

# STRATEGI PEMILIHAN KONTRAKTOR PERANGKAT LUNAK DENGAN MEMANFAATKAN PENGETAHUAN TERHADAP CAPABILITY MATURITY MODEL INTEGRATION FOR DEVELOPMENT (CMMI FOR DEV)

Bayu Adhi Tama<sup>1</sup>, Indra Silanegara<sup>2</sup>

<sup>1</sup>Fakultas Ilmu Komputer, Universitas Sriwijaya

E-mail: bayu@ilkom.unsri.ac.id

<sup>2</sup>Politeknik Negeri Jakarta

E-mail: silanegara@yahoo.com

## Abstrak

*Given out a same software development to different contractors will produce a certain different software result. This principle can be used by project manager in selecting software contractors. Therefore, a strategy in choosing the best candidate is needed. To support this activity, CMMI for Dev can be used as guidance or reference for its reader, especially for project managers and their teams about how to assess quality and productivity of activity process in a software company. This knowledge can be applied in selecting the candidate in order to work together for developing the required software. Also, the process area of CMMI for Dev promotes strategy in selecting the candidate that will develop the software in an agreement. However, due to the limitation the writer have, not all aspect can be discover and worth for further research in the future.*

**Kata kunci :** Kontraktor perangkat lunak, strategi-strategi, *Capability Maturity Model Integration for Development (CMMI for Dev)*, *Process Area (PA)*.

## I. PENDAHULUAN

Keputusan dalam memilih kontraktor piranti lunak haruslah merupakan keputusan strategis organisasi. Dua alasan utama mempercayakan sebagian atau keseluruhan pekerjaan menghasilkan piranti lunak kepada pihak lain; alasan pertama adalah bisnis inti organisasi non piranti lunak berkepentingan menyerahkan pekerjaannya kepada kontraktor yang bisnis intinya sesuai; alasan kedua adalah karena pengalaman fungsional dan teknikal organisasi tersebut tidak mencukupi untuk membangun sendiri piranti lunaknya sehingga berkepentingan untuk menunjuk kontraktor berpengalaman. Namun demikian didalam manajemen proyek organisasi itu sendiri, khususnya pada proyek piranti lunak tersebut sebaiknya bahkan seharusnya telah memiliki menejer proyek yang mempunyai kemampuan tinggi pada level organisasi yang sesuai bidangnya terhadap *acquisition/outsourcing deal*.

Bagi kebanyakan organisasi pengembang piranti lunak, menghasilkan piranti lunak yang berkualitas, mutlak harus dilakukan untuk dapat memenuhi *user requirement* dan kepuasan pelanggannya. Kualitas piranti lunak sangat dipengaruhi oleh “orang” yang mengembangkan, “teknologi” yang digunakan, dan “proses” pengembangannya. *User* yang akan menyerahkan pengembangan piranti lunaknya kepada kontraktor piranti lunak perlu memperhatikan profil kontraktor itu sendiri yang meliputi ketiga aspek diatas, apakah sudah sesuai dengan bakuan CMMI. CMMI for Dev menyediakan proses area-proses area yang bertujuan mendewasakan organisasi dalam melakukan proses

kegiatannya. Setiap proses area-proses area tersebut dikelompokkan dalam beberapa tingkat *maturity*. Walaupun tidak secara spesifik menekankan pada produk yang dihasilkan, namun diharapkan melalui peningkatan proses kegiatannya akan dihasilkan kualitas produk yang lebih baik.

Penilaian terhadap sebuah organisasi piranti lunak dapat dilakukan dengan mengetahui tingkat *maturity* organisasi tersebut dalam CMMI, sehingga dapat diketahui kualitas dan produktifitas proses kegiatannya yang berujung kepada kualitas produk piranti lunak yang dihasilkannya. Namun demikian, tentu saja diperlukan pengetahuan bagi menejer beserta tim proyek piranti lunak yang akan merekomendasikan kontraktor piranti lunak terpilih tentang bakuan CMMI itu sendiri dan pengaruhnya terhadap proses serta kualitas piranti lunak yang akan dihasilkannya. Disamping itu ternyata proses area-proses area dalam CMMI for Dev dapat pula dikembangkan dan dimanfaatkan dalam proses penilaian dan pemilihan para kandidat kontraktor piranti lunak.

Paper ini menjelaskan strategi dalam pemilihan kontraktor pengembang piranti lunak yang dimulai dengan pengetahuan tentang apa yang dimaksud dengan CMMI dalam CMMI for Dev kemudian dilanjutkan dengan bagaimana beberapa proses area dalam CMMI for Dev tersebut dapat dimanfaatkan dalam kegiatan penilaian dan pemilihan kandidat kontraktor piranti lunak, sehingga hasilnya mampu memenuhi proses bisnis yang sedang berjalan di tingkat organisasi dan memuaskan di tingkat pengguna maupun pelanggan organisasi tersebut. Paper diakhiri dengan beberapa kesimpulan yang berhasil diambil.

