

# Bekerja dengan CVS

Dalam proyek *software* skala besar, kerja sama tim yang baik sangatlah diperlukan. Untuk membantu mengelola *source code* yang dikerjakan oleh tim, kita dapat menggunakan CVS.

Satu programmer tentu saja tidak bisa menyelesaikan proyek besar sendirian sesuai target waktu, sepandai apapun orangnya, karena proyek *software* besar tidak hanya melibatkan logika, melainkan juga melibatkan analisis, pengujian, pembuatan dokumentasi, dan lainnya. Oleh karena itu, sebuah tim yang solid harus dibangun. Dalam sebuah tim, nantinya delegasi tugas dapat dilakukan sesuai kompetensi.

Apabila terdapat lebih dari satu orang di tim yang bekerja dengan *source code* program, maka ada baiknya sebuah sistem kontrol dan revisi *source code* harus diterapkan. Penerapan sistem kontrol tersebut umumnya dilakukan dengan CVS, *Concurrent Versioning System*.

Dengan menggunakan CVS, tim *developer* dapat mengambil manfaat yang besar. Sebagai contoh, tim dapat melacak setiap perubahan yang dilakukan. Perubahan juga dapat dilakukan oleh setiap anggota dan dapat digabungkan dengan anggota tim lain. Selain itu, dengan mudah *branch* baru dapat diterapkan.

Dalam *software* besar, terkadang kita perlu menerapkan sistem rilis stabil dan

sistem uji (*testing*). Ada kalanya, ketika suatu *software* dirilis, tim telah menyiapkan versi berikutnya untuk diuji. Agar tidak mengganggu *source code* stabil, *branch* baru dapat dibuat dan dikembangkan.

CVS sendiri dapat merekam *history* setiap file dan menyimpan perbedaan setiap versi. CVS juga menyimpan catatan tentang siapa yang melakukan perubahan, kapan dan alasan perubahan serta hal-hal lainnya.

Walaupun CVS dibangun di atas RCS, CVS memiliki beberapa kelebihan dibandingkan RCS. Kelebihan-kelebihan tersebut di antaranya adalah CVS termasuk sistem desentralisasi sehingga setiap anggota tim dapat memeriksa *server* dan mendapatkan versinya masing-masing secara lokal. Selain itu, CVS juga memungkinkan kita untuk melakukan editing file secara *concurrent*. Dan, ketika suatu *software* ingin dirilis, CVS dapat menandai satu *tree* proyek secara keseluruhan sebagai *software* rilis.

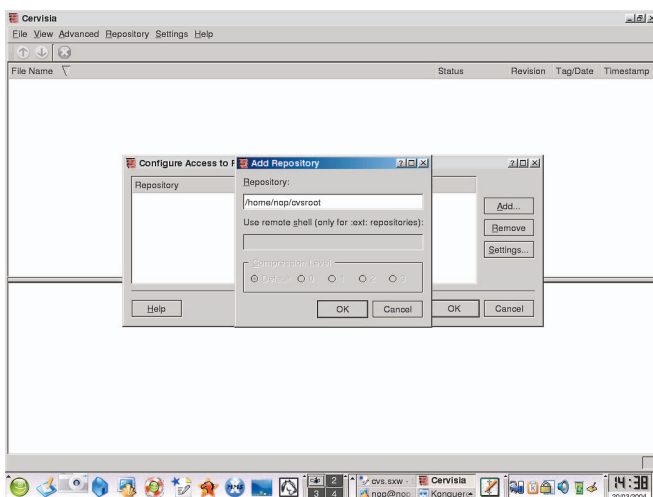
Melihat ke masa lalu, CVS adalah sistem yang rumit dan kompleks. *Software* CVS adalah *software* tingkat tinggi dan telah dikembangkan dalam kurun waktu



