**2. Urgensi & Konsep Dasar**

**Achmad Arwan, S.Kom., M.kom.**

**Manajemen konfigurasi perangkat lunak**

* SCM adalah disiplin - untuk mengelola evolusi sistem komputer di seluruh tahap pengembangan sistem. (Jessica Keys)
* SCM adalah seperangkat disiplin manajemen dalam proses rekayasa perangkat lunak untuk mengembangkan garis dasar. Setelah garis dasar ditentukan, manajemen konfigurasi meminimalkan risiko yang terkait dengan perubahan dengan menetapkan persetujuan formal dan proses pelacakan untuk perubahan. (Bernd Bruegge, Allen H. Dutoit)
* SCM adalah serangkaian kegiatan yang tujuan utamanya adalah mengidentifikasi CI (barang yang seharusnya berubah atau yang akan mengalami perubahan); untuk menemukan properti, karakteristik, dan saling ketergantungan dari item-item ini dan mencatatnya; untuk memantau barang-barang ini; untuk mengelola perubahan yang dilakukan pada item-item ini; untuk mendokumentasikan dan melaporkan proses perubahan; dan untuk memastikan bahwa barang yang dikirim lengkap dan memenuhi semua persyaratan. (Leon, Alexis)

**Semua tentang manajemen konfigurasi perangkat lunak**

* Disiplin Aturan
  + Rencana SCM
* Identifikasi item konfigurasi
  + mis., dokumentasi, program, fungsi, pustaka komponen, dan data)
  + Terkendali dengan tepat (Permintaan, analisis dampak, terapkan atau buang perubahan)
* Rekam Perubahan
  + Status permintaan perubahan harus dilacak dan dicatat dari titik asal sampai selesai.
* Mekanisme verifikasi
  + Sistem SCM harus memastikan bahwa apa yang dikembangkan dan disampaikan adalah persis apa yang diperlukan dan ditentukan.

**Jalur komunikasi**

5-10 komunikator?

**Bersama**

Perlu lebih banyak ruang kerja dan menciptakan masalah baru

**Masalah Pembaruan Bersamaan**

**Mengapa SCM?**

* Masalah:
  + Banyak orang harus mengerjakan perangkat lunak yang berubah
  + Lebih dari satu versi perangkat lunak harus didukung: ν
    - Sistem yang dirilis
    - Sistem terkonfigurasi khusus (fungsi berbeda)
    - Sistem sedang dikembangkan
* ν Perangkat lunak harus dijalankan pada mesin dan sistem operasi yang berbeda
* ν Perlu koordinasi
* ν Manajemen Konfigurasi Perangkat Lunak ν mengelola sistem perangkat lunak yang berkembang ν mengendalikan biaya yang diperlukan untuk melakukan perubahan pada suatu sistem

**Mengapa SCM**

* Melibatkan pengembangan dan penerapan prosedur dan standar untuk mengelola produk perangkat lunak yang berkembang
* Dapat dilihat sebagai bagian dari proses manajemen kualitas yang lebih umum
* Ketika dirilis ke CM, sistem perangkat lunak kadang-kadang disebut baseline karena mereka adalah titik awal untuk pengembangan lebih lanjut

**Pengembangan dan pengujian bersamaan**

* Waktu untuk pengiriman komponen sistem disepakati
* Versi baru dari sistem dibangun dari komponen-komponen ini dengan menyusun dan menautkannya
* Versi baru ini dikirimkan untuk pengujian menggunakan tes yang telah ditentukan
* Kesalahan yang ditemukan selama pengujian didokumentasikan dan dikembalikan ke pengembang sistem

**Konsep**

* Semua produk dari proses perangkat lunak mungkin harus dikelola
  + Program (misalnya, kode sumber, kode objek, program yang dapat dieksekusi, pustaka komponen, fungsi, dan subrutin);
  + Dokumentasi (misalnya, definisi persyaratan, analisis sistem, desain sistem, desain tingkat tinggi, desain tingkat rendah, spesifikasi pengujian, rencana pengujian, skrip pengujian, manual instalasi, catatan rilis, dan manual pengguna);
  + Data (data uji dan data proyek).

**Konsep: Item Konfigurasi (CI)**

* **Rencana proyek;**
* **Rencana SCM;**
* **RDD;**
* **Analisis, desain, pengkodean, pengujian, dan standar audit;**
* **Kesedihan;**
* **Dokumen desain sistem (SDD);**
* **Prototipe;**
* **Dokumen HLD;**
* **Dokumen LLD;**
* **Spesifikasi pengujian sistem;**
* **Rencana uji sistem;**
* **Kode sumber program;**
* **Kode objek dan executable;**
* **Spesifikasi tes unit;**
* **Rencana uji unit;**
* **Dokumen desain basis data;**
* **Data uji;**
* **Data proyek;**
* **Manual pengguna.**

**Konsep: Baseline**

**" *Spesifikasi atau produk yang telah ditinjau secara formal dan disetujui oleh manajemen yang bertanggung jawab, yang selanjutnya berfungsi sebagai dasar untuk pengembangan lebih lanjut, dan dapat diubah hanya melalui prosedur kontrol perubahan formal* ."**

**Contoh:**

**Baseline A: API suatu program sepenuhnya ditentukan; tubuh metode kosong.**

**Baseline B: Semua metode akses data diimplementasikan dan diuji; pemrograman GUI dapat dimulai.**

**Baseline C: GUI diimplementasikan, fase uji dapat dimulai.**

**Konsep: check-in, check out**

**Konsep: Versi, Percabangan**

Utama

Melepaskan

(Pelanggan)

Minor

Rilis Internal

(Pengembang)

Kecil

Revisi

(Pengembang)

7.5.5

**Penamaan CI**

* Nomor bagian
* Nama barang

**Nama harus deskriptif**

**Sebagai contoh, kita telah melihat bahwa versi pertama akan diidentifikasi sebagai 1.0 dan revisi berikutnya 1.1, 1.2, 2.0, 2.1, dan seterusnya. Cabang dari versi 1.1 akan diidentifikasi sebagai 1.1.1.0 dan versi selanjutnya 1.1.1.1 dan seterusnya. Cabang kedua akan diidentifikasi sebagai 1.1.2.0 dan seterusnya.**

**Melepaskan**

* File Instalasi
* File Data
* Program pengaturan
* Dokumentasi elektronik & kertas
  + Persyaratan instalasi (misalnya, spesifikasi sistem operasi, memori, dan spesifikasi yang diperlukan);
  + Cara memasang sistem dan cara menguji sistem untuk memastikan instalasi berhasil;
  + Cara meningkatkan dari versi sistem yang lebih lama;
  + Kunci atau nomor seri produk, jika nomor tersebut diperlukan untuk pemasangan;
  + Daftar kesalahan dan keterbatasan yang diketahui dari versi sistem tertentu dan daftar kesalahan yang diperbaiki dalam rilis saat ini;
  + Fitur baru yang diperkenalkan dalam rilis;
  + Instruksi untuk menghubungi pemasok sistem untuk dukungan teknis atau jika masalah muncul.

**Delta**

**Basis data SCM**

* Basis data CM digunakan untuk merekam semua informasi relevan yang terkait dengan konfigurasi.
* Basis data CM digunakan untuk mencatat rincian tentang CI, berisi informasi tentang permintaan perubahan (yang juga CI), statusnya, dan informasi mengenai proses peninjauan dan audit.
* Isi dan struktur database CM harus didefinisikan selama tahap desain sistem SCM dan harus didokumentasikan dalam rencana SCM.

