

IMPLEMENTASI KERANGKA KERJA SCRUM PADA MANAJEMEN PENGEMBANGAN SISTEM INFORMASI

M. Arif Firdaus

Teknik Informatika STMIK AMIKOM Yogyakarta
Jl Ring road Utara, Condongcatur, Sleman, Yogyakarta 55281
Email : aviroez@yahoo.com

Abstrak

Pada metode Agile, banyak kerangka kerja yang bisa dipilih seperti Scrum, eXtreme Programming, Kanban dll. Sedangkan untuk pengguna terbanyak di Amerika adalah Scrum dengan jumlah mencapai 58%.

Scrum memiliki perbedaan pada Segitiga manajemen proyek yang biasanya diisi oleh kualitas, biaya dan waktu. Pada bagian kualitas diganti oleh fungsionalitas. Hal ini disebabkan bahwa kualitas bukan menjadi salah satu variable yang sangat menentukan pada sebuah proyek yang menggunakan scrum. Fungsionalitas pada Scrum menekankan pada selesainya sebuah fungsi atau fitur yang di dalamnya sudah terdapat kualitas, testing, dokumentasi, review dan sebagainya.

Scrum bukanlah kerangka kerja yang mengharuskan anggotanya untuk mengikuti aturan dari buku secara keseluruhan, tapi juga diharapkan mampu berpikir out of the box.

Kata kunci: Agile, Scrum, sprint, backlog.

1. Pendahuluan

Metodologi merupakan salah satu hal yang sangat dibutuhkan terutama untuk membantu menyelesaikan sebuah proyek sistem informasi.

Banyak penelitian yang sudah dilakukan yang membuktikan bahwa metodologi tradisional seperti *waterfall* dianggap sebagai salah satu sebab kegagalan pada proyek sistem informasi. Hal ini disebabkan karena alur pada *waterfall* yang sangat ketat dan mengharuskan untuk menyelesaikan alur sesuai dengan aturan.

Permasalahan lain yang ada saat ini adalah permintaan yang sering berubah-ubah dari pelanggan maupun pengguna, sehingga mengharuskan pengembang menggunakan metodologi yang memungkinkan perubahan setiap saat.

Salah satu metodologi yang banyak dipakai sekarang adalah *Agile*. Metode ini, tidak seperti *waterfall* yang merupakan kumpulan langkah, menekankan pada hubungan dengan organisasi pada tim pengembang.

Pada *Agile* sendiri, banyak kerangka kerja yang bisa dipilih seperti *Scrum*, *eXtreme Programming*, *Kanban* dll. Sedangkan untuk pengguna terbanyak di Amerika adalah *Scrum* dengan jumlah mencapai 58% [9].

Oleh karena itu, perlu pembahasan lebih jauh seputar kerangka kerja *Scrum* ini. Pembahasan bisa berupa langkah-langkah yang harus dilakukan, maupun tugas-tugas tim pengembang.

2. Pembahasan

Agile

Agile adalah seperangkat metode dan metodologi yang membantu tim Anda untuk berpikir lebih efektif, bekerja lebih efisien, dan membuat keputusan yang lebih baik. Metode dan metodologi ini mengatasi semua bidang rekayasa perangkat lunak tradisional, termasuk manajemen proyek, desain perangkat lunak dan arsitektur, dan peningkatan proses. Masing-masing metode dan metodologi terdiri dari praktek-praktek yang efisien dan dioptimalkan agar mudah diadopsi [5].

Kunci untuk model *Agile* adalah bahwa keseluruhan proyek dipecah menjadi serangkaian siklus pengembangan yang singkat (biasanya disebut "iterasi" atau "*Sprint*") mulai 2 sampai 4 minggu setiap siklusnya [7].