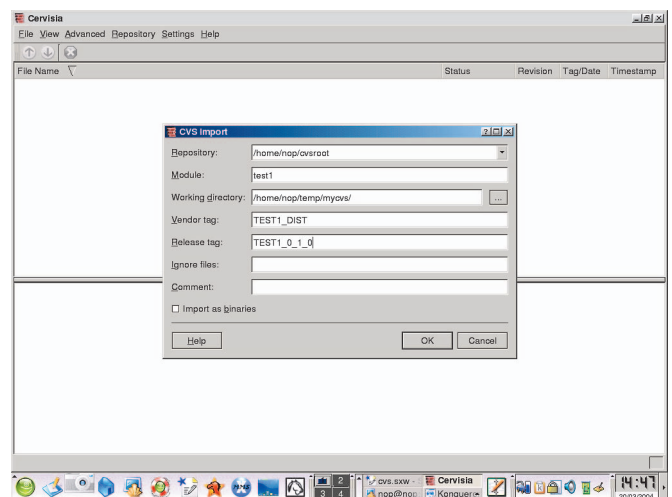
yang sangat lama. Butuh waktu 20 sampai 30 tahun untuk menentukan algoritma terbaik dan diterapkan ke dalam aplikasi. Walaupun telah dikembangkan puluhan tahun, sampai saat ini CVS masih mengalami evolusi.

Tercatat pada bulan Desember 1986, **Dick Grune** mengimplementasikan algoritma CVS ke dalam kumpulan *shell script* dan melepaskannya ke *newsgroup comp.sources.unix*. Walaupun sampai saat ini *shell script* tersebut mungkin tidak terdapat lagi dalam *source code* CVS, banyak algoritma, khususnya untuk resolusi konflik masih berdasarkan algoritma yang digunakan waktu itu. Pada bulan April 1989, **Brian Berliner** melakukan desain ulang dan menerapkannya sebagai aplikasi. Komunitas segera membantu Brian untuk menerapkan yang terbaik.

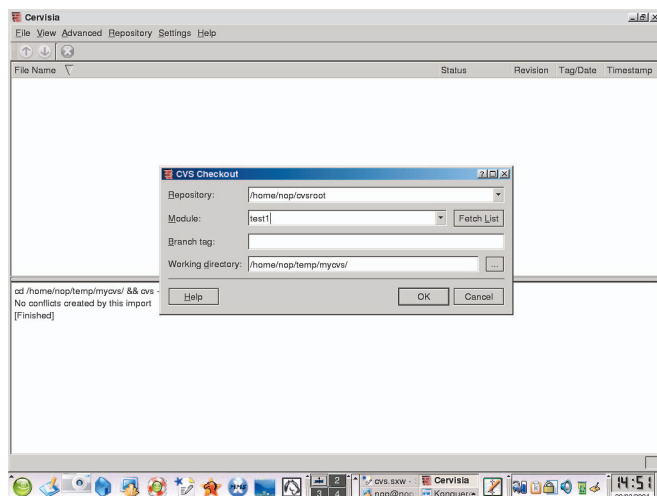
Seperti yang dikatakan sebelumnya bahwa CVS terus mengalami perkembangan, tercatat telah dikembangkan proyek dengan nama Subversion, yang merupakan penyempurnaan lebih lanjut dari



▲ Menambahkan repository.



▲ Menambahkan proyek baru.



▲ Melakukan CVS checkout.

CVS. Walau demikian, CVS sendiri masih sangat umum dan digunakan secara meluas di dunia ini. Dalam bentuk *free software* ataupun *proprietary*. Pada kenyataannya, CVS dapat memenuhi kebutuhan software besar saat ini.

Dalam artikel ini, kita akan membahas bagaimana membangun sebuah repository lokal, di mana anggota tim dapat menyimpan proyek software-nya semudah melakukan revisi dan membuat rilis baru. Sebisa mungkin, kita akan menghindari *command line interface*. Oleh karena itu, untuk *front end*, kita akan menggunakan Cervisia, front end CVS yang sangat layak digunakan.

