

1. Pendahuluan

Apa itu Version Control

Version control dikenal dengan banyak istilah. Ada yang menyebut Configuration Management Tool, Source Code Management (SCM), atau Source Control. Dalam buku ini, istilah yang digunakan adalah Version Control.

Kemampuan menggunakan version control merupakan hal yang langka di kalangan programmer Indonesia. Di bangku kuliah Teknik Informatika, Ilmu Komputer, atau jurusan lain yang sejenis, kita tidak diajarkan tentang version control. Bahkan kita tidak diberitahu kalau ada perangkat yang disebut version control.

Di dunia kerja di Indonesia, tidak banyak perusahaan yang sudah menggunakan version control. Walaupun digunakan, biasanya programmer diharapkan untuk belajar secara otodidak tanpa pelatihan.

Kalau begitu, kenapa kita harus repot-repot belajar menggunakan version control?

Terlepas dari kondisi lokal di Indonesia, perangkat ini adalah standar minimal untuk tim programmer di dunia. Di belahan dunia yang lain, ribuan programmer terlibat dalam proyek open source, masing-masing berada di negara yang berbeda, tidak saling mengenal, tidak pernah bertemu muka, berkomunikasi hanya melalui internet. Tetapi mereka mampu menghasilkan aplikasi perangkat lunak yang berkualitas tinggi dalam waktu yang wajar.

Kolaborasi dengan skala besar seperti ini dimungkinkan dengan penggunaan version control. Dengan jumlah orang yang begitu besar, sulit sekali mengelola perubahan source code yang

penting

pada tahun 2007:

version control tidak diajarkan di sekolah ataupun di sebagian besar tempat kerja

pelatihan tentang cara penggunaan version control juga belum ada di Indonesia

dihasilkan tiap programmer. Perubahan pada satu file sangat mungkin akan terhapus atau diganti dengan file lain oleh programmer lain. Apabila terjadi kesalahan, sangat sulit untuk mengembalikan kondisi aplikasi seperti sebelum terjadi error.

Version control dapat menyediakan akses untuk banyak orang sekaligus, mirip dengan sharing folder yang biasa kita gunakan. Bedanya dengan sharing folder, version control mampu;

- mencatat isi perubahan dan pelaku perubahan, dengan demikian kalau ditemukan 'keanehan' pada kode program, kita dapat melacak siapa yang membuat 'keanehan' tersebut dan kapan 'keanehan' tersebut muncul
- menyediakan fungsi undo untuk tim, yaitu dapat mengembalikan kondisi kode program ke periode yang diinginkan (minggu kemarin, sebelum bugfix, dan sebagainya)
- melihat riwayat perubahan kode, dari pertama dibuat, sampai kondisi terakhir saat ini.
- memungkinkan pengembangan kode secara paralel, misalnya aplikasi yang sama dibuat dalam berbagai versi secara bersamaan.

penting

untuk aplikasi yang dikerjakan dua orang atau lebih, version control merupakan keharusan

Saya berpendapat, jumlah maksimal programmer yang dapat bekerja sama tanpa menggunakan version control adalah satu orang. Artinya untuk aplikasi yang dikerjakan dua orang atau lebih, version control wajib digunakan.

Ada banyak aplikasi version control yang tersedia. Beberapa aplikasi yang berbayar adalah:

- Visual Source Safe. Version control buatan Microsoft. Terintegrasi dengan Visual Studio.
- Bitkeeper. Version control ini pernah digunakan untuk

mengelola kode program Linux. Skala pengembangan Linux yang raksasa menunjukkan kualitas dan keandalan aplikasi ini. Karena masalah lisensi, saat ini pengembangan Linux tidak lagi menggunakan Bitkeeper

- Perforce
- Vault. Eric Sink -pembuat Vault- menulis artikel yang sangat bagus tentang penggunaan version control.

Kalau bisa gratis mengapa harus bayar? Berikut adalah version control open source yang tersedia:

- Concurrent Versioning System (CVS). Version control paling terkenal di dunia. Usianya yang tua dan popularitas menyebabkan banyak aplikasi pendukung yang tersedia. Integrasi dengan CVS biasanya sudah dimasukkan sebagai fitur utama dalam editor-editor kode program seperti Eclipse, Netbeans, dan sebagainya.
- Subversion. Aplikasi version control yang dibuat dengan tujuan untuk menggantikan CVS.

Penggunaan Version Control

Untuk mengilustrasikan bagaimana version control digunakan, simak cerita singkat berikut.

