BAB III

KEGIATAN SELAMA KERJA PRAKTEK

UNIKOM

3.1 Jadwal Kerja Praktek

Penulis melaksanakan Kerja Praktek selama 30 hari mulai dari tanggal 12 Juli sampai dengan 25 Agustus 2004. Adapun waktu pelaksanaan Kerja Praktek dari hari senin sampai dengan hari jumat, dengan jam masuk disesuaikan dengan jadwal masuk kerja pembimbing kerja praktek

3.2 Data Kerja Praktek

3.2.1 Konsep dasar CVS

CVS (*Concurrent Versioning System*) adalah sebuah sistem kendali versi, atau suatu program manajemen versi yang mengatur perubahan versi file pada suatu proyek yang melibatkan banyak file dan atau banyak developer yang dapat bekerja pada saat yang bersamaan. Suatu program manajemen versi memungkinkan banyak developer mengerjakan suatu proyek secara bersama dan juga memungkinkan satu developer yang mengerjakan suatu proyek untuk mengontrol versi file - file yang sedang dikembangkannya.. CVS umumnya digunakan sebagai salah satu bagian penting dalam pengembangan perangkat lunak di mana sistem ini berfungsi melacak dan mengkoordinasikan setiap perubahan *source code* yang dilakukan oleh banyak developer Sebagai contoh, dalam hal pemrograman, sebuah program biasanya tidak akan dikembangkan sekaligus tapi bertahap, mungkin ada fungsi-fungsi yang belum

diimplementasikan, yang penting programnya sudah mencapai tahap tertentu. Setelah yakin bahwa semua sudah benar baru kemudian ditambahkan fungsi - fungsi yang lain. Dari banyak pengalaman, bahwa kadangkala program yang dimodifikasi menjadi *error*, tentunya kita akan mem-*back up* terlebih dahulu file itu, sebelum masuk tahap selanjutnya. Dalam hal ini sebenarnya kita telah membuat versi baru dari file kita dan kita telah melakukan manajemen versi dengan mengatur versi - versi file yang kita miliki, sehingga kita dapat memakai file (versi) yang lama dengan mudah jika terjadi kesalahan. Manajemen versi file tidak hanya dilakukan untuk *source code* program, hal ini juga bisa dilakukan untuk file apa saja.

Manajemen versi dapat dilakukan secara manual namun hal ini akan sangat menyulitkan, kita bisa saja menyimpan setiap file yang pernah dibuat atau diubah. Namun hal ini tentu saja akan menghabiskan banyak ruang di harddisk. Hal tersebut bukan masalah jika menggunakan CVS karena CVS menyimpan setiap versi file yang diciptakan di dalam satu buah file. File yang dihasilkan oleh CVS tersebut hanya menyimpan perbedaan antara setiap versi file yang pernah dibuat, maka dengan menyimpan hanya perubahan dari file yang kita buat di setiap versi, kita cukup mengatur hak akses terhadap file yang disimpan itu.

CVS adalah program manajemen versi yang dirancang untuk menangani proyek dengan file teks murni seperti program, dokumentasi

dalam format teks (misalnya dalam *txt*, *html, sgml,* atau *LaTex*) bukan file atau dokumen biner seperti dokumen milik Microsoft Office

Bug biasanya muncul pada saat sebuah perangkat lunak sedang dimodifikasi, dan mungkin tidak disadari keberadaan *bug* tersebut untuk waktu yang lama atau sampai ada orang yang melaporkan kesalahan pada program itu. Dengan menggunakan CVS, kita dapat dengan mudah mendapatkan versi lama *software* untuk mengetahui perubahan apa yang dilakukan, yang menyebabkan timbulnya *bug*. Hal ini akan sangat membantu bagi para developer *software.*

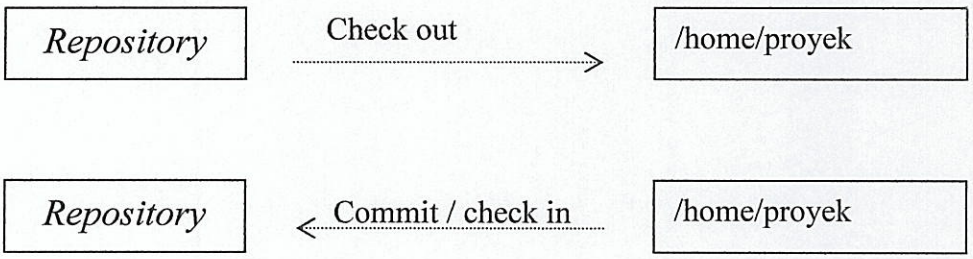
.CVS juga dapat membantu jika kita adalah bagian dari sekelompok orang yang bekerja pada suatu proyek yang sama. Akan sangat mudah terjadi proses saling menimpa (*overwrite*) program yang dikerjakan oleh satu orang dengan yang lainnya sehingga kita harus sangat berhati-hati. Beberapa editor, (misalnya *GNU Emacs*) tidak mengijinkan sebuah file yang sama dimodifikasi oleh dua orang atau lebih di saat yang sama. Namun, jika orang kedua menggunakan editor selain *Emacs* fungsi tersebut menjadi tidak berfungsi. CVS dapat mengatasi masalah tersebut dengan cara memisahkan / mengisolasi developer yang satu dengan lainnya. Setiap developer bekerja di direktori kerja masing - masing, kemudian CVS akan menggabungkan proyek saat semua developer telah menyelesaikan pekerjaannya.