Sebelum kita memulai, periksalah terlebih dahulu sistem Anda. Untuk membangun CVS repository, kita membutuhkan aplikasi CVS, yang dapat di-download di <http://www.cvshome.org>. Namun umumnya, setiap distro telah memaketkan CVS dan menginstalnya secara otomatis. Kemudian, untuk front end CVS, kita akan membutuhkan Cervisia, yang dapat di-download di <http://cervisia.sf.net>. Cervisia juga umumnya akan terinstal otomatis ketika Anda menginstal paket kdesdk.

## Membangun repository lokal

Untuk membangun *repository* lokal, siapkan sebuah direktori yang dapat diakses baca tulis oleh setiap anggota tim Anda. Sebagai contoh, penulis akan membuat sebuah repository di `/home/nop/cvsroot`. Sesuaikan

dengan konfigurasi Anda. Berikan perintah berikut ini untuk membangun repository tersebut:

```
cvs -d /home/nop/cvsroot/ init
```

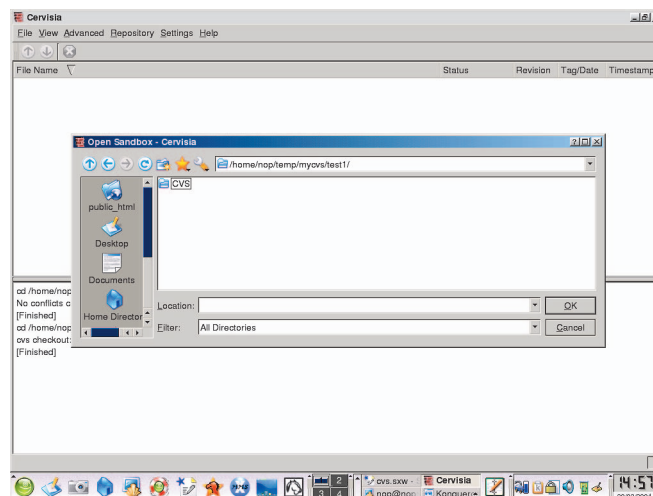
Sesederhana itu, Anda telah membangun sebuah repository CVS. Selanjutnya, setiap anggota tim Anda dapat mengakses repository tersebut. Berbagai metode dapat digunakan untuk otentikasi. Mulai dari penggunaan *password* sampai penggunaan *ssh*.

Gunakan Konqueror, Nautitus, atau file manager kesayangan Anda untuk melakukan *browsing* ke *cvsroot* tersebut. Anda akan melihat sebuah direktori dengan nama CVSROOT, yang mengandung berbagai file.

Sebagai catatan, terkadang proyek-proyek besar juga menerapkan *anonymous cvs*. Dengan demikian, siapa saja dapat masuk ke dalam repository dan mengambil yang tersedia.

Untuk bekerja dengan local repository tersebut, kita perlu menambahkannya terlebih dahulu di Cervisia. Bukalah Cervisia, dan akseslah menu *Repository | Repositories*. Sebuah dialog akan ditampilkan. Kliklah tombol *Add*, dan sebuah dialog lain akan ditampilkan.

Isilah field repository dengan alamat repository lokal kita, `/home/nop/cvsroot`. Kembali ke dialog sebelumnya, dan Anda akan melihat bahwa sebuah repository telah terdaftar. Anda bisa menambahkan repository lain apabila diinginkan. Setelah selesai, tutuplah dialog tersebut.



▲ Membuka Sandbox baru.

