

Setting CVS Server dan Cara Penggunaannya

CVS adalah sistem untuk mengontrol versi pengembangan *software*. CVS merupakan peranti lunak yang dapat menjaga sinkronisasi kumpulan file, dan membantu bagi *developer* /programer yang berada di tempat berbeda untuk mengerjakan suatu proyek secara bersama-sama.

Bagi anda yang sering mengerjakan *project* aplikasi skala menengah ke atas, maka akan merasakan hal yang merepotkan jika tidak menggunakan peranti penyinkronisasi file seperti CVS ini.

Bayangkan jika Anda bekerja secara tim (misal: lima orang atau lebih) tanpa menggunakan CVS. Pada project yang Anda kerjakan ada beberapa file yang perlu dimodifikasi atau ditambahkan secara bersamaan oleh beberapa orang. Untuk memberitahu setiap perubahan yang dilakukan oleh tim Anda, orang yang memodifikasi file harus memberitahukan pada anggota tim yang lain perubahan apa yang telah dia lakukan, dan apa saja nama file yang telah diubah. Jika perubahan yang dilakukan sering dan anggota tim Anda banyak, tentu sangatlah tidak *smart* untuk memberitahukan setiap perubahan yang telah terjadi kepada anggota tim yang lain. Maka dari itulah dibutuhkan suatu sistem pengontrol, yang salah

satunya adalah CVS. Sebelum mengerjakan project, Anda harus membuat sebuah CVS *repository* terlebih dahulu.

Untuk menyiapkan server kita agar siap menjadi CVS Repository (server), maka perlu tersedia paket cvs dalam sistem kita, Instalasinya cukup mudah baik yang menggunakan paket manager rpm, debian, maupun instalasi dari source.

Instalasi software CVS

Untuk melakukan instalasi cvs dari source, Anda bisa mengikuti petunjuk berikut ini:

1. Download terlebih dahulu, program CVS, pada situs <https://www.cvshome.org>, atau anda mengcopynya dari CD majalah *Info-LINUX* edisi ini.
2. Masuk konsol, lalu *login* sebagai root

```
$ su -  
login : root  
Password : <ketikkan password  
root anda>
```

3. Dengan mengamsumsikan Anda menyimpan file `cvs-1.12.12.tar.gz` pada directory `/home/user`, lakukan ekstraksi file tersebut :

```
# cd /usr/src
# tar -xzf /home/supri/cvs-1.12.12.tar.gz
# ./configure
# make
# make check
# make install
# make clean
```

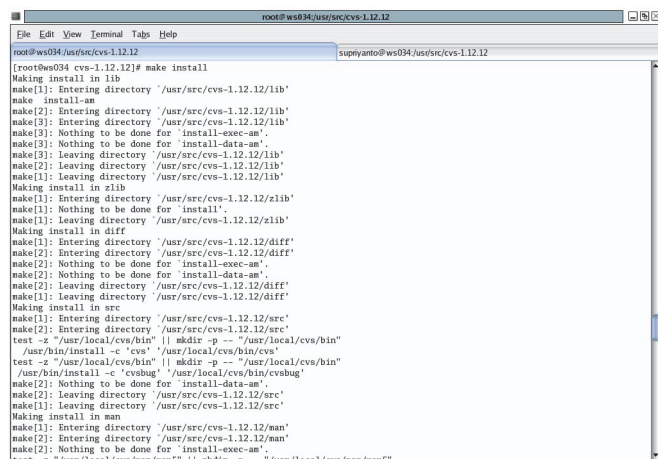
Setting CVS Server

Setelah instalasi software CVS selesai, berikutnya kami akan menjelaskan cara *setting* CVS server berikut cara penggunaannya.