3.2.2 Prinsip Kerja CVS

UNIT PELAKSANA TEKNIS PERPUST UNIKO

CVS merupakan suatu sistem yang berjalan secara *Client / Server*, jadi bila kita ingin menjalankan CVS kita harus terhubung dengan mesin server, tetapi CVS bisa juga berjalan secara *standalone* atau menggunakan server *localhost*

Dalam CVS, suatu file dalam proyek disimpan di suatu tempat yang disebut dengan *repository*. Jika ada yang ingin mengakses file dalam repository maka harus melakukan proses *login* ke dalam CVS setelah itu melakukan *checkout* yaitu menyalin file dari repository ke direktori tempat bekerja, setelah pekerjaannya selesai maka file – file tadi disalin kembali ke repository, disebut juga melakukan *checkin* atau *commit*.



Gambar 3.1 Prinsip kerja CVS

3.2.3 Fitur CVS

- Dapat mengambil kembali versi lama software.
- Dapat menyimpan semua versi dari tiap file yang dibuat.
- Dapat mengatur hak akses file
- Memungkinkan akses terhadap file secara bersama (konkuren) tidak hanya untuk dibaca tapi untuk ditulisi

- Menyimpan hanya perubahan file (tidak keseluruhan file)
- Dapat melakukan pencatatan perubahan yang ada pada file awal, sehingga *disk space* yang digunakan lebih sedikit
- Dapat melakukan *forking* (menciptakan cabang dalam pengembangan perangkat lunak sehingga ada dua versi atau lebih yang dikembangkan dengan jalur yang berbeda, yang berasal dari perangkat lunak yang sama)
- CVS juga bermanfaat jika kita adalah anggota sekelompok orang yang bekerja pada proyek yang sama. Dengan proyek yang dikerjakan oleh banyak orang, maka kemungkinan terjadinya konflik pada file - file *source code* sangat besar, misalnya satu orang merubah file yang telah dirubah oleh orang lainnya. CVS mengatasi masalah ini dengan cara mengisolasi para pengembang satu dengan yang lain. Setiap pengembang bekerja di direktorinya, dan CVS menggabungkan pekerjaannya ketika mereka telah selesai

3.2.4 CVS 1.11.2

CVS pada awalnya bekerja menggunakan *command line,* namun kini telah berkembang sehingga telah ada aplikasi CVS Client yang berbasis GUI seperti TurtoiseCVS. Namun di sini tidak akan dibahas mengenai TurtoiseCVS tersebut melainkan CVS secara keseluruhan baik sebagai Client ataupun Server. Agar kita dapat mengerti CVS secara global maka akan dijelaskan penggunaan software tersebut dalam bentuk

perintah-perintah pada *command line* dengan menggunakan CVS versi 1.11.2.

3.2.5 Penggunaan Dasar CVS

3.2.5.1. Repository

Repository CVS adalah suatu tempat atau folder dimana menyimpan secara lengkap salinan versi semua file dan direktori yang ada dalam *version control*di komputer server.

Kita tidak pernah mengakses file secara langsung dari repository, tetapi kita menggunakan perintah dari CVS untuk mendapatkan salinan dari file – file tersebut secara khusus ke dalam direktori kerja, lalu bekerja dengan file di direktori tersebut. Setelah kita selesai melakukan perubahan pada file di direktori kerja kita, kita lakukan *commit* atau *checkin* kembali ke dalam repository. Repository kemudian akan berisi perubahan yang telah dilakukan, rekaman semua perubahan, waktu peubahan dan informasi lainnya. Repository bukan subdirektori dari direktori kerja kita dan juga sebaliknya, maka keduanya harus berada di lokasi yang terpisah

CVS dapat mengakses *repository* dengan berbagai cara, dapat mengakses di komputer lokal, di computer lain dalam jaringan lokal ataupun di lingkungan jaringan *World Wide Web*. Untuk memperjelas cara aksesnya nama *repository* diawali dengan *"access method"*, contoh, metode akses `':local:'` berarti mengakses direktori *repository*,

`':local:/usr/local/cvsroot'` artinya repository berada di komputer lokal di direktoi `'/usr/local/cvsroot'`.

