

Analisis Tingkat Kematangan Implementasi Scrum Menggunakan Scrum Maturity Model pada Instansi Pemerintah

Ermawan Setyo Pambudi

Magister Sistem Informasi

Universitas Gunadarma

Jl. Margonda Raya No. 100, Depok 16424, Jawa Barat

E-mail: wawancell@gmail.com

Abstrak

Tugas dan fungsi utama Unit Pusat TIK pada Instansi Pemerintah salah satunya adalah melakukan pengembangan Sistem Informasi. Sejak pertengahan tahun 2016 Unit Pusat TIK ini mulai menerapkan metode pengembangan berbasis *agile* dengan kerangka kerja *Scrum*. Implementasi *Scrum* mulai memberikan perbaikan dalam pengembangan perangkat lunak, namun dalam penerapannya masih terdapat permasalahan yaitu tidak semua target dapat diselesaikan tepat waktu dan kualitas perangkat lunak yang dihasilkan belum cukup memenuhi harapan pengguna. Penelitian ini melakukan evaluasi dan analisis tingkat kematangan pengembangan perangkat lunak menggunakan *Scrum Maturity Model*. Proses pengumpulan data dilakukan melalui kegiatan wawancara dan kuesioner terhadap *Scrum Master* serta pengamatan terhadap dokumentasi pengembangan perangkat lunak yang menarapkan kerangka kerja *Scrum*. Kesimpulan dari hasil penelitian ini analisis tingkat kematangan berhasil dilakukan dan mencapai tingkat kematangan 2 *Scrum Maturity Model*. Rekomendasi diberikan untuk memperbaiki proses penerapan *Scrum* berdasarkan praktik-praktik yang belum dijalankan sepenuhnya sesuai *best practice* dari *Scrum Maturity Model*.

Kata Kunci: Scrum, Agile, Project Management, Maturity Level, Scrum Maturity Model.

Pendahuluan

Teknologi Informasi telah berperan sebagai *Business Enabler* [1] bagi organisasi, oleh karena itu diperlukan tata kelola yang baik agar menjamin keberlangsungan suatu proses bisnis serta mendapatkan manfaat yang optimal dari penerapan TI. Pengelolaan TI mencakup seluruh aspek antara lain pengelolaan infrastruktur TI yang handal sampai dengan pengembangan perangkat lunak yang berkualitas.

Pengembangan perangkat lunak dapat dikatakan berkualitas apabila memenuhi tiga kriteria, yaitu *on – time* (sesuai dengan waktu pengerjaan proyek), *on – budget* (sesuai dengan pembiayaan proyek), dan *on – scope* (menghasilkan produk yang sesuai dengan harapan) [2]. Pengembangan perangkat lunak tentu memerlukan sebuah mekanisme manajemen proyek agar pelaksanaan setiap tahapan kegiatan dapat terselesaikan dengan efektif dan efisien.

Sejak pertengahan tahun 2016 Instansi ini mulai menerapkan metode pengembangan berbasis *agile* dengan kerangka kerja *Scrum*. *Scrum* meru-

pakan sebuah metodologi *agile* untuk mengelola proyek melalui metode *incremental* dan *iterative* [3]. Implementasi *Scrum* mulai memberikan perbaikan dalam pengembangan perangkat lunak, namun dalam penerapannya masih terdapat permasalahan yaitu tidak semua target dapat diselesaikan tepat waktu dan kualitas perangkat lunak yang dihasilkan belum cukup memenuhi harapan pengguna.

Berdasarkan hasil evaluasi tahunan diperoleh data pelaksanaan proyek pengembangan perangkat lunak berbasis *Scrum* yang dikumpulkan dari kurun waktu 3 tahun terakhir (2018-2020) terdapat beberapa aplikasi yang penyelesaiannya terlambat, *sprint backlog* yang tidak selesai, *sprint* yang berhenti, dan *product backlog* yang tidak lengkap. Permasalahan tersebut mungkin terjadi karena penguasaan akan metodologi *Scrum* pada sebagian besar anggota tim pengembang masih terbatas. Peran di dalam anggota tim terkadang masih belum jelas dan terjadi duplikasi seperti *Scrum Master* merangkap sebagai *Product Owner*. Praktik-praktek atau kegiatan di dalam proses *Scrum* belum sepenuhnya dilaksanakan secara konsisten. Selain itu ko-

munikasi antara pengembang dengan pemilik proses bisnis belum cukup efektif sehingga seringkali terjadi kesalahan dalam mengidentifikasi kebutuhan pengguna.

Berdasarkan hal tersebut maka perumusan masalah pada penelitian ini untuk menjawab pertanyaan berikut ini: Berapakah tingkat kematangan manajemen proyek pengembangan perangkat lunak berbasis Scrum pada Unit Pusat TIK pada Instansi Pemerintah, dan rekomendasi apa saja yang dapat diberikan untuk memperbaiki tingkat kematangan dan meningkatkan kinerja dalam pengembangan perangkat lunak?

Ruang lingkup penelitian ini dibatasi pada 3 proyek perangkat lunak yang menerapkan Scrum dan dikembangkan oleh Unit Pusat TIK pada kurun waktu 2018-2020. Penelitian ini bertujuan untuk mengevaluasi tingkat kematangan manajemen proyek pengembangan perangkat lunak berbasis Scrum dan memberikan rekomendasi perbaikan, sehingga diharapkan dapat digunakan sebagai bahan evaluasi atas permasalahan-permasalahan yang dihadapi saat ini dan dapat meningkatkan kualitas pengembangan perangkat lunak pada proyek pengembangan berikutnya di Unit Pusat TIK Instansi Pemerintah.

Tinjauan Teoritis

Konsep dari metodologi *Agile* pada awalnya berasal dari konsep *iterative* dan *incremental* pada proses pengembangan perangkat lunak, dimana *requirement* dan solusi berkembang melalui rencana-rencana eksekusi yang dibagi secara tahapan-tahapan iteratif dan dilakukannya penilaian terhadap *output* pada setiap tahapan. *Agile* memiliki arti cepat, ringan, dan lincah, dan digunakan sebagai kata yang menggambarkan model proses yang berbeda dari model-model proses yang sudah ada sebelumnya.

Konsep *Agile software development* dicetuskan oleh Kent Beck dan 16 rekannya dengan menyatakan bahwa *Agile Software Development* adalah cara membangun perangkat lunak dengan melakukan pengembangan perangkat lunak serta membantu orang lain yang membangunnya sekaligus *Scrum* adalah salah satu dari metodologi *agile* untuk memanajemen proyek pengembangan melalui metode iterasi dan *incremental* [3]. *Scrum* bukan hanya sebuah metodologi, melainkan juga adalah sebuah proses yang sederhana untuk pengembangan aplikasi yang kompleks dan untuk meningkatkan kinerja sebuah organisasi.

