

EVALUASI PROSES PENGEMBANGAN PERANGKAT LUNAK PADA *VIRTUAL TEAM DEVELOPMENT* MENGGUNAKAN CMMI Versi 1.3

Wahyu Widodo

Jurusan Magister Teknik Informatika Fakultas Teknik Industri

Universitas Islam Indonesia

Yogyakarta, Indonesia

Email : wahyusoft@yahoo.com

Abstrak

Proses pengembangan perangkat lunak menggunakan outsourcing virtual team development (VTD) memiliki banyak kelebihan diantaranya biaya produksi lebih murah, tersedia banyak tenaga ahli, dan waktu lebih kompetitif. Namun demikian banyak temuan resiko yang berdampak pada kualitas produk perangkat lunak yang dihasilkan. Tujuan dari penulisan makalah ini adalah untuk mengetahui tingkat kematangan dan tingkat kapabilitas proses pengembangan perangkat lunak pada VTD. Model yang digunakan untuk evaluasi dengan menggunakan CMMI for development versi 1.3. Metode yang digunakan dalam penelitian ini menggunakan representasi continuous dengan memilih praktik-praktik spesifik tertentu pada area proses CMMI level 2 dan level 3. Pengambilan data dilakukan dengan kuisioner online pada responden yang merupakan agensi kontraktor pada situs freelance market place upwork.com. Hasil dari penelitian ini menjadi strategi bagi unit organisasi atau perusahaan yang menggunakan jasa outsourcing VTD dalam mengembangkan perangkat lunak yang berorientasi pada kualitas produk perangkat lunak yang dihasilkan.

Kata kunci : *Evaluasi, Perangkat Lunak, Virtual Team Development, CMMI.*

1. PENDAHULUAN

Pertumbuhan persaingan bisnis perusahaan atau organisasi membutuhkan dukungan dari industri perangkat lunak yang dapat menyesuaikan kebutuhan pengguna dalam menunjang semua aktifitas bisnis. Perangkat lunak akan terus berevolusi seperti penambahan kompleksitas fitur, perubahan proses bisnis, perbaikan error, penerapan teknologi terbaru, sehingga kualitas sistem perangkat lunak yang tinggi menjadi kebutuhan vital [1]. Perusahaan atau organisasi terutama di negara maju mengembangkan perangkat lunak menggunakan cara outsourcing kepada *virtual team development* (VTD) [1]. Beberapa situs yang cukup bereputasi menyediakan jasa outsourcing dalam mengembangkan perangkat lunak antara lain : elance.com, upwork.com, dan freelancer.com. Dengan cara outsourcing, perusahaan mendapatkan keuntungan diantaranya biaya produksi pengembangan jauh lebih murah, tersedia banyak tenaga ahli, waktu lebih kompetitif, dan dapat fokus pada bisnis utama perusahaan [2]. Namun demikian terdapat banyak kekurangan dan hambatan dalam menggunakan outsourcing VTD seperti perbedaan latar belakang budaya, perbedaan bahasa, minimnya pengawasan, lemahnya koordinasi dengan tim pengembang, kurangnya intensitas pertemuan antara VTD dengan *client* yang berakibat kesalahpahaman dengan kebutuhan yang diinginkan yang dipastikan akan berdampak pada kualitas produk perangkat lunak yang dihasilkan [3].

Secara umum menentukan kualitas produk perangkat lunak dibagi menjadi dua pendekatan yaitu pendekatan proses produksi dan pendekatan hasil pengujian produk. Pendekatan proses produksi memegang peranan signifikan dalam menghasilkan suatu produk perangkat lunak berkualitas yang dilakukan oleh tim pengembang [4]. Sangat penting bagi perusahaan atau organisasi memahami setiap tahapan proses (*process area*) dalam pengembangan perangkat lunak sehingga dapat menggunakan outsourcing VTD yang tetap berorientasi pada kualitas produk. Terdapat beberapa model yang dirancang fokus

