

LAPORAN SOFTWARE CONFIGURATION MANAGEMENT
e-Commerce Macarina
MATA KULIAH MANAGEMENT KUALITAS
PERANGKAT LUNAK



Disusun Oleh :

Achmad Iqbal Baidlowi	(E41181040)
M.Birril Febrian Aziz	(E41180702)
Eilham Wayu Pratama	(E41180954)
Luqman Hakim	(E41180965)
Ratna Dwi Kristina Sari	(E41180434)

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2020

BAB I

PENDAHULUAN

Perubahan adalah hal yang tidak dapat dihindarkan ketika perangkat lunak komputer sedang dibuat. Perubahan-perubahan tersebut meningkatkan tingkat kebingungan di antara para software engineer yang berkerja pada proyek tersebut. Kebingungan muncul bila perubahan-perubahan tersebut tidak dianalisis sebelum perubahan tersebut dilaksanakan; dicatat sebelum diimplementasi, dilaporkan kepada yang ingin mengetahui, atau dikontrol dengan suatu cara yang akan meningkatkan kualitas & mengurangi error. Software configuration management (SCM) adalah kegiatan payung (umbrella activities) yang dilaksanakan selama proses perangkat lunak.

Tujuan SCM

- Mengidentifikasi perubahan
- Mengontrol perubahan
- Memastikan perubahan yang telah diimplementasikan

BAB II

DASAR TEORI

Manajemen konfigurasi perangkat lunak (SCM), sering disebut juga manajemen perubahan (change management), adalah serangkaian kegiatan yang dirancang untuk mengelola perubahan dengan mengidentifikasi produk/hasil kerja yang kemungkinan besar akan mengalami perubahan, membuat hubungan di antara mereka, menentukan mekanisme untuk mengelola berbagai versi produk kerja tersebut, mengendalikan perubahan yang terjadi, dan mengaudit dan melaporkan perubahan yang dilakukan.

Kategori output

- Program komputer untuk sumber dan dieksekusi
- Dokumen untuk praktisi teknis dan pemakai
- Data yang diisi kedalam program dan keluaran dari program

Sumber perubahan mendasar

- Kondisi pasar
- Tuntutan pelanggan
- Reorganisasi/perubahan struktur tim pengembang
- Redefenisi sistem atau produk

Configuration Management (CM)

- Versi baru perangkat lunak dibuat karena ada perubahan:
 - Sistem operasi yang berbeda pada mesin;
 - Adanya perubahan pada kemampuan (menawarkan kemampuan yang berbeda;
 - Untuk kebutuhan tertentu.
- Kaitan-kaitan CM untuk memenej pengembangan perangkat lunak meliputi:
 - Penggantian sistem pada sebuah aktifitas tim;
 - Bertujuan untuk mengontrol biaya dan melibatkan membuat perubahan pada sistem.

- Melibatkan standar dan prosedur pengembangan aplikasi untuk memenej pengembangan perangkat lunak.
- Merupakan bagian dari proses manajemen kualitas.
- Hubungan CM, sistem perangkat lunak disebut juga *baselines* sebagai titik awal untuk mengembangkan lebih
a
njut.
- Merupakan sebuah konsep manajemen konfigurasi perangkat lunak untuk mengontrol perubahan selama perubahan masih dapat dibenarkan

Standarisasi CM

- CM selalu berdasarkan kepada standar yang diaplikasikan didalam sebuah organisasi.
- Standar didefinisikan bagaimana item-item diidentifikasi, bagaimana perubahan dikendalikan dan bagaimana memenej versi yang baru.
- Standar external yang mungkin mempengaruhi seperti: standar IEEE.
- Beberapa standar menerapkan proses pengembangan seperti model air terjun (waterfall), standar CM yang baru membutuhkan pengembangan yang evolusioner.

Frequent system building

- Memudahkan untuk menemukan masalah yang berasal dari interaksi komponen dimulai awal proses.
- Sebuah proses manajemen perubahan yang diperlukan untuk menelusuri masalah yang telah ditemukan dan diperbaiki.

Perencanaan CM

- Seluruh produk proses perangkat lunak mempunyai :
 - Spesifikasi;
 - Disain;
 - Program-program;
 - Test data;
 - User manuals.
- Semakin kompleks sistem perangkat lunak, semakin banyak dokumen-dokumen yang diperlukan dan perlu dibuat pemisahan (layak nya modul-modul).