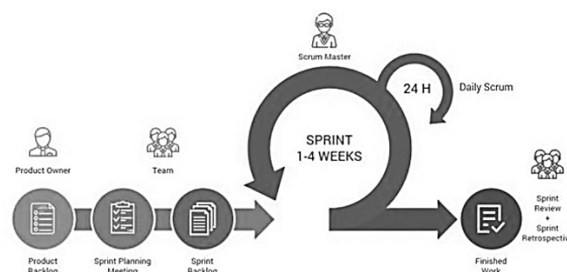
Scrums sangat sesuai digunakan untuk pengembangan perangkat lunak yang kompleks karena proses *Scrum* menggunakan metode empiris atau dengan kata lain setiap tahapan di dalamnya melibatkan proses inspeksi dan adaptasi. *Scrum* memperkenalkan konsep dari *Sprint* yang merepresentasikan iterasi dari pengembangan dalam kurun

waktu tertentu [4].

Tiga role utama yang ada pada *Scrum* yaitu *Scrum Master* yang bertanggung jawab pada proses *Scrum* dan penggunaan *Scrum* yang tepat guna, *Product Owner* yang bertanggung jawab untuk penyalarsan pengembangan dan definisi tujuan bisnis, dan Tim yang bertugas untuk implementasi proses pengembangan. Sebuah tim terdiri dari lima sampai sembilan anggota dengan lintas fungsional keterampilan.

Scrum juga mengidentifikasi empat artefak yang dioperasikan oleh seluruh tim *Scrum* pada siklus pengembangan yaitu *Product Backlog* yang merupakan sebuah daftar prioritas mengenai segala sesuatu yang diperlukan untuk menyimpulkan suatu produk; *Sprint Backlog* yaitu merupakan daftar tugas yang harus dilakukan selama *Sprint*; *Release Burndown Charts* yang berupa grafik yang menunjukkan kemajuan proyek dari waktu ke waktu; dan *Sprint Burndown Charts* yaitu grafik yang menampilkan kemajuan *Sprint* dari waktu ke waktu.

Kegiatan pertemuan/*ceremonial Scrum*, terdiri dari *Release Planning Meeting* yang merupakan *kick – off Meeting* untuk menentukan item awal dari *Product Backlog* serta analisis jadwal dan anggaran untuk proyek pembangunan oleh tim *Scrum*; *Sprint Planning Meeting* dimana tim pengembangan dan klien akan membahas masalah dan menentukan tujuan untuk *Sprint* berikutnya; *Daily Scrum* yang merupakan pertemuan singkat dan santai selama lima belas menit bagi pengembang untuk mengupdate tugas yang telah diselesaikan, yang akan di kerjakan dan permasalahan yang dihadapi; *Sprint Review* yang dilakukan setiap akhir *Sprint* dengan mendemonstrasi perangkat lunak yang telah berjalan untuk *user* dan *stakeholder*; dan *Sprint Retrospective* dimana tim melakukan review terhadap pelaksanaan *Sprint* terakhir dalam rangka mencari perbaikan pada penggunaan Metodologi Scrum dan kolaborasi pada umumnya. Siklus dan proses *Scrum* seperti terlihat pada gambar 1 dibawah ini.



Gambar 1: Siklus dan Proses Scrum [4]

Beberapa penelitian terdahulu dijadikan sebagai referensi dan digunakan untuk menganalisis dan membantu membuat solusi terhadap masalah yang terjadi pada objek penelitian ini. M. Ashraf dan N. Ali [5] melakukan penelitian tentang identifikasi

dan analisis permasalahan dalam pengimplementasian metode *scrum*, hasil penelitiannya menyimpulkan bahwa dalam implementasi *Scrum*, dengan memberikan pelatihan yang tepat, partisipasi aktif setiap anggota tim, komunikasi yang efektif dan koordinasi antara anggota tim akan dapat meminimalkan berbagai masalah, sebaliknya bekerja secara individu, kurangnya koordinasi dan roles yang tumpang tindih menyebabkan masalah dalam implementasi *Scrum*.

R. Akif dan M. Hammad [6] meneliti tentang isu dan tantangan dalam pengimplementasian metode *Scrum* dengan melakukan survey terhadap perusahaan, hasil penelitiannya menunjukkan bahwa terdapat beberapa masalah yang dapat mempengaruhi dalam mengimplementasikan *Scrum* yaitu integrasi, kualitas code, gangguan dalam kerja tim, durasi sprint dan kurangnya pelatihan. J. Wan, Y. Zhu dan M. Zeng [7] membahas studi kasus mengenai faktor penentu utama keberhasilan dalam implementasi *Scrum*. Hasil dari penelitian ini adalah implementasi *Scrum* dapat berjalan dengan baik apabila perusahaan menerapkan 12 prinsip *Agile Development*, meningkatkan kualitas pengelolaan proyek, dan mendapatkan dukungan dari pihak top manajemen.

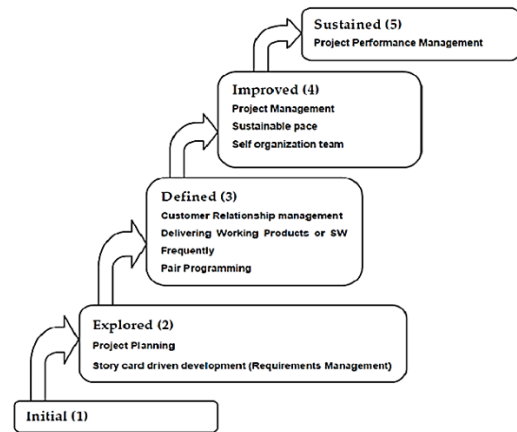
Ahlijati Nuraimah [8] melakukan penelitian untuk mengukur tingkat kematangan manajemen proyek pengembangan perangkat lunak menggunakan *Scrum Maturity Model* oleh karena adanya masalah ketidakefektifan penerapan kerangka kerja *Scrum* pada perusahaan PT. XYZ yang dilihat dari rata-rata sisa pekerjaan pada *sprint* yang selalu tidak tuntas. Hasil dari penelitian ini perusahaan perlu meningkatkan pemahaman Scrum untuk seluruh anggota tim melalui pelatihan dan penerapan scrum secara konsisten agar tingkat kematangannya dapat ditingkatkan. Pemecahan masalah pada penelitian ini dilakukan kajian terhadap metodologi-metodologi yang relevan yaitu *Capability Maturity Model Integration* (CMMI), *Agile Maturity Model* (AMM), dan *Scrum Maturity Model*.