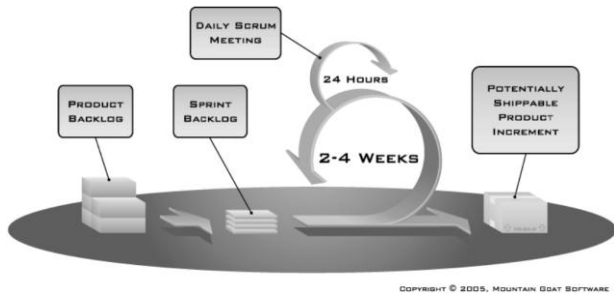
Tujuan dari *Agile* adalah bukan memberi instruksi dan perintah, tapi untuk memberikan inspirasi dan pengaruh agar orang-orang di dalam organisasi dapat bekerja secara mandiri dan kolaboratif untuk menghantarkan produk yang berkualitas [2].

Scrum

Scrum merupakan salah satu model dari metodologi *Agile* pada manajemen pengembangan proyek [1].

Scrum bisa digunakan untuk pengembangan system secara keseluruhan, pengembangan system sebagian maupun proyek internal/pelanggan [1].

Tujuan utama *Scrum* adalah untuk inspect & adapt yang berarti bahwa melihat permasalahan yang ada, dan melakukan adaptasi terhadap masalah tersebut. Pengembangan perangkat lunak menggunakan *Scrum* menekankan untuk mengambil setiap langkah pada pengembangan perangkat lunak secara singkat [8].



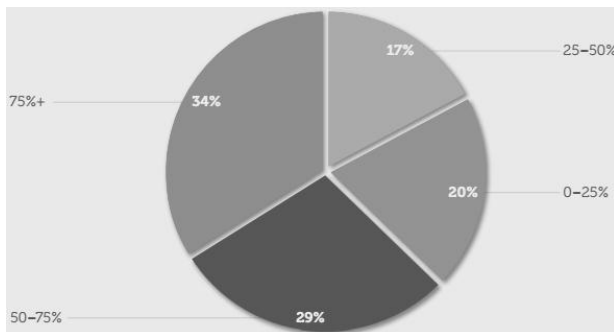
Gambar 1. Siklus Scrum [7]

Adapun sifat-sifat dari *scrum* adalah [4]:

- Ringan
- Mudah dipelajari
- Sulit dikuasai

Scrum tidak boleh dikaitkan dengan metode apapun, karena *Scrum* merupakan sebuah kerangka kerja dan bukan metodologi. *Scrum* lebih menekankan orang-orang yang menjalankan proses pengembangan perangkat lunak itu sendiri daripada alur tahapan proses perangkat lunak [2].

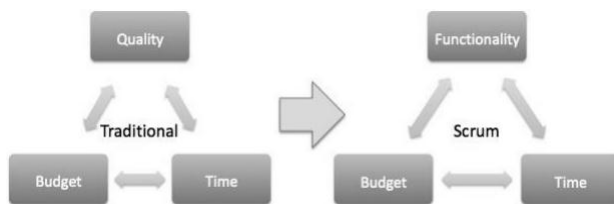
Pembinaan sumber daya manusia menjadi focus utama pada *Scrum* daripada hanya sekedar mengikuti alur sebuah metode. Langkah-langkah yang diberikan pada kerangka kerja *scrum* hanyalah untuk memudahkan tim berkolaborasi dengan masing-masing anggotanya.



Gambar 2. Grafik kesuksesan penggunaan Scrum [6]

Dari Gambar 2 di atas, bisa dipahami bahwa dari partisipan keseluruhan, 34% dari 75% atau lebih proyeknya berhasil dihantar setelah menggunakan *scrum*. Sedangkan 29% dari 50%-75% dari proyek yang dikerjakan berhasil dihantar dengan baik.