## **II. KONSEP CAPABILITY MATURITY MODEL INTEGRATION FOR DEVELOPMENT (CMMI FOR DEV)**

Konsep CMMI for Dev dikeluarkan oleh *Software Engineering Institute (SEI)* dari *Carnegie Mellon University* pada akhir tahun 2001 dan di-publish Agustus 2006 guna menggantikan konsep serupa yaitu CMM yang telah dipakai untuk proses *assessment* sejak tahun 1990-an. Kegiatan pengembangan ini sendiri disponsori oleh US DoD (*Department of Defense*) [1]. Dalam dokumen resminya, CMMI bertujuan meningkatkan kematangan organisasi dengan memberikan panduan (*guidance*) mengenai peningkatan proses pengembangan suatu produk dan layanan. CMMI for Dev juga terdiri dari hal-hal praktis yang menunjuk pada aktifitas pemeliharaan dan pengembangan, termasuk daur hidup (*lifecycle*) produk piranti lunak dari mulai konsep, pengembangan, hingga pemeliharannya dengan penekanan pada aktifitas kegiatan yang diperlukan untuk membangun dan memelihara keseluruhan total produk.

CMMI for Dev dibuat untuk menghindari penggunaan berbagai model CMM secara terpisah. Oleh karena itu model yang telah terintegrasi di dalam CMMI for Dev merupakan model-model dalam bakuan CMM yang meliputi [7]:

- SW-CMM (*Capability Maturity Model for Software*)
- SE-CMM (*System Engineering Capability Maturity Model*)
- SECAM (*System Engineering Capability Assessment Model*)
- SECM (*System Engineering Capability Model*)
- SA-CMM (*Software Acquisition Capability Maturity Model*)

- IPD-CMM (*Integrated Product Development Capability Maturity Model*)

Pemilihan untuk mengintegrasikan model-model diatas karena adopsi terhadap model-model tersebut yang sudah semakin meluas didalam komunitas rekayasa piranti lunak (*software engineering*) dan rekayasa sistem (*system engineering*) dan masing-masing model tersebut memiliki pendekatan yang berbeda-beda untuk peningkatan proses aktifitas kegiatan suatu organisasi. Penggunaan model-model CMM yang terpisah memiliki beberapa kekurangan, karena [7]:

- Masing-masing model memiliki struktur, kondisi, format, dan pengukuran kedewasaan yang berbeda
- Dapat membingungkan, terutama pada penggunaan lebih dari satu model
- Sangat sulit untuk menggabungkan masing-masing model tersebut
- Sulit digunakan sebagai referensi dalam pemilihan *supplier*.

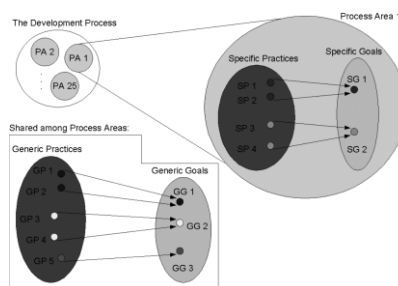
Sedangkan, jika menggunakan CMMI for Dev dapat diperoleh keuntungan sebagai berikut [7]:

- Peningkatan dalam penilaian (*assessment*) yang efektif dan efisien pada berbagai disiplin
- Mengurangi biaya pelatihan dan biaya *assessment*
- Secara umum, merupakan peningkatan visi yang terintegrasi pada seluruh elemen organisasi
- Integrasi dari rekayasa sistem dan lingkungan piranti lunak sehingga dapat menambah produktivitas dan kualitas produk.

#### A. Struktur CMMI for Development

CMMI for Dev disusun berdasarkan tiga konsep yaitu : *Process rea* (PA), *goals*, dan *practices* [1]. Gambar.1 dapat dijadikan ilustrasi struktur dari CMMI for Dev. CMMI for Dev terdiri dari 22 *Process Area*, dimana masing-masing PA tersebut terdiri dari *Specific Practices* (SP) dan *Specific Goals* (SG). Selain itu, terdapat pula *goals* dan *practices* lain yang disebut *Generic Practices* (GP) dan *Generic Goals* (GG). Adapun GG dan GP sama dengan SP dan SG, dengan pengecualian bahwa keduanya tidak hanya spesifik ke PA tertentu tetapi keduanya fokus kepada lebih dari satu PA.

CMMI for Development sendiri terdiri dari dua model yaitu CMMI for Dev + IPPD (*Integrated Products and Process Development*) dan CMMI for Dev tanpa IPPD. Keduanya memiliki persamaan pada *shared area*, tetapi bagaimanapun CMMI For Development + IPPD memiliki tambahan *goals* dan *practices* yang tidak tercakup dalam CMMI for Dev tanpa IPPD.



Gambar.1. Struktur CMMI for Development [3]

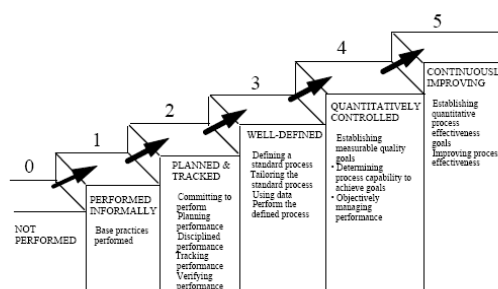
## B. Pemilihan Representasi

Pemilihan representasi dipengaruhi oleh tiga hal yaitu faktor bisnis, faktor budaya, dan juga faktor pengalaman dalam penggunaan model-model sebelumnya. CMMI for Dev memungkinkan penggunaannya untuk menggunakan dua representasi yang berbeda, yaitu: *staged* dan *continues*. Pada representasi *continues* dimungkinkan sebuah organisasi untuk memilih PA (atau suatu kelompok PA) dan meningkatkan proses yang sesuai dengannya, sedangkan pada representasi *staged* menggunakan kumpulan *predefined* dari PA untuk mendefinisikan alur peningkatan untuk suatu organisasi. Alur peningkatan proses ini dicirikan dalam suatu tingkat kematangan (*maturity levels*) dan masing-masing tingkat kematangan ini menyediakan kumpulan PA yang mencirikan perilaku organisasi yang berbeda [1].