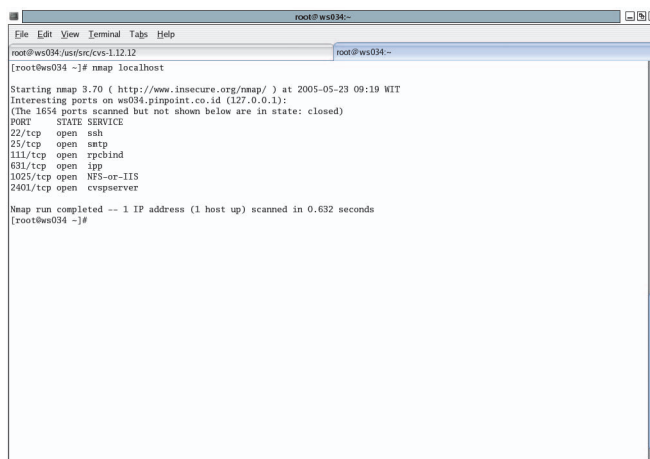
- ### 1. Tambah User Dan Group CVS

Buat user dan group CVS, dengan mengetikkan perintah di bawah ini:

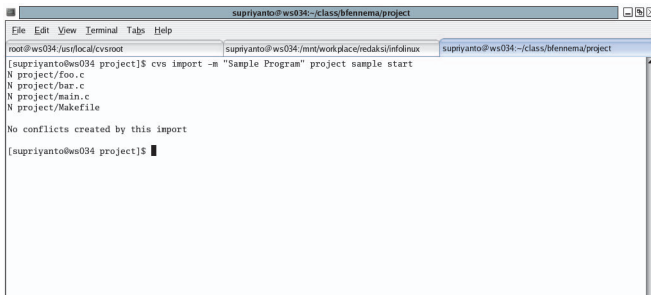
```
# useradd -M -s /sbin/nologin  
CVS
```



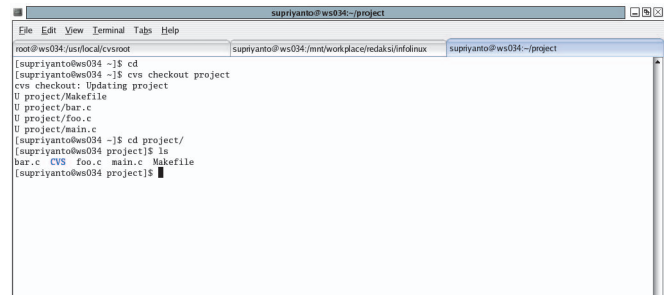
Gambar 1. Instalasi CVS Server.



Gambar 2. Menge-*check* port CVS Server



Gambar 3. CVS import.



Gambar 4. CVS checkout.

```
# useradd -M -s /sbin/nologin
anonymous
```

Lalu edit file /etc/group untuk menambahkan user lain ke CVS group

```
# vi /etc/group

cvs:x:502:supri,anonymous
```

2. Check file /etc/services

Check apakah baris berikut, sudah terdapat di file /etc/services

```
cvspserver 2401/tcp # CVS
client/server operations
cvspserver 2401/udp # CVS
client/server operations
```

Kalau sudah ada biarkan, kalau belum ada tambahkan baris di atas ke dalam file tersebut.

3. Buat file cvs di /etc/xinetd.d

Agar service cvs server dapat berjalan buat file dengan nama cvs dan letakkan di directory /etc/xinetd.d

```
# vi /etc/xinetd.d/cvs
service cvspserver
{
    disable = no
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    passenv = PATH
    server = /usr/bin/cvs
    server_args = -f --allow-root=/usr/local/cvsroot
    pserver
}
```

4. Restart service xinetd

Service xinetd harus di-restart, agar settingan di atas dibaca oleh system.

```
# /etc/init.d/xinetd restart
```

Check apakah port untuk service CVS server sudah aktif atau belum, dengan mengetikkan perintah berikut:

```
# nmap localhost
```

5. Buat Repository Server

Repository adalah folder file-file kerja yang akan dikerjakan secara bersama-sama. Sebagai contoh, penulis menetapkan directory /usr/local/cvsroot sebagai tempat repository.