Repository dibagi menjadi dua bagian, direktoi CVSROOT berisi file – file administrative untuk CVS, sedangkan direktori lainnya berisi modul yang *user defined.*

3.2.5.2. Contoh Singkat

Untuk mempermudah dalam mempelajari CVS, kita akan melakukan beberapa langkah contoh kerja menggunakan CVS. Yang pertama harus anda ketahui adalah CVS menyimpan semua file dalam repository terpusat. Di sini diasumsikan bahwa repository telah dikonfigurasikan sebelumnya.

Misalnya anda bekerja pada sebuah compiler sederhana. Sourcenya terdiri dari file berbahas C dan sebuah `'Makefile'`. Compilernya bernama `'tc'` (Trivial Compiler), dan repository diset sehingga ada module yang bernam `'tc'`.

1. Mendapatkan Source

Yang pertama harus anda lakukan adalah mendapatkan copy dari file `'tc'` dengan menggunakan perintah checkout.

```
$ cvs checkout tc
```

Hal ini akan menciptakan sebuah direktori baru dengan nama `'tc'` yang berisi file source.

```
$ cd tc
$ ls
CVS  Makefile  backend.c  driver.c  frontend.c  parser.c
```

Direktori CVS digunakan secara internal oleh CVS. Biasanya, anda tidak diperkenankan untuk memodifikasi ataupun menghapus satupu file di dalamnya.

Mulailah memodifikasi dengan editor yang biasa anda gunakan. Misalnya kita modifikasai file `'backend.c'`. Catatan untuk pengguna RCS dan SCCS : tidak perlu melakukan lock pada file yang akan anda edit.

2. Commit Perubahan File

Jika ternyata program yang anda modifikasi dapat di compile dengan sukses, anda ingin membuat versi baru dari `'backend.c'`. Di sini `'backend.c'` akan disimpan di repository sehingga dapat diakses oleh orang lain yang menggunakan repository tesebut.

```
$ cvs commit backend.c
```

CVS membuka sebuah editor, yang memungkinkan anda untuk memasukkan log message. Anda ketik di `"Added an optimization pass."`, save file temporary tersebut, lalu keluar dari editor.

Environtment variable `'$CVSEDITOR'` menunjukkan editor apa yang telah dibuka. Jika `'$CVSEDITOR'` tidak ditentukan, tetapi variabel `'$EDITOR'` ditentukan maka editor tersebut yang akan digunakan. Jika kedua variabel tersebut tidak ditentukan maka ada editor default tergantung pada Sistem Operasi yang anda pakai, misalnya 'vi' untuk Unix atau 'notepad' untuk Windows NT/95.

Sebagai pelengkap, CVS memeriksa environment variable `'$VISUAL'`. Ada beberapa pendapat mengenai perlu atau tidaknya

variabel tersebut disertakan. Diset atau tidaknya variabel tersebut tidak akan berpengaruh.

Saat CVS membuka editor, daftar file-file yang telah dimodifikasi disertakan. Untuk CVS client, daftar ini didasarkan pada pembandingan waktu modifikasi file tersebut sekarang dengan waktu modifikasi terakhir file tersebut diupdate. Oleh karena itu, jika waktu modifikasi file berubah walaupun isi file tidak berubah, status dari file tersebut akan menjadi modified. Anda tidak perlu khawatir dengan hal itu karena saat commit CVS dapat mendeteksi bahwa isi file tidak dimodifikasi sehingga statusnya menjadi unmodified. Saat update selanjutnya, CVS dapat mengetahui file tersebut tidak dimodifikasi, dan akan mereset timestamp yang dimilikinya sehingga file itu tidak akan muncul lagi di sesi editor di masa yang akan datang.

Jika anda tidak ingin ada editor, maka anda dapat menentukan log message pada command line menggunakan flag `'-m'` seperti berikut :

```
$ cvs commit -m "Added an optimization pass" backend.c
```

3. Pembersihan

Sebelum anda berpindah ke pekerjaan lain, jika anda ingin menghapus copy `'tc'` yang anda kerjakan caranya adalah :

```
$ cd ..
$ rm -r tc
```

Tapi lebih baik menggunakan perintah release :

```
$ cd ..
$ cvs release -d tc
M driver.c
? tc
You have [1] altered files in this repository.
Are you sure you want to release (and delete) directory 'tc' : n
```

```
** 'release' aborted by users choice.
```

Perintah release memeriksa bahwa semua modifikasi anda telah dicommit. Jika history logging diaktifkan maka akan dibuat pula sebuah catatan di file history. Saat anda menggunakan flag `'-d'` dengan release, copy file kerja anda akan dihapus.

Pada contoh di atas, perintah release menghasilkan dua baris output. `'? tc'` memiliki arti bahwa file `'tc'` tidak dikenali oleh CVS. Anda tidak perlu khawatir karena `'tc'` adalah compiler yang executable, dan tidak disimpan di repository.