Anton dan Endy mengembangkan sebuah aplikasi untuk menghitung pemakaian internet (billing) di hotspot. Aplikasi ini dikembangkan atas permintaan toko kopi terkenal Starblog Coffee. Anton dan Endy masing-masing mempunyai pekerjaan full time, sedangkan aplikasi billing ini dikembangkan di waktu luang mereka, sehingga mereka jarang sekali memiliki waktu luang yang berbarengan.

Anton menginstal repository Subversion pada komputer di rumahnya, yang kebetulan terhubung ke internet 24 jam sehari. Semua kode program aplikasi disimpan di komputer tersebut dan dapat diakses oleh Anton dan Endy.

Pengembangan aplikasi membutuhkan waktu dua minggu. Selama selang waktu tersebut, kedua programmer silih berganti menambahkan fungsi dan fitur ke dalam aplikasi tersebut.

Setelah dua minggu, Endy menginstal aplikasi di Starblog Coffee. Semua kode program yang terbaru diambil dari repository dan dimasukkan ke dalam USB disk.

Segera setelah tiba di Starblog Coffee, kode program tersebut dikompilasi. Ternyata terjadi error pada saat kompilasi. Tapi Endy tidak panik. Ini adalah masalah yang sangat umum terjadi. Masalah ini telah diperkirakan sebelumnya dan telah diantisipasi dengan penggunaan version control.

Dengan menggunakan internet, dia segera mengakses repository server dan melihat catatan perubahan. Terlihat bahwa malam sebelumnya Anton menambahkan satu fitur bonus pada aplikasi billing. Pasti itu penyebabnya, karena hari sebelumnya Endy telah mencoba melakukan kompilasi dan pengetesan terhadap aplikasi dan aplikasi berjalan dengan baik.

Untungnya Anton tidak lupa memberikan penanda (tag) pada repository sebelum dia menambahkan fitur bonus tersebut. Dengan cepat Endy dapat mengambil kode program sesuai kondisi sebelum penambahan fitur bonus.

Kali ini aplikasi berhasil diinstal dengan lancar. Berkat digunakannya version control dengan disiplin, masalah yang terjadi dapat diselesaikan dengan mudah dan cepat.

Cerita di atas menggambarkan secara singkat bagaimana

penting

tag adalah batas penanda yang dibuat di repository.

dengan menggunakan tag, kita dapat mengambil semua file sesuai kondisi pada waktu tag dibuat.

penggunaan version control dapat membantu mengatasi masalah dengan cepat. Masih banyak lagi fitur version control yang akan kita bahas pada buku ini.

Apa itu Subversion

Sebagai version control open source paling terkenal di dunia, CVS telah diakui keandalan dan popularitasnya. Banyak aplikasi pendukung yang tersedia untuk CVS. Akan tetapi, CVS juga memiliki banyak keterbatasan yang fatal, misalnya;

- tidak mendukung atomic commit
- tidak mendukung penyimpanan file binary
- tidak mendukung rename file atau folder
- ijin akses tidak dapat diatur per folder

Subversion diciptakan untuk mengatasi semua keterbatasan tersebut. Dengan desain baru yang lebih modular, Subversion tidak hanya mengatasi keterbatasan CVS, tapi juga memiliki fitur tambahan seperti:

- protokol http
- otentikasi menggunakan LDAP

Profil Pembaca

Buku ini ditulis untuk:

- **Project Manager**

Sebagai orang yang mengkoordinasikan aktivitas pelanggan, programmer, dan system administrator, project manager harus memahami garis besar penggunaan version control. Sehingga project manager dapat mengetahui

prosedur apa yang mungkin atau tidak mungkin dilakukan dalam timnya. Dengan bermodalkan buku ini, project manager juga bisa membuat keputusan yang berwawasan tentang aplikasi version control mana yang harus digunakan.

Project manager juga dapat mengelola dokumen proyek dalam Subversion. Hal ini sulit dilakukan menggunakan CVS, karena CVS tidak mendukung file binary. Tetapi Subversion dapat mengelola file binary (misalnya Microsoft Word atau Microsoft Project) dengan efisien.

- **Programmer**

Agar dapat memenuhi standar *de facto* internasional, programmer wajib menguasai penggunaan version control. Apabila ingin terlibat dalam project open source, penguasaan version control merupakan hal wajib.

- **System Administrator**

Pada organisasi yang mengimplementasikan version control, biasanya tugas instalasi dan konfigurasi adalah tanggung jawab system administrator. Dalam buku ini, dibahas tentang cara instalasi server dan client, berbagai kemungkinan pilihan konfigurasi protokol, dan akses kontrol yang tersedia.