CMMI (*Capability Maturity Model Integration*) merupakan suatu model pendekatan dalam penilaian tingkat kematangan dan kemampuan sebuah organisasi. CMMI dipublikasikan oleh SEI (*Software Engineering Institute*) of Carnegie Mellon University. Tujuan dari CMMI adalah untuk peningkatan proses dalam membangun produk perangkat lunak berdasarkan proses pengembangannya.

Agile Maturity Model [9] bertujuan untuk memberikan fokus pada proses *Agile* sebagai dasar untuk studi mereka. Model kematangan ini masih terinspirasi oleh CMMI seperti terlihat pada Gambar 2. Masing-masing tingkat memiliki satu set tujuan untuk dipraktikkan. Metode ini terinspirasi dari beberapa metodologi *Agile*.

Tujuan utama AMM adalah untuk program peningkatan impementasi pengembangan berbasis *Agile*, oleh karena itu mengidentifikasi tingkat kematangan merupakan aktivitas penting dalam pen-

dekatan AMM. Penilaian mandiri merupakan cara yang paling umum untuk melakukan penilaian terhadap proses pengembangan perangkat lunak [10]. Pendekatan ini menjadi populer karena penilaian mandiri tidak membutuhkan biaya yang tinggi, kemudahan akses, dan kepemilikan atas data [10].



Gambar 2: Agile Maturity Model [9]

Tipe jawaban kuesioner pada AMM adalah “Ya” (poin pertanyaan dijalankan/ada seluruhnya), “Tidak” (poin pertanyaan tidak dijalankan/tidak ada seluruhnya), “Sebagian” (poin pertanyaan hanya dijalankan sebagian), dan “Tidak Berlaku (N/A)” (poin pertanyaan tidak dapat diimplementasikan) [9]. Presentase untuk masing-masing KPA dihitung menggunakan persamaan seperti rumus 1:

$$\frac{\sum(Y_n) + \frac{1}{2} \sum(P_n)}{\sum(T_n) - \sum(NA_n)} \times 100 \quad (1)$$

dengan Y_n adalah jumlah jawaban “Ya”, P_n adalah jumlah jawaban “Sebagian”, T_n adalah total jumlah pertanyaan, dan NA_n adalah jumlah jawaban “Tidak Berlaku”. Berdasarkan nilai persentase KPA Rating yang diperoleh, maka dilakukan penafsiran penilaian ke dalam kategori *Fully Achieved* (Tercapai Sepenuhnya) untuk nilai 86% sampai 100%, *Largely Achieved* (Sebagian Besar Dicapai) untuk nilai 51% sampai 85%, *Partially Achieved* (Sebagian Dicapai) untuk 16% sampai 50%, dan *Not Achieved* (Tidak Tercapai) untuk 0% sampai 15%.

Sebagian besar pendekatan pengukuran *maturity level* yang ada memiliki ruang lingkup yang terlalu luas. Hal ini dilakukan agar memungkinkan Model *Maturity* yang ada dapat secara luas diterima dan dilaksanakan. Namun, seringkali ini diikuti oleh tujuan yang sangat generik dan praktek-praktek yang tidak dapat dengan mudah dipahami dan diterapkan khususnya untuk metodologi *Agile* yang spesifik pada prakteknya. Oleh karena itu, metode *Scrum Maturity Model* muncul untuk mengklasifikasikan kematangan dalam lingkup yang difokuskan pada proses metodologi *Scrum*.

Tujuan utama *Scrum Maturity Model* adalah untuk membantu dan membimbing organisasi yang mengembangkan proyek TI di dalam mengadopsi *Scrum* dan mendorong perbaikan dengan mengambil pendekatan khusus untuk peran *client* atau *user*. Selain itu, *Scrum Maturity Model* juga membantu organisasi yang tidak terbiasa dengan *Scrum*, tapi ingin mengadopsi praktek *Scrum* dengan menggunakan pendekatan yang bertingkat (*staging*) [11]. Level-level yang terdapat pada *Scrum Maturity Model* seperti terlihat pada Tabel 1 dibawah ini.

Tabel 1: Level Scrum Maturity Model [11]

Level	Maturity	Goals
1	Initial	-
2	Managed	Basic Scrum Management Software Requirement Engineering
3	Defined	Customer Relationship Management Iteration Management
4	Quantitatively Managed	Standardized Project Management Measurement and Analysis Management
5	Optimizing	Performance Management

Level 1 (*Initial*) merupakan tingkat pertama dan terendah pada SMM dapat diterapkan untuk setiap organisasi yang menggunakan *Scrum*. Pada tingkat ini tidak ada Goal untuk mendefinisikan proses atau perbaikan. Tidak adanya definisi yang eksplisit mengenai proses pengembangan secara *Agile* dengan metodologi *Scrum* dalam organisasi diklasifikasikan sebagai tingkatan ini.

Level 2 (*Managed*) memperlihatkan praktek pengembangan perangkat lunak yang lebih terstruktur dan lengkap dibandingkan pada Level 1. Pada level ini terdiri dari dua tujuan utama yaitu *Basic Scrum Management* yang memastikan bahwa organisasi harus dapat mencapai penara-

pan minimum yang dapat diterima metodologi *Scrum*, dan *Software Requirement Engineering* yang mengharuskan organisasi untuk mematuhi langkah-langkah yang ada dan menghasilkan produk yang berkualitas.