Untuk membuat repository, ketikkan perintah di bawah ini:

```
# cvs -d /usr/local/cvsroot
init
```

Buat direktori dan set hak aksesnya :

```
# export CVSROOT=/usr/local/
cvsroot
# chown -R cvs.cvs $CVSROOT
# chmod ug+rw $CVSROOT
# chmod o-rwx $CVSROOT
# chmod g+s $CVSROOT
```

Buat symbolic link. Link menyesuaikan "allow-root" di /etc/xinetd.d/cvspserver

```
# ln -s $CVSROOT /cvs
```

6. Buat password untuk user-user CVS

Pindah ke directory CVSROOT

```
# cd /usr/local/cvsroot/
CVSROOT
```

Buat file yang bernama random.pl untuk menggenerate password user CVS.

```
# vi random.pl

#!/usr/bin/perl
srand (time());
my $randletter = "(int (rand
(26)) + (int (rand (1) + .5)
% 2 ? 65 : 97))";
my $salt = sprintf ("%c%c",
```

```
eval $randletter, eval
$randletter);
my $plaintext = shift;
my $crypttext = crypt
($plaintext, $salt);
print "${crypttext}\n";
```

Berikutnya ubah hak akses file random.pl agar bisa di executable, dan jalankan file itu untuk mendapatkan random password. (Misal : saya ingin memberikan password supria untuk user supri)

```
# chmod 777 random.pl
# chown cvs.cvs random.pl
# ./random.pl supria
Axixk2JstQPvU
```

Copy-kan hasil random ini ke file passwd.

```
# vi /usr/local/cvsroot/
CVSROOT/passwd

supri:Axixk2JstQPvU:cvs
anonymous::cvs
```

Lalu rubah perizinan file passwd ini agar tidak bisa dibaca oleh sembarang user.

```
# chmod 400 /usr/local/
cvsroot/CVSROOT/passwd
# chown cvs.cvs /usr/local/
cvsroot/CVSROOT/passwd
```

Restart kembali service xinetd Anda

```
# /etc/init.d/xinetd restart
```

Cara penggunaan CVS

Setelah anda men-setting server CVS di komputer anda, selanjutnya saya akan menjelaskan cara pemakaian CVS itu sendiri.

1. Menaruh file project dalam CVS

Pertama-tama copy-kan file latihan yang ada di CD yang berada pada directory tuto-

rial, namanya latihan.tar.gz. Dimisalkan file ini berada di home direktori Anda.

Login sebagai user biasa, misal di sini saya memakai contoh user supri.

```
$ cd /home/supri
$ tar -xzf latihan.tar.gz
$ mkdir class
$ cd class
$ mkdir bfennema
$ cd bfennema
$ mkdir project
$ cd /home/supri/class/bfennema/project
$ cp /home/supri/Makefile .
$ cp /home/supri/main.c .
$ cp /home/supri/bar.c .
$ cp /home/supri/foo.c .
```

Selanjutnya kita akan menaruh keempat file tersebut ke CVS server, ganti IP 192.168.2.80 dan username-nya sesuai dengan hostname dan user di sistem Anda.

```
$ cd /home/supri/class/bfennema/project
$ export CVSROOT=:pserver:supri@192.168.2.80:/cvs
$ cvs login
password : supriaje

$ cvs import -m "Sample Program" project sample start
```

CVS seharusnya merespon seperti ini :

```
N project/Makefile
N project/main.c
N project/bar.c
N project/foo.c

No conflicts created by this import
```

2. Dasar penggunaan CVS

Tadi Anda telah menambahkan 'project' ke CVS repository Anda, lalu bagaimana jika anda ingin mengubah file yang telah ditaruh di CVS repository tersebut.

Untuk melakukan hal itu, lakukan cvs checkout. Agar lebih paham ikuti langkah berikut:

```
$ cd
$ cvs checkout project
```

CVS seharusnya merespon dengan:
cvs checkout: Updating project

```
U project/Makefile
U project/bar.c
U project/foo.c
U project/main.c
```

Maka akan tercipta folder project yang berisikan keempat file Makefile, bar.c, foo.c, and main.c, yang telah kita letakkan di CVS repository.