`'M driver.c'` merupakan peringatan yang lebih berarti, karena berarti bahwa file `'driver.c'` telah berubah semenjak di-check out.

Perintah release selalu diakhiri dengan pemberitahuan berapa banyak file yang telah dimodifikasi pada copy file kerja anda, dan kemudian memberikan pertanyaan konfirmasi penghapusan file dan membuat beberapa catatan di file history.

Pada contoh di atas, anda memberikan jawaban `'n <RET>'` saat release meminta konfirmasi.

4. Melihat Perbedaan

Anda tidak ingat pernah memodifikasi `'driver.c'`, jadi anda ingin mengetahui apa yang terjadi pada file tersebut.

```
$ cd tc
$ cvs diff driver.c
```

Perintah tadi menjalankan diff untuk membandingkan versi `'driver.c'` yang anda checkout dengan yang anda di copy file kerja

anda. Saat anda melihat outputnya anda ingat pernah menambahkan command line option yang mengaktifkan optimization pass. Anda lakukan checkin, dan melakukan release pada module tersebut.

```
cvs commit -m "Added an optimization pass" driver.c
Checking in driver.c;
/usr/local/cvsroot/tc/driver.c,v  <--  driver.c
new revision: 1.2; previous revision: 1.1
done
$ cd ..
$ cvs release -d tc
You have [1] altered files in this repository.
Are you sure you want to release (and delete) directory 'tc' : y
```

3.2.6 Penggunaan CVS Lebih Lanjut

Pada bagian ini akan dijelaskan mengenai struktur perintah-perintah pada CVS secara lebih menyeluruh dan terperinci.

1. Struktur

Format keseluruhan dari semua perintah (command) CVS adalah :

```
cvs [cvs_options] cvs_command [command_option] [command_args]
```

`'cvs'`

Nama dari program CVS.

`'cvs_options'`

Beberapa option yang mempengaruhi semua sub-perintah CVS.

`'cvs_command'`

Salah satu diantara beberapa sub-perintah yang berbeda. Beberapa perintah memiliki nama lain (alias) yang bisa digunakan. Ada dua situasi di mana anda dapat tidak memakai `'cvs_command'` yaitu : `'cvs -H'` untuk menampilkan daftar perintah yang ada, dan `'cvs -v'` untuk menampilkan informasi versi dari CVS itu sendiri.

'cvs_options'

Option yang bersifat spesifik untuk perintah (command).

'command_args'

Argumen untuk perintah yang diberikan.

Namun ada beberapa kebingungan antara 'cvs_options' dan 'command _options'. Misalnya '-l' jika digunakan sebagai 'cvs_options' hanya mempengaruhi beberapa perintah, sedangkan jika digunakan sebagai 'command_options' akan memiliki arti yang berbeda dan diterima oleh lebih banyak perintah.

2. Exit Status

CVS dapat menunjukkan keberhasilan atau kegagalan yang terjadi kepada calling environment dengan cara mengatur "exit status"-nya. Cara memeriksa exit status akan berbeda pada tiap sistem operasi. Contohnya pada unix shell script, variabel '$?' akan bernilai 0 jika perintah terakhir memberikan exit status yang sukses, dan akan bernilai lebih besar dari 0 jika exit status mengindikasikan kegagalan.

Jika CVS sukses, maka akan menghasilkan status sukses. Jika terjadi error, akan keluar pesan kesalahan dan memberikan status kegagalan. Satu pengecualian adalah pada perintah 'cvs diff' di mana akan memberikan status sukses jika tidak menemukan perbedaan, atau status kegagalan jika ditemukan perbedaan ataupun error.

3. ~/.cvsrc

Ada beberapa `command_options' yang sangat sering digunakan sehingga anda membuat sebuah alias untuk memastikan option tersebut selalu

dipakai. Contoh, banyak orang kesulitan membaca output default dari perintah `'diff'`, dan tidak ada yang lebih mudah dimengerti antara konteks diff ataupun unidiff.

File `` `~/.cvsrc' `` adalah suatu cara sehingga anda dapat menambahkan default option pada `` `cvs_command' `` dalam cvs, daripada harus bergantung pada alias ataupun shell script lainnya.

Format `file` `` `~/.cvsrc' `` adalah sederhana. File akan dicari untuk sebuah baris yang diawali dengan nama yang sama seperti `` `cvs_command' `` yang sedang dieksekusi. Jika ada yang cocok, maka baris tersebut akan dipecah (dengan karakter spasi) menjadi option yang terpisah dan ditambahkan ke argumen perintah sebelum option lainnya dari command line.