Perbandingan Metode Tradisional dan Scrum



Gambar 3. Metode Tradisional dan Scrum [1]

Pada Gambar 3 di atas bisa dipahami bahwa *Scrum* memiliki perbedaan pada segitiga manajemen proyek

yang biasanya diisi oleh kualitas, biaya dan waktu. Pada bagian kualitas diganti oleh fungsionalitas. Hal ini disebabkan bahwa kualitas bukan menjadi salah satu variabel yang sangat menentukan pada sebuah proyek yang menggunakan *scrum*. Fungsionalitas pada *Scrum* menekankan pada selesainya sebuah fungsi atau fitur yang di dalamnya sudah terdapat kualitas, testing, dokumentasi, review dan sebagainya.

Product Backlog

Merupakan sebuah daftar *item* yang diperlukan pada produk dan merupakan sumber dari persyaratan yang dibutuhkan untuk membuat sebuah produk[4]. Produk *backlog* harus dilakukan dan sudah disusun berdasarkan prioritas tertentu. *Item* tersebut bisa berupa hasil dari eksplorasi kebutuhan pelanggan, deskripsi secara functional dan non functional dan hal-hal yang diperlukan untuk merilis sebuah produk jadi [3].

Backlog yang akan dibuat harus terdiri dari 4 kategori, yaitu [3]:

- *Detailed Appropriately*

Backlog yang ingin dibuat, haruslah dijelaskan sedetail-detailnya. Seperti pada Gambar 1, produk dengan detail yang tinggi, dijelaskan lebih detail. Produk yang mendapatkan prioritas menengah memiliki penjelasan yang agak detail. Sedangkan produk yang rendah prioritasnya mendapat deskripsi yang kurang detail [4].

Hal ini berlangsung terus-menerus hingga produk yang menempati prioritas tinggi selesai dikerjakan, dan digantikan oleh produk dengan prioritas di bawahnya. Penentuan *product backlog* yang memiliki prioritas tinggi harus dipecah menjadi beberapa bagian kecil, sehingga memudahkan tim untuk membagi pekerjaan sesuai dengan kompetensinya.



Gambar 4. Product Backlog dengan prioritasnya

- *Estimated*

Produk harus selalu diestimasi. Estimasi biasanya diperkirakan menggunakan poin-poin tertentu ataupun jumlah hari. Dengan mengetahui hal ini, memudahkan untuk memberikan prioritas maupun rencana rilis.

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		30	

Gambar 5. Product Backlog dengan estimasi [1]

- *Emergent*

Backlog harus selalu dinamis, yang berarti bahwa siap menerima perubahan yang diberikan. *Backlog* diharapkan selalu mampu untuk berkembang karena isinya selalu berubah berdasarkan kebutuhan pelanggan maupun umpan balik yang didapatkan.

- *Prioritized*

Seperti sudah dijelaskan sebelumnya, semua pekerjaan pada *backlog* harus diberikan prioritas dari yang tertinggi ke terendah. Prioritas tertinggi menandakan bahwa pekerjaan harus segera diselesaikan, dan ketika sudah selesai, pekerjaan harus segera dihapus dari *backlog* dan digantikan oleh pekerjaan dengan prioritas di bawahnya.

Product Owner merupakan satu-satunya orang yang bertanggung-jawab untuk mengelola *Product Backlog* [4]. Adapun tugas-tugas yang harus dilakukan oleh *product owner* adalah sbb:

- Mengekspresikan dengan jelas *item Product Backlog*;
- Mengurutkan *item* di dalam *Product Backlog* untuk mencapai tujuan dan misi dengan cara terbaik;
- Mengoptimalkan nilai dari hasil pekerjaan Tim Pengembang
- Memastikan *Product Backlog* transparan, jelas, dan dapat dilihat semua pihak, dan menunjukkan apa yang akan dikerjakan oleh Tim *Scrum* selanjutnya;
- Memastikan Tim Pengembang dapat memahami *item* dalam *Product Backlog* hingga batasan yang diperlukan;

Sprint Backlog

Sprint Backlog adalah sekumpulan *item Product Backlog* yang telah dipilih untuk dikerjakan di *Sprint*, juga di dalamnya rencana untuk mengembangkan potongan tambahan produk dan merealisasikan *Sprint Goal*. *Sprint Backlog* adalah perkiraan mengenai fungsionalitas apa yang akan tersedia di iterasi selanjutnya dan pekerjaan yang perlu dikerjakan untuk menghantarkan fungsionalitas tersebut menjadi potongan produk yang dianggap selesai [4].