## Working directory dan proyek sederhana

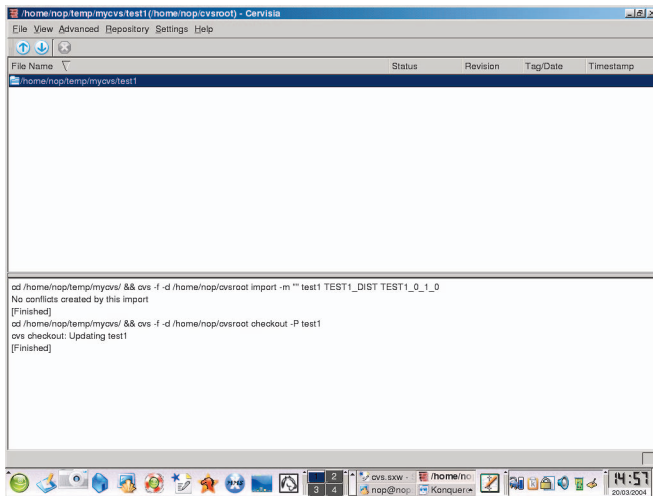
Kita mengasumsikan CVS repository kita belum terisi proyek apapun. Dalam hal ini, kita akan memasukkan satu proyek baru ke dalamnya. Sebelum melakukan hal tersebut, buatlah terlebih dahulu sebuah *working directory*. *Working directory* adalah istilah di mana Anda akan menyimpan setiap local copy untuk proyek yang Anda ambil dari CVS repository. Di artikel ini, penulis mendefinisikan *working directory* di `/home/nop/temp/mycvs`. Sesuaikan dengan konfigurasi Anda.

Selanjutnya, bukalah Cervisia, dan kita akan mendaftarkan proyek baru kita di CVS repository. Untuk menambahkan suatu proyek, kita menyebutnya sebagai *import*. Bukalah menu *Repository | Import*. Sebuah dialog akan ditampilkan.

Pada field repository, pilihlah repository lokal kita sebelumnya. Untuk module, isilah dengan nama proyek kita, misalnya `test1`. Untuk *working directory*, masukkan *working directory* kita yang telah dibuat sebelumnya, yaitu `/home/nop/temp/mycvs`. Sekali lagi, sesuaikan dengan konfigurasi Anda.

Pada vendor tag, masukkanlah nilai berikut ini: `TEST1_DIST`. Pada release tag, masukkanlah informasi rilis seperti `TEST1_0_1_0`. Anda dapat mengisikan komentar di *field comment*. Kliklah tombol *OK*, dan Anda baru saja mendaftarkan sebuah proyek dengan nama `test1`.

Lihatlah ke bagian bawah window Cervisia, dan pastikan tidak ada pesan kesalahan yang tertulis di sana.



▲ Daftar file Sandbox.

Selanjutnya, setelah kita mendaftarkan proyek baru di repository lokal kita sebelumnya, kita mungkin ingin mendapatkan perubahan terbarunya. Memeriksa CVS disebut juga dengan istilah *checkout*. Untuk melakukan checkout, pilihlah menu *Repository | checkout*. Sebuah dialog akan ditampilkan, masukkan nama module yang ingin diperiksa, yaitu test1. Pastikan working directory Anda diisi dengan benar. Kliklah tombol OK, dan lihatlah log di bagian bawah Cervisia. Pastikan tidak terjadi kesalahan. Apabila sukses, maka kita telah mendapatkan satu copy lokal dari modul test1 tersebut. Periksa working directory Anda, dan Anda akan menemukan sebuah direktori dengan nama test1, yang mengandung direktori CVS di dalamnya.

Untuk bekerja dengan file-file kita, kita perlu membuka sebuah sandbox terlebih dahulu dengan mengakses menu *File | Open Sandbox*. Pilihlah direktori test1 yang terletak di dalam working directory. Apabila sukses, maka Anda akan melihat direktori Anda di area kerja utama Cervisia.

Saat ini, proyek kita belum memiliki satu source code pun. Kita akan membuatnya terlebih dahulu agar bisa bekerja dengan CVS nantinya. Buatlah test1/test1.c di working directory kita, dan isilah dengan *source code* berikut:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    return 0;
```