Jika sebuah perintah memiliki dua nama (misal, `'checkout'` dan `'co'`), nama aslinya, tidak perlu sama dengan yang digunakan di command line akan digunakan untuk dicocokkan ke file tersebut. Jadi jika berikut ini adalah isi dari file `` `~/.cvsrc' `` anda :

```
log -N
diff -uN
rdiff -u
update -Pd
checkout -P
release -d
```

Perintah `` `cvs checkout foo' `` akan memiliki option `'-P'` yang ditambahkan ke argumennya, sama saja untuk `` `cvs co foo' ``.

Dengan contoh file di atas, output dari `` `cvs diff foobar' `` akan dalam format unidiff. `` `cvs diff -c foobar' `` akan memberikan context diffs. Mendapatkan format diff yang lama ("old") akan lebih rumit, karena `'diff'`

tidak memiliki option untuk menggunakan format "old", jadi anda akan membutuhkan `cvs -f diff foobar'.

Pada nama command anda dapat menggunakan 'cvs' untuk menentukan option global. Misalnya berikut adalah baris dalam file '.cvsrc'

```
cvs -z6
```

ini akan membuat CVS menggunakan kompresi level 6.

4. Option Global

Berikut ini adalah daftar 'cvs_options' (sebelah kiri 'cvs_command') yang dapat digunakan :

'--allow-root=ROOTDIR'

Menentukan direktori CVSROOT yang legal.

'-a'

Mengautentikasi semua komunikasi antara client dengan server. Hanya berpengaruh pada CVS client (diimplementasikan menggunakan koneksi GSSAPI). Autentikasi mencegah berbagai macam serangan yang berupa pembajakan koneksi TCP yang aktif. Mengaktifkan autentikasi tidak mengaktifkan enkripsi.

'-b BINDIR'

Pada CVS 1.9.18 dan yang lebih lama, option ini menspesifikasikan bahwa program RCS berada dalam direktori BINDIR. Versi CVS sekarang tidak menjalankan program RCS, untuk kompatibilitas option ini dapat digunakan meskipun tidak menjalankan apapun.

'-T TEMPDIR'

Memakai TEMPDIR sebagai direktori di mana file-file temporary disimpan. Tidak mengacuhkan pengaturan pada variabel environment '$TMPDIR' dan

semua precompiled directory. Parameter ini harus ditentukan sebagai pathname yang absolut. Saat menjalankan client/server '-T' hanya mempengaruhi proses lokal, memakai '-T' untuk client tidak akan berpengaruh pada server dan sebaliknya.

'-d CVS_ROOT_DIRECTORY'

Menggunakan CVS_ROOT_DIRECTORY sebagai pathname untuk direktori root pada repository. Tidak mengacuhkan seting pada variabel '$CVSROOT'.

'-e EDITOR'

Menggunakan EDITOR untuk memasukkan informasi log revisi. tidak mengacuhkan pengaturan pada variabel environment '$CVSEDITOR' dan '$EDITOR'.

'-f'

Tidak membaca file '~/.cvsrc'. Option ini lebih sering digunakan karena non-ortogonalitas dari penggunaan option CVS. Contohnya, '-N' berfungsi untuk mematikan tampilan nama tag, pada 'cvs log' tidak memiliki option untuk mematikan tampilan. Jadi jika anda memiliki '-N' dalam '~/.cvsrc' untuk 'log', anda dapat menggunakan '-f' untuk menampilkan nama tag.

'-H'

'--help'

Menampilkan informasi penggunaan perintah 'cvs_command' yang diberikan tanpa menjalankan perintah tersebut. Jika anda menuliskan 'cvs -H' maka akan ditampilkan help untuk CVS secara keseluruhan.

'-l'

Tidak mencatat 'cvs_command' ke dalam command history (perintah tetap dijalankan).

'-n'

Tidak mengubah file apapun. Mencoba menjalankan 'cvs_command', tetapi hanya untuk memberikan report, tanpa menghapus, update, merge, ataupun membuat file baru.

'-Q'

Menyebabkan perintah berjalan secara sangat quiet, perintah tersebut hanya akan memberikan output jika terjadi masalah serius.

'-q'

Menyebabkan perintah berjalan secara agak quiet.

'-r'

Membuat file kerja baru read-only. Hasilnya sama seperti jika anda mengaktivkan variabel environment '$CVSREAD'. Defaultnya adalah membuat file kerja dapat ditulisi (writable), kecuali watches diaktifkan.

'-s VARIABLE=VALUE'

Menetapkan sebuah variabel user.

'-t'

Melacak (trace) eksekusi program, menampilkan pesan yang menunjukkan langkah-langkah aktifitas CVS. Berguna digunakan bersama '-n' untuk memeriksa apa yang akan terjadi terhadap sebuah perintah yang tidak dikenal.

'-v'

'--version'

Menampilkan versi dan informasi hak cipta CVS.

'-w'

Membuat file kerja baru read-write. Tidak mengacuhkan pengaturan pada '$CVSREAD'. file akan dibuat read-write secara default, kecuali '$CVSREAD'

di aktifkan atau diberikan option `'-r'`.