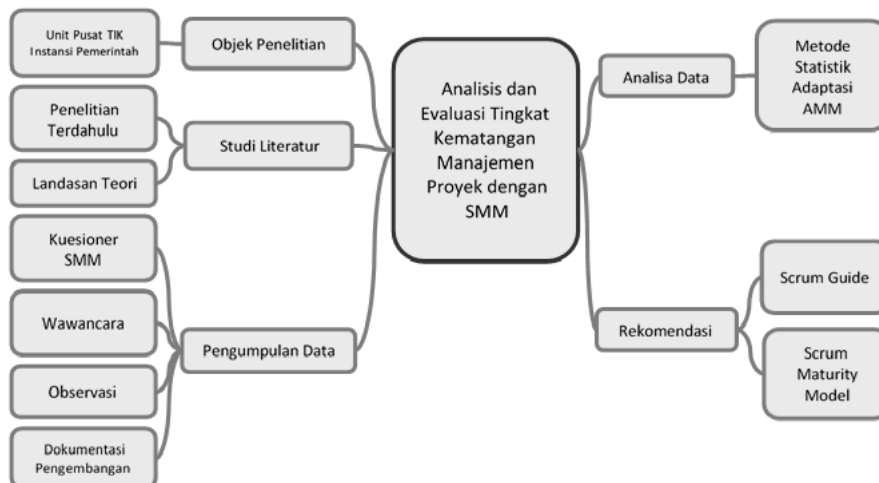
Level 3 (*Defined*) memiliki fokus utama pada hubungan tim pengembang dengan user dan penyelesaian perangkat lunak tepat waktu. Pada level ini terdiri dari dua tujuan utama yaitu *Customer Relationship Management* yang menekankan pentingnya *user* dan upaya yang diperlukan untuk memaksimalkan kerjasama dengan pelanggan, dan *Iteration Management* untuk menjalankan serangkaian praktek yang harus dipenuhi sehingga organisasi selalu menyelesaikan *Sprint* secara tepat waktu dan sesuai dengan *budget*.

Level 4 (*Quantitatively Managed*) menuntut sebuah organisasi untuk menetapkan suatu standar dan proses pengembangan perangkat lunak yang teratur yang dibantu oleh manajemen kinerja, melalui analisis pengukuran dan praktek.

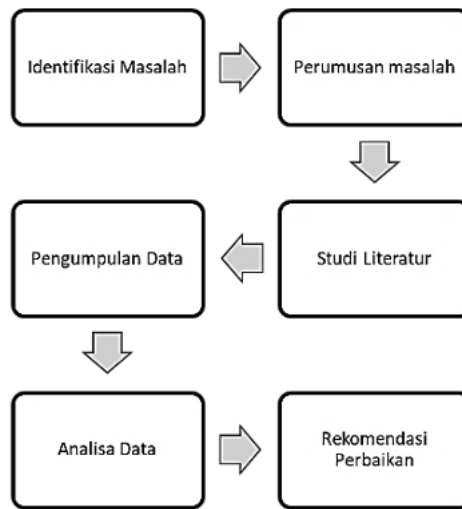
Level 5 (*Optimizing*) merupakan kategori pengembang perangkat lunak kelas atas yang mendorong organisasi terus fokus pada perbaikan diri untuk kompetisi dan membawa tingkat kepuasan dari user, tim pengembangan dan seluruh stakeholder.

Metode Penelitian

Penelitian yang dilakukan menggabungkan penelitian lapangan melalui pengamatan dan terlibat langsung didalam proses pengembangan perangkat lunak Diagram kerangka pemikiran ditunjukkan seperti pada Gambar 3. Adapun tahapan penelitian yang dilakukan seperti terlihat pada Gambar 4.



Gambar 3: Diagram Kerangka Pemikiran



Gambar 4: Tahapan Penelitian

Identifikasi permasalahan dilakukan identifikasi berdasarkan kondisi dan latar belakang yang penulis temukan melalui pengamatan langsung dan studi dokumentasi pengembangan proyek perangkat lunak yang menerapkan *Scrum*. Dari hasil pengamatan tersebut kemudian merumuskan masalah dan menentukan tujuan dari penelitian.

Melalui tinjauan pustaka yang besumber dari buku teks, artikel dan jurnal internasional dan penelitian terdahulu dilakukan studi literatur untuk menggali landasan teori dan menemukan metodologi yang tepat dengan pertanyaan penelitian yang telah dirumuskan pada tahapan sebelumnya. Pengumpulan data dilakukan pada bulan Januari 2021 setelah memiliki rumusan masalah, konsep dan kerangka pemikiran.

Terdapat dua jenis data yang digunakan yaitu data primer dan data sekunder. Data primer didapatkan dari observasi, pembagian kuesioner dan wawancara terhadap *Scrum* Master. Kuesioner dan wawancara di tujukan untuk 3 orang *Scrum* Master yang menangani langsung Proyek pengembangan aplikasi sesuai objek penelitian ini.

Kuesioner dibuat menggunakan alat bantu Google Form yang disampaikan kepada 3 responden pada minggu pertama bulan Januari 2021 dan telah di peroleh hasilnya secara lengkap pada minggu kedua bulan Januari 2021. Wawancara dilakukan terhadap 3 orang *Scrum* Master pada minggu ketiga bulan Januari 2021 untuk menggali informasi tentang proyek yang dikembangkan, peran dan tanggung jawab anggota di dalam tim *Scrum*, praktek-praktek *Scrum* yang telah dilakukan dan kendala-kendala yang dihadapi. Data sekunder didapatkan dari observasi atas dokumentasi 3 proyek pengembangan perangkat lunak pada Unit Pusat TIK kurun waktu 2018-2020 yang dilakukan pada minggu keempat bulan Januari 2021 melalui tools manajemen proyek Microsoft *Team Foundation Server* (TFS) dengan melihat artifak-artifak pengemban-

gan yang ada yaitu *Product Backlog*, *Sprint Backlog*, dan *Sprint Burndown Chart*.

Analisis data dilakukan dari hasil kuesioner, wawancara, dan pengamatan terhadap dokumentasi pengembangan yang ada. Analisis dilakukan menggunakan metode statistik yang diadaptasi dari pendekatan Agile Maturity Model [9], sehingga diperoleh tingkat kematangan proyek pengembangan perangkat lunak yang menerapkan proses kerja *Scrum*. Nilai persentase KPA Rating yang didapatkan kemudian diinterpretasikan ke dalam kategori *Fully Achieved* (Tercapai Sepenuhnya), *Largely Achieved* (Sebagian Besar Dicapai), *Partially Achieved* (Sebagian Dicapai), dan *Not Achieved* (Tidak Tercapai).

Suatu proses pengembangan perangkat lunak dinyatakan telah mencapai tingkat kematangan level tersebut apabila seluruh KPA-nya dilakukan sepenuhnya yaitu nilai KPA Rating mencapai 86% atau lebih. Rekomendasi perbaikan dilakukan berdasarkan hasil penilaian pada masing-masing tingkat kematangan, selanjutnya dilakukan identifikasi area perbaikan.

Saran perbaikan ditentukan berdasarkan jawaban dari kuesioner yang bernilai “Tidak”, “Sebagian”, atau “Tidak Berlaku (N/A)”. Dari setiap jawaban tersebut kemudian dipetakan dengan praktek-praktek yang seharusnya dilakukan sesuai *Best Practice Scrum Maturity Model* agar bisa mencapai tingkat kematangan yang lebih tinggi. Obyek penelitian yang ditetapkan adalah proyek pengembangan perangkat lunak yang sudah menerapkan *Scrum*.