**Database SCM harus dapat memberikan jawaban**

* Apa konfigurasi saat ini? Apa statusnya?
* Orang mana yang telah menerima versi sistem tertentu?
* Konfigurasi perangkat keras dan sistem operasi apa yang diperlukan untuk menjalankan a
* diberikan versi sistem?
* Berapa banyak versi sistem yang telah dibuat dan tanggal pembuatannya?
* Perubahan apa yang telah dilakukan pada perangkat lunak, dokumentasi, dan item lain dalam proyek? Siapa yang membuatnya, dan kapan mereka dibuat?
* Apakah perubahan disetujui oleh seseorang atau hanya dilakukan secara informal?
* Versi sistem apa yang mungkin terpengaruh jika komponen tertentu diubah?
* Berapa banyak permintaan perubahan yang tertunda pada item tertentu?
* Berapa banyak kesalahan yang dilaporkan ada di versi tertentu?
* Bisakah saya membuat ulang yang asli dari versi yang diubah atau versi yang diubah dari yang asli?
* Bisakah saya mencari tahu apa yang terjadi pada item tertentu di beberapa titik waktu, seperti perubahan apa yang dilakukan padanya dan seterusnya?
* Apakah perubahan saya memengaruhi orang lain?

**Manfaat SCM**

* Peningkatan daya saing organisasi;
* Layanan pelanggan yang lebih baik dan niat baik pelanggan yang ditingkatkan;
* Pengembalian investasi yang lebih baik;
* Peningkatan kontrol manajemen atas aktivitas pengembangan perangkat lunak;
* Peningkatan produktivitas pengembangan perangkat lunak;
* Penanganan kompleksitas perangkat lunak yang lebih mudah;
* Keamanan yang ditingkatkan;
* Penggunaan kembali perangkat lunak yang lebih tinggi;
* Biaya pemeliharaan perangkat lunak lebih rendah;
* Jaminan kualitas yang lebih baik;
* Pengurangan cacat dan bug;
* Identifikasi masalah lebih cepat dan perbaikan bug;
* Pengembangan yang bergantung pada proses daripada pengembangan yang tergantung pada orang;
* Jaminan bahwa sistem yang benar dibangun.

**Ringkasan**

* SCM adalah kegiatan yang penting, karena membantu kami untuk terus mencatat evolusi CI.
* SCM juga perlu direncanakan untuk memastikan semua kode telah dicatat sejak dikembangkan.
* SCM memiliki terminologi yang dikenal, mereka adalah item Konfigurasi, Baseline, Check In / Out, Versi, Percabangan, Rilis, Delta, Database SCM.

**Referensi**

* Lan sommervile
* Bernd Bruegge, Allen H. Dutoit
* Jessica Keys
* Leon, Alexis

**Item Konfigurasi, Baseline**

**Achmad Arwan, S.Kom., M.kom.**

**Item Konfigurasi**

* Item konfigurasi perangkat lunak (SCI) atau item konfigurasi (CI) Unit kode perangkat lunak yang disetujui, dokumen atau perangkat keras yang dirancang untuk manajemen konfigurasi dan diperlakukan sebagai entitas yang berbeda dalam proses manajemen konfigurasi perangkat lunak. (Galin, D. (2004))
* *"Kumpulan perangkat keras, perangkat lunak, atau keduanya, yang ditujukan untuk manajemen konfigurasi dan diperlakukan sebagai satu kesatuan dalam proses manajemen konfigurasi." (* Bernd Bruegge, Allen H. Dutoit)
* Item konfigurasi ditentukan oleh manajer konfigurasi
* Kegiatan ini memberikan dasar untuk kegiatan SCM lainnya

**Manfaat & Tujuan Item Konfigurasi**

* Untuk memilih produk berdasarkan fungsionalitas, verifikasi, dukungan, kompleksitas, risiko, dan aktivitas manajemen;
* Untuk menentukan struktur (hierarki) suatu produk dan organisasi serta hubungan dokumentasi konfigurasinya dan informasi produk lainnya;
* Untuk mendokumentasikan kinerja, antarmuka, dan atribut lainnya dari suatu produk;
* Untuk menentukan tingkat yang tepat dari tanda identifikasi produk dan
* dokumentasi;
* Untuk memberikan identitas unik pada suatu produk atau ke bagian komponen suatu produk;
* Untuk memberikan identitas unik pada dokumen teknis yang menggambarkan suatu produk;
* Untuk memodifikasi identifikasi produk dan dokumen untuk mencerminkan penggabungan perubahan besar;

**Manfaat & Tujuan Item Konfigurasi**

* Pemeliharaan kontrol rilis dokumen untuk manajemen baseline;
* Untuk membedakan antara versi produk;
* Untuk mengkorelasikan produk dengan pengguna terkait atau instruksi perawatan;
* Untuk mengelola informasi termasuk dalam format digital;
* Untuk mengkorelasikan masing-masing unit produk dengan jaminan dan kewajiban masa pakai;
* Untuk mengkorelasikan tingkat revisi dokumen dengan versi / konfigurasi produk;
* Untuk memberikan titik referensi untuk mendefinisikan perubahan dan tindakan korektif.

**Definisikan Item Konfigurasi**

* Tidak setiap entitas harus berada di bawah kendali manajemen konfigurasi sepanjang waktu
* Dua masalah:
  + **Apa:** Pemilihan Item Konfigurasi
    - Apa yang harus di bawah kendali konfigurasi?
  + **Kapan:** Kapan Anda mulai menempatkan entitas di bawah kontrol konfigurasi?
* Pilihan untuk Manajer Proyek:
  + Dimulai dengan Item Konfigurasi terlalu dini memperkenalkan birokrasi
  + Dimulai dengan Item Konfigurasi yang terlambat menyebabkan kekacauan.

**Definisikan Item Konfigurasi (lanjutan)**

* Memilih item konfigurasi yang tepat adalah keterampilan yang membutuhkan latihan
  + Sangat mirip dengan pemodelan objek
  + Gunakan teknik yang mirip dengan pemodelan objek untuk menemukan item konfigurasi!
    - Temukan item konfigurasi
    - Temukan hubungan antara item konfigurasi.

**dampak Pemilihan Ci**

* **Efek Memilih Terlalu Banyak Cis**
  + Meningkatnya beban administrasi dalam persiapan, pemrosesan, dan pelaporan
  + Peningkatan waktu dan biaya pengembangan serta potensi penciptaan desain yang tidak efisien.
  + Peningkatan potensial dalam upaya manajemen, kesulitan menjaga koordinasi, dan pembuatan persyaratan, desain, pengujian, dan spesifikasi sistem yang tidak perlu untuk setiap CI yang dipilih.
* **Efek Memilih Terlalu Sedikit Cis**
  + Kehilangan visibilitas ke tingkat yang diperlukan untuk melakukan pemeliharaan atau modifikasi.
  + Kesulitan dalam mengelola perubahan secara efektif (misalnya, mengelola perubahan pada masing-masing item, yang merupakan bagian dari CI).

**Calon Item Konfigurasi**

Item Konfigurasi

Calon

Model

Subsistem

Dokumen

Model objek

Model Dinamis

Basis data

Antarmuka pengguna

**. . . .**

Kode

Data

Tes Unit

RAD

ANEH

**. . . .**

**. . . .**

**. . . .**

"Proyek"

**Kemungkinan Pilihan Item Konfigurasi**

* Pernyataan masalah
* Rencana Pengelolaan Proyek Perangkat Lunak (SPMP)
* Dokumen Analisis Persyaratan (RAD)
* Dokumen Desain Sistem (SDD)
* Perjanjian Proyek
* Dokumen Desain Objek (ODD)
* Model Dinamis
* Model objek
* Model Fungsional
* Tes unit
* Strategi uji integrasi
* Kode sumber
* Spesifikasi API
* Input data dan basis data
* Rencana uji
* Uji data
* Perangkat lunak pendukung (bagian dari produk)
* Perangkat lunak pendukung (bukan bagian dari produk)
* Panduan pengguna
* Manual administrator

**Item konfigurasi umum**

* **Mendesain dokumen**
* **Kode perangkat lunak**
* \* Kode sumber
* \* Kode objek
* \* Perangkat lunak prototipe
* **File data**
* \* Uji kasus dan skrip uji
* \* Parameter, kode, dll.
* **Alat pengembangan perangkat lunak** (versi yang diterapkan pada tahap pengembangan dan pemeliharaan)
* \* Kompiler dan debugger
* \* Aplikasi generator
* \* Alat KASUS

**Contoh item konfigurasi dengan versi**

**Terminologi: Baseline**

**" *Spesifikasi atau produk yang telah ditinjau secara formal dan disetujui oleh manajemen yang bertanggung jawab, yang selanjutnya berfungsi sebagai dasar untuk pengembangan lebih lanjut, dan dapat diubah hanya melalui prosedur kontrol perubahan formal* ."**

**Contoh:**

**Baseline A: API suatu program sepenuhnya ditentukan; tubuh metode kosong.**

**Baseline B: Semua metode akses data diimplementasikan dan diuji; pemrograman GUI dapat dimulai.**

**Baseline C: GUI diimplementasikan, fase uji dapat dimulai.**

**Baseline**

* Baseline dapat ditentukan menggunakan bahasa konfigurasi, yang memungkinkan Anda untuk menentukan komponen apa yang termasuk dalam versi sistem tertentu.
* Baseline penting karena Anda sering harus membuat ulang versi spesifik dari sistem yang lengkap.
  + Misalnya, lini produk dapat dipakai sehingga ada versi sistem individual untuk pelanggan yang berbeda. Anda mungkin harus membuat ulang versi yang dikirimkan ke pelanggan tertentu jika, misalnya, bahwa pelanggan melaporkan bug di sistem mereka yang harus diperbaiki.