Susunan Buku

Untuk memudahkan segmen pembaca yang berbeda, buku ini disusun menjadi lima bagian utama, yaitu:

- 1. Instalasi dan Konfigurasi Server**

Bagian ini biasanya intensif digunakan oleh System

Administrator. Pada bagian ini akan dijelaskan tentang:

- instalasi Subversion
- membuat repository
- mengaktifkan protokol http dan https
- mengatur akses kontrol
- menggunakan LDAP server untuk otentikasi

2. Instalasi dan Konfigurasi Client

Bagian ini memuat tentang aplikasi yang digunakan di sisi client. Biasanya yang berminat membaca ini adalah para pengguna akhir (*end-user*), misalnya programmer atau project manager. Topik yang dijelaskan pada bagian ini adalah:

- instalasi TortoiseSVN. Aplikasi ini terintegrasi dengan Windows Explorer, sangat mudah digunakan.
- integrasi dengan Eclipse.
- integrasi dengan Macromedia Dreamweaver.

3. Penggunaan Subversion

Bagian ini membahas tentang operasi yang umum dilakukan pengguna Subversion. Pengguna akhir (*end-user*) seperti programmer dan project manager dianjurkan menguasai bagian ini. Topik yang dijelaskan antara lain:

- mendownload (checkout) dari repository
- mengupload (commit) perubahan ke repository
- menambah file/folder baru
- menghapus file/folder

- mengganti nama (rename) file/folder
- tag, branch, dan merge

4. Pengembangan aplikasi dengan Subversion

Bagian ini menjelaskan cara mengembangkan software dengan menggunakan Subversion. Bagian ini akan sangat bermanfaat bagi Project Manager yang mengelola tim programmer. Materi yang dibahas adalah:

- Tiga langkah pengembangan: Development, Testing, Production. Ini adalah siklus klasik pengembangan aplikasi.
- Mengelola proyek dengan bantuan version control.
- Mengelola produk dan berbagai versinya.

5. Aplikasi Tambahan

Untuk meningkatkan kemampuan dan fungsi Subversion, ada beberapa aplikasi tambahan yang menarik untuk diketahui.

Buku ini dapat dibaca dari depan ke belakang untuk mendapatkan pemahaman menyeluruh, atau langsung ke bab yang relevan dengan kebutuhan.

Format Penulisan

Di dalam buku ini kita akan banyak menggunakan kode program, perintah command-line, dan juga mencetak *output* dari komputer.

Perintah Command-Line

Perintah yang diberikan ke komputer ditulis dengan huruf fixed-width dengan format tebal dan latar belakang abu-abu seperti ini:

```
svnadmin create --fs-type fsfs /home/andy/svnroot
```

Perintah yang harus dilakukan oleh root (dalam Linux) ditandai dengan prompt berbentuk #, seperti ini:

```
# apt-get install subversion
```

Perintah yang dapat dilakukan oleh user biasa, ditandai dengan prompt berbentuk \$, seperti ini:

```
$ svn help
```

Perintah yang dilakukan oleh user tertentu ditandai dengan prompt seperti ini:

```
andy> svnadmin create --fs-type fsfs /home/andy/svnroot
```

Windows tidak mengenal perbedaan antara root dan user biasa, sehingga bentuk prompt dapat diabaikan. Perlu diperhatikan, pemisah folder di Windows adalah *backslash* (\), berbeda dengan Linux yang menggunakan *slash* (/). Perintah di atas dilakukan di Windows seperti ini:

```
C:\> svnadmin create --fs-type fsfs c:\andy\svnroot
```

Untuk selanjutnya, semua perintah command line akan ditulis dengan aturan Linux, pengguna Windows harap menyesuaikan.

Output komputer

Output yang dikeluarkan oleh komputer juga ditulis dengan huruf fixed-width, tapi tidak menggunakan huruf tebal dan latar belakang, seperti ini:

```
bash: svnadmin: command not found
```

Kode Program

Kode program ditulis dengan huruf fixed-width tebal tanpa latar belakang, seperti ini:

```
package com.artivisi.tutorial;  
public class ContohKode {  
    public static void main(String[] xxx) {  
        System.out.println("Contoh kode program");  
    }  
}
```

penting

hal penting ditulis
seperti ini

Hal Penting

Poin yang penting akan ditulis pada kotak abu-abu di samping paragraf, seperti contoh berikut.