Ada tiga proyek akan diteliti yaitu Proyek A (PA), Proyek B (PB), dan Proyek C (PC). Deskripsi dari ketiga proyek dapat dilihat pada Tabel 2 berikut ini:

Tabel 2: Deskripsi Proyek Perangkat Lunak dalam Obyek Penelitian

Proyek	Keterangan
Proyek A (PA)	Proyek pengembangan aplikasi Kepegawaian berbasis Web yang digunakan oleh Unit Pengelola Kepegawaian dengan teknologi (NET, AngularJS, SQL Server) di kerjakan oleh Scrum tim beranggotakan 6 orang.
Proyek B (PB)	Proyek pengembangan aplikasi Tata Naskah Dinas berbasis Web digunakan oleh Unit Pengelola Organisasi dan Tatalaksana dengan teknologi (NET, AngularJS, SQL Server) dikerjakan oleh Scrum tim beranggotakan 5 orang.
Proyek C (PC)	Proyek pengembangan aplikasi Office Automation berbasis Web dan Mobile digunakan oleh Unit Pengelola Organisasi dan Tatalaksana dengan teknologi (NET Core, Angular, SQL Server) dikerjakan oleh Scrum tim beranggotakan 7 orang.

Hasil dan Pembahasan

Pengumpulan data dari hasil pengisian kuesioner dan wawancara oleh *Scrum Master*, observasi langsung dan dokumentasi yang terdapat pada Tools Kolaborasi Pengembangan Perangkat Lunak yaitu Microsoft TFS (*Team Foundation Server*). Daftar pertanyaan untuk setiap tingkat kematangan Level 2 dan Level 3 pada masing-masing sasaran umum disusun mengacu pada Scrum Maturity Model (SMM) [12]. Penilaian kuesioner dihitung menggunakan rumus pada persamaan 1.

Penilaian Maturity Level 2 Sasaran Umum Basic Scrum Management

Hasil rekapitulasi jawaban pada kuesioner *Scrum Maturity Model* untuk tingkat kematangan 2 sasaran umum *Basic Scrum Management* seperti terlihat pada Tabel 3.

Berdasarkan persamaan (1) hasil perhitungan nilai KPA Rating pada sasaran umum *Basic Scrum Management* untuk Proyek A (PA) sebesar 68,75%, Proyek B (PB) sebesar 79,86% dan Proyek C (PC) sebesar 81,25%. Nilai tersebut merupakan rata-rata dari setiap sasaran khusus yang ada di dalam sasaran umum Basic Scrum Management. Rata-rata KPA Rating gabungan dari 3 proyek sebesar 81,25%. Mengacu pada kategori penafsiran nilai KPA rating, maka 81,25% masuk dalam kelompok *Largely Achieved* karena di dalam rentang nilai 51% s.d 85% sehingga dapat dikatakan bahwa sasaran umum ini “Sebagian Besar Dicapai”. Hasil rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum *Basic Scrum Management* seperti terlihat pada Tabel 4.

Penilaian Maturity Level 2 Sasaran Umum Software Requirements Engineering

Hasil rekapitulasi jawaban pada kuesioner Scrum Maturity Model untuk tingkat kematangan 2 sasaran umum Software Requirement Engineering

seperti terlihat pada Tabel 5. Berdasarkan persamaan (1) hasil perhitungan nilai KPA Rating pada sasaran umum Software Requirement Engineering untuk Proyek A (PA) sebesar 83,85%, Proyek B (PB) sebesar 90,80% dan Proyek C (PC) sebesar 93,23%. Nilai tersebut merupakan rata-rata dari setiap sasaran khusus yang ada di dalam sasaran umum *Software Requirement Engineering*. Rata-rata KPA Rating gabungan dari 3 proyek sebesar 93,23%. Mengacu pada kategori penafsiran nilai KPA rating, maka 93,23% masuk dalam kelompok *Fully Achieved* karena di dalam rentang nilai 86% s.d 100% sehingga dapat dikatakan bahwa sasaran umum ini “Tercapai Sepenuhnya”.

Tabel 3: Rekapitulasi jawaban kuesioner SMM sasaran umum Basic Scrum Management

Sasaran Khusus	Praktek	Nama Proyek		
		PA	PB	PC
2.1.1 A danya peran Scrum	Individu dari sisi client diasumsikan sebagai role Product Owner?	S	S	S
	Individu dari organisasi berperan sebagai Scrum Master?	Y	Y	Y
	A danya Tim dari organisasi sebagai Tim Scrum?	Y	Y	Y
	A danya Product Backlog?	Y	Y	Y
	Product Backlog diperbaharui sesuai otorisasi Product Owner?	Y	Y	Y
	A danya Sprint?	Y	Y	Y
	Sprint Backlog diperbaharui oleh tim?	S	Y	Y
2.1.2 A danya Scrum Artifak	Sprint Backlog diperbaharui sepengetahuan anggota tim?	Y	Y	Y
	A danya Release Bumdown Chart?	S	Y	S
	Release Burndown Chart diperbaharui sesuai kemajuan proyek?	S	S	S
	A danya Sprint Bumdown Chart?	Y	Y	Y
	Sprint Bumdown Chart diperbaharui sesuai kemajuan proyek?	Y	S	Y
	Release Planning Meeting minimal dihadiri oleh Scrum Master dan Product Owner?	S	S	S
	Sprint Planning Meeting dilakukan tepat sekali setiap Sprint?	Y	Y	Y
2.1.3 A danya Scrum Meeting dan diparticipasi	Sprint Planning Meeting dihadiri oleh Scrum Master, Product Owner dan Tim?	Y	Y	Y
	Daily Scrum Meeting diadakan tepat sekali setiap hari kerja (hari adanya Sprint)?	Y	Y	Y
	Daily Scrum Meeting sekurangnya dihadiri oleh Scrum Master dan tim?	Y	Y	Y
	Sprint Review Meeting diadakan tepat sekali setiap Sprint?	Y	Y	Y
	Sprint Review Meeting dihadiri oleh pada pengambil keputusan, Stakeholder, Scrum Master dan tim?	S	S	S
	Sprint Retrospective Meeting diadakan sekali per-Sprint?	S	S	S
	Sprint Retrospective Meeting minimal dihadiri oleh Scrum Master dan tim. Sebagai tambahan, Product Owner dapat menghadiri Meeting?	S	S	S
2.1.4 Sprint Dilaksanakan dengan benar	Software development proyek dimulai dengan release plan Meeting, jika ada?	S	S	S
	Siklus dari Sprint?	Y	Y	Y