**Lebih lanjut tentang Baseline**

* Ketika sistem dikembangkan, serangkaian baseline dikembangkan, biasanya setelah tinjauan (tinjauan analisis, tinjauan desain, tinjauan kode, pengujian sistem, penerimaan klien, ...)
  + *Baseline pengembangan* (RAD, SDD, Tes Integrasi, ...)
    - Tujuan: Mengkoordinasikan kegiatan rekayasa.
  + *Garis dasar fungsional* (prototipe pertama, rilis alpha, rilis beta)
    - Sasaran: Dapatkan pengalaman pelanggan pertama dengan sistem fungsional.
  + *Garis dasar* produk (produk)
    - Sasaran: Mengkoordinasikan penjualan dan dukungan pelanggan.
* Ada banyak skema penamaan untuk baseline (1.0, 6.01a, ...)
* Skema 3 digit:

Utama

Melepaskan

(Pelanggan)

Minor

Rilis Internal

(Pengembang)

Kecil

Revisi

(Pengembang)

7.5.5

**Baseline dalam SCM**

Rilis Resmi

Baseline A (pengembangan)

Baseline B (fungsional)

Baseline C (uji beta)

Semua perubahan relatif terhadap baseline A

Semua perubahan relatif terhadap baseline B

Semua perubahan relatif terhadap baseline C

**Skema identifikasi versi tiga digit**

MUE.1.0.0: Rilis

MUE.1.2.1: Lepaskan

MUE.2.0.3: Lepaskan

MUE.0.0.1: Rilis

Rilis besar pertama

Rilis minor kedua

dengan perbaikan bug

Mayor kedua

rilis dengan tiga

serangkaian perbaikan bug

Rilis uji alfa

Skema identifikasi versi tiga digit

<version> :: = <konfigurasi item nama>. <major>. <minor>. <revision>

<major> :: = <bilangan bulat negatif>

<minor> :: = <bilangan bulat negatif>

<revision> :: = <bilangan bulat negatif>

**Sistem manajemen versi**

* Identifikasi versi dan rilis
  + Versi yang dikelola adalah pengidentifikasi yang diberikan ketika mereka dikirimkan ke sistem.
* Manajemen Penyimpanan
  + Untuk mengurangi ruang penyimpanan yang diperlukan oleh beberapa versi komponen yang hanya berbeda sedikit, sistem manajemen versi biasanya menyediakan fasilitas manajemen penyimpanan.
* Ubah rekaman riwayat
  + Semua perubahan yang dilakukan pada kode sistem atau komponen dicatat dan didaftar.

**Check-in dan check-out dari repositori versi**

Ver 2.1 IN

Ver 2.0 BL

Ver 1.0 BL

Ver 2.2 IN

Ver 3.0 BL

Ver 4.0 BL

Ver 4.1 IN

Model evolusi linier

Ver a1.0 BL

Ver e1.1 BL

Ver c2.0 BL

Ver c1.1 BL

Ver d1.1 IN

Ver d1.0 BL

Ver e1.0 BL

Ver b1.0 BL

Ver c1.0 BL

Ver b1.1 IN

Model evolusi pohon

Umum

Printer

Printer hitam

Printer berwarna

Printer-faks

Konfigurasi perangkat lunak

model evolusi

**Versi rilis Firefox**

**Ubah Log**

**Ringkasan**

* Item konfigurasi mencakup semua (kode, dokumen, tes data, manual, dll).
* Item konfigurasi juga memiliki versi, yang perubahannya didasarkan pada garis dasar.
* Salah satu versi penamaan Perangkat Lunak adalah skema 3 digit;
  + Melepaskan
  + Versi
  + Revisi
* Ada beberapa jenis evolusi pada versi (linear, pohon)

**Referensi**

* Galin, Daniel. *Jaminan kualitas perangkat lunak: dari teori ke implementasi* . Pearson Education India, 2004.
* Lan sommervile
* Bernd Bruegge, Allen H. Dutoit
* Jessica Keys

Google Terjemahan

**Teks asli**

Software configuration item (SCI) or configuration item (CI) An approved unit of software code, a document or piece of hardware that is designed for configuration management and treated as a distinct entity in the software configuration management process.

Sumbangkan terjemahan yang lebih baik

**Ubah Manajemen & Alat**

**Achmad Arwan, S.Kom., M.Kom.**

**Ubah Manajemen**

**Proses perencanaan, pengorganisasian, koordinasi dan pengendalian komposisi lingkungan , internal dan eksternal; untuk memastikan bahwa perubahan proses dilaksanakan sesuai dengan rencana yang disetujui dan tujuan keseluruhan memperkenalkan perubahan dicapai dengan sesedikit mungkin gangguan .**

* Proses **perencanaan, pengorganisasian, koordinasi dan pengendalian** komposisi **lingkungan** , internal dan eksternal; untuk memastikan bahwa perubahan proses dilaksanakan sesuai dengan **rencana yang disetujui** dan tujuan keseluruhan memperkenalkan perubahan dicapai dengan **sesedikit mungkin gangguan** .

Ubah Manajemen

Oseni, Yehezkiel

10/26/08,

"Ubah Manajemen dalam Perubahan Proses," 2007, Jurnal Online

**Sebelum kita membahas lebih lanjut tentang CM, maka kita harus tahu siapa yang berpartisipasi dalam proses manajemen konfigurasi**

* Sebelum kita membahas lebih lanjut tentang CM, maka kita harus tahu siapa yang berpartisipasi dalam proses manajemen konfigurasi
  + Manajer Konfigurasi
    - Bertanggung jawab untuk mengidentifikasi item konfigurasi
    - Juga sering bertanggung jawab untuk menentukan prosedur untuk membuat promosi dan rilis
  + Ubah Anggota Dewan Kontrol
    - Bertanggung jawab untuk menyetujui atau menolak permintaan perubahan
  + Pengembang
    - Membuat promosi yang dipicu oleh permintaan perubahan atau aktivitas normal pembangunan. Pengembang memeriksa perubahan dan menyelesaikan konflik
  + Auditor
    - Bertanggung jawab untuk pemilihan dan evaluasi promosi untuk rilis dan untuk memastikan konsistensi dan kelengkapan rilis ini

Ubah Manajemen

**Manajemen perubahan adalah penanganan permintaan perubahan**

* Manajemen perubahan adalah **penanganan** permintaan perubahan
  + Permintaan perubahan mengarah pada pembuatan rilis baru
* Proses manajemen perubahan umum:
  + Perubahan diminta (ini dapat dilakukan oleh siapa saja termasuk pengguna dan pengembang)
  + Permintaan perubahan dinilai berdasarkan persyaratan dan kendala proyek
  + Setelah penilaian, permintaan perubahan diterima atau ditolak
  + Jika diterima, perubahan ditugaskan ke pengembang dan diterapkan
  + Perubahan yang diterapkan diaudit
* Kompleksitas proses manajemen perubahan bervariasi sesuai dengan proyek. Proyek-proyek kecil dapat melakukan permintaan perubahan secara informal dan cepat sementara proyek-proyek kompleks memerlukan formulir permintaan perubahan yang terperinci dan persetujuan resmi oleh satu manajer lagi.