Jika sebelumnya telah menggunakan CMM dan sangat mengenal representasi tertentu, disarankan untuk menggunakan representasi tersebut, karena proses transisi dari CMM ke CMMI sangat mudah. Sekali waktu jika sudah terbiasa menggunakan CMMI disarankan untuk menggunakan representasi yang lain. Organisasi biasanya menggunakan representasi kedua-keduanya untuk peningkatan proses mereka. Bagian berikut akan menjelaskan perbandingan dari masing-masing representasi tersebut dan representasi mana yang cocok bagi suatu organisasi.

## C. Representasi Continues

Representasi *continues* memberikan fleksibilitas yang tinggi ketika menggunakan CMMI untuk peningkatan proses. Sebuah organisasi bisa meningkatkan performa dari proses yang dilakukannya. Representasi *continues* juga memungkinkan sebuah organisasi untuk meningkatkan banyak proses-proses yang berbeda pada berbagai tingkat yang berbeda pula. Representasi *continues* ini dapat diilustrasikan seperti pada Gambar.2.

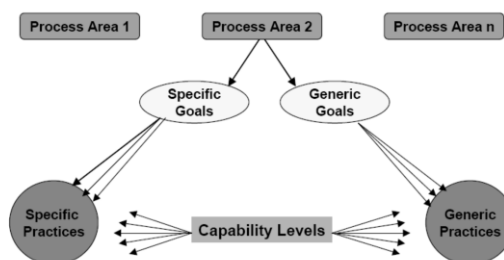


Gambar.2. CMMI for Dev Representasi Continues [5]

Jika telah mengetahui proses yang perlu ditingkatkan dan juga memahami ketergantungan (*dependency*) diantara seluruh PA yang ada di CMMI for Dev, representasi *continues* sangat sesuai untuk organisasi Anda. Gambar.3 memperlihatkan struktur dari representasi *continues*.

Secara kategorial, PA dari CMMI for Dev dengan representasi *continues* dapat dijelaskan pada gambar berikutnya (Gambar.4). PA sendiri disusun berdasarkan level kemampuan (*capability level*). Seluruh 25 PA representasi *continues* dikategorikan

kedalam 5 (lima) proses kategori yaitu Process Management (5 PAs), Project Management (8 PAs), Engineering (6 PAs), dan Support (6 PAs).



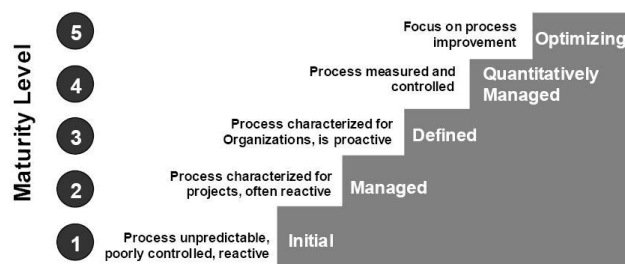
Gambar.3. Struktur dari Representasi Continues CMMI for Dev [5]

Process Area Categories	Process Area
Process Management	Organizational Process Focus Organizational Process Definition Organizational Training Organizational Process Performance Organizational Innovation and Deployment
Project Management	Project Planning Project Monitoring and Control Supplier Agreement Management Integrated Project Management (IPPD) Risk Management Integrated Teaming Integrated Supplier Management Quantitative Project Management
Engineering	Requirement Management Requirements Development Technical Solution Product Integration Verification Validation
Support	Configuration Management Process and Product Quality Assurance Measurement and Analysis Decision Analysis and Resolution Organizational Environment for Integration Causal Analysis and Resolution

Gambar.4. Process Area CMMI for Dev Representasi Continues Berdasarkan Kategori [5]

#### D. Representasi Staged

Pada representasi ini memberikan cara yang lebih sistematis dan terstruktur untuk peningkatan proses berdasarkan model satu tingkat dalam satu waktu. Pencapaian pada masing-masing tingkat dapat dijadikan dasar untuk pencapaian ke tingkat selanjutnya. PA disusun berdasarkan tingkat kematangan (*maturity level*). Representasi *staged* menentukan suatu susunan/urutan untuk implementasi PA berdasarkan tingkat kematangannya, yang akan memberikan alur peningkatan proses bagi suatu organisasi dari *Initial level* hingga *Optimizing level* (Gambar.5). Jika Anda tidak mengetahui proses mana yang perlu ditingkatkan, representasi *staged* merupakan pilihan yang terbaik bagi organisasi Anda.