pada evaluasi perbaikan proses, menilai level kematangan dan level kapabilitas tim pengembang perangkat lunak seperti Six Sigma, ISO 9001, Bootstrap, Kipi, dan CMMI *for Development*. CMMI *for development* merupakan model yang cukup populer digunakan pada perbaikan proses pengembangan perangkat lunak [5, 6]. Model CMMI dapat digunakan untuk mengetahui level kapabilitas sekaligus level kematangan sebuah tim pengembang perangkat lunak yang disebut sebagai *equivalent staging*. Dengan demikian target area proses yang akan digunakan untuk evaluasi dimulai dari level 2 dan level 3. Pada level kematangan 2 merupakan tahap awal dimana seluruh proses pengembangan perangkat lunak oleh VTD seharusnya telah terencana, terukur, dan terkontrol dengan baik. Sedangkan area proses pada level 3, VTD seharusnya telah memiliki standarisasi prosedur, *tools*, dan metode kerja sehingga ada jaminan konsistensi dalam pelaksanaan proyek.

Pada penelitian ini, responden yang akan digunakan adalah agen kontraktor pada upwork.com. Dengan memahami tahapan pada area proses pengembangan perangkat lunak, diharapkan organisasi atau perusahaan dapat memiliki strategi pada setiap tahapan proses pengembangan perangkat lunak skema outsourcing VTD yang berorientasi pada kualitas produk.

2. LANDASAN TEORI

Perbaikan proses pengembangan perangkat lunak mencakup serangkaian aktivitas menuju yang lebih baik, sebagai akibatnya kualitas perangkat lunak menjadi lebih tinggi. Kualitas perangkat lunak bukan hanya berdampak pada kepuasan pelanggan (*user satisfaction*) saja, tetapi dampaknya justru lebih besar ada pada tim pengembang [7]. Dapat dianalogikan bahwa perangkat lunak adalah sebuah makhluk yang terus berevolusi sesuai dengan kebutuhan pengguna. Penambahan fitur, pengurangan fitur, perubahan proses bisnis yang akan berakibat pada *reengineering*. Apabila kualitas perangkat lunak tinggi, maka akan lebih mudah bagi para engineer untuk melakukan perubahan. Perbaikan proses adalah “Penerapan praktik-praktik secara konsisten dan mengubah praktik-praktik yang menjadi penyebab masalah” [8]. Tujuan dari perbaikan proses pengembangan perangkat lunak adalah untuk memahami proses yang ada saat ini, memperkenalkan perubahan proses untuk meningkatkan kualitas, mengurangi biaya, atau mempercepat jadwal, tuntutan industri terhadap peningkatan kualitas produksinya, dan untuk mengurangi resiko cacat perangkat lunak yang diakibatkan dari kesalahan yang terjadi saat proses pengembangan.

A. Capability Maturity Model Integration

The Capability Maturity Model Integration memainkan peran penting dalam upaya perbaikan proses perangkat lunak. Proses ini dikembangkan oleh Software Engineering Institute [9] di Carnegie Mellon University pada tahun 1986. Tujuannya adalah untuk meningkatkan secara bertahap (*staged*) dan berkelanjutan (*continuous*) penerapan teknologi perangkat lunak pada level organisasi. Model ini memberikan panduan bagi organisasi untuk memilih strategi perbaikan proses perangkat lunak dengan memfasilitasi penentuan kemampuan saat ini dan mengidentifikasi isu-isu kritis yang perlu segera diperbaiki [10].

Dalam mengimplementasi dan menafsirkan model CMMI dapat ditempuh dua representasi yang berbeda yaitu *continouos* atau *staged*. Pemilihan representasi dipengaruhi oleh tiga hal yaitu : faktor bisnis, faktor budaya, dan juga faktor pengalaman dalam penggunaan model-model sebelumnya. Pendekatan *continouos* ditempuh bila upaya peningkatan fokus pada area proses tertentu saja, sedangkan pendekatan *staged* dirancang sebagai suatu tahapan-tahapan baku bagi peningkatan sekumpulan proses dalam sebuah organisasi. Pada penelitian ini akan menggunakan representasi *continouos*.