Ubah Manajemen

**Ubah Manajemen**

**Mengontrol Perubahan**

Pengguna

Melepaskan

Perangkat lunak

Gudang

Menguasai

Direktori

Programmer

Promosi

* Dua jenis perubahan kontrol:
  + Promosi: Keadaan pengembangan internal suatu perangkat lunak diubah
  + Rilis: Sistem perangkat lunak yang diubah dibuat terlihat di luar organisasi pengembangan.

**Terminologi: Direktori SCM**

* Direktori Programmer (IEEE: Dynamic Library)
  + Perpustakaan untuk memegang entitas perangkat lunak yang baru dibuat atau dimodifikasi
  + Ruang kerja programmer dikendalikan oleh programmer saja
* Master Directory (IEEE: Controlled Library)
  + Mengelola baseline saat ini dan untuk mengendalikan perubahan yang dilakukan padanya
  + Perubahan harus diotorisasi
* Gudang Perangkat Lunak (IEEE: Static Library)
  + Arsip untuk berbagai baseline yang dirilis untuk penggunaan umum
  + Salinan baseline ini dapat dibuat tersedia untuk organisasi yang meminta.

**Direktori SCM Standar**

Foo'95

Foo'98

* Direktori Programmer
  + (IEEE Std: "Dynamic Library")
  + Sepenuhnya di bawah kendali satu programmer
* Direktori Induk
  + (IEEE Std: "Perpustakaan Terkendali")
  + Direktori pusat semua promosi
* Gudang Perangkat Lunak
  + (IEEE Std: "Static Library")
  + Baseline dirilis secara eksternal.

Sumber pusat

arsip kode

Melepaskan

Promosi

**Promosi dan Rilis adalah Operasi pada CI**

**. . . .**

**Item Konfigurasi**

memajukan()

melepaskan()

Dokumen

ANEH

Item Konfigurasi

Model

Subsistem

Model objek

Model Dinamis

Basis data

Antarmuka pengguna

Kode

Data

Tes Unit

RAD

**. . . .**

**. . . .**

**. . . .**

"Proyek"

**Mari Membuat Model Obyek untuk Manajemen Konfigurasi**

Melepaskan

Gudang

Promosi

Direktori Induk

\*

\*

**"Promosi disimpan di direktori master dan rilis disimpan di repositori"**

Masalah: Mungkin ada banyak promosi dan banyak rilis

Solusi: Gunakan Multiplicity

**Mari Membuat Model Obyek untuk Manajemen Konfigurasi**

Melepaskan

Promosi

Gudang

Direktori Induk

\*

\*

**Wawasan: Promosi dan Rilis keduanya versi**

Versi

Solusi: Gunakan Warisan

**Mari Membuat Model Obyek untuk Manajemen Konfigurasi**

Melepaskan

Promosi

Gudang

Direktori Induk

\*

\*

**Masalah: Item konfigurasi dapat memiliki beberapa versi**

Versi

Item Konfigurasi

\*

Solusi: Buat asosiasi 1-banyak antara Item Konfigurasi dan Versi

**Mari Membuat Model Obyek untuk Manajemen Konfigurasi**

Melepaskan

Promosi

Gudang

Direktori Induk

\*

\*

**Masalah: Item konfigurasi dapat dikelompokkan sendiri**

Versi

Solusi: Gunakan pola desain komposit

Item Konfigurasi

\*

\*

Konfigurasi

barang

Item yang dikendalikan

CM Agregat

**Model Objek Akhir untuk Manajemen Konfigurasi (Diagram Kelas UML)**

Versi

\*

Terkendali

barang

\*

CM Agregat

Konfigurasi

barang

Melepaskan

Promosi

Gudang

Menguasai

Direktori

\*

\*

**Alat Manajemen Konfigurasi Perangkat Lunak**

* RCS: Yang pertama di blok [Tichy 1975]
* CVS (Kontrol Versi Serentak)
  + berdasarkan RCS, memungkinkan konkurensi tanpa mengunci
  + http://www.cvshome.org/
* Subversi
  + Berdasarkan pada CVS
  + Proyek Sumber Terbuka ( [http://subversion.tigris.org/](https://translate.google.com/translate?hl=id&prev=_t&sl=en&tl=id&u=http://subversion.tigris.org/project_packages.html) )
* Terpaksa
  + Server repositori, melacak aktivitas pengembang
  + http://www.perforce.com
* ClearCase
  + Beberapa server, pemodelan proses, mekanisme pemeriksaan kebijakan
  + http://www.rational.com/products/clearcase/

**Alat Manajemen Konfigurasi Perangkat Lunak**

* VSS - Sumber visual aman
* CVS- Sistem versi bersamaan
* Kasus Yang Jelas Rasional
* SVN- Subversi
* Terpaksa
* TortoiseSVN
* Konser tim IBM Rational
* Manajemen versi manajemen Konfigurasi IBM
* Pisau cukur
* Sistem kontrol versi Quma
* SourceAnywhere

**Manajemen Konfigurasi Perangkat Lunak Pemodelan**

* Model IEEE
* Model RCS
* Model CVS
* Model SVN

**Model IEEE (Diagram Kelas UML)**

Versi

\*

Terkendali

Barang

\*

CM Agregat

Konfigurasi

Barang

Melepaskan

Promosi

Gudang

Menguasai

Direktori

\*

\*

**Model RCS (Diagram Kelas UML)**

Versi

\*

Mengajukan

Menandai

\*

\*

Tidak ada agregat CM

Konkurensi dikendalikan dengan penguncian

Nomor versi digunakan untuk membedakan promosi dari rilis

Tag digunakan untuk mengidentifikasi garis dasar

Pengguna

kunci

\*

0..1

Gudang

\*

Copy pekerjaan

\*

1

1

leluhur

\*

1

**Penomoran Versi RCS (diagram instance UML)**

1.1: Revisi

1.3.1.1: Revisi

1.3.2.1: Revisi

1.2: Revisi

1.3: Revisi

1.4: Revisi

1.3.1.2: Revisi

Cabang 1.3.1

Cabang 1.3.2

Bagasi

**Model CVS (Diagram Kelas UML)**

Revisi

Barang Terkendali

\*

Direktori

Mengajukan

Gudang

Superset dari RCS

Direktori juga dikendalikan

Perubahan bersamaan ditangani dengan salin / ubah / gabungkan (sebagai lawan penguncian)

Menandai

\*

\*

\*

\*

Copy pekerjaan

Pengguna

\*

1

1

leluhur

\*

1

**Salin / Ubah / Gabung dalam CVS**

:Gudang

memodifikasi()

Alice

: Pengembang

lapor masuk (ac)

Periksa()

aliceCopy: WorkingCopy

Periksa()

joe

: Pengembang

memodifikasi()

perbarui (jc)

checkin (jc)

menggabungkan (ac)

joeCopy: WorkingCopy

Jika perubahan tumpang tindih, Joe CVS menyelesaikan konflik dengan mengedit salinan gabungan.

**Mengganti nama dalam CVS**

* Cvs tidak mendukung penggantian nama
* Sebaliknya, pengguna akhir
  + mengganti nama file dalam copy pekerjaan
  + menghapus nama lama dari repositori
  + menambahkan nama baru di repositori
* Dua pohon sejarah terputus

editor1.3

editor1.4

myEditor1.1

myEditor1.2

cvs hapus editor

cvs tambahkan myEditor

Hubungan antara dua pohon revisi hilang

**Subversi**

* Proyek Sumber Terbuka [http://subversion.tigris.org/](https://translate.google.com/translate?hl=id&prev=_t&sl=en&tl=id&u=http://subversion.tigris.org/project_packages.html)
* Berdasarkan pada CVS
  + Antarmuka dan fitur subversi mirip dengan CVS
  + Perintah: checkout, tambah, hapus, komit, beda
* Perbedaan dengan CVS
  + Versi mengontrol pemindahan, penggantian nama, penyalinan file dan direktori
  + Metadata versi terkontrol dari file dan direktori
* Opsi Server
  + Instalasi mandiri
  + Terintegrasi ke dalam server web Apache
* Waktu untuk manajemen cabang tidak tergantung pada ukuran sistem (CVS membuat salinan fisik file, Subversion hanya menggunakan tag).