Tabel 4: Rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum *Basic Scrum Management*

N	Sasaran Khusus	KPA Rating %			Rata-rata
		PA	PB	PC	
1	Ada <i>Scrum Role</i>	83.33	83.33	83.33	83.33
2	Ada <i>Scrum Artifact</i>	83.33	88.89	88.89	87.04
3	Ada <i>Scrum Meeting</i> yang diparticipasi	77.78	77.78	77.78	77.78
4	<i>Sprint</i> dilaksanakan dengan benar	75.00	75.00	75.00	75.00
Rata-Rata		68.75	79.86	81.25	81.25
Interpretasi		<i>Largely Achieved (51 % s.d 85%)</i>			

Keterangan: PA: Proyek A, PB: Proyek B, PC: Proyek C

Tabel 5: Rekapitulasi jawaban kuesioner SMM sasaran umum *Software Requirement Engineering*

Sasaran Khusus	Praktek	Nama Proyek		
		PA	PB	PC
2.2.1 Definisi Product Owner jelas	Product Owner mempunyai kontak langsung dengan tim <i>Scrum</i> ?	Y	Y	Y
	Product Owner memiliki kontak langsung dengan stakeholder di sisi Client?	Y	Y	Y
	Product Owner bertanggung jawab untuk menentukan prioritas?	Y	Y	Y
	Product Owner memiliki pengetahuan untuk mendefinisikan prioritas?	Y	Y	Y
	Product Owner memiliki visi akan produk?	Y	Y	Y
2.2.2 Manajemen Product Backlog	Product Backlog item Sesuai dengan visi produk Product Owner?	Y	Y	Y
	Product Backlog items diprioritaskan oleh Product Owner?	Y	Y	Y
	Product Backlog items teratas didefinisikan sesuai nilai bisnis?	Y	Y	Y
2.2.3 Sprint Planning Meeting yang sukses	Product Owner menyajikan Product Backlog terbaru?	S	S	S
	Product Owner bekerjasama secara aktif?	Y	Y	Y
	Semua tim bekerja sama secara aktif?	Y	Y	Y
	Hasil disepakati pada saat perencanaan <i>Sprint</i> ?	Y	Y	Y
	Product Owner cukup puas dengan prioritas yang disetujui?	Y	Y	Y
	Seluruh tim percaya bahwa rencana yang disusun dapat tercapai?	S	Y	Y

Hasil rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Software Requirement Engineering seperti terlihat pada Tabel 6.

Penilaian Maturity Level 3 Sasaran Umum Customer Relationship Management

Hasil rekapitulasi jawaban pada kuesioner *Scrum Maturity Model* untuk tingkat kematangan 3 sasaran umum *Customer Relationship Management* seperti terlihat pada Tabel 7.

Tabel 6: Rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Software Requirements

N	Sasaran Khusus	KPA Rating %			Rata-rata
		PA	PB	PC	
1	Definisi Product Owner jelas	100	100	100	100
2	Manajemen Product Backlog	100	100	100	100
3	Sprint Planning Meeting yang sukses	83.33	91.67	91.67	88.89
Rata-Rata		83.85	90.80	93.23	93.23
Interpretasi		<i>Fully Achieved (86% s.d 100%)</i>			

Keterangan: PA: Proyek A, PB: Proyek B, PC: Proyek C

Berdasarkan persamaan (1) hasil perhitungan nilai KPA Rating pada sasaran umum Customer Relationship Management untuk Proyek A (PA) sebesar 62,50%, Proyek B (PB) sebesar 81,94% dan Proyek C (PC) sebesar 76,39%. Nilai tersebut merupakan rata-rata dari setiap sasaran khusus yang ada di dalam sasaran umum Customer Relationship Management. Rata-rata KPA Rating gabungan dari 3 proyek sebesar 73,61%. Mengacu pada kategori penafsiran nilai KPA rating, maka 73,61% masuk dalam kelompok *Largely Achieved* karena di dalam rentang nilai 51% s.d 85% sehingga dapat dikatakan bahwa sasaran umum ini “Sebagian Besar Dicapai”.

Tabel 7: Rekapitulasi jawaban kuesioner SMM sasaran umum *Customer Relationship Management*

Sasaran Khusus	Praktek	Nama Proyek		
		PA	PB	PC
3.1.1 Ada definisi “Selesai”	Definisi dari “Selesai” disetujui oleh semua anggota tim?	S	S	S
	Definisi dari Selesai dicapai pada setiap iterasi?	S	Y	S
	Tim menghormati definisi dari “Selesai”?	S	Y	Y
3.1.2 Product Owner tersedia	Product Owner selalu hadir untuk tim?	Y	Y	Y
	Product Owner mudah dihubungi?	Y	Y	Y
3.1.3 Sprint Review Meeting yang sukses	Demonstrasi Aplikasi yang Berjalan?	T	S	S
	Demonstrates Aplikasi yang ditest?	S	S	S
	Product Owner dan stakeholder yang lain memberikan umpan balik?	S	S	S
	Product Owner dan stakeholder menerima aplikasi yang disajikan?	S	Y	Y

Keterangan: S: Sebagian, Y: Ya, T: Tidak

Hasil rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Customer Relationship Management seperti terlihat pada Tabel 8.

Tabel 8: Rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Customer Relationship Management

N	Sasaran Khusus	KPA Rating %			Rata-rata
		PA	PB	PC	
1	Ada definisi "Selesai"	50.00	83.33	66.67	66.67
2	Product Owner tersedia	100	100	100	100
3	Sprint Review Meeting yang sukses	37.50	62.50	62.50	54.17
Rata-Rata		62.50	81.94	76.39	73.61
Interpretasi		<i>Largely Achieved (51% s.d 85%)</i>			

Keterangan: PA: Proyek A, PB: Proyek B, PC: Proyek C

Penilaian Maturity Level 3 Sasaran Umum Iteration Management

Hasil rekapitulasi jawaban pada kuesioner Scrum Maturity Model untuk tingkat kematangan 3 sasaran umum Iteration Management seperti terlihat pada Tabel 9.