B. Standar Evaluasi Proses CMMI

Dalam penerapan CMMI, dilakukan pengukuran untuk mengevaluasi hasil pencapaian penerapannya. Proses evaluasi ini disebut dengan istilah *appraisal*. Metode standar yang digunakan dalam melakukan appraisal, yang dikeluarkan oleh SEI adalah *Standard CMMI Appraisal Method for Process Improvement* [11]. SEI menggolongkan

metode appraisal menjadi 3 kelas, masing-masing dengan dengan tingkat kelengkapan, besar usaha pelaksanaan dan requirement yang berbeda-beda. Kelas tersebut adalah class A, class B dan class C. Pada Tabel 1 berikut ini penjabaran detail mengenai perbandingan 3 kelas appraisal method.

Tabel 1. Perbandingan Kelas *Appraisal Method*

	Class A	Class B	Class C
Ukuran tim evaluasi	8-10	3-4	1-2
Waktu evaluasi	10 hari	3-4 hari	1-2 hari
Minimal jumlah sumber data	3	2	1
Keandalan dan keberhasilan	Tinggi	Sedang	Rendah
Usaha dan biaya yang diperlukan	Tinggi	Sedang	Rendah
Wawancara langsung	Ya	Ya	Tidak
SCAMPI	A	B	C

(Sumber : Kneuper, 2009)

Walaupun hasil dari appraisal kelas C tidak sama tingkat keandalannya dan kebenarannya dibandingkan dengan kelas A dan B, namun untuk tujuan-tujuan tertentu, tingkat keandalannya telah cukup baik untuk proses evaluasi.

3. METODE PENELITIAN

3.1. Studi Pustaka

Studi pustaka merupakan kegiatan untuk mempelajari literatur-literatur dan teori yang mendukung dalam melakukan penelitian ini. Studi literatur melalui paper, jurnal, artikel, buku, website yang terkait dengan perbaikan proses pengembangan perangkat lunak outsourcing pada tim terdistribusi atau VTD, resiko-resiko dan pencegahan pada proyek *outsourcing VTD*, dan karakteristik pengembangan perangkat lunak VTD. Pada tahap ini akan dilakukan pembuatan proposal penelitian.

3.2. Menginterpretasikan CMMI pada Kasus Outsourcing VTD

Pada tahapan ini meneliti tentang implementasi CMMI pada pengembangan perangkat lunak *outsourcing VTD*. Model CMMI *for Development* versi 1.3 memungkinkan organisasi dan manajemen senior untuk menginterpretasikan pada situasi yang berbeda sesuai dengan spesifikasi kebutuhan, tujuan organisasi dan target proyek yang akan dicapai. Sub proses pada tahapan ini antara lain :

3.2.1. Memilih Representasi Continuous

Pada penelitian ini, penggunaan model CMMI menggunakan representasi *continuous* untuk memperoleh fleksibilitas dalam menentukan area proses yang akan dievaluasi.

3.2.2. Penentuan Sampel Proyek

Sampel Proyek memberikan pandangan ragam cara kerja yang dilakukan agen kontraktor dalam menyelesaikan proyeknya. Sampel proyek yang sudah selesai dilakukan oleh agen kontraktor dapat dilihat pada bagian halaman profile *work history* pada upwork.com. Dengan menentukan sampel proyek, maka akan diketahui jenis pekerjaan apakah *project development*, *project management*, atau *change request*.

3.2.3. Menentukan Area Proses Evaluasi

Pada siklus pengembangan perangkat lunak dengan pendekatan metode agile dengan menggunakan 10 area proses yang berada pada level 2 dan 3 (Mellon, 2010). Daftar area proses tersebut seperti yang terlihat pada Tabel 2 berikut.

