

# LECTURE NOTES

## Software Engineering

### Minggu 7

## Software Configuration Management

## LEARNING OUTCOMES

Setelah menyelesaikan pembelajaran ini, mahasiswa akan mampu:

- LO 4 – Menganalisa proyek *management software*

***Outline Materi (Sub-Topic) :***

1. *The Software Configuration*
2. *SCM Repository*
3. *The SCM Process*
4. *SCM for Web Engineering*
5. Studi Kasus

## ISI MATERI

### 1. *The Software Configuration*

Software configuration merupakan rangkaian komponen yang saling berhubungan dengan software yang kita bangun. Terdapat istilah configuration items yaitu software dan elemen konfigurasinya, contoh:

- Program atau source code itu sendiri
- Dokumentasi
- Data

Jadi *Software* tidak hanya merupakan program dari *computer* dalam bentuk *source code*, tapi meliputi semua dokumen terkait dengan pengembangan *software* tersebut dan *database* yang digunakan.

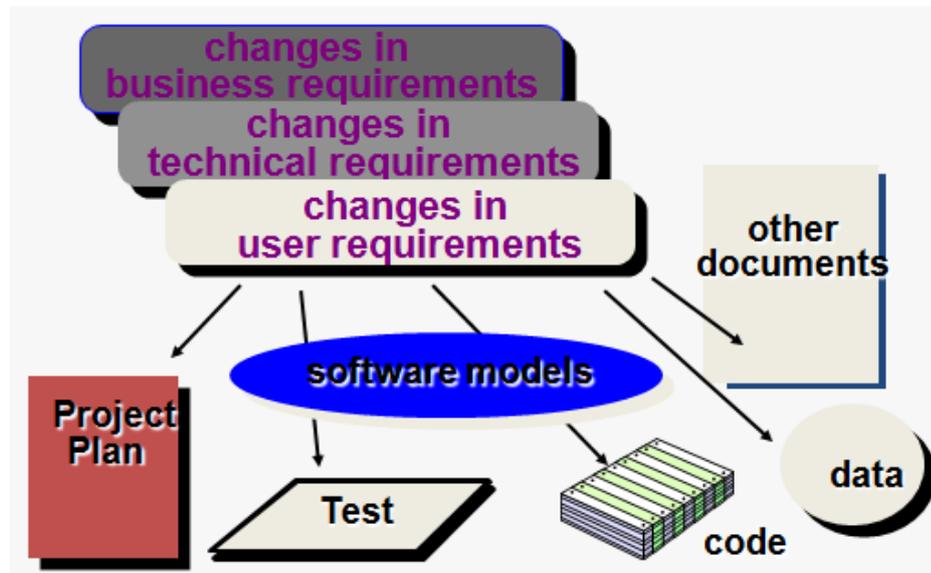
Contoh:

Anda membangun aplikasi penjualan berbasis web pada perusahaan Anda, maka yang termasuk komponen dari *software configuration* adalah:

- Source code dari software yang memiliki versi-versi tertentu, misalnya versi 1.0, versi 2.0 dan seterusnya
- Dokumentasi yang dihasilkan dan dibutuhkan, misalnya ada beberapa dokumen dengan versi-versi tertentu
- Data yang digunakan, dapat berupa *database* atau pun data mentah, misalnya data dalam bentuk *spreadsheet* atau bentuk lainnya.

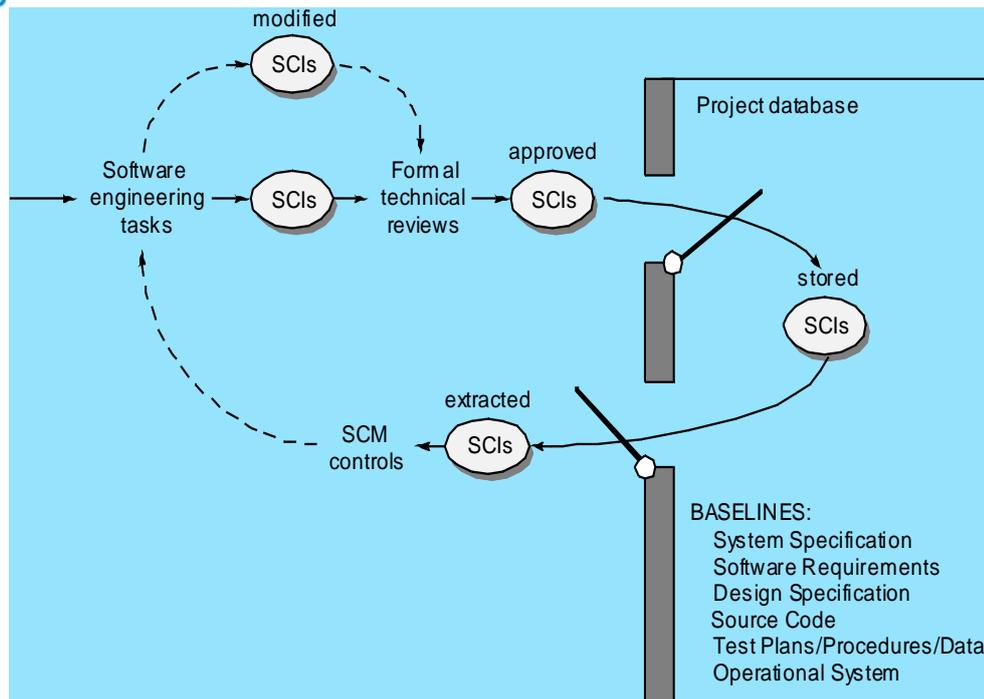
Software konfigurasi bertujuan untuk mengakomodasi perubahan yang mungkin terjadi, perubahan dapat terjadi dalam:

- *Requirement* bisnis, proses bisnis yang berubah sehingga menuntut penyesuaian dari program yang dibuat.
- *Requirement* teknis, perubahan dari kebutuhan teknis seperti perubahan platform menyebabkan penyesuaian program.
- *Requirement user*, perubahan kebutuhan dari user secara specific.



Pada *software configuration management* dilakukan dengan mengubah baseline yang ada. Baseline adalah spesifikasi atau produk yang secara formal telah direview dan disetujui, sehingga menjadi dasar dari pengembangan selanjutnya. Baseline hanya dapat diubah setelah mendapatkan persetujuan secara formal melalui prosedur control perubahan.

Baseline juga disebut sebagai milestone dalam proses pengembangan *software* yang ditandai dengan mengirimkan satu atau lebih item *software* konfigurasi.



## 2. *SCM Repository*

*Software Configuration Management repository* adalah sekumpulan mekanisme dan struktur data yang memungkinkan tim software untuk mengatur perubahan dalam cara yang *effective*. *SCM repository* menjalankan fungsi-fungsi berikut:

- *Data integrity*
- *Information sharing*
- *Tool integration*
- *Data integration*
- *Methodology enforcement*
- *Document standardization*

*Repository feature* merupakan *feature-feature* yang dapat digunakan untuk melakukan fungsi dari pengaturan konfigurasi software. Berikut adalah beberapa *feature* dari *SCM repository*:

a. *Versioning*

Pada *feature versioning*, dilakukan penyimpanan semua versi *software* untuk memungkinkan pengaturan secara efektif untuk produk yang akan direlease dan untuk mengijinkan *developer* untuk kembali ke versi sebelumnya.

b. *Dependency tracking and change management*

*Repository* mengatur hubungan yang sangat beragam antar data elemet yang disimpan di dalamnya.

c. *Requirement tracing*

Menyediakan kemampuan untuk melakukan pelacakan semua perancangan dan konstruksi komponen yang dihasilkan dari proses spesifikasi *requirement*.

d. *Configuration management*

Menjaga *track* dari sekumpulan konfigurasi yang merepresentasikan milestone dari proyek tertentu.

e. *Audit trails*

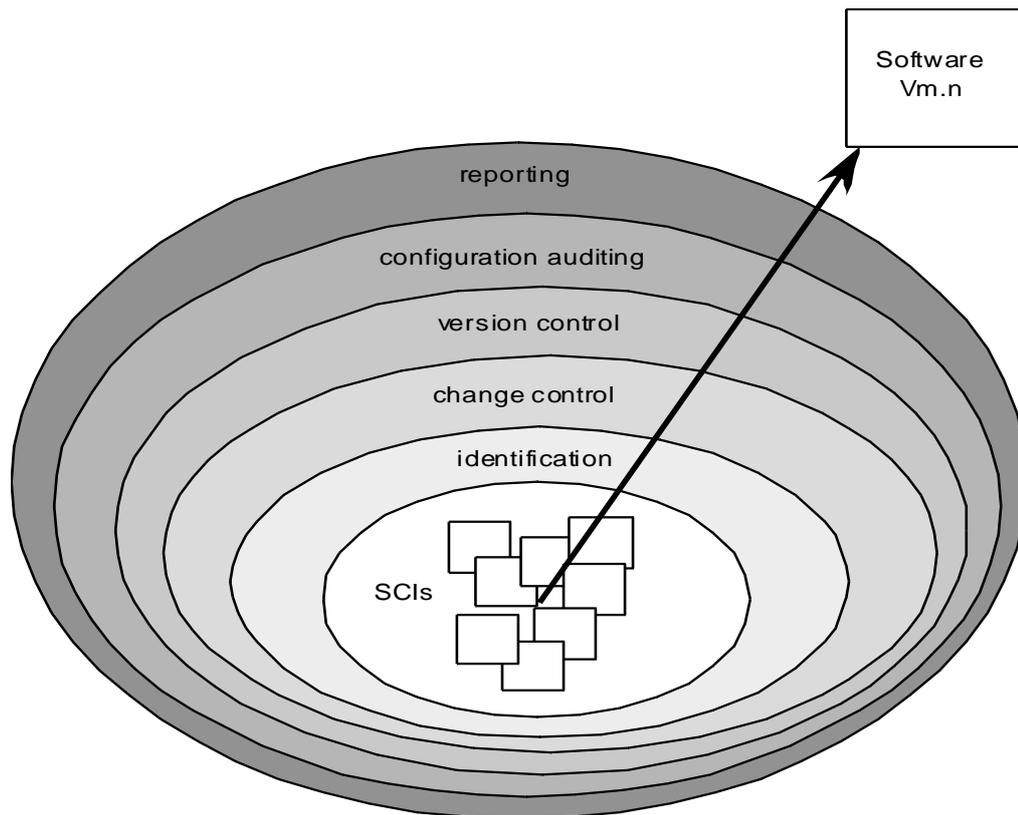
Menetapkan informasi tambahan mengenai kapan, bagaimana dan oleh siapa perubahan dibuat.