Tabel 9: Rekapitulasi jawaban kuesioner SMM sasaran umum Iteration Management

Sasaran Khusus	Praktek	Nama Proyek		
		PA	PB	PC
3.2.1 Pengelolan Sprint Backlog	Sprint Backlog mudah dilihat?	Y	Y	Y
	Sprint Backlog diperbaharui secara harian?	S	S	S
	Sprint Backlog hanya tersedia untuk tim?	S	Y	Y
	Sprint Backlog items dibagi menjadi tugas-tugas?	Y	Y	Y
	Seluruh Anggota tim mengestimasi?	S	S	S
	Seluruh Sprint Backlog item diestimasi?	S	S	S
	Seluruh tugas diestimasi?	S	S	S
3.2.2 Iterasi direncanakan	Remaining effort untuk setiap task diperbaharui setiap hari?	S	S	S
	Sprint berakhir sesuai jadwal?	S	S	S
	Daily Scrum dilakukan setiap hari ketika Sprint berlangsung?	S	Y	S
	Pada saat Daily Scrum Meeting diidentifikasi permasalahan dan hambatan?	Y	Y	Y
	Sprint yang tidak dapat tercapai sesuai jadwal harus dibatalkan sesegera mungkin?	T	Y	T
	Tim tidak diganggu oleh pihak luar?	T	S	S
	Tim mengirimkan hasil yang dijanjikan?	S	Y	S
3.2.3 Daily Scrum yang sukses	Batasan waktu 15 menit untuk Meeting?	Y	S	S
	Setiap anggota tim mengetahui apa yang dikerjakan oleh anggota lain?	Y	Y	Y
3.2.4 Velocity yang tenkur	Sprint Burndown Chart mudah dilihat?	S	S	S
	Sprint Burndown Chart diperbaharui harian?	S	S	S
	Scrum Master melakukan analisis secara berkala terhadap kemajuan Sprint?	S	S	S

Berdasarkan persamaan (1) hasil perhitungan nilai KPA Rating pada sasaran umum Iteration Management untuk Proyek A (PA) sebesar 63,54%, Proyek B (PB) sebesar 69,27% dan Proyek C (PC) sebesar 60,94%. Nilai tersebut merupakan rata-rata dari setiap sasaran khusus yang ada di

dalam sasaran umum Customer Iteration Management. Rata-rata KPA Rating gabungan dari 3 proyek sebesar 64,58%. Mengacu pada kategori penafsiran nilai KPA rating, maka 64,58% masuk dalam kelompok Largely Achieved karena di dalam rentang nilai 51% s.d 85% sehingga dapat dikatakan bahwa sasaran umum ini "Sebagian Besar Dicapai". Hasil rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Iteration Management seperti terlihat pada Tabel 10.

Tabel 10: Rekapitulasi KPA Rating untuk setiap sasaran khusus pada sasaran umum Iteration Management

N	Sasaran Khusus	KPA Rating %			Rata-rata
		PA	PB	PC	
1	Pengelolaan Sprint Backlog	62.50	68.75	68.75	66.67
2	Iterasi direncanakan	41.67	83.33	50	58.33
3	Daily Scrum yang sukses	100	75	75	83.33
4	Velocity yang terukur	50	50	50	50.00
Rata-Rata		63.54	69.27	60.94	64.58
Interpretasi		<i>Largely Achieved (51% s.d 85%)</i>			

Keterangan: PA: Proyek A, PB: Proyek B, PC: Proyek C

Tabel 11: Hasil Rekapitulasi KPA Rating untuk setiap sasaran umum pada tingkat kematangan 2 dan 3.

Sasaran Umum	KPA Rating %			Avg	Ket
	PA	PB	PC		
L2.Basic Scrum Management	68.75	79.86	81.25	81.25	Largely Achieved
L2.Software Requirements Engineering	83.85	90.80	93.23	93.23	Fully Achieved
L3.Customer Relationship Management	62.50	81.94	76.39	73.61	Largely Achieved
L3.Iteration Management	63.54	69.27	60.94	64.58	Largely Achieved

Keterangan: PA: Proyek A, PB: Proyek B, PC: Proyek C

Analisa Data

Setelah dilakukan penilaian KPA Rating untuk masing-masing sasaran umum dan sasaran khusus dari setiap proyek yang dijadikan objek penelitian, selanjutnya dilakukan rekapitulasi terhadap nilai KPA Rating untuk memperoleh nilai rata-rata dari seluruh proyek dan ditetapkan tingkat kematangannya. Nilai KPA Rating yang diperoleh pada setiap sasaran umum akan diinterpretasikan kedalam 4 kategori untuk menentukan apakah sebuah sasaran umum memenuhi kriteria Fully Achieved (Tercapai Sepenuhnya), Largely Achieved (Sebagian Besar Dicapai), Partially Achieved (Sebagian Dicapai), atau Not Achieved (Tidak Tercapai). Tingkat kematangan secara keseluruhan akan

dinyatakan telah mencapai level tersebut apabila seluruh KPA dari sasaran umum tercapai sepenuhnya (Nilai KPA Rating mencapai 86% atau lebih). Berdasarkan hasil perhitungan rata-rata KPA Rating dari seluruh sasaran umum yang telah dinilai pada tingkat 2 dan 3, didapatkan hasil rekapitulasi nilai KPA Rating seperti terlihat pada Tabel 11.

Hasil pengukuran tingkat kematangan dapat disimpulkan bahwa manajemen proyek pengembangan perangkat lunak Unit Pusat TIK berada pada tingkat kematangan 2, dan belum mencapai level 3 karena nilai rata-rata KPA Rating sasaran umum di level 3 masih dibawah 86%, sehingga masih diperlukan beberapa perbaikan pada proses pengembangan perangkat lunak agar bisa mencapai tingkat kematangan yang lebih tinggi.

Tabel 12: Hasil pemetaan rekomendasi saran perbaikan untuk tingkat kematangan 3.

