

# ***Capability Maturity Model Sebagai Mekanisme Kualifikasi Sebuah Software Development House***

Oleh :

**Johan Max**

[johansmax@yahoo.com](mailto:johansmax@yahoo.com)

*Dipublikasikan dan didedikasikan  
untuk perkembangan pendidikan di Indonesia melalui*

## **MateriKuliah.Com**

***Lisensi Pemakaian Artikel:***

*Seluruh artikel di MateriKuliah.Com dapat digunakan, dimodifikasi dan disebarluaskan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut Penulis. Hak Atas Kekayaan Intelektual setiap artikel di MateriKuliah.Com adalah milik Penulis masing-masing, dan mereka bersedia membagikan karya mereka semata-mata untuk perkembangan pendidikan di Indonesia. MateriKuliah.Com sangat berterima kasih untuk setiap artikel yang sudah Penulis kirimkan.*

# ***Capability Maturity Model Sebagai Mekanisme Kualifikasi Sebuah Software Development House***

## ***Bagian 1***

Dokument ini adalah bagian pertama dari tulisan bersambung mengenai Capability Maturity Model (CMM). Tujuan penulisan dokumen ini adalah untuk memberikan gambaran sederhana tentang CMM beserta kegunaanya.

Capability Maturity Model adalah sebuah model yang dikembangkan oleh Software Engineering Institute atas permintaan Departement of Defense(DOD) Amerika Serikat dengan tujuan membuat ujian saringan masuk bagi kontraktor yang mendaftarkan diri untuk menjadi konsultan DOD.

Capability diterjemahkan menjadi kapabilitas yang berarti kemampuan yang bersifat laten. Capability lebih mengarah kepada integritas daripada kapabilitas itu sendiri. Definisi integritas adalah kemampuan untuk menepati janji. Maturity berarti matang atau dewasa. Matang merupakan hasil proses. Dewasa merupakan hasil pertumbuhan. Model didefinisikan sebagai suatu penyederhanaan yang representatif terhadap keadaan di dunia nyata.

Jadi secara keseluruhan CMM dapat didefinisikan sebagai berikut : CMM adalah sebuah penyederhanaan yang representatif yang digunakan untuk mengukur tingkat kematangan sebuah software development house dalam menyajikan/membuat/mengembangkan perangkat lunak sebagaimana telah dijanjikan secara tertulis dalam perjanjian kerja sama.

Keyword utama dari CMM adalah mengukur. Mengukur didefinisikan sebagai suatu proses untuk memetakan sebuah kondisi ke dalam sebuah skala/ukuran. Berbicara mengukur kita berbicara :

- Apa yang diukur (parameter)
- Bagaimana cara mengukurnya (metode)
- Bagaimana standard penilaiannya (skala penilaian)
- Bagaimana interpretasinya (artinya bagi manusia)

Parameter adalah attribut dari sebuah entity yang akan diukur. Interpretasi adalah pemetaan dari hasil pengukuran ke dalam bahasa manusia. Mungkin sulit dipahami untuk itu saya beri ilustrasi sebagai berikut : Nilai rata-rata IP mahasiswa perguruan tinggi X adalah 3,5 dari nilai maximum 4. Fakta di atas dapat diinterpretasikan sebagai :

- Mahasiswanya pintar
- Materinya terlalu dangkal
- Mahasiswa jago nyontek

- Nilai bisa dibeli

Untuk mengukur tingkat kematangan sebuah organisasi tidaklah mudah. Proses ini melibatkan serangkaian parameter untuk diukur. Mekanisme yang digunakan adalah mekanisme observasi oleh lembaga sertifikasi independen yang melakukan pengamatan terhadap organisasi yang akan dinilai. Untuk menilainya digunakan metode analitical hirarki process (AHP). Sebagai ilustrasi dari metode AHP adalah sistem SKS di kampus kita masing-masing yaitu dengan ada sebuah parameter, nilai dan bobot (sks matakuliah)

Capability Maturity Model membuat 5 level/skala kematangan yaitu :

- Initial
- Repeatable
- Defined
- Managed
- Optimized

Level initial bercirikan sebagai berikut :

- Tidak adanya manajemen proyek
- Tidak adanya quality assurance
- Tidak adanya mekanisme manajemen perubahan (change management)
- Tidak ada dokumentasi
- Adanya seorang guru/dewa yang tahu segalanya tentang perangkat lunak yang dikembangkan.
- Sangat bergantung pada kemampuan individual

Level Repeatable bercirikan sebagai berikut :

- Kualitas perangkat lunak mulai bergantung pada proses bukan pada orang
- Ada manajemen proyek sederhana
- Ada quality assurance sederhana
- Ada dokumentasi sederhana
- Ada software configuration managemen sederhana
- Tidak adanya knowledge managemen
- Tidak ada komitmen untuk selalu mengikuti SDLC dalam kondisi apapun
- Tidak ada statiskal control untuk estimasi proyek
- Rentan terhadap perubahan struktur organisasi.

Level Defined bercirikan :

- SDLC sudah dibuat dan dibakukan

- Ada komitmen untuk mengikuti SDLC dalam keadaan apapun
- Kualitas proses dan produk masih bersifat kwalitatif bukan kualitatif (tidak terukur hanya kira-kira saja)
- Tidak menerapkan Activity Based Costing
- Tidak ada mekanisme umpan balik yang baku

Level Managed bercirikan :

- Sudah adanya Activity Based Costing dan digunakan untuk estimasi untuk proyek berikutnya
- Proses penilaian kualitas perangkat lunak dan proyek bersifat kuantitatif.
- Terjadi pemborosan biaya untuk pengumpulan data karena proses pengumpulan data masih dilakukan secara manual
- Cenderung bias. Ingat effect thorne, manusia ketika diperhatikan maka prilakunya cenderung berubah.
- Tidak adanya mekanisme pencegahan defect
- Ada mekanisme umpan balik

Level Optimized bercirikan :

- Pengumpulan data secara otomatis
- Adanya mekanisme pencegahan defect
- Adanya mekanisme umpan balik yang sangat baik
- Adanya peningkatan kualitas dari SDM dan peningkatan kualitas proses.

Ekivalensi CMM dengan programming.

Programming atau pembuatan program dapat dibuat kesamaannya dengan CMM. Programming in small (coding red) ekivalen dengan CMM level 1. Programming in large (proyek manajemen, dokumentasi, dll) ekivalen dengan CMM level 2. Keduanya dapat dikelompokkan menjadi programming as art proccess karena tidak memiliki unsur engineering. Unsur engineering yang perlu ditambahkan adalah standarisasi (pembakuan) dan pengukuran. Jika sudah dilakukan standarisasi maka ekivalen dengan CMM level 3. Jika sudah ada pengukuran maka ekivalen dengan CMM level 4. Jika sudah sampai di level 4 maka programming dapat dianggap sebagai engineering process. Keseluruhan level dari 1-4 dapat dipandang sebagai programming as discreet process dimana tidak ada pengembangan berkelanjutan (life time quality improvment) . Baru pada level 5 programming dapat dianggap sebagai continues process dimana peningkatan kualitas sumber daya manusia dan proses dilakukan secara terus menerus.

Kegunaan CMM meliputi :

- Menilai tingkat kematangan sebuah organisasi pengembang perangkat lunak
- Memfilter kontraktor yang akan menjadi pengembang perangkat lunak
- Memberikan arah untuk peningkatan organisasi bagi top manajemen di dalam sebuah organisasi pengembang perangkat lunak
- Sebagai alat bantu untuk menilai keunggulan kompetitif yang dimiliki sebuah perusahaan dibandingkan perusahaan pesaingnya.