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

Sapridin (1695114057)

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 . Perusahaan atau organisasi terutama di negara maju mengembangkan perangkat lunak menggunakan cara outsourcing kepada v*irtual team development* (VTD).

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. 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 focus 1141 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*.

1. **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. 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 *reenginering*.

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”. 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

1. *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 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

1. *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*.

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.

1. **Hasil dan Pembahasan**
2. **Pengambilan Data Responden**

Data responden diambil dari agen kontraktor kategori pengembangan perangkat unak 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 *feeback* pada situs upwork.com.

1. **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 kuisioner.

1. **Pembahasan**
2. Strategi Meningkatkan Area Proses CM

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

1. Strategi Meningkatkan Area Proses PI

Area proses ini masuk pada bagian kategori *enginering.* Pada proses kolaborasi pengembangan perangkat lunak setiap saat ada penambahan kode, bahkan satu data file dikerjakan lebih dari satu programer. 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 :

1. 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 masingmasing anggota tim.

1. Memastikan Kompabilitas Sistem

Dalam mengintegrasikan kode harus dipastikan setiap perubahan yang dikirim oleh anggota tim programer 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.

1. Assembly dan Packing Produk

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

1. Strategi Meningkatkan Area Proses RD

Pada area proses ini, seorang analis sistem mengidentifikasi kebutuhan *client* kemudian diterjemahkan dalam persyaratan produk. Persyaratan produk dianalisis untukmenghasilkan solusi pada level konseptual. Contoh persyaratan produk antara lain: fiturdesain perangkat lunak, kinerja perangkat lunak, dan verifikasi persyaratan. Artefakyang dihasilkan pada area proses ini berupa *data flow diagram* (DFD), *business process modeling* (BPM), atau UML untuk mengambarkan alur proses perangkat lunak.

1. 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:

1. Mengidentifikasi Sumber Resiko
2. Melakukan Penilaian Resiko
3. Pengendalian Resiko
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.

1. **Kesimpulan dan saran**
2. Kesimpulan

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

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

**Daftar Pustaka**

Casey, V., *Developing Trust In Virtual Software Development Teams.* Journal of Theoretical and Applied Electronic Commerce Research, 2010.

Minhas, N.M., et al., *An Improved Framework for Requirement Change Management in Global Software Development.* Journal of Software Engineering and Applications,2014.

Verner, J.M., et al., *Risks and risk mitigation in global software development: A tertiary study.* Elsevier, 2014. **56**: p. 54–78.

Unterkalmsteiner, M., et al., *Evaluation and Measurement of Software Process Improvement A Systematic Literature Review.* IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, 2012. **38**(2).

Dakrory, M.I. and H.A. Abdou, *Virtual teams processes: a conceptualization and application.* Problems and Perspectives in Management, 2009. **7**(3).

Fang, Y., B. Han, and W. Zhou, *Research and Analysis of CMMI Process Improvement Based on SQCS System.* TELKOMNIKA, 2012.

Team, S.U., *Standart CMMI Appraisal Method for Process Improvement ( SCAMPI)* *A , Version 1.3: Method Definition Document*. 2011: Carnegie Mellon University.