Tabel 2. Domain Area Proses yang akan digunakan

Level	Kategori	Nama Area Proses	Singkatan
2	<i>Support</i>	Manajemen konfigurasi	CM
	<i>Project Management</i>	Pengawasan dan pengendalian proyek	PMC
	<i>Project Management</i>	Perencanaan proyek	PP
	<i>Support</i>	Jaminan proses dan kualitas produk	PPQA
	<i>Engineering</i>	Manajemen persyaratan	REQM
3	<i>Engineering</i>	Integrasi produk	PI
	<i>Engineering</i>	Kebutuhan pengembangan	RD
	<i>Project Management</i>	Manajemen resiko	RSKM
	<i>Engineering</i>	Solusi teknis	TS
	<i>Engineering</i>	Verifikasi	VER

3.2.4. Mengidentifikasi Praktik Spesifik yang Relevan pada Area Proses

Setelah menentukan area proses, pada tahapan ini menentukan praktik-praktik spesifik yang relevan diterapkan pada proses pengembangan perangkat lunak VTD untuk digunakan sebagai indikator evaluasi. Kemudian indikator tersebut dikonversi kedalam kalimat tanya sebagai bahan pengambilan data kuisioner. Dari 10 area proses tersebut diekstrak menjadi 19 jumlah pertanyaan yang akan digunakan pada pengambilan data responden. Pada Tabel 3 rekapitulasi jumlah pertanyaan setiap area proses.

Tabel 3. Rekapitulasi Daftar Pertanyaan

Area Proses	Jumlah Pertanyaan
CM	2
PMC	2
PP	2
PPQA	3
REQM	2
PI	2
RD	1
RSKM	2
TS	1
VER	2
Total Pertanyaan	19

Karena keterbatasan dalam penulisan, maka pada makalah ini penulis menampilkan contoh pertanyaan pada satu area proses manajemen konfigurasi saja. Tujuan dari area proses ini adalah untuk menjaga integritas produk dan mendukung frekuensi perubahan, frekuensi pengembangan yang didukung oleh personal, tim pengembang, ataupun even *pair-programming*. Pada Tabel 4 terdapat dua praktik spesifik yang akan digunakan sebagai proses evaluasi.

Tabel 4. Daftar Praktik dan Pertanyaan pada Area Proses CM

Kode	Praktik Spesifik	Pertanyaan
CM-SP1.1	Mengidentifikasi konfigurasi produk.	Apakah manajer proyek menerapkan standarisasi penggunaan tools, design interface, dan penjelasan proses bisnis software?
CM-SP2.1	Melacak perubahan permintaan.	Apabila terjadi perubahan permintaan, apakah terdapat kesepakatan antara tim dengan pelanggan tentang estimasi waktu, prioritas, biaya, dan resiko dampak perubahan?

3.2.5. Pengumpulan Data

Pengumpulan data dengan menggunakan kuisioner online pada 30 agen kontraktor upwork, berdasarkan metode SCAMPI C reponden yang dibutuhkan setiap 1 agen kontraktor yaitu diwakili minimal 1 freelancer yang memiliki kriteria sebagai berikut :

1. Freelancer pada kategori *Web, Mobile & Software Development*.
2. Telah memiliki nilai *feedback* minimal 4,5.

Pada Tabel 5 responden akan menilai setiap proses apakah diimplementasikan semua, sebagian besar, sebagian atau tidak diimplementasikan sama sekali [11]. Kriteria penilaian *specific practice* (SP) terdapat pada tabel berikut ini :

Tabel 5. Penilaian *Specific Practice*

Kode	Nama	Point Bobot
FI	Diimplementasikan penuh	3
LI	Diimplementasikan sebagian besar	2
PI	Diimplementasikan sebagian	1
NI	Tidak diimplementasikan	0

3.2.6. Identifikasi Peluang Perbaikan Proses Pengembangan Perangkat Lunak

Pada tahapan ini dilakukan dengan mengidentifikasi dari jawaban kuisioner responden untuk menemukan celah proses yang harus diperbaiki. Celah proses diketahui dari hasil nilai rata-rata responden tiap area proses dengan menggunakan rumus sebagai berikut :

$$\text{Nilai Pencapaian Area Proses} = \frac{\text{Jumlah Point Bobot Jawaban}}{\text{Jumlah Responden}}$$

Setelah diketahui angka nilai pencapaian tiap area proses, selanjutnya menentukan area proses tersebut tercapai atau tidak tercapai (S. U. Team, 2011). Pada Tabel 5 penjelasan detail dan rentang nilai yang digunakan.