Sekarang kita akan merubah file main.c, lalu mengimportnya kembali ke CVS repository.

```
$ cd /home/supri/project
```

Sekarang tambahkan baris printf("DONE\n"); setelah function call ke bar() yang ada pada file main.c. (cara mudahnya lihat saja file main2.c)

```
$ vi main.c
```

tambahkan baris printf("DONE\n"); atau

```
$ cp /home/supri/main2.c main.c
```

Sekarang anda dapat check in copy file terbaru.

```
$ cvs commit -m "Added a DONE message." main.c
```

Seharusnya CVS merespon dengan:

```
Checking in main.c;
/class/'username'/cvsroot/project/main.c,v <-- main.c
new revision: 1.2; previous revision: 1.1
done
```

Catatan: option -m memberikan Anda *checking message* di command line. Jika tidak memakai option itu, maka pesan akan ditampilkan di editor di mana dapat mentipkan checking message.

3. Menggunakan CVS untuk beberapa programmer

Untuk menyimulasikan penjelasan dengan beberapa programmer, pertama ciptakan terlebih dahulu direktori untuk programmer kedua Anda.

Misal namanya Programmer2.

Klik Konsol baru, kemudian login ke CVS Server.

```
$ export CVSROOT=:pserver:supri@192.168.2.80:/cvs
```

```
$ cvs login
password : supriaje
```

Buat folder untuk programmer kedua ini di home direktori.

```
$ cd
$ mkdir devel2
```

Berikutnya, Check Out, copy project dari CVS Server.

```
$ cd devel2
$ cvs checkout project
```

Selanjutnya, dalam direktori devel2/project, tambahkan kalimat printf("YOU\n"); setelah printf("BAR\n"); yang ada dalam file bar.c

```
$ vi bar.c
```

tambahkan baris printf("YOU\n"); setelah printf("BAR\n");

```
atau
$ cp /home/supri/bar2.c bar.c
```

Selanjutnya Check in bar.c yang baru saja diubah oleh Programmer2.

```
$ cvs commit -m "Added a YOU" bar.c
```

Sekarang kembali ke Programmer1.

```
$ cd /home/supri/project
```

Lalu kita akan mengupdate file bar.c

```
$ cvs update bar.c
```

CVS seharusnya akan merespon dengan:
P bar.c

Sekarang lihat file bar.c. Seharusnya file itu isinya akan sama dengan yang dimiliki oleh Programmer2.

Selanjutnya, edit foo.c dan tambahkan baris printf("YOU\n"); setelah baris printf("FOO\n");

```
$ vi foo.c
```

atau

```
$ cp /home/supri/rgps/documentation/cvs/cvstutorial/foo2.c foo.c
```

lalu check in file foo.c tersebut.

```
$ cvs commit -m "Added YOU" foo.c
```

```

supriyanto@ws034:~/project
[supriyanto@ws034 project]$ cvs commit -m "Added a DONE message." main.c
Checking in main.c;
/cvs/project/main.c,v <-- main.c
new revision: 1.2; previous revision: 1.1
done
[supriyanto@ws034 project]$

```

Gambar 5. CVS commit.

```

supriyanto@ws034:~/project
[supriyanto@ws034 ~]$ cd
[supriyanto@ws034 ~]$ mkdir devel2
[supriyanto@ws034 ~]$ cd devel2/
[supriyanto@ws034 devel2]$ cvs checkout project
cvs checkout: Updating project
U project/Makefile
U project/bar.c
U project/foo.c
U project/main.c
[supriyanto@ws034 devel2]$ ls
project
[supriyanto@ws034 devel2]$ cd project/
[supriyanto@ws034 project]$ vi bar.c
[supriyanto@ws034 project]$ cvs commit -m "Added a YOU" bar.c
Checking in bar.c;
/cvs/project/bar.c,v <-- bar.c
new revision: 1.2; previous revision: 1.1
done
[supriyanto@ws034 project]$

```

Gambar 6. Check In bar.c oleh Programmer-2.