**Model Subversi (Diagram Kelas UML)**

Revisi

Barang Terkendali

\*

Direktori

Mengajukan

Gudang

Fungsi serupa dari CVS

Pengubahan nama / pemindahan file dikendalikan

Nomor versi global

Cabang dan tag adalah salinan

\*

\*

Copy pekerjaan

Pengguna

\*

1

1

leluhur

\*

1

salinan

\*

1

**Skema Nomor Revisi Global SVN**

Nomor revisi global berlaku untuk seluruh pohon

Nomor revisi ditingkatkan untuk setiap check in, terlepas dari tingkat perubahannya.

**Mengganti nama dalam Subversion**

* subversi memperlakukan nama item konfigurasi sebagai atribut
* Perubahan nama diberikan sama dengan perubahan konten
* Ada satu pohon sejarah
* Mengganti nama konflik (dua pengguna mengganti nama file yang sama pada saat yang sama) terdeteksi

editor 3

editor 4

myEditor 5

svn ganti nama editor myEditor

**Menyalin dalam Subversi**

* Subversi melacak salinan dengan cara yang mirip dengan mengganti nama.
* Riwayat item konfigurasi mencakup riwayat dokumen asli yang disalin.
* Salinan murah, perbedaan (yaitu, perubahan nama) disimpan.
* Cabang dalam subversi adalah salinan sederhana.

editor 3

editor 4

myEditor 5

svn copy editor myEditor

editor 6

**Tapi tunggu, apakah semua itu cukup untuk proses CM?**

* Tapi tunggu, apakah semua itu cukup untuk proses CM?
  + CM **menangani** permintaan perubahan, bukan?
  + Jadi, apa sebenarnya **penanganannya** ?
    - perencanaan, pengorganisasian, koordinasi dan pengendalian
* Apakah Anda ingat proses CM?
  + Perubahan diminta (ini dapat dilakukan oleh siapa saja termasuk pengguna dan pengembang)
  + Permintaan perubahan dinilai berdasarkan persyaratan dan kendala proyek
  + Setelah penilaian, permintaan perubahan diterima atau ditolak
  + Jika diterima, perubahan ditugaskan ke pengembang dan diterapkan
  + Perubahan yang diterapkan diaudit

Ubah Manajemen

**Jadi, kita masih memiliki beberapa topik “panas” untuk dibahas dalam kuliah ini**

* Jadi, kita masih memiliki beberapa topik “panas” untuk dibahas dalam kuliah ini
  + Promosikan / lepaskan kebijakan
  + Kegiatan dan tanggung jawab

Ubah Manajemen

**Ubah Kebijakan**

**Tujuan dari kebijakan perubahan adalah untuk menjamin bahwa setiap promosi atau rilis sesuai dengan kriteria yang diterima secara umum**

* Tujuan dari kebijakan perubahan adalah untuk **menjamin** bahwa setiap promosi atau rilis **sesuai** dengan kriteria yang diterima secara umum
* Contoh untuk kebijakan perubahan:
  + "Tidak ada pengembang yang diizinkan mempromosikan kode sumber yang tidak dapat dikompilasi tanpa kesalahan dan peringatan."
  + "Tidak ada garis dasar yang dapat dirilis tanpa diuji beta oleh setidaknya 500 orang eksternal."

Ubah Kebijakan

**Mengontrol Perubahan**

Pengguna

Melepaskan

Perangkat lunak

Gudang

Menguasai

Direktori

Programmer

Promosi

* Dua jenis perubahan kontrol:
  + Promosi: Keadaan pengembangan internal suatu perangkat lunak diubah
  + Rilis: Sistem perangkat lunak yang diubah dibuat terlihat di luar organisasi pengembangan
* Pendekatan untuk mengendalikan perubahan (Ubah Kebijakan):
  + Informal (baik untuk lingkungan dan promosi jenis penelitian)
  + Pendekatan formal (baik untuk CI yang dikembangkan secara eksternal dan untuk rilis)

***Promosi***

***Kebijakan***

***Melepaskan***

***Kebijakan***

**Terminologi: Versi, Revisi, Rilis**

* Versi
  + Rilis awal atau rilis ulang item konfigurasi yang terkait dengan kompilasi lengkap atau kompilasi ulang item. Versi yang berbeda memiliki fungsi yang berbeda
* Revisi
  + Ubah ke versi yang hanya memperbaiki kesalahan dalam desain / kode, tetapi tidak mempengaruhi fungsi yang didokumentasikan
* Melepaskan
  + Distribusi formal dari versi yang disetujui

**Kegiatan dan Tanggung Jawab**

**Kegiatan CM**

* Identifikasi Item Konfigurasi
  + Telah dibahas pada slide awal kuliah ini
* Kontrol Konfigurasi
* Status Konfigurasi, Akuntansi
* Audit dan Ulasan Konfigurasi

**Kontrol Konfigurasi**

* Tetapkan langkah-langkah berikut
  + Cara mengidentifikasi kebutuhan untuk perubahan (tata letak formulir permintaan perubahan)
  + Analisis dan evaluasi permintaan perubahan
  + Persetujuan atau penolakan atas suatu permintaan
  + Verifikasi, implementasi, dan rilis perubahan

**mengubah proses manajemen**

**Mengenali**

**Analisis**

**Memutuskan**

**Menerapkan**

**Ubah Permintaan**

* Menentukan prosedur untuk meminta perubahan ke item konfigurasi yang ditentukan dan informasi yang akan didokumentasikan:
  + Nama dan versi item konfigurasi tempat masalah muncul
  + Nama dan alamat Originator
  + Tanggal permintaan
  + Indikasi urgensi
  + Kebutuhan akan perubahan
  + Deskripsi perubahan yang diminta

**Evaluasi Perubahan**

* Menentukan analisis yang diperlukan untuk menentukan dampak perubahan yang diusulkan dan prosedur untuk meninjau hasil analisis.

**Ubah Persetujuan atau Penolakan**

* Menjelaskan pengaturan papan kontrol konfigurasi (CCB)
  + Dapat seorang individu atau kelompok
  + Beberapa level CCB juga dimungkinkan, tergantung pada
  + Dalam upaya pengembangan kecil satu tingkat CCB sudah cukup
* Juga menunjukkan tingkat otoritas CCB dan tanggung jawabnya
  + Secara khusus, harus ditentukan ketika CCB dipanggil.

**Menerapkan Perubahan**

* Menentukan aktivitas untuk memverifikasi dan menerapkan perubahan yang disetujui
* Permintaan perubahan yang telah diisi harus mengandung informasi ini:
  + Permintaan perubahan asli
  + Nama dan versi item konfigurasi yang terpengaruh
  + Tanggal verifikasi dan pihak yang bertanggung jawab
  + Pengidentifikasi versi baru
  + Tanggal rilis atau instalasi dan pihak yang bertanggung jawab.
* Tentukan juga kegiatan untuk
  + Pengarsipan permintaan perubahan selesai
  + Perencanaan dan kontrol rilis
  + Bagaimana mengoordinasikan banyak perubahan
  + Cara menambahkan item konfigurasi baru ke konfigurasi
  + Cara menyampaikan garis dasar baru.

**Status Konfigurasi, Akuntansi**

* Jawab pertanyaan-pertanyaan berikut:
  + Elemen apa yang harus dilacak dan dilaporkan untuk baseline dan perubahan?
  + Apa jenis laporan akuntansi status yang akan dihasilkan? Berapa frekuensi mereka?
  + Bagaimana informasi dikumpulkan, disimpan, dan dilaporkan?
  + Bagaimana akses ke data status manajemen konfigurasi dikendalikan?