```
}
```

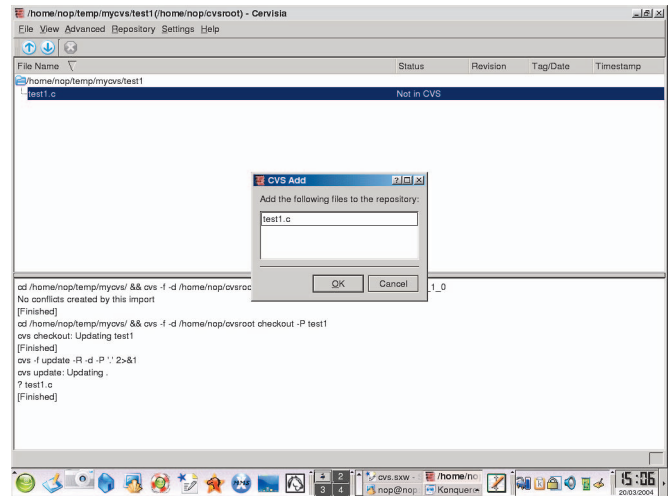
## Menambahkan dan menghapus file

Kita telah membuka sebuah sandbox, dan kita telah memiliki sebuah source code dengan nama file test1.c. Kita akan meng-upload file tersebut ke repository lokal kita. Namun, sebelum itu, karena kita baru saja mengubah direktori test1 kita, perbarui dahulu sandbox kita dengan memilih *File | Update*. Perhatikan perubahan di area kerja kita. Kini, kita telah memiliki test1.c. Perhatikan status dari file tersebut, yaitu Not in CVS. Hal ini wajar karena file tersebut masih kita miliki secara lokal, dan belum disimpan ke repository.

Pilihlah test1.c tersebut dan akseslah menu *File | Add to Repository*. Sebuah dialog akan tampil, dan pastikan Anda telah memilih test1.c. Kembalilah perhatikan status dari file test1.c tersebut, yaitu *Locally Added*. Perhatikan juga *Revision*-nya, yaitu 0.

Sampai di sini, kita masih belum benar-benar menyimpan dan mendaftarkan file kita di CVS. Untuk benar-benar melakukannya, kita akan melakukan CVS commit. Pilihlah *File | Commit*. Sebuah dialog akan tampil. Isilah *Log message* sesuai keinginan. Dan kliklah tombol OK.

Perhatikan status file test1.c tersebut. Kini statusnya telah berubah menjadi *Up to Date*. Perhatikan juga *revision*-nya, yang kini berubah menjadi 1.1. Kita telah menambahkan file ke dalam repository CVS kita.



▲ Menambahkan file ke repository.

Untuk menghapus file, pilihlah *File | Remove from Repository*. Sebuah dialog akan tampil untuk konfirmasi. Harap diperhatikan, apabila Anda menghapus file, maka file lokal Anda juga akan dihapus.

Bagaimana? Sampai di sini, kita telah mempelajari bagaimana menambahkan dan menghapus file. Berikut kita akan melihat bagaimana untuk mengirimkan file test1.c yang kita modifikasi ke repository.

## Melakukan perubahan

Modifikasilah file test1.c. Kini, file test1.c berikan source code berikut:

```
#include <stdio.h>
```

```
int main()
```

```
{
```

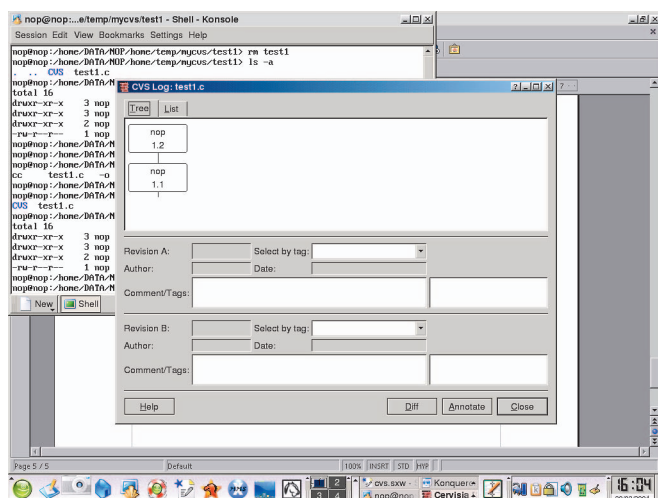
```
    printf("hello world!!!");
```