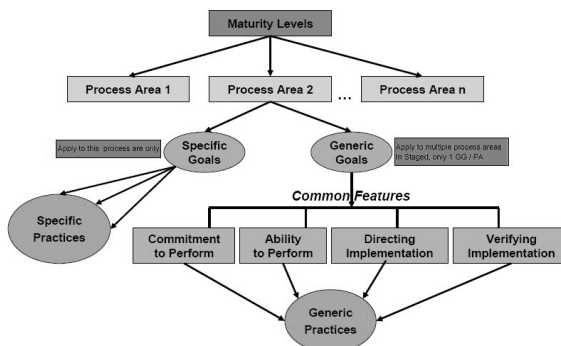
Gambar.5. CMMI for Dev Representasi Staged [5]

Tingkat kematangan ini dapat dijelaskan sebagai berikut [3]:

- *Maturity level 1 - Initial.* Pada ML1 ini proses biasanya berbentuk ad hoc. Sukses pada level ini didasarkan pada kerja keras dan kompetensi yang tinggi orang-orang yang ada didalam organisasi tersebut
- *Maturity level 2 - Managed.* Pada ML2 ini sebuah organisasi telah mencapai seluruh *specific* dan *generic goals* pada Level 2. Dengan kata lain seluruh proses dalam organisasi telah direncanakan, dilaksanakan, diukur, dan dikontrol dengan baik
- *Maturity level 3 - Defined.* Pada ML3 ini sebuah organisasi telah mencapai seluruh *specific* dan *generic goals* pada Level 2 dan Level 3. Proses dicirikan dan dipaparkan dalam standar, prosedur, *tool*, dan metode
- *Maturity level 4 - Quantitatively Managed.* Pada ML4 ini, sebuah organisasi telah mencapai seluruh *specific* dan *generic goals* yang ada pada Level 2, 3, dan 4. Sebuah subproses dipilih yang secara signifikan terlibat dalam keseluruhan proses. Subproses yang terpilih ini kemudian dikontrol dengan menggunakan statistik atau teknik kuantitative lainnya
- *Maturity level 5 - Optimizing.* Pada ML5 ini suatu organisasi telah mencapai seluruh *specific* dan *generic goals* yang ada di Level 2, 3, 4, dan 5. ML 5 fokus kepada peningkatan proses secara berkesinambungan melalui inovasi teknologi.

Seluruh PA yang ada dalam representasi *stage* maupun *continues* adalah sama. Tetapi yang pada representasi *staged* tidak ada persyaratan (*requirements*) untuk Level 1, tidak seperti pada representasi *continues* yang memiliki persyaratan bagi organisasi yang berada pada Level 1. Peningkatan kemampuan ini memungkinkan suatu organisasi lebih terlihat jelas kemajuan yang telah dicapainya Struktur dari representasi dan PA yang ada di representasi *staged* dapat dijelaskan pada Gambar.6 dan Gambar.7 dibawah ini.

Pada representasi *staged* ini, masing-masing PA juga dikategorikan kedalam 5 (lima) kategori seperti dijelaskan diatas. Masing-masing tingkat memiliki PA, kecuali pada Level 1. Pada Level 2 memiliki 7 (tujuh) PA, pada Level 3 memiliki 14 (empat belas ) PA, dan masing-masing pada Level 4 dan Level 5 memiliki 2 (dua) PA.



Gambar.6. Struktur dari CMMI for Dev Representasi Staged [5]

Maturity Level	Focus	Process Areas
5 Optimizing	Continuous Process Improvement	<ul style="list-style-type: none"> <li>• Organization Innovation and Deployment</li> <li>• Causal Analysis and Resolution</li> </ul>
4 Quantitatively Managed	Quantitative Management	<ul style="list-style-type: none"> <li>• Organizational Process Performance</li> <li>• Quantitative Project Management</li> </ul>
3 Defined	Process Standardization	<ul style="list-style-type: none"> <li>• Requirements Development</li> <li>• Technical Solution</li> <li>• Product Integration</li> <li>• Verification</li> <li>• Validation</li> <li>• Organizational Process Focus</li> <li>• Organizational Process Definition</li> <li>• Organizational Training</li> </ul> <ul style="list-style-type: none"> <li>• Integrated Project Management</li> <li>• Risk Management</li> <li>• Decision Analysis and Resolution</li> <li>• Organizational Environment for Integration</li> <li>• Integrated Teaming</li> <li>• Integrated Supplier Management</li> </ul>
2 Managed	Basic Project Management	<ul style="list-style-type: none"> <li>• Requirements Management</li> <li>• Project Planning</li> <li>• Project Monitoring and Control</li> <li>• Supplier Agreement Management</li> </ul> <ul style="list-style-type: none"> <li>• Measurement and Analysis</li> <li>• Product and Process Quality Assurance</li> <li>• Configuration Management</li> </ul>
1 Initial		

Gambar.7. Process Area dari CMMI for Dev Representasi Staged [5]

### III. STRATEGI PEMILIHAN KONTRAKTOR PIRANTI LUNAK DENGAN MEMANFAATKAN PROSES AREA CMMI FOR DEVELOPMENT

#### A. Memanfaatkan *Process Area Causal Analysis and Resolution*

*Causal analysis and Resolution* (CAR) adalah sebuah proses area kategori pendukung (*support*) pada representasi *staged* pada CMMI for Dev *maturity level 5* yang bertujuan mengidentifikasi kasus cacat (*causes of defects*) dan masalah lainnya dan membuat suatu tindakan pencegahannya sehingga tidak terjadi dimasa yang akan datang [1]. Sebagai sebuah strategi dalam pemilihan kontraktor piranti lunak, proses area CAR dapat dimanfaatkan karena dua alasan berikut:

1. Guna menilai *track and record* kandidat kontraktor piranti lunak dalam menangani cacat dan permasalahan lainnya
2. Sarana pembelajaran bagi organisasi pemilih kandidat tersebut.