Selanjutnya kembali lagi ke direktori Programmer2.

```
$ cd /home/supri/devel2/project
```

Berikutnya tambahkan kalimat `printf("TOO\n");`; setelah baris `printf("FOO\n");`;

```
$ vi foo.c
```

tambahkan baris `printf("TOO\n");`; setelah `printf("FOO\n");`; atau

```
$ cp /home/foo3.c foo.c
```

Lalu ketikkan:

```
$ cvs status foo.c
```

CVS seharusnya merespon dengan:

```

=====
File: foo.c      Status: Needs
Merge

Working revision: 1.1.1.1
'Some Date'
Repository revision: 1.2 /
class/'username'/cvsroot/
project/foo.c,v
Sticky Tag:      (none)
Sticky Date:     (none)
Sticky Options:  (none)

```

Status yang lainnya adalah:

1. Up-to-date
File identik sama dengan revision terakhir yang ada direpository.
2. Locally Modified
Anda dapat mengedit file, dan belum meng-commit perubahan Anda.
3. Needing Patch
Seseorang telah meng-commit *revision* yang lebih baru ke repository.
4. Needs Merge

Seseorang telah mengcommit revision yang lebih baru ke repository.

Meski demikian, Programmer2 dapat memerge perubahan yang dia buat dengan perubahan yang dibuat oleh Programmer1.

Ketikkan perintah ini:

```
$ cvs update foo.c
```

CVS seharusnya meresponnya dengan:

```

RCS file: /class/'username'/'
cvsroot/project/foo.c,v
retrieving revision 1.1.1.1
retrieving revision 1.2
Merging differences between
1.1.1.1 and 1.2 into foo.c
rcsmerge: warning: conflicts
during merge
cvs update: conflicts found in
foo.c
C foo.c

```

Ketika perubahan yang kita buat di tiap version keluar bersamaan, kita harus membandingkan secara manual `foo.c` untuk melihat apa yang ingin kita lihat.

Lihat pada file `foo.c`, kita akan melihat :

```

void foo()
{
    printf("FOO\n");
<<<<<<< foo.c
    printf("TOO\n");
=====
    printf("YOU\n");
>>>>>> 1.2
}

```

Kita lihat kalau text yang telah ditambahkan oleh Programmer1 antara:

```
===== dan >>>>>> 1.2.
```

Sedangkan text yang Anda tambahkan

antara:

```
===== dan <<<<<<< foo.c
```

Untuk membetulkannya, pindahkan baris `printf("TOO\n");` ke setelah baris `printf("YOU\n");`; dan hapus tambahan baris yang CVS tambahkan.

```
$ vi foo.c
```

lalu ubah sesuai dengan perintah di atas. atau

```
$ cp /home/supri/foo4.c foo.c
```

Berikutnya, commit `foo.c`

```
$ cvs commit -m "Added T00"
foo.c
```

Branches in CVS

Salah satu kegunaan dari CVS, terutama ketika memanajemen beberapa software produk secara bersamaan, CVS dapat membuat percabangan ke dalam *revision tree*.

Versi number dari RCS tidak dapat melakukan release number dari suatu produk software. Meski demikian, CVS memiliki perintah `tag` yang membiarkan Anda memberikan nama symbolic ke revision tertentu dari suatu file. Option `-v` ke perintah `cvs status` membiarkan anda untuk melihat setiap tags dalam file.

Kembali lagi ke contoh kasus, pindahkan ke directory programmer1, dan ketikkan :

```
$ cvs update foo.c
```

Untuk mendapatkan revision terbaru dari `foo.c`

Selanjutnya Anda ingin merilis produk software Anda. Untuk melakukan itu, kita ingin memberikan tag semua file dengan tag `release-1`.

```
$ cvs tag release-1 .
```

```

supriyanto@ws034:~/project
File Edit View Terminal Tabs Help
root@ws034:/usr/local/cvsroot  supriyanto@ws034:~/devel2/project  supriyanto@ws034:~/project
[supriyanto@ws034 project]$ cat bar.c
void bar()
{
    printf("BAR\n");
}
[supriyanto@ws034 project]$ cvs update bar.c
P bar.c
[supriyanto@ws034 project]$ cat bar.c
void bar()
{
    printf("BAR\n");
    printf("YOU\n");
}
[supriyanto@ws034 project]$ ls
bar.c  CVS  foo.c  main.c  Makefile
[supriyanto@ws034 project]$