```
    return 0;
```

```
}
```

Karena kita telah mengubah file test1.c, maka kita perlu meng-update sandbox kita. Sekali lagi, akseslah perintah *File | Update*. Perhatikan kembali status file test1.c, yang berubah menjadi *Locally Modified*.

Selanjutnya, perubahan tersebut akan kita kirimkan ke repository. Cukup lakukan saja cvs commit dengan mengakses *File | Commit*. Isikan komentar Anda, dan perhatikan kembali status file test1.c, yang kini berubah menjadi *Up to Date*. Perhatikan juga *revision*-nya yang berubah menjadi 1.2. Semua hal tersebut menunjukkan Anda telah melakukan perubahan.



▲ Dialog Browse Log.

Berikutnya, kita akan melihat bagaimana Cervisia dapat menampilkan perubahan yang terjadi diantara dua revisi dengan tampilan yang memukau.

## Mengamati perubahan

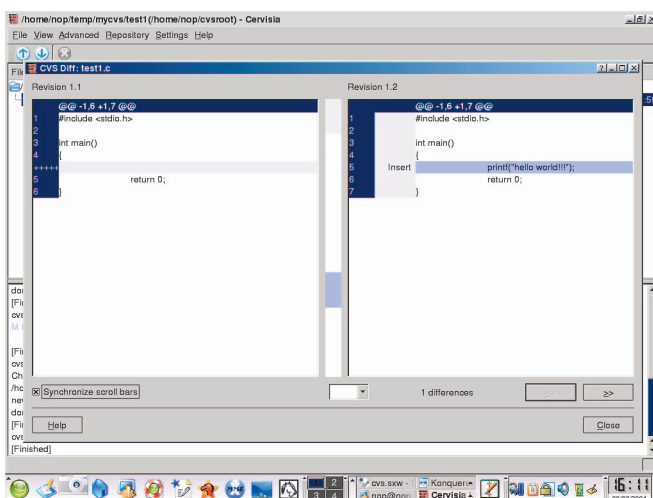
Untuk mengamati perubahan antara revisi, akseslah menu *View/Browse Log*. Sebuah dialog akan tampil. Di dialog ini, Cervisia menyajikan dua modus untuk mengamati revisi-revisi yang ada.

Pertama adalah modus *Tree*. Pada modus tree ini, setiap perubahan akan berusaha ditampilkan dalam hirarki pohon secara mendetail. Sementara, pada modus list, setiap revisi akan ditampilkan secara mendetail.

Masuk ke hal yang paling menyenangkan. Semisal, kita akan melihat apa saja yang berubah dari revisi 1.1 dan 1.2. Pilihlah 1.1 untuk Revision A dan 1.2 untuk Revision B. Untuk memilih revision A, gunakan tombol kiri mouse. Sementara, untuk memilih revision B, gunakan tombol tengah mouse Anda. Kemudian, kliklah tombol *Diff*.

Sebuah dialog akan ditampilkan untuk kita. Di dialog ini, Anda dapat melihat perubahan di antara dua revisi. Perubahan disajikan dengan sangat mendetail. Penggunaan warna-warni indah dan atribut teks lainnya benar-benar luar biasa. Di dialog ini, kita juga akan melihat informasi berapa saja perbedaan yang ada.

Aktifkan *Synchronize scroll bars* untuk mengunci scroll bar. Anda dapat melakukan navigasi dengan mudah dan cepat



▲ Mengamati perubahan yang terjadi.

untuk melihat perubahan-perubahan yang ada.

Pada dialog *Browse Log* sebelumnya, apabila Anda hanya ingin melihat isi dari suatu revisi, Anda dapat menekan tombol *Annotate*.

Demikianlah perkenalan kita dengan CVS. Tentu saja, perkenalan ini tidak mampu mencakup setiap aspek dari CVS

yang sangat kompleks tersebut. Namun, paling tidak, Anda bisa segera memulai proyek yang lebih besar, yang melibatkan berbagai pihak, untuk hasil yang lebih baik.

Segeralah buat repository baru, dan marilah sama-sama mengembangkan sesuatu yang besar dan berguna bagi semua pihak.🐱

**Noprianto** ([noprianto@infolinux.co.id](mailto:noprianto@infolinux.co.id))

# IKLAN