Beberapa elemen dari *Software Configuration Management*:

- **Component element**, sekumpulan dari *tool* yang dipasarkan dengan manajemen sistem data (seperti *database*) yang memungkinkan akses ke data tersebut dan melakukan pengaturan terhadap setiap item konfigurasi *software*.
- **Prosess element**, merupakan kumpulan dari prosedur dan tugas yang mendefinisikan pendekatan yang efektif untuk melakukan manajemen perubahan melibatkan semua yang terlibat dalam manajemen.

- **Construction elements**, kumpulan dari *tool-tool* yang mengautomasi pembuatan *software* dengan memastikan validasi komponen yang layak.
- **Human element**, untuk mengimplementasikan SCM yang efektif, tim *software* harus menggunakan sekumpulan *feature tool* dan proses.

### 3. The SCM Process



Urutan proses dalam SCM:

- Dimulai dari pengenalan kebutuhan akan perubahan
- Permintaan akan perubahan dari user
- Evaluasi dari *developer*
- Laporan perubahan dibuat
- Memutuskan otoritas *control* dari perubahan

- f. Permintaan dimasukkan kedalam daftar antrian untuk dieksekusi (jika diterima)
- g. Menugaskan individu pada setiap obyek konfigurasi
- h. “*check out*” obyek konfigurasi
- i. Melakukan perubahan
- j. Melakukan review/audit terhadap perubahan
- k. “*check in*” konfigurasi item yang telah diubah
- l. Menentukan baseline untuk pengujian
- m. Menjalankan pengujian kualitas dan aktifitas pengujian
- n. Mempromosikan perubahan untuk penyertaan di revisi berikutnya
- o. Membangun kembali versi *software* yang benar
- p. Melakukan review/audit terhadap perubahan dari semua item konfigurasi
- q. Mengikutkan perubahan di versi terbaru
- r. Melakukan distribusi versi baru

#### **4. *SCM for Web Engineering***

Dalam melakukan pengaturan konfigurasi untuk aplikasi *mobile* atau *web*, beberapa faktor yang harus diperhatikan:

##### **a. *Content***

Sebuah aplikasi web berisi berbagai content seperti *text*, grafik, *applet*, *script*, *audio/video*, *form*, *element* dari *page*, *table*, dan data

*streaming*. Tantangan dalam mengorganisasikan kumpulan konten menjadi kumpulan konfigurasi yang rasional dan kemudian menentukan mekanisme *control* konfigurasi yang layak.

b. *People*

Faktor manusia perlu diperhatikan karena pengembangan aplikasi web memiliki persentase yang *signifikan* untuk dilakukan dengan cara ad hoc, dimana setiap orang dapat membuat content dalam web tersebut.

c. *Scalability*

Dengan pertumbuhan ukuran dan kompleksitas, perubahan kecil dapat menyebabkan efek yang dapat menyebabkan masalah. Oleh karena itu, mekanisme konfigurasi control harus dapat diarahkan secara proposional dan *scalable*.

d. Politik

Siapa yang memiliki *web application*? Siapa yang bertanggung jawab dalam akurasi informasi yang diberikan di *web*? Siapa yang memastikan proses *quality control*? Siapa yang bertanggung jawab membuat perubahan?

## 5. Studi kasus

Praktik software configuration yang efektif pada perusahaan adalah penggunaan aplikasi manajemen pengembangan aplikasi, yang di dalamnya sudah terdapat version untuk melakukan manajemen konfigurasi ini. Terdapat beberapa fitur yang sangat berguna, misalnya:

- *Version control*
- *Access control* dari konfigurasi yang ada

- Manajemen untuk *source code*
- Fitur kolaborasi antar team di dalam pengelolaan *source code* tersebut

Penggunaan alat bantu atau tools untuk software configuration management ini sangat membantu para stakeholder untuk melakukan manajemen source code dan fungsi lainnya, dibandingkan dengan melakukan proses konfigurasi secara manual.

## DAFTAR PUSTAKA

- Software configuration Management,  
<http://www.sei.cmu.edu/reports/87cm004.pdf>
- Software engineering : a practitioners approach