```

Gambar 7. Check Out bar.c oleh Programmer-1.

```

supriyanto@ws034:~/project
File Edit View Terminal Tabs Help
root@ws034:/usr/local/cvsroot  supriyanto@ws034:~/devel2/project  supriyanto@ws034:~/project
[supriyanto@ws034 project]$ cvs status foo.c
=====
File: foo.c      Status: Needs Merge
Working revision: 1.1.1.1
Repository revision: 1.2 /cvs/project/foo.c,v
Sticky Tag: (none)
Sticky Date: (none)
Sticky Options: (none)
[supriyanto@ws034 project]$ cvs update foo.c
RCS file: /cvs/project/foo.c,v
retrieving revision 1.1.1.1
retrieving revision 1.2
Merging differences between 1.1.1.1 and 1.2 into foo.c
rcsmerge: warning: conflicts during merge
cvs update: conflicts found in foo.c
C foo.c
[supriyanto@ws034 project]$ vi foo.c
[supriyanto@ws034 project]$ clear

```

Gambar 8. Check Status Foo.c.

CVS seharusnya meresponnya dengan:

```

cvs tag: Tagging .
T Makefile
T bar.c
T foo.c
T main.c

```

Jika beberapa programmer lain ingin menge-check out release-1, dia dapat mengetikkan perintah cvs berikut:

Misal ada programmer3, dan dia ingin menge-check out release-1, maka sebelum dia ingin menge-check out dia harus login ke cvs server terlebih dahulu.

```

$ cd
$ mkdir devel3
$ cd devel3
$ export CVSR00T=:pserver:
supri@192.168.2.80:/cvs
$ cvs login
password : supriaje
$ cvs checkout -r release-1
project

```

Dia akan mendapatkan *release* versi, di samping versi terbaru. (jika lebih dari satu).

Sekarang kita memiliki release version, berikutnya kita akan memulainya ke version 2. Misalkan kita berandai-andai, kustomer kita komplain tentang kesalahan fatal di versi 1. Versi 2 mungkin akan mengalami hal yang sama enam bulan kemudian, dan bugs di version 1 perlu dibetulkan sekarang.

Pertama, Anda dapat menghapus dua buah *development trees* dari tutorial sebelumnya. Caranya mudah untuk melakukannya dengan CVS. Pindah ke directory dimana directory Programmer1 berada, dan ketikkan:

```

$ cd
$ cvs release -d project

```

CVS seharusnya merespon dengan:

```

You have [0] altered files in
this repository.
Are you sure you want to release
(and delete) module `project':
To this response: y

```

Option -d memberitahukan release cvs yang akan dihapus di direktori kerja.

Anda dapat melakukan hal yang sama untuk directory programmer2.

Selanjutnya kita akan menciptakan branch off version 1.

Kembali ke directory home, dan ketikkan:

```

$ cd
$ cvs rtag -b -r release-1
release-1-patches project

```

CVS seharusnya meresponnya dengan

```

cvs rtag: Tagging project

```

Selanjutnya kita memerlukan copy direct dari branch yang telah kita ciptakan.

```

$ cvs checkout -r release-1-
patches project

```

CVS seharusnya meresponnya dengan

```

cvs checkout: Updating project
U project/Makefile
U project/bar.c
U project/foo.c
U project/main.c

```

Sekarang anda dapat membetulkan bugs di version 1. Setiap perubahan yang telah di-commit akan di-commit kembali ke branch dan bukannya ke *main trunk*.

