story point* dan *work capacity* 56 hari, maka nilai *focus factor* pada *sprint 2* adalah 0,36 atau 37%.

C. Pelaksanaan Sprint 3

Sprint 3 dilaksanakan mulai tanggal 30 November 2016. *Sprint 3* terdiri dari 2 *developer* yang akan mengerjakan pengembangan produk. *Sprint 3* akan mengerjakan *Sprint Backlog* dengan nilai *value* 18 sampai dengan 10.

Sprint backlog pada *sprint 3* dilakukan *refinement* untuk mendeskripsikan *Product Backlog Item* lebih detail dan jelas. Hasil *refinement* jumlah *Product Backlog Item* mejadi 12 dan *story point* menjadi 29. *Sprint backlog* pada *sprint 3* dijelaskan pada Tabel 4.

Pada *sprint 3* target pencapaian *sprint* secara keseluruhan adalah menambahkan modul *Go No Go Evaluation & Tracker*. Pencapaian utama yang harus selesai dalam *sprint 3* adalah proses pendataan *opportunity* dijadikan sebuah proposal untuk dilakukan proses *assessment* oleh *evaluator* dan persetujuan oleh *approver*. Di pertengahan pelaksanaan *sprint* tim mendapatkan permintaan dari *product owner* untuk menambahkan 2 modul tambahan yaitu modul *Currencies* dan modul *SC Office*. Catatan hasil *daily scrum* disajikan pada Tabel 4.

TABEL 4. CATATAN DAILY SCRUM SPRINT 3

Id	Product Backlog Item	Story Point	Selesai
20	Opportunity CRUD	4	7-12-2016
21	Go noGo Create Assessment	4	23-12-2016
22	Go noGo Assessment	3	27-12-2016
23	Go noGo proposal	2	30-12-2016
24	Go noGo Assessment Export	2	21-01-2016
25	Approval Index dan Assessment Result	2	27-12-2016
26	Export Opportunity	2	18-01-2017
27	Import Opportunity	3	18-01-2017
28	Notification	2	17-01-2017
29	Recommendation Type	1	16-01-2017
30	Question Category CRUD	2	10-12-2016
31	Question CRUD	2	10-12-2016
32	Currencies CRUD	2	13-01-2017
33	SC Office CRUD	2	13-01-2017

Proses *development* pada *sprint 3* lebih cepat dari estimasi awal 48 hari kerja menjadi 35 hari kerja. Jumlah hari kerja tim sebenarnya adalah 70 hari kerja. Masing-masing modul tambahan ini memiliki nilai *story point* 2. Jadi total *product backlog* pada *sprint 3* adalah 14 dengan *story point* 33. Nilai *focus factor* pada *sprint 3* adalah 0,47 atau 47%.

Product backlog bisa ditambahkan di pertengahan *sprint* jika *product owner* menemukan kebutuhan pengguna baru yang tidak direncanakan sebelumnya. Hal ini dilakukan jika nilai *value product backlog item* baru bernilai besar. Jika *product backlog item* memiliki nilai *value* kecil dan tidak ada keterkaitan dengan capaian *sprint*, maka *product backlog item* bisa di kerjakan di *sprint* berikutnya.

D. Demo dan Review Produk

Demo dan *review* produk dilaksanakan pada tanggal 10 April 2017. *Demo* produk dilaksanakan dengan tutorial langsung dengan pengguna aplikasi. Setelah *demo* dan tutorial dilaksanakan *user* memberikan *review* terhadap aplikasi yang sudah dilaksanakan.