Adalah kurang bijaksana menilai kualitas kerja kontraktor piranti lunak dari kepadatan cacat (*defect density*) produk piranti lunaknya karena cacat adalah suatu keniscayaan atau lumrah ada dalam setiap piranti lunak. Bahkan pada piranti lunak yang mapan sekalipun, seperti Microsoft Windows atau Macintosh cacat pernah teridentifikasi. Yang perlu menjadi bahan pertimbangan adalah, bagaimana pola penanganan oleh kontraktor piranti lunak tersebut terhadap cacat dan permasalahan lainnya yang mungkin muncul terutama saat piranti lunak tersebut digunakan oleh organisasi pengguna. Dan yang lebih penting lagi adalah apakah kontraktor piranti lunak tersebut memiliki dokumen tentang sejarah cacat produk dan masalah-masalah lainnya serta tindakan yang telah dilakukan untuk mengatasinya selama menangani proyek-proyek sebelumnya yang akan digunakan untuk mencegah kejadian serupa yang mungkin terjadi pada proyek yang sedang/akan berjalan.

Dengan kata lain, proses area CAR dapat dimanfaatkan untuk menilai kemampuan kontraktor piranti lunak dalam menganalisa dan mengidentifikasi kecenderungan-kecenderungan terjadinya cacat dan masalah lainnya akan berimplikasi kepada penilaian kemampuan mereka dalam mengambil tindakan pencegahan terhadap masalah-masalah yang mungkin akan muncul dalam perjalanan kegiatan proyek tersebut.

Proses area CAR dapat pula digunakan sebagai sarana pembelajaran bagi organisasi karena langkah-langkah dalam *specific goal and practices* (SG dan SP) pada CAR juga relevan untuk mengatasi masalah diluar cacat, misalnya pada permasalahan pemilihan kontraktor piranti lunak. Jenis-jenis kesalahan yang pernah muncul dalam proses pemilihan dan penunjukkan kontraktor piranti lunak sebaiknya dianalisa dan diidentifikasi kecenderungannya. Berdasarkan pemahaman mendefinisikan proses tersebut dan bagaimana hal tersebut terjadi, akar penyebab masalah dan akibat yang ditimbulkannya dapat segera diketahui.

SG dan SP tersebut menyediakan sebuah mekanisme bagi tim pemilihan kontraktor piranti lunak untuk mengevaluasi proses-proses kegiatannya pada tingkat lokal dan mencari bentuk perbaikan yang dapat diimplementasikan. Bila bentuk perbaikan dinilai efektif, informasi tersebut dapat diteruskan ke level organisasi.

#### B. Memanfaatkan *Proses Area Decision Analysis and Resolution*

*Decision nalysis and Resolution* (DAR) adalah sebuah proses area kategori *Support* pada CMMI for Dev *maturity level 3* yang bertujuan untuk menganalisa keputusan yang mungkin akan diambil dengan menggunakan proses evaluasi formal yang akan mengevaluasi alternatif yang telah ditemukan terhadap kriteria-kriteria yang telah ditentukan sebelumnya [1]. Sedangkan yang dimaksud dengan proses evaluasi formal adalah sebuah pendekatan terstruktur untuk mengevaluasi solusi alternatif terhadap kriteria yang telah ditentukan untuk menentukan sebuah solusi yang direkomendasikan yang menunjuk pada suatu isu [1].

Proses area DAR mencakup pembuatan *guideline* untuk menentukan isu-isu yang mana yang merupakan subyek pada proses evaluasi dan kemudian menerapkannya kedalam isu-isu tersebut. Yang dimaksud dengan isu-isu tersebut adalah sebuah keputusan yang harus diambil dari beberapa solusi alternatif keputusan yang ada pada pengembangan sebuah produk atau pada sebuah tahapan dalam *System Development Life Cycle* (SDLC). Salah satu isu yang dimaksud adalah bagaimana memilih kontraktor piranti lunak yang merupakan salah satu dari solusi alternatif apakah organisasi akan membuat sendiri (*make/in house*), membeli (*buy/COTS*) atau menyerahkan pada pihak ketiga (*acquisition/outsourcing*). Isu-isu spesifik tersebut perlu diidentifikasi dalam sebuah proses evaluasi formal.