Konfigurasi Data base

- Seluruh informasi CM harus di maintenance di dalam sebuah konfigurasi data base.
- Database (db) CM terutama dihubungkan untuk pengaturan perangkat lunak

Implementasi Db CM

- Menjadi bagian dari sebuah lingkungan terpadu untuk mendukung pengembangan perangkat lunak.
 - Db CM dan dokumen yang tertata seluruhnya dimainten pada sistem yang sama.
- Case tool terintegrasi dengan Db CM, sehingga memiliki hubungan erat antara CASE Tool dan CM Tool.
- Secara umum, dB CM dirawat secara terpisah sebagai bagian yang lebih fleksibel dan murah.

BAB III

PEMECAHAN MASALAH

Manajemen Perubahan

- Sistem perangkat lunak tunduk kepada perubahan yang berkelanjutan, kebutuhan perubahan berasal dari:
 - User.
 - Pengembang.
 - Kekuatan pangsa pasar.
- Manajemen perubahan mempunyai kaitan dengan pengawasan bahwa perubahan telah diterapkan dengan cara yang lebih hemat dan efektif.

Manajemen versi dan release

- Membuat suatu rencana versi sistem perangkat lunak.
- Rencanakan ketika suatu sistem dibuat versi yang baru.
- Pastikan prosedur manajemen versi dan tool diterapkan dengan baik.
- Rencanakan dan distribusikan release sistem yang baru.

Version/Variant/Release

- Version merupakan instansiasi dari sebuah sistem dimana secara fungsional berbeda cara dengan sistem yang lain.
- Variant merupakan instansiasi dari sebuah sistem dimana identik secara fungsional tetapi non-fungsional berbeda dengan sistem yang lain.
- Release merupakan sebuah instansiasi dari sebuah sistem yang telah didistribusikan ke user (pengguna) berada diluar lingkungan tim pengembang.

Identifikasi Versi

- Prosedur untuk identifikasi versi perlu menggambarkan suatu cara yang jelas untuk menjelaskan versi komponen yang ada.
- Ada tiga teknik dasar untuk mengidentifikasi komponen:
 - Penomoran versi;
 - Identifikasi berdasarkan atribut;
 - Identifikasi berorientasikan perubahan.

Penomoran versi

- Rencanakan penamaan yang sederhana menggunakan secara linear V1, V1.1, V1.2, V2.1, V2.2 dan lain-lain.
- Struktur asal secara aktual merupakan sebuah pohon atau sebuah jaringan yang dibandingkan secara urutan sequence.
- Nama-nama tidak mengandung arti.
- Rencana penamaan secara hirarki mengarahkan kearah yang lebih sedikit kesalahan didalam versi yang telah diidentifikasi.

Identifikasi Berdasarkan Atribut

- Atribut dapat dihubungkan dengan sebuah versi yang dikombinasikan atribut-atribut yang telah diidentifikasi,
- Seperti: atribut tanggal, creator (pencipta), bahasa pemrograman, pelanggan, status dan lain-lain.
- Lebih fleksibel dari skema penamaan yang eksplisit untuk pengembalian versi; bagaimanapun dapat menyebabkan permasalahan-permasalahan yang unik, serangkaian atribut yang dipilih untuk seluruh versi dapat dikenali dengan keunikannya.
- Didalam prakteknya, sebuah versi juga membutuhkan sebuah nama yang dihubungkan untuk memudahkan referensi

Identifikasi berorientasi perubahan

- Integrasi versi dan perubahan membuat perubahan versi awal.
- Digunakan untuk sistem dibandingkan komponen.
- Masing-masing usulan perubahan telah memiliki penjelasan untuk diimplementasikan.
- Perubahan diaplikasikan secara prinsip dalam sebuah versi sistem.

TOOLS SCM

- Briefcase toolkit
- Emacs
- Aegis
- BCS
- CVS
- ICE
- ODE
- Dll

BAB IV

KESIMPULAN

Software Configuration Management sangat penting bagi pengembang untuk mengetahui parameter-parameter mana yang harus diubah sesuai dengan kondisi yang terjadi pada waktu tersebut. Jika kita tidak mengendalikan perubahan, maka perubahan yang akan mengendalikan kita. Dan itu bukan hal yang baik. Perubahan yang tidak terkendali akan sangat mudah membuat menjadikan project software kita menjadi kacau. Akibatnya, kualitas perangkat lunak menjadi buruk dan proses delivery menjadi tertunda. Karena itu, manajemen perubahan (change management) adalah bagian penting dari manajemen mutu. Ketika masing-masing produk kerja bisa dimonitor, dilacak, dan dikendalikan; ketika setiap perubahan dapat dilacak dan dianalisis; ketika semua orang yang perlu tahu tentang perubahan telah diinformasikan - kita telah melakukannya dengan benar.

BAB V
DAFTAR PUSTAKA

- <https://beritati.blogspot.com/2019/04/manajemen-konfigurasi-software-scm.html>
- <https://docplayer.info/36101709-10-software-configuration-management.html>