Tabel 5 Penilaian *Specific Goals*

Kode	Nama	Rentang Nilai Rata-rata	Keterangan
S	Tercapai	3 – 2	Semua SP yang terkait dengan tujuan diimplementasikan penuh (FI) dan diimplementasikan sebagian besar (LI).
NS	Tidak tercapai	1,9 - 0	Agregasi kelemahan terkait dengan tujuan tidak memiliki dampak negatif yang signifikan terhadap pencapaian tujuan.

3.2.7. Rekomendasi Perbaikan Proses Pengembangan Perangkat Lunak

Pada tahapan ini, memberikan rekomendasi dan prioritas yang perbaikan pada pengembangan perangkat lunak VTD. Rekomendasi ini didasarkan pada hasil peluang perbaikan saat pengolahan data kuisisioner. Rekomendasi berupa saran agar area proses tercapai, dan contoh artefak produk yang dapat mendukung pada area proses tersebut.

4. HASIL DAN PEMBAHASAN

4.1. Pengambilan Data Responden

Data responden diambil dari agen kontraktor kategori pengembangan perangkat lunak dan *web mobile* pada situs upwork.com yang berasal dari Indonesia. Semua agen kontraktor (AK) yang dipilih tersebut telah memiliki nilai (*rating*) dan *feedback* yang baik dari *client*. Data tersebut dapat dilihat dari *history works* dan *feedback* pada situs upwork.com.

4.2. Penentuan Sampel Proyek

Pada penelitian ini, sampel proyek dilakukan dengan menganalisa secara online *history works* masing-masing pada 30 agen kontraktor untuk mengambil satu contoh data proyek yang telah selesai dilakukan dan relevan untuk dievaluasi proses pengembangannya menggunakan CMMI. Pada Tabel 6 menampilkan 5 contoh sampel proyek masing-masing agen kontraktor yang akan digunakan sebagai dasar pengisian kuisisioner.

Tabel 6. Tabel Sampel Proyek

AK	Contoh Proyek
AK 1	<i>Full Stack Independent Ruby on Rail Developer</i>
AK 2	<i>Project Manager for Website Development</i>
AK 3	<i>IOS & Android Podcasting / Radio App Dev Project</i>
AK 4	<i>Convert SlimStarter to Laravel and Expand Functionality</i>
AK 5	<i>Ongoing Site Maintenance</i>

Pada Gambar 2 aplikasi kuisisioner yang digunakan untuk pengambilan data menggunakan kwiksurveys.com.



Gambar 2. Aplikasi Kuisisioner

Hasil nilai rata-rata area proses dari 30 responden agen kontraktor terdapat pada Tabel 7 berikut ini.

Tabel 4.1. Hasil Nilai Responden

Level	Area Proses	Nilai Rata-Rata	Tercapai
2	CM	1,98	NS
2	PMC	2,48	S
2	PP	2,33	S
2	PPQA	2,24	S
2	REQM	2,23	S
3	PI	1,81	NS
3	RD	1,8	NS
3	RSKM	1,65	NS
3	TS	1,8	NS
3	VER	2,18	S

4.3. Pembahasan

Berdasarkan hasil dari nilai rata-rata area proses 30 responden agen kontraktor, diketahui bahwa masih terdapat 5 area proses yang belum tercapai antara lain CM, PI, RD, RSKM dan TS. Sebagian besar area proses yang belum tercapai ada pada level 3. Dengan demikian perlu adanya strategi meningkatkan area proses tersebut.

4.3.1. Strategi Meningkatkan Area Proses CM

Area proses pada manajemen konfigurasi sangat bermanfaat sebagai repositori informasi proyek. Dalam proses pengembangan perangkat lunak, menghasilkan banyak informasi seperti dokumentasi analisis, spesifikasi desain, panduan penggunaan, source code, hasil pengujian, dan dokumentasi teknis.