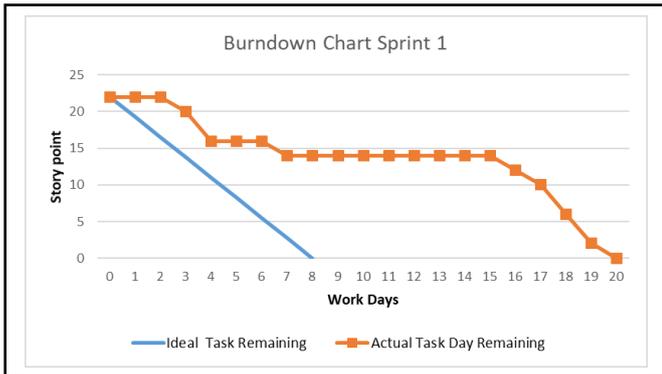
Hasil dari *review* dan *feedback* dari *user* dikerjakan pada *sprint* selanjutnya sebelum *delivery product* atau merilis program aplikasi untuk siap digunakan oleh *user*.

E. Analisis Scrum

Pengembangan program aplikasi IDMS dilaksanakan dalam 3 *sprint*. *Sprint 1* dilaksanakan pada tanggal 11 Oktober 2016 sampai dengan 8 November 2016. *Sprint 2* dilaksanakan pada tanggal 10 November 2016 sampai dengan 29 November 2016. *Sprint 3* dilaksanakan pada tanggal 30 November 2016 sampai dengan 20 Januari 2017.

1) Analisis Sprint 1

Sprint 1 dengan jumlah *story point* 22 dan waktu pelaksanaan *sprint* adalah 20 hari. Gambar 1 adalah grafik *burndown chart sprint 1*.

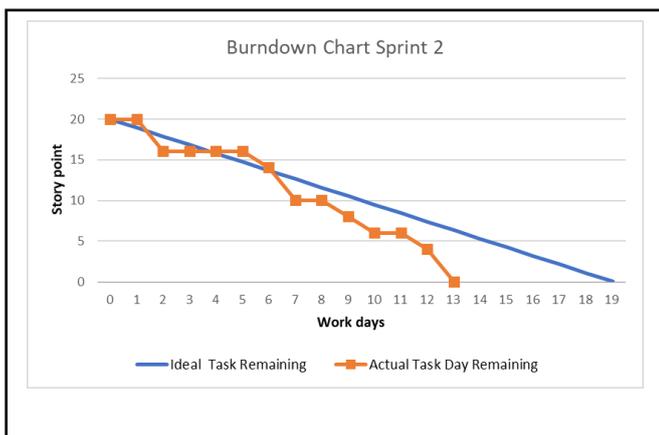


Gambar 1. *Burndown Chart Sprint 1*

Pada awal pelaksanaan *sprint 1* penurunan garis *actual task day remaining* menjauh dan berada di atas garis *ideal task remaining*. Hal ini menunjukkan bahwa pelaksanaan tidak sesuai dengan estimasi waktu yang sudah direncanakan. Pada Gambar 1 *work days* ke 8 sampai dengan ke 15 garis *actual task day remaining* tidak mengalami penurunan. Kondisi seperti yang terjadi pada *work days* ke 8 sampai ke 15 disebabkan karena tim masih mempelajari *framework* yang dipakai dan kondisi beberapa anggota tim yang masih belum fokus dengan pekerjaannya.

2) Analisis Sprint 2

Sprint 2 dengan jumlah *story point* 20 dan waktu pelaksanaan *sprint* adalah 19 hari. Gambar 2 adalah grafik *burndown chart sprint 2*.