*Guideline* tersebut kemudian digunakan untuk menghasilkan sebuah proses evaluasi formal yang merupakan pendekatan terstruktur untuk mengevaluasi solusi alternatif berapa para kandidat kontraktor piranti lunak yang terdaftar terhadap kriteria-kriteria yang telah ditetapkan sebelumnya guna menghasilkan sebuah solusi rekomendasi proses penunjukkan kontraktor piranti lunak. Dengan menggunakan sebuah proses evaluasi formal, penunjukkan kontraktor piranti lunak akan menurunkan subyektifitas dalam pengambilan keputusan pemilihan kontraktor piranti lunak dengan probabilitas penyeleksian yang lebih ketat dan relevan terhadap berbagai tuntutan pihak-pihak terkait (*stakeholder*).

Beberapa *specific practices* (SP) dari proses area DAR dapat pula dimanfaatkan dalam membuat kriteria-kriteria pemilihan kontraktor piranti lunak, seperti *SP Establish Guidelines for Decision Analysis*, dapat dimanfaatkan untuk membuat *guideline* yang nantinya akan dikembangkan dalam sebuah proses evaluasi formal pemilihan kontraktor



piranti lunak. *SP Establish Evaluation Criteria* menyediakan *practices* yang dapat dimanfaatkan untuk membuat kriteria evaluasi, alasan-alasan terpilih atau ditolaknya, dan ranking setiap kriteria berdasarkan nilai kebutuhan, obyektif, prioritas dan tingkat kepentingannya. *SP Select Evaluation Methods* dapat dimanfaatkan untuk menentukan metoda apakah yang akan digunakan dalam mengevaluasi kandidat kontraktor piranti lunak dihadapkan kepada kriteria-kriteria yang telah ditentukan dengan cara simulasi menggunakan model probalistik dan teori keputusan atau simulasi lainnya.

*SP Evaluate Alternatives* dapat dimanfaatkan untuk mengevaluasi kembali solusi-solusi penggunaan kriteria kriteria dan metoda-metoda yang telah ditetapkan melalui analisa-analisa, diskusi-diskusi, dan *review*. Mungkin pula eksperimen, *prototype*, pilot atau simulasi dibutuhkan untuk mendukung substansi penilaian dan kesimpulan akhir. *SP Select Solutions* dapat dimanfaatkan untuk sebagai pertimbangan akhir hasil-hasil dari evaluasi alternatif diatas. Resiko sehubungan dengan penerapan solusi-solusi harus menjadi perhatian. Produk akhir (*deliverable product*) yang dihasilkan berupa rekomendasi solusi akhir kriteria dan metoda penunjukkan kontraktor piranti lunak.

### C. Mamanfaatkan Proses Area Supplier Agreement Management

*Supplier Agreement Management* (SAM) adalah sebuah proses area kategori *Project Management* pada CMMI for Dev maturity level 2 yang bertujuan untuk mengatur pengadaan barang dari supplier [1]. Sedangkan maksud dari "pengadaan" adalah proses mendapatkan produk (barang dan layanan) melalui kontrak [1].

Beberapa *specific practices* (SP) dari proses area SAM dapat dimanfaatkan sebagai suatu strategi dalam pemilihan kontraktor piranti lunak seperti *SP Determine Acquisition Type* yang dapat dimanfaatkan untuk menentukan jenis pengadaan produk piranti lunak atau komponen piranti lunak yang akan dipesan. Terdapat banyak pengadaan yang dapat digunakan untuk menghasilkan piranti lunak atau komponennya, diantaranya:

- *Custom Development*: Proses pengembangan piranti lunak dilakukan sendiri oleh pihak yang membutuhkannya
- *Package Software*: Proses pengembangan piranti lunak dilakukan dengan cara membeli produk piranti lunak yang tersedia dipasaran (COTS) termasuk lisensinya kemudian dikustomisasi sesuai dengan kebutuhan organisasi
- *Outsourcing/acquisition*: Proses pengembangan piranti lunak dilakukan oleh pihak ketiga berdasarkan kesepakatan (*agreement*).

Kegiatan yang sebaiknya dilakukan adalah dengan membuat daftar jenis pengadaan yang akan digunakan, apakah akan dibuat sendiri (*make*), dengan cara membeli (*buy*) atau dilakukan oleh pihak lain (*outsourcing*) seperti dijelaskan diatas untuk semua piranti lunak atau komponen piranti lunak. *SP Select Supplier* dapat dimanfaatkan untuk memilih kontraktor yang didasarkan kepada evaluasi kemampuan mereka yang sesuai dengan spesifikasi persyaratan yang diminta dan kriteria yang telah ditentukan. Kriteria tersebut harus ditentukan merujuk pada factor-factor penting pada proyek piranti lunak. Contoh faktor faktor penting tersebut adalah:

- Lokasi geografis dari kontraktor piranti lunak
- *Track record* kontraktor dalam menangani pekerjaan yang serupa

- Kemampuan teknik dari kontraktor piranti lunak
- Staf dan fasilitas yang disediakan kontraktor tersebut untuk melakukan mgas tersebut
- Pengalaman dalam mengaplikasikan kegiatan yang serupa.