`'-x'`

Mengenkripsi seluruh komunikasi antara client dengan server. Hanya berpengaruh terhadap pada CVS client (diimplementasikan menggunakan koneksi Kerberos). Dukungan enkripsi tidak tersedia secara default, harus diaktifkan menggunakan configure option khusus yaitu option `'--enable-encryption'` saat anda membangun CVS.

`'-z GZIP-LEVEL'`

Mengatur level kompresi. nilainya antara 1 (cepat, kompresi rendah) sampai 9 (lambat,kompresi tinggi), atau 0 untuk tidak mengaktifkan kompresi. Hanya memberikan efek pada CVS client.

5. Option Umum

Berikut ini adalah beberapa `'command_options'` yang tersedia pada perintah-perintah CVS.

Perhatikan bahwa perintah `'history'` adalah satu pengecualian karena bisa terjadi beberapa konflik jika menggunakan option standar berikut.

`'-D DATE_SPEC'`

Gunakan revisi terakhir sebelum tanggal `DATE_SPEC`. Spesifikasi ini bersifat "sticky", saat anda mendapatkan file kerja anda menggunakan `'-D'`, CVS akan merekam tanggal yang anda tentukan dan menggunakannya untuk update di masa yang akan datang. `'-D'` bisa digunakan bersama perintah `'annotate'`, `'checkout'`, `'diff'`, `'export'`, `'history'`, `'rdiff'`, `'rtag'`, dan `'update'`. Contoh perintah yang menggunakan flag `'-D'` :

```
$ cvs diff -D "1 hour ago" cvs.textinfo
```

'-f'

Saat anda menetapkan tanggal atau tag pada perintah CVS, file yang tidak memiliki tag yang ditentukan akan diabaikan. Gunakan '-f' agar file dapat diperbaiki meskipun tidak memiliki tag ataupun tanggal yang ditentukan.

'-f' digunakan bersama perintah 'annotate', 'checkout', 'export', 'rdiff', 'rtag', dan 'update'.

'-k KFLAG'

Jika anda menggunakannya bersama perintah 'checkout' atau 'update', CVS akan menghubungkan KFLAG dengan file anda dan terus digunakan di masa depan. '-k' dapat digunakan untuk perintah 'add', 'checkout', 'diff', 'import', dan 'update'.

'-l'

lokal ; hanya berjalan di direktori kerja yang sedang anda pakai, tidak terhadap sub-direktorinya. Digunakan untuk perintah 'annotate', 'checkout', 'commit', 'diff', 'edit', 'editors', 'export', 'log', 'rdiff', 'remove', 'rtag', 'status', 'tag', 'unedit', 'update', 'watch' dan 'watchers'.

'-m MESSAGE'

Menggunakan MESSAGE sebagai informasi log. Digunakan untuk 'add', 'commit', dan 'import'.

'-n'

Tidak menjalankan program checkout/commit/tag. Disediakan untuk perintah 'checkout', 'commit', 'export', dan 'rtag'.

'-P'

Memangakas (prune) direktori kosong.

'-p'

Mengalirkan file dari repository ke output standar, bukan menuliskannya ke direktori yang sedang dipakai. Digunakan bersama '-checkout' dan 'update'.

'-R'

Memproses direktori secara rekursif. Digunakan untuk perintah 'annotate', 'checkout', 'commit', 'diff', 'edit', 'editors', 'export', 'rdiff', 'remove', 'rtag', 'status', 'tag', 'unedit', 'update', 'watch' dan 'watchers'.

'-r TAG'

Menggunakan revisi yang ditentukan oleh argumen TAG, bukan menggunakan revisi "head" default. Ada dua tag khusus : 'HEAD' menunjukkan versi terbaru yang ada di repository, dan 'BASE' menunjukkan revisi terakhir yang anda check out ke dalam direktori kerja sekarang. Spesifikasi tag ini bersifat "sticky". Memberikan option global '-q' bersama option perintah '-r' berguna untuk mencegah pesan kesalahan saat file RCS tidak memiliki tag yang diminta.

'-r' digunakan bersama perintah 'checkout', 'commit', 'diff', 'history', 'export', 'rdiff', 'rtag', dan 'update'.

'-w'

Menetapkan nama file yang harus di-filter. Anda dapat menggunakannya secara berulang. Ketetapannya bisa berupa pola nama file dengan tipe yang sama seperti yang dapat anda atur di file '.cvswrappers'. Perintah '-w' digunakan bersama perintah 'import' dan 'update'.