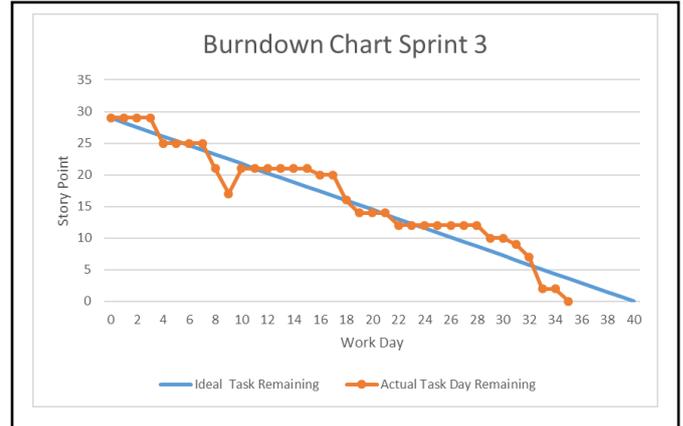


Gambar 2. *Burndown Chart Sprint 2*

Garis *actual task days remaining* pada grafik *burndown chart sprint 2* di permulaan bisa mengikuti atau tidak berada jauh dari garis *ideal task remaining* hal ini menandakan pelaksanaan dan perencanaan sesuai. Beda halnya yang terjadi pada *work days* ke 7 sampai ke 13 *actual task day remaining* berada di bawah garis *ideal task remaining*. Kondisi seperti ini menandakan *sprint* akan selesai lebih awal pada dari estimasi waktu yang sebelumnya ditetapkan.

3) Analisis Sprint 3

Sprint 3 dengan jumlah *story point* 33 dan waktu pelaksanaan *sprint* adalah 19 hari. Gambar 3 adalah grafik *burndown chart sprint 3*.



Gambar 3. *Burndown Chart Sprint 3*

Pada *sprint 3* terjadi penambahan tugas di tengah pelaksanaan *sprint*. Penambahan tugas bisa dilihat pada *work day* ke 9 garis *actual task day remaining* mengalami kenaikan dari 17 menjadi 21. Penambahan *product backlog* pada *sprint 3* tidak mempengaruhi pada estimasi waktu. Hal ini dikarenakan pada saat di tambahkan *product backlog* baru, garis *actual task day remaining* berada di bawah garis *Ideal task day remaining*.

Burndown chart sprint 3 ada dua waktu di mana garis membentuk garis lurus. Hal ini disebabkan karena ketidaksiplinan tim dalam melaksanakan *daily scrum* sehingga pencapaian kerja tidak dilaporkan.

4) Analisis Focus Factor

Nilai *focus factor* pada setiap *sprint* berbeda-beda dan mengalami peningkatan. *Sprint 1* memiliki nilai *focus factor* 0,27, *sprint 2* memiliki nilai *focus factor* 0,36 dan *sprint 3* memiliki nilai *focus factor* 0,47. Ringkasan pelaksanaan proses pengembangan aplikasi IDMS disajikan pada Tabel 4.13.

TABEL 5. RINGKASAN PELAKSANAAN SCRUM

Sprint	Tanggal Mulai	Tanggal Selesai	JD ^a	JSP ^b	WDT ^c	FF ^d
1	11-10-2016	08-11-2016	4	22	80	0,27
2	10-11-2016	29-11-2016	4	20	56	0,36
3	30-11-2016	20-01-2017	2	33	70	0,47

^a Jumlah Developer

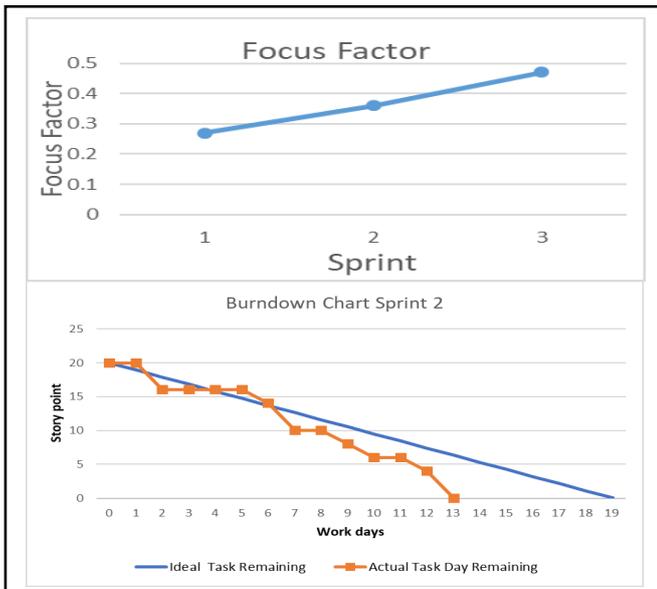
^b Jumlah Story point

^c Work Day Total

^d Focus Factor

Pencapaian *focus factor* pada setiap *sprint* dapat dilihat pada grafik Gambar 4. Dari Gambar 4 kita dapat melihat pada setiap *sprint focus factor* tim mengalami peningkatan.