4.3.2. Strategi Meningkatkan Area Proses PI

Area proses ini masuk pada bagian kategori *engineering*. Pada proses kolaborasi pengembangan perangkat lunak setiap saat ada penambahan kode, bahkan satu data file dikerjakan lebih dari satu programmer. Penambahan fungsionalitas, penghapusan bagian kode tertentu, penataan ulang kode program (*refactoring*) selalu ada pada saat bekerja dengan tim kolaborasi. Area proses ini membantu dalam upaya *assembly* beberapa file kode menjadi satu produk yang siap dikirim ke *client* dengan beberapa strategi sebagai berikut :

- a. Membuat Perencanaan Integrasi Perangkat Lunak
Langkah awal pada area proses ini adalah membuat rencana integrasi produk. Menentukan repositori atau server yang akan digunakan untuk integrasi, perangkat lunak yang akan digunakan, prosedur pengiriman kode masing-masing anggota tim.
- b. Memastikan Kompabilitas Sistem
Dalam mengintegrasikan kode harus dipastikan setiap perubahan yang dikirim oleh anggota tim programmer tidak mengganggu sistem yang telah ada. Dalam praktik pengembangan perangkat lunak biasanya menggunakan unit testing yang dipakai sebagai validasi bahwa kode program aman untuk diintegrasikan tanpa mengganggu fungsionalitas yang lain. Tim programmer direkomendasikan menggunakan *tools* unit testing. Unit testing digunakan untuk menguji penambahan, perubahan, dan modifikasi fungsi. Contoh software unit testing yang sering digunakan yaitu PHPUnit dan CodeCaption.

c. Assembly dan Packing Produk

Pada tahap akhir, file kode dari beberapa anggota tim programmer diintegrasikan (*merger*), diverifikasi, dan divalidasi. Verifikasi dan validasi digunakan untuk memastikan kinerja, kesesuaian, dan kesiapan perangkat lunak untuk dipacking kemudian dikirim ke *client*. Pada proses verifikasi dan validasi biasanya dilakukan oleh seorang *quality assurance* (QA) untuk melakukan pengujian fungsionalitas dan kualitas perangkat lunak.

4.3.2. Strategi Meningkatkan Area Proses RD

Pada area proses ini, seorang analis sistem mengidentifikasi kebutuhan *client* kemudian diterjemahkan dalam persyaratan produk. Persyaratan produk dianalisis untuk menghasilkan solusi pada level konseptual. Contoh persyaratan produk antara lain: fitur desain perangkat lunak, kinerja perangkat lunak, dan verifikasi persyaratan. Artefak yang dihasilkan pada area proses ini berupa *data flow diagram* (DFD), *business process modeling* (BPM), atau UML untuk menggambarkan alur proses perangkat lunak. Tujuan pada area proses ini agar tim programmer lebih memahami dan mudah menerjemahkan kedalam bahasa kode.

4.3.3. Strategi Meningkatkan Area Proses RSKM

Resiko biasanya terjadi diluar perencanaan pengembangan perangkat lunak. Kegagalan pengembangan perangkat lunak bisa terjadi karena perubahan persyaratan, anggota tim *resign* mendadak, dan sebagainya. Area proses ini ditujukan untuk mengidentifikasi potensi masalah sebelum terjadi dan meminimalkan dampak resiko. Strategi yang dapat dilakukan pada area proses ini antara lain :

a. Mengidentifikasi Sumber Resiko

Pada proses pengembangan perangkat lunak VTD sumber resiko bisa berasal dari beberapa faktor seperti :

- 1) Faktor waktu : proyek tidak selesai sesuai estimasi waktu pada perencanaan.
- 2) Faktor kualitas dan fungsionalitas : Proyek tidak sesuai dengan fitur yang telah direncanakan, atau kualitas rendah saat rilis produk.
- 3) Faktor biaya : Anggaran biaya proyek jauh lebih tinggi dari anggaran yang ditetapkan.

b. Melakukan Penilaian Resiko

Setelah ditemukan sumber resiko dan dikelompokkan kedalam faktor resiko, tahapan selanjutnya dilakukan penilaian resiko. Penilaian resiko digunakan untuk menghasilkan peringkat yang akan digunakan sebagai bahan pengambilan keputusan prioritas pencegahan resiko yang akan dilakukan. Semakin tinggi tingkat resiko yang akan berpengaruh terhadap kegagalan proyek pengembangan perangkat lunak, maka semakin diprioritaskan mitigasi resikonya.

c. Pengendalian Resiko

Pengendalian resiko dapat meminimalisir potensi ancaman kegagalan proyek dengan cara melakukan pengawasan selama proses pengembangan perangkat lunak berlangsung. Pengawasan progres selama proses pengembangan pada VTD dapat dilakukan dengan menggunakan *tools* seperti atlantis jira, redmine, atau trelo, sehingga selalu ada komunikasi antara project manager dengan anggota tim.