6. admin

* Kebutuhan : repository, direktori kerja

* Perubahan pada : repository

* Sinonim : `rcs`

Pada Unix, jika ada grup bernama `'cvsadmin'`, hanya anggota grup tersebut yang dapat menjalankan `'cvs admin'` (kecuali perintah `'cvs admin -k'` yang bisa dijalankan semua orang). Grup ini harus ada pada server, atau sistem manapun yang menjalankan CVS non-client/server. Agar semua pengguna tidak bisa menjalankan perintah `'cvs admin'`, buatlah sebuah grup yang tidak memiliki user di dalamnya. Pada NT, fitur `'cvsadmin'` tidak ada sehingga semua user bisa menjalankan `'cvs admin'`.

Option untuk admin

Beberapa option mungkin tidak jelas kegunaanya untuk CVS, namun berguna untuk keperluan historis. Beberapa option bahkan bisa membuat CVS tidak bisa berjalan sama sekali sampai anda meng-undo efeknya.

`'-AOLDFILE'`

Mungkin tidak bekerja dengan CVS. Menambahkan (Append) daftar akses `OLDFILE` ke daftar akses pada file RCS.

`'-aLOGINS'`

Mungkin tidak bekerja dengan CVS. Menambahkan (Append) nama-nama login pada daftar `LOGINS` yang dipisahkan tanda koma ke daftar akses pada file RCS.

`'-b[REV]'`

Mengatur branch secara default ke `REV`. Pada CVS, biasanya kita tidak

memanipulasi branch default, untuk menentukan di branch mana kita akan bekerja sebaiknya menggunakan sticky tag. Alasan penggunaan perintah `'cvs admin -b'` adalah untuk mengembalikan ke versi vendor saat menggunakan branch milik vendor. Jangan menggunakan spasi di antara `'-b'` dengan argumennya.

`'-cSTRING'`

Mengatur komentar utama pada `STRING`. Komentar utama (comment leader) tidak digunakan oleh CVS versi sekarang ataupun RCS 5.7.

`'-e[LOGINS]'`

Mungkin tidak bekerja dengan CVS. Menghapus (erase) nama-nama login pada daftar `LOGINS` yang dipisahkan tanda koma dari daftar akses pada file RCS. Jika `LOGINS` tidak digunakan, hapus semua daftar akses yang ada. Jangan menggunakan spasi di antara `'-e'` dengan argumennya.

`'-I'`

Berjalan secara interaktif, meskipun input standarnya bukan sebuah terminal. Tidak bisa dijalankan pada CVS client/server.

`'-i'`

Tidak berguna dalam CVS. Option ini membuat dan menginisialisasi file RCS baru, tanpa menyimpan sebuah revisi. Dengan CVS, penambahan file menggunakan perintah `'cvs add'`.

`'-kSUBST'`

Mengatur keyword pengganti secara default ke SUBST. Memberikan option `'-k'` pada `'cvs update'`, `'cvs export'`, atau `'cvs checkout'` agar default tersebut tidak digunakan.

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

'-l[REV]'

Mengunci (lock) revisi bernomor REV. Jika ada branch, maka akan dilakukan penguncian pada revisi terakhir di branch tersebut. Jika tidak menggunakan REV, maka yang akan dikunci adalah revisi terakhir pada branch default.

'-L'

Mengatur penguncian (locking) menjadi strict. Strict locking berarti pemiliki file RCS tidak dibebaskan dari locking untuk checkin. Pada CVS, strict locking harus digunakan.

'-mREV:MSG'

Mengganti pesan log revisi REV menjadi MSG.

'-NNAME[:[REV]]'

Bekerja seperti '-n', namun tidak mempedulikan pengaturan NAME sebelumnya.

-nNAME[:[REV]]'

Menghubungkan nama simbolik NAME dengan branch dari revisi REV. Namun lebih baik menggunakan 'cvs tag' atau 'cvs rtag'. Hapus nama simbolik jika ':' dan 'REV' tidak ditulis, jika tidak akan keluar pesan kesalahan jika NAME sudah terhubung dengan nomor yang lain. Jika REV simbolik, ini akan diperluas sebelum penghubungan (association) . Sebuah REV yang berisi nomor branch diikuti tanda '.' berarti revisi terbaru pada branch tersebut. Tanda ':' diikuti REV yang kosong berarti revisi terbaru pada branch default, biasanya trunk. Contohnya, 'cvs admin -nNAME:' menghubungkan NAME dengan revisi paling baru dari semua file RCS, sedangkan 'cvs admin -nNAME:$' menghubungkan NAME dengan nomor revisi yang di-ekstrak dari string keyword pada file kerja yang bersangkutan.

'-oRANGE'

Menghapus (outdates) revisi yang ditentukan RANGE.

Gunakan salah satu cara berikut untuk menentukan RANGE :

'REV1::REV2'

Mematikan semua revisi di antara REV1 dengan REV2, sehingga CVS hanya menyimpan perbedaan keduanya. Contoh, setelah '-o 1.3::1.5' bisa mengambil (retrieve) revisi 1.3, revisi 1.5, atau perbedaan yang didapat dari 1.3 ke 1.5, tetapi revisi 1.4 atau perbedaan antara 1.3 dengan 1.4 tidak termasuk. Contoh lain, '-o 1.3::1.4' dan '-o 1.3::1.3' tidak memiliki dampak, karena tidak ada revisi antara keduanya yang akan dihapus.