Langkah-langkah awal yang dapat dilakukan adalah:

- Studi pasar (*market study*)
- Daftar kandidat kontraktor piranti lunak
- Daftar kontraktor yang sedang atau pernah bekerja sama sebelumnya
- *Trade study* atau catatan evaluasi kriteria lainnya untuk mengetahui kelebihan dan kekurangan kandidat kontraktor piranti lunak, dan alasan alasan pemilihan kontraktor piranti lunak
- Materi permintaan dan persyaratan

Langkah selanjutnya yang dapat dilakukan

- Tetapkan dan dokumentasikan kriteria untuk mengevaluasi kontraktor piranti lunak yang potensial
- Identifikasi kontraktor potensial dan materi permintaan serta persyaratan kepada kontraktor tersebut
- Evaluasi proposal menurut kriteria evaluasi
- Evaluasi risiko terhadap setiap penawaran dari para kontraktor
- Evaluasi daya-tawar kontraktor dalam melakukan pekerjaan
- Pilih kontraktor piranti lunak

Contoh metode evaluasi daya-tawar kontraktor piranti lunak dalam melakukan pekerjaan tersebut adalah:

- Evaluasi pengalaman terdahulu dengan aplikasi serupa
- Evaluasi performa terdahulu dalam menangani pekerjaan serupa
- Evaluasi kapabilitas manajemennya
- Evaluasi kapabilitasnya
- Evaluasi staf yang tersedia untuk menangani pekerjaan tersebut
- Evaluasi fasilitas-fasilitas dan sumber daya lainnya yang dimiliki kontraktor tersebut
- Evaluasi kemampuan tim proyek untuk bekerja sama dengan kontraktor tersebut
- Evaluasi pengaruh kandidat piranti lunak terhadap rencana proyek dan komitmen yang telah dibuat.

#### D. Memanfaatkan *Proses Area Technical Solution*

*Technical Solution* (TS) adalah sebuah proses area kategori *Engineering* pada CMMI for Dev *maturity level* 3 yang bertujuan mendesign, membangun, menerapkan solusi. solusi, design, dan implementasi yang meliputi produk. komponen produk, dan proses daur hidup produk yang sesuai baik yang berdiri sendiri maupun terkait dengan yang lain [1]. Jika proses area DAR dapat dimanfaatkan untuk membuat kriteria-kriteria penilaian sebuah solusi alternatif piranti lunak, maka proses area TS dapat dimanfaatkan untuk memilih kontraktor piranti lunak dengan menilai solusi alternatif piranti lunaknya.

Proses area TS menyediakan bagaimana sebuah produk seperti piranti lunak dipersiapkan dengan cara mendesain dan mengembangkan desain tersebut (pada tahap Desain) sebelum kemudian dibuatkan produk piranti lunak aktual atau yang sesungguhnya (pada tahap Implementasi). Proses area TS ini dapat dimanfaatkan sebagai suatu strategi dalam memilih kontraktor piranti lunak dengan menilai produk arsitektur yang mereka tawarkan berupa *prototype* atau *pilot*. *Prototype* atau *pilot* yang diperagakan oleh kontraktor piranti lunak dapat dipergunakan sebagai suatu bentuk penilaian yang cukup berharga terhadap kemampuan dan pengetahuan mereka dalam mengembangkan sebuah paket data teknik atau satu set persyaratan yang lengkap menjadi piranti lunak yang sesungguhnya.

Beberapa *specific practices* (SP) pada TS dapat pula dimanfaatkan sebagaimana proses areanya seperti pada *SP Develop lternative Solution and Selection Criteria*. Dalam SP ini pihak pemesan/organisasi akan memiliki solusi alternatif yang dibutuhkan sebagai solusi pembanding terhadap piranti lunak yang akan dibuat. Solusi alternatif ini dapat berupa produk arsitektur (sebagai contoh piranti lunak yang disiapkan oleh pihak kontraktor) atau produk komersial (COTS/produk yang beredar dipasaran) sebagai produk arsitekturnya. Solusi alternatif tersebut dapat mengurangi kesenjangan dari segi biaya, jadwal, dan performa terhadap kandidat piranti lunak yang akan dikembangkan/dipesan atau dapat dipergunakan sebagai *design issues*, *constraints*, dan kriteria piranti lunak yang akan dikembangkan/dipesan tersebut. Kriteria seleksi dapat berupa biaya (waktu, SDM, dan uang), *benefit* (performa, kapabilitas, dan keefektifannya), dan resiko (teknik, ekonomi, organisasi), dan lain-lain.

Pertimbangan lebih terperinci terhadap solusi alternatif dan kriteria seleksi dapat dijabarkan sebagai berikut:

- Biaya pengembangan, manufaktur, lelang, pemeliharaan serta dukungan, dan lain-lain
- Performa
- Kompleksitas piranti lunak termasuk selama dalam proses masapaknya
- Daya tahan terhadap kondisi dan operasi penggunaan, mode-mode pengoperasiannya, lingkungan, dan variasi yang kelak diterapkan selama proses masa pakainya
- Kemampuan ekspansi dan dikembangkan
- Keterbatasanteknologinya
- Sensitifitas terhadap metode dan mated konstruksinya
- Risiko
- Perkembangan requirement dan teknologi
- Disposal
- Kemampuan dan keterbatasan pengguna dan operator yang akan menggunakan piranti lunak tersebut

Biaya masa pakai piranti lunak yang sebenarnya sangat perlu diminimalkan, sering terlewatkan dari pantauan organisasi terutama pada organisasi yang sedang berkembang. Sering terjadi piranti lunak tidak dapat mengikuti perkembangan organisasi dan fitur-fitur yang tersedia hanya bermanfaat dalam waktu singkat. Pelanggan tentu tidak

bersedia membayar fitur-fitur jangka pendek dengan biaya lebih mahal sehingga jika manfaatnya telah tidak ada pada akhirnya jika fitur-fitur tersebut dihilangkan akan menurunkan biaya keseluruhan dalam masa pakai piranti lunak.