Sprint

Sprint merupakan jantung dari *Scrum* yang berarti bahwa batasan waktu selama satu bulan atau kurang, di mana sebuah pekerjaan dianggap “Selesai”, bisa digunakan, dan berpotensi untuk dirilis. *Sprint* biasanya memiliki durasi yang konsisten sepanjang proses pengembangan produk [2].

Pada *Scrum*, terdapat beberapa bagian, yaitu

- *Sprint Planning*

Sprint Planning adalah sebuah aktivitas untuk membuat rencana pada *Product Backlog Item* (PBI) yang akan dan siap dikembangkan oleh tim pengembang pada satu *Sprint* [2].

Ada beberapa hal yang perlu dipersiapkan sebelum melakukan *Sprint Planning*, yaitu:

- Menentukan tujuan *Sprint*
- Mempersiapkan pekerjaan sesuai dengan waktunya
- Memperkecil *item* dari *product backlog* sehingga cukup untuk dikerjakan pada satu *sprint*.
- Memastikan Kejelasan, kemampuan untuk dites, dan Kelayakan

- *Daily Scrums*

Daily Scrum meeting atau biasa juga disebut *daily Stand up meeting* adalah kegiatan/pertemuan dengan batasan waktu maksimum selama 15 menit agar Tim Pengembang dapat mensinkronisasikan pekerjaan mereka dan membuat perencanaan untuk 24 jam ke depan [4].

Daily Scrum memungkinkan tim untuk memudahkan mengelola pekerjaan dan mengungkapkan segala hambatan yang diperoleh tiap harinya. *Product Owner* diharapkan selalu mengikuti *daily scrum* agar bisa menyaksikan secara langsung progress yang sudah dilakukan oleh tim pengembang [2].

- *Sprint Review*

Sprint Review adalah sebuah aktivitas atau acara yang dilaksanakan pada akhir *sprint* yang tujuannya untuk meninjau hasil pekerjaan tim pengembang pada *sprint* yang terakhir dikerjakan [2].

- *Sprint Retrospective*

Sprint Retrospective adalah sebuah kesempatan untuk tim *Scrum* meninjau dirinya sendiri dan melakukan peningkatan yang akan diimplementasikan pada *sprint* selanjutnya [2].

Tim Scrum

Tim *scrum* harus fleksibel, kreatif dan produktif. Tim juga diharapkan mampu menghantarkan product secara berkala, memaksimalkan kesempatan dan menerima masukan [1].

Tim *scrum* terdiri dari [4]:

- *Product Owner*

Product Owner bertanggung jawab untuk memaksimalkan nilai produk dan hasil kerja Tim Pengembang [4]. *Product Owner* adalah bagian dari tim *Scrum* dan bisa bekerja sama dengan anggota tim lainnya dan memastikan bahwa pekerjaan yang dibutuhkan, berhasil selesai sesuai keinginan[3].

Product Owner diharapkan memiliki criteria di bawah ini [3]:

- Memiliki visi dan misi
- Bisa memimpin tim
- Komunikator and Negosiator
- Memiliki wewenang and berkomitmen
- Selalu ada dan memiliki kualifikasi

- *Scrum Master*

Scrum Master bekerja dengan memastikan bahwa tim dapat melewati segala hambatan yang ditemukan dan sudah meminta bantuan sebelumnya [5].