4.3.4. Strategi Meningkatkan Area Proses TS

Dukungan dokumentasi perangkat lunak sangat dibutuhkan bagi pengguna. Isi dokumentasi antara lain, persyaratan kebutuhan hardware, petunjuk instalasi, pengaturan konfigurasi, dan penggunaan detail produk perangkat lunak. Menyediakan dokumentasi dapat dengan membuat manual book, bisa berupa file pdf, chm, video panduan, atau berupa halaman web. Dukungan solusi teknis yang baik sangat berpengaruh terhadap kepuasan pengguna terhadap produk perangkat lunak yang digunakan.

5. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Kesimpulan yang dapat diperoleh dari hasil penelitian ini adalah sebagai berikut :

- a. Model CMMI dapat digunakan untuk evaluasi pada tim pengembang perangkat lunak skala kecil dengan menggunakan representasi *continuos*.
- b. Evaluasi proses berbasis kuisioner dapat diterapkan pada VTD dengan mengkonversi praktik-praktik spesifik pada area proses yang dipilih menjadi kalimat pertanyaan.
- c. Untuk mengetahui level kapabilitas dan level kematangan, digunakan *equivalent staging* dengan dimulai dari level 2.
- d. Hasil penelitian diketahui bahwa level kematangan dan kapabilitas sebagian besar baru tercapai pada level 2.

5.2. Saran

Saran yang dapat peneliti berikan sebagai rekomendasi untuk meningkatkan proses pengembangan perangkat lunak yaitu :

- a. Dalam membudayakan dokumentasi dalam proses pengembangan perangkat lunak VTD, perlu adanya komitmen antara project manager dengan anggota tim.
- b. Perlu adanya peningkatan kualitas proses pada area konfigurasi manajemen, integrasi produk, kebutuhan pengembangan, dan manajemen resiko.
- c. Resiko dalam mengembangkan perangkat lunak pada VTD dapat diminimalisir bahkan dihilangkan apabila ada monitoring dan pengendalian selama proses pengembangan perangkat lunak.

DAFTAR PUSTAKA

- [1]. Casey, V., *Developing Trust In Virtual Software Development Teams*. Journal of Theoretical and Applied Electronic Commerce Research, 2010.
- [2]. Minhas, N.M., et al., *An Improved Framework for Requirement Change Management in Global Software Development*. Journal of Software Engineering and Applications, 2014.
- [3]. Verner, J.M., et al., *Risks and risk mitigation in global software development: A tertiary study*. Elsevier, 2014. **56**: p. 54–78.
- [4]. Unterkalmsteiner, M., et al., *Evaluation and Measurement of Software Process Improvement A Systematic Literature Review*. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2012. **38**(2).
- [5]. Mellon, C., *CMMI for Development Version 1.3*. 2010: Software Engineering Institute.
- [6]. Herbsleb, J., et al., *Benefits of CMM-Based Software Process Improvement : Initial Results*, C.M. University, Editor. 1994, Software Engineering Institute.
- [7]. Lavallée, M. and P.N. Robillard, *The Impacts of Software Process Improvement on Developers : A Systematic Review*. Software Engineering (ICSE), 2012: p. 113 - 122.
- [8]. Wiegers, K.E., *Process Improvement that Works*. 1999.
- [9]. Dakrory, M.I. and H.A. Abdou, *Virtual teams processes: a conceptualization and application*. Problems and Perspectives in Management, 2009. **7**(3).
- [10]. Fang, Y., B. Han, and W. Zhou, *Research and Analysis of CMMI Process Improvement Based on SQCS System*. TELKOMNIKA, 2012.
- [11]. Team, S.U., *Standart CMMI Appraisal Method for Process Improvement (SCAMPI) A , Version 1.3: Method Definition Document*. 2011: Carnegie Mellon University.