Misalkan anda membetulkan bug antara swap YOU and TOO printf's di foo.c

```

Ketikkan saja perintah berikut :
$ vi foo.c

```

lakukan seperti perintah di atas

atau

```

$ cp /home/supri/rgps/
documentation/cvs/cvstutorial/
foo5.c foo.c

```

Selanjutnya check in kembali foo.c.

```

$ cvs commit -m "Fixed printf
bug" foo.c

```

Seharusnya CVS akan merespon dengan hasil berikut:

```

Checking in foo.c;
/usr/local/cvsroot/project/
foo.c,v <-- foo.c
new revision: 1.3.2.1; previous
revision: 1.3
done

```

Hal ini juga memungkinkan untuk *merge branch* kembali ke main tree. Untuk melakukan ini, pertama release patch version.

```

$ cd
$ cvs release -d project

```

Berikutnya, gunakan option -j 'branch' agar cvs checkout ke merge release-1 dan release-1-patches.

```

$ cvs checkout -j release-1-
patches project

```

CVS seharusnya meresponnya dengan:

```

cvs checkout: Updating project
U project/Makefile
U project/bar.c
U project/foo.c
RCS file: /class/'username'/
cvsroot/project/foo.c,v
retrieving revision 1.3
retrieving revision 1.3.2.1
Merging differences between 1.3
and 1.3.2.1 into foo.c
U project/main.c

```


Jika ada konflik yang terjadi di sana, maka kita harus membetulkannya secara manual.

Setelah hal ini dilakukan Anda dapat meng-commit-nya, dengan mengetikkan perintah berikut:

```
$ cvs commit -m "Merged patch"
cvs commit: Examining project
Checking in project/foo.c;
/usr/local/cvsroot/project/
foo.c,v <-- foo.c
new revision: 1.4; previous
revision: 1.3
done
cvs commit: Examining 2.4routing
```

Untuk meng-commit semua file yang berubah adalah dengan cara me-merge release-1-patch ke dalam source tree sekarang.

Tambahan perintah CVS

Menambahkan sebuah file ke module:

Ambil file copy kerja dari modules dengan menggunakan cvs checkout

```
$ cd
```

```
$ mkdir testing
$ cd testing
$ cvs checkout -j release-1-
patches project
$ cvs commit -m "Merged patch"
```

Buat sebuah file baru ke dalam directory, misalkan tambah.c

```
$ vi tambah.c
```

Isikan, lalu save

```
$ cvs add tambah.c
```

Gunakan cvs commit untuk check in file ke repository.

```
$ cvs commit baru
```

Menghapus sebuah file dari module:

Yakinkan bahwa tidak ada *uncommitted* modifikasi file.

Lalu hapus sebuah file dari direktori kerja anda, misal, main.c

```
$ cd
$ cd testing/project
$ rm main.c
```

Untuk menghapus file tersebut gunakan perintah cvs remove

```
$ cvs remove main.c
$ cvs commit main.c
```

Demikian tutorial CVS yang dapat saya tulis. Memang pada awalnya terasa sulit untuk memahami konsep-konsep yang ada di CVS. Tetapi setelah Anda memahami konsep yang ada di CVS ini, maka segala kesulitan itu akan terbayar dengan berkurangnya waktu yang digunakan untuk sinkronisasi file yang terdapat dalam pembuatan *software*. Contoh konkretnya seperti yang ada di situs *sourceforge.net*, hampir semua project open source yang ada di situs tersebut menyediakan fasilitas akses CVS server bagi para developer yang berminat ikut serta.

Semoga dengan contoh sederhana yang ada pada artikel ini, dapat membantu anda untuk penerapan pengerjaan software development yang lebih besar secara efektif dan efisien. 🙏

Supriyanto (supriyanto@infolinux.co.id)

Colocation dengan 3 Internet Backbone (USA, EROPA & SINGAPORE)

harga mulai dari Rp. 1.325.000,-



Anda mendapatkan :

- 🔗 128 Kbps Bandwidth Internasional
- 🔗 Unlimited 100 Mbps IIX Bandwidth
- 🔗 8 Public IP Address
- 🔗 Technical Support 24 x 7 x 365
- 🔗 99.9% Uptime Guarantee
- 🔗 UPS Backup

Email : sales@saranadata.com

Telp : +62 (21) 526 9257

Business Services :

- Colocation Server
- Webhosting Colocation
- Dedicated Server
- Rack Solution
- Internet Service Solution
- Corp. Application Hosting
- System Mirroring & DRC
- Data Storage Services
- Custom Made Solution

Personal Services :

- Download Services
- E- File Storage
- E- Photo Album