Peningkatan *focus factor* pada setiap *sprint* menandakan kalau tim mengalami peningkatan pada kinerja dan kemampuan tim dalam mengembangkan produk. Nilai rata-rata *focus factor* pada pengembangan program aplikasi IDMS selama 3 *sprint* adalah 0,37.



Gambar 4. Pencapaian focus factor pada setiap sprint

Untuk sebuah tim yang sehat nilai *focus factor* rata-rata adalah 0,8 dan jika sebuah tim memiliki nilai *focus factor* kurang dari 0,8 menandakan tim terganggu oleh pekerjaan atau kejadian-kejadian eksternal di luar pengembangan produk [5]. Beberapa penyebab tim *developer* pada penelitian kali ini tidak mencapai nilai *focus factor* 0,8 berdasarkan hasil *daily scrum* dan *retrospective* adalah :

- Tim menerapkan standar nilai *story point* yang terlalu kecil.
- Tim *developer* yang masih mempelajari teknologi yang dipakai dalam membuat produk program aplikasi IDMS.
- Terdapat gangguan terhadap masing-masing anggota tim pengembangan produk dengan tugas di luar pengembangan.
- Kedisiplinan tim untuk menjalankan *daily scrum*.
- Kondisi tim *developer* yang masih baru, sehingga perlu ada penyesuaian sistem dan cara kerja tim.

IV. KESIMPULAN

Berdasarkan hasil pelaksanaan metodologi *scrum* pada aplikasi IDMS dengan jumlah sprint 3 dan hasil analisis pada setiap sprint maka penulis dapat ditarik kesimpulan :

- *Scrum Framework* bisa diterapkan pada proses pengembangan program aplikasi skala menengah dengan jumlah tim *deveoper* terbatas atau kecil.
- *Scrum Framework* dapat memfasilitasi kebutuhan pengguna yang dinamis pada saat pelaksanaan pengembangan berlangsung.
- Perhitungan nilai *focus factor* dari rumus
$$\text{Focus Factor} = \frac{\text{Velocity}}{\text{Work Capacity}}$$
 didapat nilai *focus factor* masing-masing *sprint* adalah 0,27 atau 27%, 0,37 atau 37% dan 0,48 atau 48%. Nilai *focus factor* pada ketiga sprint rata rata 0,37 atau 37%. Nilai *focus factor* ini yang nantinya bisa digunakan oleh tim *scrum* di pengembangan program aplikasi selanjutnya.
- Nilai *focus factor* pada setiap sprint mengalami peningkatan. Peningkatan nilai *focus factor* pada setiap sprint menunjukkan kemampuan dan fokus terhadap tugas yang meningkat.

DAFTAR PUSTAKA

- [1] Kenneth, E. K., & Julie, E. K. (2006). *System Analysis and Design* (8 ed.). New Jersey: Prentice Hall.
- [2] Leau, Y. B., Loo, W. K., Tham, W. Y., & Tan, S. F. (2012). Software Development Life Cycle AGILE vs Traditional Approaches. *International Conference on Information and Network Technology (ICINT 2012)*, 37, p. 162. Singapore: IACSIT Press.
- [3] Primadharna, A. U., Doewes, A., & suryani, E. (2015). Pengembangan Aplikasi Skripsi (Tugas Akhir) Berbasis Web Mengguakan Metode Scrum. *Elinvo*, 9.
- [4] Schwaber, K., & Sutherland, J. (2013). *Panduan Scrum*. Scrum.Org and ScrumInc.
- [5] Downey, S., & Sutherland, J. (2013). Scrum Metrics for Hyperproductive Teams:How They Fly like Fighter Aircraft. *Hawaii International Conference on System Sciences*, 4870-4878.
- [6] Rubin, K. S. (2013). *ESSENTIAL SCRUM*. United States: Pearson Education.
- [7] Schwaber, K. (2004). *Agile Project Management with Scrum*. Washington: Microsoft Press.