**Audit dan Ulasan Konfigurasi**

* Identifikasi audit dan ulasan untuk proyek
  + Audit menentukan untuk setiap item konfigurasi jika memiliki karakteristik fisik dan fungsional yang diperlukan
  + Tinjauan adalah alat manajemen untuk menetapkan garis dasar.
* Untuk setiap audit atau tinjauan rencana harus mendefinisikan:
  + Tujuan
  + Item Konfigurasi yang sedang ditinjau
  + Jadwal untuk ulasan
  + Prosedur untuk melakukan tinjauan
  + Peserta berdasarkan jabatan
  + Dokumentasi yang diperlukan
  + Prosedur untuk mencatat kekurangan dan cara memperbaikinya
  + Kriteria persetujuan.

**Terima kasih**

**Akuntansi status**

**pengantar**

**Status konfigurasi Pendahuluan AKUNTANSI**

* + CSA adalah elemen CM yang terdiri dari perekaman dan pelaporan informasi yang diperlukan untuk mengelola sistem perangkat lunak dan karakteristiknya secara efektif.
  + Informasi ini termasuk daftar identifikasi konfigurasi yang disetujui, status perubahan yang diusulkan untuk konfigurasi, dan status implementasi perubahan yang disetujui.

**Status konfigurasi Pendahuluan AKUNTANSI**

* + Dengan kata lain, fungsi akuntansi status adalah pencatatan dan pelaporan informasi yang diperlukan untuk mengelola CI secara efektif, termasuk, tetapi tidak terbatas pada, catatan dokumentasi konfigurasi yang disetujui dan nomor identifikasi, status perubahan yang diusulkan, status implementasi perubahan yang disetujui , status CR dan PR yang tertunda atau terbuka, dan status pembangunan semua unit CI.

**Status konfigurasi Pendahuluan AKUNTANSI**

* + Tujuan dari akuntansi status adalah untuk memberi informasi kepada manajer, pengguna, pengembang, dan pemangku kepentingan proyek lainnya tentang berbagai tahap konfigurasi dan evolusi mereka.
  + Ini menyiratkan tiga tugas dasar: pengambilan data, perekaman data, dan pembuatan laporan
  + Kegiatan akuntansi status penting untuk menjaga kelangsungan proyek dan menghindari duplikasi usaha.

**Status konfigurasi Pendahuluan AKUNTANSI**

* + CSA melibatkan penyimpanan dan pemeliharaan hal-hal berikut:
    - Informasi tentang dokumentasi konfigurasi (seperti pengidentifikasi dokumen dan tanggal efektif);
    - Informasi tentang konfigurasi produk (seperti nomor komponen atau perubahan yang dipasang pada unit tertentu);
    - Informasi tentang dokumentasi operasional dan pemeliharaan produk (seperti dokumen yang dipengaruhi oleh setiap perubahan dan status pembaruan mereka)
    - Informasi tentang proses CM (seperti status CR).

**Status konfigurasi ACCOUNTING status yang baik**

* + Sistem akuntansi status yang baik harus dapat menjawab pertanyaan seperti berikut dan banyak lagi:
    - Apa status suatu barang?
    - Sudahkah CR tertentu disetujui?
    - Apa status CR dan PR yang tertunda atau terbuka?
    - Item apa saja yang dipengaruhi oleh CR tertentu?
    - Kapan CR disetujui, dan siapa yang menyetujuinya?
    - Siapa yang melakukan perubahan untuk CR tertentu, dan kapan itu selesai?

**Status konfigurasi AKUNTANSI Status Bagus**

* + Sistem akuntansi status yang baik harus dapat menjawab pertanyaan seperti berikut dan banyak lagi:
    - Siapa yang memeriksanya? Siapa yang menyetujuinya?
    - Versi item mana yang mengimplementasikan CR yang disetujui?
    - CR apa yang ditugaskan kepada siapa?
    - Berapa banyak CR prioritas tinggi yang saat ini tidak diterapkan?
    - Apa yang berbeda dari versi sistem yang baru?
    - Berapa banyak CR yang diinisiasi setiap bulan, dan berapa tingkat persetujuannya?
    - Berapa PR yang diajukan setiap bulan, dan bagaimana status masing-masing?
    - Apa penyebab utama masalah atau cacat tersebut?

**Database akuntansi status**

**status Basis Data AKUNTANSI**

* + Daftar berikut memberikan informasi yang diperlukan untuk laporan akuntansi status sederhana:
    - Nama CI dan nomor identifikasi;
    - Nama CI tingkat lebih tinggi berikutnya;
    - Tanggal mulai desain;
    - Merancang tanggal persetujuan dan nomor revisi;
    - Tanggal mulai pengkodean;
    - Tanggal selesai pengkodean;
    - Tanggal mulai pengujian;
    - Tanggal selesai pengujian;
    - Membangun tanggal mulai dan selesai dan nomor revisi;

**status Basis Data AKUNTANSI**

* + Daftar berikut memberikan informasi yang diperlukan untuk laporan akuntansi status sederhana:
    - Tanggal penggabungan sistem;
    - Tanggal pengiriman sistem dan nomor revisi;
    - Tanggal CR, nomor CR, dan nama pemohon;
    - Ubah tanggal disposisi;
    - Ubah tanggal pendirian, nama pelaksana, dan nomor revisi.

**status Basis Data AKUNTANSI**

* + Fitur penting dari basis data adalah kemampuan untuk melacak sistem perangkat lunak ke atas dan ke bawah.
  + Ini adalah kemampuan untuk melacak hubungan persyaratan perangkat lunak turun melalui berbagai tingkatan (misalnya, sistem, program, modul, dan unit-proses yang dikenal sebagai menelusuri dalam terminologi manajemen pengetahuan-atau dalam arah yang berlawanan (yaitu, melacak hubungan dari elemen terkecil (unit) ke atas melalui modul, program, sistem, desain, dan akhirnya, persyaratan).

**Pentingnya status AKUNTANSI**

* + Akuntansi status mengacu pada fungsi manajemen informasi (atau manajemen data) dalam sistem SCM. Untuk setiap CI yang dirancang, dikembangkan, ditinjau, disetujui, dirilis, dan didistribusikan, kegiatan yang dilakukan dan informasi lainnya seperti bagaimana mereka dilakukan; mengapa, di mana, dan kapan mereka selesai; dan siapa yang melakukannya harus direkam.
  + Rincian ini akan berguna untuk semua orang yang terlibat dalam proyek dengan berbagai cara.
    - Kebutuhan informasi pengembang berbeda dari manajer proyek, tetapi setiap anggota tim proyek dan fungsi dukungan akan memerlukan setidaknya beberapa informasi.

**Pentingnya status AKUNTANSI**

* + Akuntansi status mengacu pada fungsi manajemen informasi (atau manajemen data) dalam sistem SCM. Untuk setiap CI yang dirancang, dikembangkan, ditinjau, disetujui, dirilis, dan didistribusikan, kegiatan yang dilakukan dan informasi lainnya seperti bagaimana mereka dilakukan; mengapa, di mana, dan kapan mereka selesai; dan siapa yang melakukannya harus direkam.
  + Rincian ini akan berguna untuk semua orang yang terlibat dalam proyek dengan berbagai cara.
    - Kebutuhan informasi pengembang berbeda dari manajer proyek, tetapi setiap anggota tim proyek dan fungsi dukungan akan memerlukan setidaknya beberapa informasi.

**Pentingnya status AKUNTANSI**

* + Misalnya, informasi yang diberikan oleh fungsi akuntansi status membantu manajemen proyek
  + mengidentifikasi masalah, menentukan sumber masalah, dan mengambil tindakan korektif sebelum situasi menjadi tidak terkendali.
    - Dari laporan yang dihasilkan dan dengan membuat permintaan khusus, manajemen proyek dapat menentukan bagaimana kinerja proyek dan membandingkan kinerja dengan rencana. Seseorang juga dapat melihat jenis perubahan, laju perubahan, penyebab perubahan, biaya, dan banyak faktor lainnya dan mengambil tindakan yang diperlukan.

**Laporan status akuntansi**

**status Laporan AKUNTANSI**

* + Faktor-faktor yang harus dipertimbangkan saat merancang persyaratan pelaporan dan laporan sistem meliputi:
    - Audiens untuk laporan;
    - Informasi yang terkandung dalam setiap laporan;
    - Kebutuhan akan laporan rutin atau laporan yang disediakan secara ad hoc;
    - Frekuensi laporan;
    - Daftar distribusi.