Dalam beberapa hal, pelanggan seharusnya sedikit-banyak mendapat masukan tentang beberapa potensi untuk menurunkan biaya selama masa pakai piranti lunak. Kriteria tersebut dapat digunakan dalam seleksi atau sebagai solusi terakhir yang sebaiknya diadakan sebagai pendekatan pembandingan terhadap biaya dan keuntungan serta resiko. Pada akhirnya kontraktor piranti lunak yang mampu menyediakan/menjelaskan solusi alternatif terbaik (berupa *prototype* atau *pilot*) dapat dijadikan kandidat utama.

Langkah-langkah strateginya dijelaskan dalam *SP Select Product Component Solutions* yang dapat dimanfaatkan sebagai suatu sistem penilaian terhadap solusi alternatif piranti lunak kandidat kontraktor piranti lunak yang pada akhirnya nanti turut menentukan terpilih tidaknya kontraktor piranti lunak tersebut. Langkah langkah tersebut dapat dijabarkan sebagai berikut:

- Evaluasi setiap solusi alternatif piranti lunak (berupa *prototype* atau *pilot*) terhadap kriteria seleksi yang telah ditentukan sebelumnya dalam bentuk konsep dan skenario pengoperasian piranti lunak
- Berdasarkan hasil evaluasi, beri nilai terhadap kriteria seleksi yang memenuhi dan update kriteria tersebut jika perlu.
- Identifikasi dan masukkan isu-isu yang berkembang dan relevan seputar requirement dan solusi alternatif piranti lunak serta kandidat piranti lunak yang akan dibuat.
- Pilih solusi alternatif piranti lunak terbaik yang paling memenuhi dan memuaskan dalam kriteria seleksi tersebut.
- Perhatikan persyaratan dan hubungannya dengan setiap set alternatif terpilih sebagai satu set alokasi persyaratan pada piranti lunak dan komponen piranti lunak tersebut.
- Identifikasi kembali setiap solusi alternatif komponen piranti lunak yang mana yang akan digunakan kembali (*reused*) dan yang mana akan dibuat *bam* (*acquired*).
- Dokumentasikan hasil solusi, evaluasi, dan alasan alasannya.

#### IV. KESIMPULAN

Keputusan dalam memilih kontraktor piranti lunak haruslah merupakan keputusan strategis organisasi. Oleh karena itu, proyek tersebut sebaiknya dipimpin oleh menejer proyek dengan kemampuan tinggi pada level organisasi yang sesuai bidangnya terhadap *acquisition/outsourcing deal* karena menyerahkan pengembangan piranti lunak yang sama kepada kontraktor yang berbeda akan menghasilkan beberapa piranti lunak yang pasti berbeda, termasuk berbeda dari segi kualitas dan produktifitasnya.

Menejer proyek dan timnya harus mengerti bagaimana strategi-strategi dalam memilih kandidat terbaik. Salah satu penilaian dalam pemilihan kontraktor piranti lunak dapat dilakukan dengan mengetahui tingkat *maturity* organisasi tersebut dalam CMMI, sehingga dapat diketahui kualitas dan produktifitas proses kegiatannya yang berujung

kepada kualitas produk piranti lunak yang akan dihasilkannya. Oleh sebab itu, menejer proyek dan timnya sebaiknya telah membekali diri dengan pengetahuan tentang CMMI.

Sebagai strategi tambahan, pokok-pokok bahasan dalam setiap proses area CMMI for Dev dapat digali dan dikembangkan guna dimanfaatkan dalam proses penilaian dan pemilihan kandidat yang kelak ditunjuk untuk mengembangkan piranti lunak seperti dibuktikan dalam penulisan ini.

## **V. DAFTAR PUSTAKA**

- [1] CMMI Product Team, Software Engineering Institute, Carnegie Mellon University, CMMI for Development, Version 1.2: Improving Processes for Better Products, CMU/SEI-2006-TR-008, ESC-TR-2006-008, Pittsburgh, PA, August 2006.
- [2] Dennis, A., Wixom, B.H., Tegarden, D., System Analysis and Design with UML Version 2.0, John Wiley & Sons Inc, 2005.
- [3] Hoggerl, Martin and Sehorz, Bernhard, An Introduction to CMMI and its Assessment Procedures, Seminar Paper, Department of Computer Science University of Salzburg, Februari 2006.
- [4] Foegen, Malte and Richter, Jurgen, CMM, CMMI, and ISO 15504 (SPICE), IT Maturity Services, Wibas, Germany, 2003.
- [5] Schneider, Henry., Overview of CMMI, Texas Symposium on Software Engineering, Teraquest, Texas, February 2004.
- [6] Reitzig, Rolf W., et all., Achieving Capability Maturity Model Integration Level 2 using IBM's Rational Software Solution, Rational Software Whitepaper, Rational Software Corporation, 2003.
- [7] Unknown, CMMI v1.1 vs SW-CMM Overview.