Scrum Master bukan berarti orang menguasai *Scrum*, tapi merupakan sebuah peran tertentu dalam kerangka kerja ini, karena *Scrum Master* juga memiliki peran yang berhubungan dengan *Product Owner* [6].

Tanggung jawab dari *Scrum Master* adalah sebagai berikut[1]:

- Menjaga anggota tim dari gangguan eksternal
- Bertindak sebagai agen perubahan dan selalu beradaptasi untuk memaksimalkan produktivitas tim
- Memberi pelatihan kepada tim
- Meminimalisir hambatan pada tim
- Memastikan komunikasi yang efisien dan efektif antara tim dan *product owner*
- Memberikan fasilitas untuk setiap kegiatan yang berhubungan dengan *scrum*

- *Scrum Team*

Scrum team merupakan sekelompok individu yang bekerja sama untuk menghantarkan ataupun menaikkan nilai dari produk yang diminta oleh pelanggan [1].

Prasyarat dari tim *Scrum* adalah sbb[2]:

- Berfungsi Antarlintas

- Semua anggota berpartisipasi
- Mampu berkomunikasi dengan baik
- *Meaningful Conversation*
- Memiliki tujuan yang sama

Pekerjaan Dianggap Selesai

Pada proyek *Scrum*, pekerjaan dianggap selesai apabila memenuhi kriteria sebagai berikut [7]:

- Lingkup testing telah dilakukan dan telah berhasil melewati persyaratan dan kriteria test
- Kode sudah direview sebelumnya dan memenuhi persyaratan yang sudah ditentukan
- Sesuai dengan dokumentasi yang telah disepakati sebelumnya

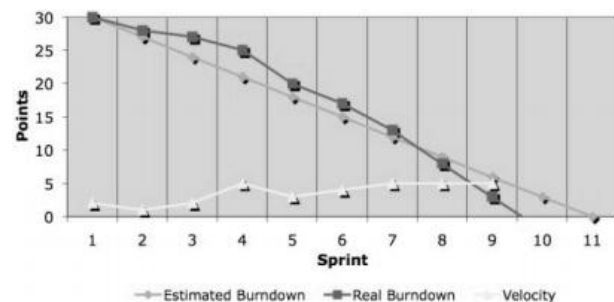
Pada bagian ini, hal yang perlu diperhatikan bukan hanya fitur atau modul yang sudah dihantarkan oleh pengembang, tapi juga kualitas dari fitur atau modul tersebut. Walaupun begitu, pengertian bahwa pekerjaan sudah dianggap selesai bisa berbeda antara tim *Scrum* satu dengan lainnya [1].

Pengertian pekerjaan dianggap selesai bisa juga dipisah antara siklus pada *scrum*, Seperti pada[1]:

- Pengertian selesai pada *Product Backlog*
- Pengertian selesai pada *Sprint*
- Pengertian selesai pada rilis

Scrum Burndown Chart

The *Scrum Burndown Chart* adalah alat pengukuran visual yang menunjukkan pekerjaan selesai per hari terhadap tingkat proyeksi penyelesaian pembebasan proyek ini. Tujuannya adalah untuk memungkinkan bahwa proyek ini tetap pada jalurnya untuk memberikan solusi yang diharapkan dalam jadwal yang ditentukan [1].



Gambar 6. Product Backlog dengan estimasi [1]

Velocity merupakan satuan pada *Scrum burndown Chart* yang menandakan tingkat kemajuan pada tim *Scrum*. *Velocity* dihitung jika dan hanya jika pekerjaan sudah selesai dikerjakan dan sudah dirilis.

Implementasi

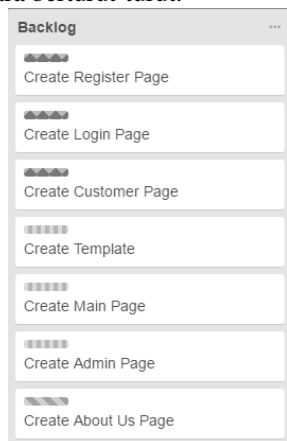
Sistem Informasi yang akan dibangun berupa sistem informasi di mana anggotanya bisa melakukan *login* ke dalam sistem.