**status Laporan AKUNTANSI**

* + Ada 2 kategori laporan yang dapat dihasilkan dalam Status Accounting:
    - Laporan rutin
    - Laporan ad hoc

**status AKUNTANSI, Laporan, ROUTIN**

* + Contoh laporan rutin adalah:
    - Ubah Log
    - Laporan perkembangan
    - Laporan Status CI
    - Log Transaksi

**Laporan rutin, ubah log**

* + Log perubahan harus berisi semua informasi tentang CR di sistem
  + Frekuensi distribusi yang biasa adalah bulanan. Laporan ini harus berisi informasi seperti nomor CR, status, nama pencetusnya, item yang terkena dampak, tanggal asal, deskripsi perubahan, dan nama pelaksana.

**Laporan kemajuan Laporan Rutin**

* + Laporan kemajuan, yang merupakan ringkasan dari kemajuan pembangunan sejak laporan terakhir dikeluarkan, digunakan terutama oleh manajemen untuk memantau kemajuan proyek.
  + Laporan ini harus mencakup informasi seperti periode pelaporan (dari dan sampai tanggal), ID1 tugas, deskripsi singkat pekerjaan yang dilakukan selama periode tugas, dan status tugas (misalnya, selesai atau persen selesai).

**Laporan Status Laporan rutin**

* + Laporan ini disiapkan untuk meringkas status semua CI dalam sistem dan harus mencakup informasi seperti daftar CI, deskripsi, dan lokasi Cis (perpustakaan terkontrol tempat mereka disimpan)
  + Deskripsi CI harus mencakup nama, nomor versi, dan detail item yang tergantung.

**Log transaksi Laporan Rutin**

* + Log ini berisi transaksi yang telah terjadi pada barang, dicatat dalam urutan kronologis
    - Log harus berisi perincian seperti nomor transaksi, tanggal, pencetus (orang yang membuat entri), sifat entri (apa yang dianggap entri), barang yang terkena dampak, aktivitas (misalnya, CR, persetujuan CCB, analisis, dan PR) , deskripsi, peserta (orang-orang yang terlibat), item yang terkena dampak (item yang dipengaruhi oleh aktivitas), dan komentar.
  + Tujuan dari log transaksi adalah untuk mengetahui apa yang terjadi selama periode tertentu, katakanlah, “Apa kegiatan yang dilakukan pada mm / hh / tttt?” Di sini, idenya bukan untuk memberikan deskripsi terperinci tentang bagaimana segala sesuatu dilakukan, tetapi untuk memberi seseorang gambaran tentang apa yang terjadi selama periode tertentu.

**status Laporan AKUNTANSI AD Hoc**

* + Beberapa contoh laporan ad hoc tercantum sebagai berikut:
    - Daftar semua CR yang telah disetujui tetapi tidak diimplementasikan;
    - Daftar semua CR yang dimulai dalam empat bulan terakhir;
    - Daftar berapa banyak orang yang mengerjakan CR tertentu;
    - Catatan berapa banyak waktu yang dibutuhkan untuk mengimplementasikan perubahan tertentu;
    - Jumlah dan detail CR yang tertunda.

**Otomatisasi akuntansi status**

**status Alat SCM otomasi AKUNTANSI**

* + Kita semua tahu bahwa Akuntansi Status terdiri dari pencatatan dan pelaporan informasi yang diperlukan untuk mengelola sistem perangkat lunak dan karakteristiknya secara efektif
    - Begitu banyak laporan dan proses yang harus dihasilkan dan dicatat
  + Untuk mengatasi masalah tersebut, kita perlu menggunakan perangkat lunak SCM Tools untuk melakukan proses SA secara efisien dan efektif
    - Alat SCM dapat membuat Proses SA lebih otomatis, misalnya alat SCM dapat digunakan untuk menghasilkan laporan rutin

**status Alat SCM otomasi AKUNTANSI**

* + Ada beberapa kategori laporan umum yang didukung oleh sebagian besar alat SCM
    - Ubah dan Pelacakan Masalah Laporan
    - Pelaporan Perbedaan
    - Kueri Ad Hoc
    - Jurnal

**Ubah dan Pelacakan Masalah Laporan**

* + Laporan ini berisi perincian seperti siapa yang membuat perubahan dan kapan, siapa yang memprakarsai itu, mengubah riwayat, dan status CR. Di sini, keuntungannya, seperti yang disebutkan sebelumnya, adalah bahwa pengguna dapat menyesuaikan informasi yang diambil dalam format apa pun yang mereka inginkan. Jadi seseorang dapat menghasilkan laporan dari semua CR yang belum ditugaskan; semua CR yang tertunda; semua CR ditugaskan untuk orang tertentu; dan CR diurutkan berdasarkan faktor-faktor seperti tanggal, tingkat keparahan, prioritas, klasifikasi, tanggal penyelesaian, dan status.

**Pelaporan Perbedaan**

* + Penting untuk melacak perbedaan antara versi dan rilis, karena hal itu akan membuat lebih mudah untuk memasukkan perubahan dari satu versi ke versi berikutnya.
  + Sebagian besar alat SCM memiliki fasilitas untuk menghasilkan laporan perbedaan, yang akan berisi perbedaan (perubahan) antara dua versi item atau set item.

**Kueri Ad Hoc**

* + Kegunaan memiliki kemampuan permintaan ad hoc tidak pernah bisa dilebih-lebihkan.
  + Kueri ad hoc memungkinkan pengguna sistem SCM untuk mendapatkan informasi yang mereka inginkan, saat mereka menginginkannya, dan dalam bentuk yang mereka inginkan.
    - Alat pelaporan sangat canggih sehingga banyak dari mereka memiliki antarmuka pengguna grafis yang membantu pengguna menulis pertanyaan mereka sendiri dengan memilih item yang mereka minati.
    - Selain itu, alat pelaporan ini memiliki fitur penelusuran, sehingga pengguna dapat menelusuri dari laporan ringkasan hingga tingkat detail yang diperlukan

**Jurnal**

* + Fitur jurnal membedakan alat SCM dari proses akuntansi status manual
  + Jurnal mencatat semua peristiwa yang terjadi pada semua item konfigurasi saat terjadi, sehingga memberikan pengguna dengan gambaran lengkap dan komprehensif tentang apa yang terjadi selama periode waktu tertentu.
    - Ini adalah versi yang lebih canggih dan lebih komprehensif dari log transaksi manual.

**RINGKASAN**

* + Akuntansi status adalah aktivitas perekaman yang berfungsi sebagai tindak lanjut dari hasil aktivitas SCM untuk identifikasi konfigurasi dan kontrol perubahan.
    - Ini melacak dokumen identifikasi konfigurasi saat ini, konfigurasi saat ini dari perangkat lunak yang dikirimkan, status perubahan yang sedang ditinjau, dan status penerapan perubahan yang disetujui.
  + Fungsi status akuntansi memainkan peran penting dalam manajemen dan pengendalian proyek yang efisien
  + Alat SCM mengotomatiskan fungsi status akuntansi dan membantu memberikan informasi yang akurat, tepat waktu, dan relevan kepada pengguna.

**Verifikasi & audit**

**pengantar**

**verifikasi dan pengenalan audit**

* + Tujuan dari verifikasi konfigurasi dan audit adalah untuk memverifikasi bahwa sistem perangkat lunak cocok dengan deskripsi CI dalam spesifikasi dan dokumen dan bahwa paket yang ditinjau selesai
  + V&A memastikan kinerja dan persyaratan fungsional telah dicapai oleh desain dan bahwa desain telah didokumentasikan secara akurat dalam dokumentasi konfigurasi

**verifikasi dan pengenalan audit**

* + Tujuan dan manfaat V&A meliputi:
    - Memastikan bahwa desain produk memberikan kemampuan kinerja yang disepakati;
    - Memvalidasi integritas dokumentasi konfigurasi;
    - Memverifikasi konsistensi antara produk dan dokumentasi konfigurasinya;
    - Menentukan bahwa ada proses yang memadai untuk memberikan kontrol konfigurasi yang berkelanjutan;
    - Memberikan kepercayaan diri dalam menetapkan garis dasar produk;
    - Memastikan konfigurasi yang diketahui sebagai dasar untuk hal-hal seperti instruksi pengoperasian dan pemeliharaan, pelatihan, dan suku cadang dan perbaikan.