Sasaran Umum	Sasaran Khusus	Sasaran Perbaikan
3.1 Customer Relationship Management	Ada definisi "Selesai"	-Definisi dari "Selesai" harus disetujui oleh semua anggota tim proyek sehingga memiliki standar yang sama. -Setiap proyek harus bisa mencapai definisi dari Selesai pada setiap sprint.
	Sprint Review Meeting yang sukses	-Setiap Sprint Review harus dapat di demokan aplikasi yang telah berjalan. -Setiap Sprint Review harus dapat di demokan aplikasi yang telah diuji. -Setiap kegiatan Sprint Review Meeting, Product Owner dan stakeholder lain selalu memberikan umpan balik
	Iterasi direncanakan	-Sprint Backlog harus selalu diperbaharui tiap hari. -Seluruh Sprint Backlog item harus diestimasi -Seluruh Task/tugas harus diestimasi -Remaining effort untuk setiap task harus diperbaharui setiap hari setiap ada perkembangan progress.
3.2 Iteration Management	Pengelolaan Sprint Backlog	-Setiap Sprint harus berakhir sesuai jadwal. -Daily Scrum harus dilakukan setiap hari secara konsisten di tempat dan jam yang sama ketika Sprint berlangsung -Sprint yang tidak dapat tercapai sesuai jadwal harus dibatalkan sesegera mungkin -Tim dijaga tetap fokus dan tidak diganggu oleh pihak luar. Jalus komunikasi hanya melalui product owner atau scrum master. -Tim harus mengirimkan hasil yang dijanjikan sesuai jadwal.
	Daily Scrum yang sukses	-Dilakukan pembatasan waktu maksimal 15 menit untuk Daily Scrum Meeting.
	Velocity yang terukur	-Sprint Burndown Chart dibuat dan dapat diakses oleh semua anggota tim -Sprint Burndown Chart diperbarui setiap hari -Scrum Master melakukan analisis rutin terhadap progres Sprint

Rekomendasi Saran Perbaikan

Saran perbaikan untuk mencapai tingkat kematangan yang lebih tinggi didapatkan dengan melakukan identifikasi praktek-praktek apa saja yang harus dilakukan pada setiap sasaran umum dimana KPA Rating belum tercapai sepenuhnya. Pemilihan praktek didasarkan pada hasil penilaian kuesioner terhadap butir-butir pertanyaan yang memiliki nilai "Tidak", "Sebagian", atau "Tidak Berlaku". Daftar rekomendasi sasaran perbaikan untuk tingkat kematangan 3 ditunjukkan pada Tabel 12.

Penutup

Penelitian atas analisis dan evaluasi tingkat kematangan manajemen proyek pengembangan perangkat lunak berbasis *scrum* menggunakan *Scrum Maturity Model* pada Unit Pusat TIK Instansi Pemerintah telah berhasil dilakukan, dan dapat diambil kesimpulan bahwa Unit Pusat TIK Instansi Pemerintah telah mencapai tingkat kematangan 2 (*Managed*) pada *Scrum Maturity Model*. Pada tingkat kematangan 3 (*Defined*), instansi ini belum sepenuhnya memenuhi dan mendapatkan interpretasi *Largely Achieved* (Sebagian Besar Tercapai).

Rekomendasi perbaikan yang disarankan ditetapkan untuk mencapai tingkat kematangan yang lebih tinggi yaitu Level 3 dengan mengacu pada praktek-praktek yang belum dijalankan sepenuhnya. Pada tingkat kematangan 3 ini disarankan untuk memperbaiki praktek-praktek pada sasaran khusus yang terkait dengan manajemen hubungan dengan pelanggan dan manajemen iterasi dalam proses pengembangan perangkat lunak.

Saran yang dapat diberikan untuk Unit Pusat TIK Instansi Pemerintah adalah untuk dapat menindaklanjuti secara bertahap rekomendasi perbaikan atas praktek-praktek dalam pengembangan perangkat lunak berbasis *Scrum*. Pemahaman tentang kerangka kerja *Scrum* beserta seluruh praktek didalamnya perlu ditingkatkan untuk anggota tim dan dapat di usulan dalam program pelatihan baik formal maupun nonformal, dan perlunya penerapan secara disiplin dan konsisten untuk semua proyek pengembangan yang menggunakan kerangka kerja *Scrum* agar mendapatkan hasil yang maksimal. Selain itu kerangka kerja *Scrum* dapat disesuaikan dan dikombinasikan dengan manajemen proyek lain sesuai dengan praktik dan tata kelola yang berlaku pada Instansi.

Untuk penelitian selanjutnya, berikut saran-saran yang dapat diberikan dapat dikembangkan dengan tidak hanya terkait penilaian proses *Scrum* namun juga hasil dari implementasi *Scrum*, sehingga dapat dilihat tingkat efektifitas penerapan *Scrum* dalam meningkatkan kualitas perangkat lunak dibandingkan dengan metodologi yang digunakan sebelumnya, selain itu penelitian dapat dilakukan untuk jumlah proyek yang lebih banyak se-

hingga dapat menggambarkan tingkat kematangan yang lebih baik dan menyeluruh untuk organisasi tersebut.

Daftar Pustaka

- [1] L. M. Applegate, R. D. Austin and D. Soule, "Corporate Information Strategy and Management", 8th ed., New York: McGraw-Hill, 2008.
- [2] S. Kathy, "Information Technology Project Management", 6th ed., Canada: Thomson Course Technology, 2006, p. 7.
- [3] K. Schwaber and M. Beedle, "Agile Software Development with Scrum", United States: Prentice Hall, 2002.
- [4] K. Schwaber and J. Sutherland, "Scrum Guide", Scrum Alliance, October 2011.
- [5] M. Ashraf and N. Ali, "Impact Agile Project Management: Identification And Analysis Of Problems In Scrum Implementation", VAWKUM Transactions on Computer Sciences, vol. 2, no. 2, pp. 01-08, November 2013.
- [6] R. Akif and M. Hammad, "Issue and Challenges In Scrum Implementation", International Journal of Scientific and Engineering Research, vol. 3, no. 8, pp. 1-4, Agustus 2012.
- [7] J. Wan, Y. Zhu and M. Zeng, "Case Study on Critical Success Factors of Running Scrum", Journal of Software Engineering and Application, vol. 6, pp. 59-64, February 2013.
- [8] A. Nuraminah, "Analisis Tingkat Kematangan Manajemen Proyek Pengembangan Perangkat Lunak Menggunakan Scrum Maturity Model: Studi Kasus PT. XYZ", STIMIK ESQ, vol. 2, no. 2, pp. 1-4, Agustus 2016.
- [9] C. Patel and R. Muthu, "Agile Maturity Model (AMM): A Software Process Improvement framework for Agile Software Development Practices", International Journal of Software Engineering., vol. 2, no. 1, 2009.
- [10] S. Dutta, M. Lee, L. Van Wassenhove, "Software engineering in Europe: A Study of Best Practices", IEEE Software, Volume: 16, Issue: 3, pp. 82-90, 1999.
- [11] Alexandre Yin, Soraia Figueiredo, M. Silva, "Scrum Maturity Model", Computer Science, Corpus ID: 59785048, 2011.
- [12] A. Yin, S. Figueiredo and M. Silva, "Scrum Maturity Model: Validation for IT organizations' roadmap to develop software centered on the client role", Barcelona: International Conference on Software Engineering Advances (ICSEA), 2011.