Membuat *Product Backlog*

Salah satu langkah awal yang harus dilakukan pada permulaan scrum adalah dengan membuat *product backlog*.

Pembuatan *product backlog* bisa dilakukan menggunakan buku catatan, papan tulis, maupun aplikasi. Pada Gambar 7, aplikasi yang digunakan adalah *Trello*.

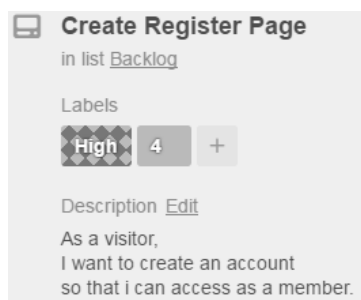
Daftar pekerjaan sudah harus disusun berdasarkan prioritasnya dengan prioritas tertinggi selalu ditempatkan di posisi teratas dan prioritas sedang dan rendah di bawahnya secara berturut-turut.



Gambar 7. Contoh *Product Backlog* menggunakan *Trello*

Backlog yang sudah dibuat harus di deskripsikan sesuai kebutuhan fungsionalnya atau biasa disebut juga *user story*. Disyaratkan menyebutkan pengguna, hal yang ingin dikerjakan, dan tujuan yang harus dicapai. Misal *As a/an <User>, I want to <task>, so that <goal>*.

Prioritas juga harus sudah ditentukan dan diurutkan, misalnya menggunakan prioritas *High, Medium, Low*. Estimasi juga harus ditentukan dan dituliskan dengan angka. Misal pada Gambar 8 yang memberikan estimasi untuk backlog pembuatan halaman registrasi sebanyak 4 poin. Untuk kasus ini, poin ini mewakili jam.



Gambar 8. Contoh *Product Backlog* menggunakan *Trello*

Melakukan *Scrum Meeting*

Scrum meeting dilakukan dalam sebuah siklus *sprint* untuk 2-4 minggu dengan studi kasus yang akan dilakukan adalah menggunakan siklus tiap 2 minggu.

Sebelum melakukan *meeting*, para tim diharapkan untuk mengisi *form sprint* yang sudah dibuat, tujuannya adalah untuk mengetahui apa yang akan sudah diselesaikan oleh tim pada hari sebelumnya, yang akan dilakukan hari ini, dan permasalahan yang sudah dihadapi.

Selain itu, form *sprint* juga bisa membantu melakukan dokumentasi terhadap *Scrum* harian yang dilakukan. Contoh *form* yang bisa digunakan adalah seperti Tabel 1. *Form sprint* dipisahkan sesuai dengan siklus *sprint* yang sudah ditetapkan sebelumnya. *Form* yang dibuat juga diharapkan untuk membantu masing-masing anggota mengingat apa yang akan dilakukan dan kebutuhan pada hari itu dan mencatat pencapaian serta permasalahan yang sudah dilakukan.

Tabel 1. Tabel *Form sprint*

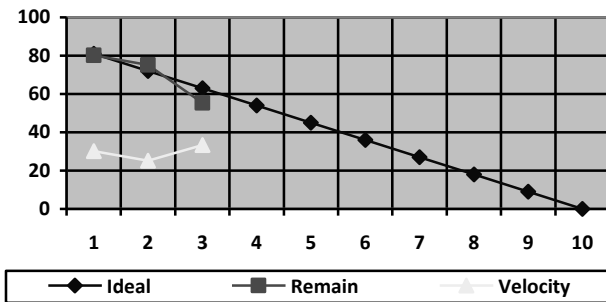
Sprint 1 1 Jun – 14 Jun 2016	01 Juni 2016		Comments
	To do Requirement	Done Problems	
Andre	Assign Job		
Brian	Create testing scenario	Register is tested	Some more validations needed for required fields
John	Create register view	Register view is almost done	
	Scenario from brian	Some problems with javascript	Probably needs library update
Doe	Create register function	Register is functioned properly	
	View from John	Lacks of validation	