**Verifikasi & Audit**

**proses verifikasi dan audit**

* + Proses verifikasi konfigurasi (CV) dan audit (CA) meliputi yang berikut:
    - Verifikasi konfigurasi dari konfigurasi awal CI, dan penggabungan perubahan yang disetujui, untuk memastikan bahwa CI memenuhi kinerja yang disyaratkan dan persyaratan konfigurasi yang terdokumentasi;
    - CA catatan verifikasi konfigurasi dan produk fisik untuk memvalidasi bahwa program pengembangan telah mencapai persyaratan kinerja dan dokumentasi konfigurasi atau sistem / CI yang diaudit konsisten dengan produk yang memenuhi persyaratan.

**Input verifikasi dan audit**

* + Input untuk verifikasi konfigurasi dan aktivitas audit tercantum sebagai berikut:
    - Konfigurasi, status, dan informasi jadwal dari akuntansi status;
    - Dokumentasi konfigurasi yang disetujui (yang merupakan produk dari proses identifikasi konfigurasi);
    - Hasil pengujian dan verifikasi;
    - CI perangkat keras fisik atau perangkat lunak CSCI dan perwakilannya;
    - Buat instruksi dan alat yang digunakan untuk mengembangkan, memproduksi, menguji, dan memverifikasi produk.
  + V&A yang sukses → sistem dan CI (s) dan satu set dokumentasi yang mungkin dianggap sebagai dasar produk.

**Verifikasi & Audit siapa yang melakukan audit?**

* + Audit biasanya dilakukan oleh perwakilan atau tim perwakilan dari manajemen, departemen QA, atau pelanggan atau klien, dalam beberapa kasus, audit dilakukan oleh lembaga eksternal

**Verifikasi & Audit apa yang dibutuhkan untuk audit?**

* + Sumber daya dan bahan yang diperlukan untuk melakukan audit mencakup item-item berikut sejauh yang sesuai untuk jenis dan ruang lingkup audit:
    - Rencana dan agenda audit;
    - Fasilitas yang memadai dan akses tidak terbebani;
    - Tugas dan ketersediaan personel;
    - Spesifikasi, gambar, manual, jadwal, dan data desain yang berlaku, hasil uji, laporan inspeksi, lembar proses, lembar data, prosedur keselamatan, dan dokumentasi lain yang dianggap perlu;
    - Alat dan peralatan inspeksi yang diperlukan untuk evaluasi dan verifikasi;
    - Akses ke produk dan komponen terperinci untuk ditinjau.

**Verifikasi & Audit Jenis CA**

* + Ada beberapa jenis tentang CA:
    - Jenis utama:
      * CA Fungsional (FCA) dan CA Fisik (PCA)
    - Jenis Alternatif:
      * Ulasan (MRR)
      * Test readiness reviews (TRRs)
      * Pengujian alfa dan beta

**Verifikasi & Audit FCA & PCA**

* + Tujuan dari FCA adalah untuk memverifikasi bahwa kinerja aktual CI setuju dengan persyaratan yang ditentukan dalam definisi persyaratan dan dokumen desain sistem.
  + Tujuan PCA adalah untuk memverifikasi bahwa CI, seperti yang dibangun, sesuai dengan dokumentasi teknis yang mendefinisikannya
    - PCA biasanya dilakukan setelah berhasil menyelesaikan FCA.

**Verifikasi & Audit MRR & TRRS**

* + MRR dilakukan untuk mengonfirmasi bahwa distribusi, layanan, pemeliharaan, dukungan teknis, dan orang-orang lapangan sudah siap; bahwa manual instalasi, operasi, dan troubleshooting sudah siap; dan bahwa pengujian produk dan uji coba berhasil, di antara tujuan lain.
  + TRR dilakukan untuk mengevaluasi hasil uji pendahuluan untuk satu atau lebih CI, untuk memverifikasi bahwa prosedur pengujian untuk masing-masing CI lengkap, untuk mematuhi rencana dan deskripsi pengujian, untuk memenuhi persyaratan pengujian, dan untuk memverifikasi bahwa suatu proyek dipersiapkan untuk melanjutkan ke pengujian formal CI.

**Verifikasi & Audit ALPHA & pengujian beta**

* + Pengujian alfa dilakukan ketika sistem atau produk memiliki banyak fitur baru yang belum teruji sebelumnya
  + Pengujian beta diperlukan ketika tim pengembangan memutuskan bahwa beberapa tingkat evaluasi pelanggan diperlukan sebelum rilis akhir produk.

**Verifikasi & Verifikasi Audit**

* + Verifikasi konfigurasi adalah proses berkelanjutan yang umum untuk CM, rekayasa sistem, rekayasa desain, manufaktur, dan QA.
  + Verifikasi bahwa suatu desain mencapai tujuannya dicapai dengan perbandingan persyaratan yang sistematis dengan hasil pengujian, analisis, atau inspeksi.

**Verifikasi & verifikasi Audit**

* + Metode verifikasi harus direncanakan dengan hati-hati untuk memastikan bahwa semua persyaratan dipenuhi dan metode verifikasi individu yang dipilih sudah tepat.
  + Selain itu, hal-hal berikut harus diverifikasi:
    - Sistem kontrol perpustakaan dokumentasi;
    - Keunikan pengidentifikasi produk;
    - Validitas antarmuka;
    - Audit internal mencatat proses dan prosedur CM.

**AUDIT SISTEM SCM**

**AUDIT SISTEM SCM**

* + Audit sistem SCM dilakukan untuk memastikan bahwa implementasi SCM tetap konsisten dengan kebijakan dan prosedur yang ditetapkan.
  + Aspek umum yang dapat dianggap sebagai bagian dari audit sistem tercantum sebagai berikut:
    - Proses kontrol perubahan operasional, termasuk fungsi CCB;
    - Implementasi permintaan perubahan;
    - Penelusuran perubahan yang disetujui terhadap spesifikasi dan persyaratan asli;
    - Ketersediaan data desain dan dokumentasi untuk mendukung perubahan yang disetujui;
    - Keterlacakan keputusan desain dengan persyaratan awal.

**SISTEM AUDIT SCM PERAN TIM & Alat SCM di ca**

* + Merupakan tanggung jawab tim SCM untuk menjadwalkan audit dan menemukan personil yang memenuhi syarat untuk melaksanakannya.
  + Kemampuan pengumpulan informasi otomatis dari alat SCM membuat audit menjadi proses yang sangat sederhana, karena informasi yang diperlukan dapat dihasilkan untuk keperluan verifikasi dan auditor dapat mengkonfirmasi apakah sistem atau produk yang diaudit lengkap dan memenuhi semua persyaratan.

**Ringkasan**

* + CA dilakukan untuk memastikan bahwa sistem perangkat lunak berfungsi dengan benar dan untuk memastikan bahwa konfigurasi telah diuji untuk menunjukkan bahwa mereka memenuhi persyaratan fungsional mereka dan berisi semua entitas yang dapat dikirim
  + Dua jenis audit utama adalah PCA dan FCA
    - Sedangkan FCA mengotentikasi bahwa perangkat lunak berkinerja sesuai dengan persyaratan dan sebagaimana dinyatakan dalam dokumentasi, PCA mengautentikasi bahwa komponen yang akan dikirim benar-benar ada dan bahwa mereka lengkap dalam segala hal dan mengandung semua item yang diperlukan
  + Sistem SCM juga harus menjalani audit untuk memastikan bahwa implementasi SCM tetap konsisten dengan kebijakan dan prosedur yang ditetapkan
  + Alat SCM mengotomatiskan sebagian besar tugas audit dan menjadikan audit pekerjaan yang mudah dan tidak menyakitkan