Membuat *Burndown Chart*

Untuk melakukan evaluasi dan melihat seberapa jauh pekerjaan sudah diselesaikan, pembuatan *burndown chart* merupakan salah satu cara untuk membantu memberikan informasi seputar evaluasi dari penerapan scrum dalam bentuk grafik.

Burndown chart ditandai dengan sumbu Y untuk story poin [10] dan sumbu X untuk iterasi. *Burndown chart* dimulai dari garis ideal sebagai pembatas bahwa pekerjaan diselesaikan sesuai dengan estimasi yang sudah ditentukan. Pengisian chart dilakukan setelah ada

pekerjaan yang sudah dianggap selesai berdasarkan hari sprint di mana pekerjaan itu dilakukan.



Gambar 9. Contoh Burndown Chart berdasarkan pekerjaan yang sudah diselesaikan

Perhitungan Remaining effort pada burndown chart adalah berdasarkan jumlah poin pekerjaan yang berhasil dilakukan. Perhitungan dimulai dari titik tertinggi dari ideal effort hingga angkanya menjadi 0.

Perhitungan velocity merupakan rata2 dari pekerjaan yang sudah diselesaikan dibagi dengan iterasinya. Misal tim berhasil menyelesaikan 30 poin pada iterasi 1, maka velocity nya adalah sebesar 30. Pada iterasi ke-2, tim berhasil menyelesaikan 20 poin, maka velocitynya adalah $(20+30)/2=25$. Hal ini berlaku seterusnya.

3. Kesimpulan

Tujuan utama dari penerapan kerangka kerja *Scrum* adalah untuk memenuhi *Agile* metodologi. *Scrum* bukanlah kerangka kerja yang mengharuskan anggotanya untuk mengikuti aturan dari buku secara keseluruhan, tapi juga diharapkan mampu berpikir *out of the box*. Oleh karena itu, para anggota tim *Scrum*, dianjurkan untuk selalu belajar dan mengembangkan diri.

Daftar Pustaka

- [1] Evans, Jenny, *Scrum Revealed*, International Scrum Institute™.
- [2] Partogi, Joshua, *Manajemen Modern dengan Scrum*, Yogyakarta: Penerbit Andi, 2015.
- [3] Pichler, Roman, *Agile product management with Scrum : creating products that customers love*, New York: Addison-Wesley, 2010.
- [4] Schwaber, Ken and Sutherland, Jeff, *The Scrum Guide*, Scrum.org, 2013.
- [5] Stellman, Andrew & Greene, Jennifer, *Learning Agile: Understanding Scrum, Xp, Lean, And Kanban*, California: O'Reilly Media, 2015.
- [6] Kim, Don, *The State of Scrum: Benchmarks and Guidelines*, ScrumAlliance, 2013.
- [7] Edwards, Ian; Bickerstaff, Roger; Bartsch, Christian, *Contracting for agile software development projects*, Bird & Bird.
- [8] Deemer, Pete; Benefield, Gabrielle; Larman, Craig; Vodde, Bas, *A Lightweight Guide To The Theory And Practice Of Scrum, Version 2.0*, 2012, Info Queue
- [9] VersionOne, *the 10th Annual State of Agile™ Report*, 2016, VersionOne.com
- [10] James, Michael, *Scrum Reference Card*, 2016, VersionOne.com

Biodata Penulis

M. Arif Firdaus, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AKAKOM Yogyakarta, lulus tahun 2013. Saat ini sedang mengambil gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Teknik Informatika di STMIK AMIKOM Yogyakarta dan menjadi Software Developer di printerous.com.