'::REV'

Mematikan revisi antara awal branch yang mengandung REV dan REV itu sendiri. Branchpoint dan REv dibiarkan ada. Contohnya, '-o ::1.3.2.6' menghapus revisi 1.3.2.1, revisi 1.3.2.5, da semua yang ada diantaranya, namun membiarkan 1.3 dan 1.3.2.6 tetap ada.

'REV::'

Mematikan revisi di antara REV dengan akhir branch yang mengandung REV. Revisi REV dibiarkan ada tetapi revisi head dihapus.

'REV'

Menghapus revisi REV. Contohnya, '-o 1.3' ekuivalen dengan '-o 1.2::1.4'.

'REV1:REV2'

Menghapus revisi dari REV1 ke REV2, di dalam branch yang sama. REV1 dan REV2 beserta revisi yang ada diantaranya tidak akan bisa di-retrieve.

Contoh, perintah `'cvs admin -oR_1_01:R_1_02'`, berarti menghapus sampai dengan revisi `R_1_02` termasuk revisi itu sendiri, beserta tag `R_1_02`. Jika ada file yang belum diubah di antara `R_1_02` dan `R_1_03`, file tersebut akan memiliki nomor revisi yang sama dengan yang diberikan pada tag `R_1_02` dan `R_1_03`. Jadi, `R_1_02` tidak akan bisa di-retrieve dan `R_1_03` juga harus di-restore kembali.

`':REV'`

Menghapus revisi dari awal branch yang mengandung `REV` termasuk menghapus `REV`.

`'REV:'`

Menghapus revisi-revisi dari revisi `REV`, termasuk `REV` itu sendiri, sampai akhir branch yang mengandung `REV`.

Revisi yang akan dihapus tidak boleh memiliki branch ataupun lock. Jika revisi yang akan dihapus ada yang memiliki nama simbolik, dan menggunakan sintaks `':::'`, CVS akan error dan tidak akan menghapus revisi manapun. Jika anda ingin menghapus revisi beserta nama simboliknya, pertama hapus nama simboliknya menggunakan `'cvs tag -d'`, kemudian jalankan `'cvs admin -o'`. Jika ada yang memakai sintaks bukan `':::'`, maka CVS akan menghapus revisi tersebut tetapi tetap membiarkan nama simbolik menunjuk revisi yang tidak ada.

Anda harus memastikan tidak ada orang yang melakukan check out revisi yang anda update karena akan terjadi kesalahan saat dia mencoba commit salinan revisi tersebut.

'-q'

Bekerja secara quiet, tanpa mencetak diagnostik.

'-sSTATE[:REV]'

Dapat digunakan bersama CVS. Mengatur state attribute dari revisi REV ke STATE. Jika REV berupa nomor branch, maka revisi terbaru akan dianggap berada pada branch tersebut. Jika REV tidak terisi, maka revisi terbaru dianggap berada pada default branch. Identifier apapun dapat diterapkan pada STATE. State yang dapat digunakan antara lain 'Exp' (experimental), 'Stab' (stable), dan 'Rel' (released). Secara default, state dari revisi baru akan diatetapkan menjadi 'Exp' saat revisi dibuat. State tersebut dapat dilihat pada output CVS LOG, dan keyword 'Log' dan '$State$'. CVS menggunakan state yang 'dead' untuk keperluannya, untuk mendapatkan file dari atau ke state 'dead' ,gunakan perintah seperti 'cvs remove' dan 'cvs add', bukan menggunakan 'cvs admin -s'.

'-t[FILE]'

Dapat digunakan bersama CVS. Menuliskan teks deskriptif tentang isi dari FILE ke dalam file RCS, dan menghapus teks yang sudah ada sebelumnya. nama path dari FILE tidak boleh diawali '-'. Teks tersebut bisa dilihat pada output dari 'cvs log'. Tidak boleh ada spasi antara '-t' dengan argumennya. Jika FILE dikosongkan, teks didapat dari input standar yang diakhiri end-of-file atau baris yang diakhiri '.' dengan sendirinya.

'-t-STRING'

Sama seperti '-tFILE'. Menuliskan teks deskriptif dari STRING ke dalam file RCS, dan menghapus teks yang sudah ada sebelumnya. Tidak boleh ada spasi antara '-t' dengan argumennya.

'-U'

Mengatur locking menjadi non-strict. Non-stricy locking berarti bahwa pemilik sebuah file harus tidak me-lock sebuah revisi untuk checkin. Pada CVS, strict locking harus digunakan.

'-u[REV]'

Lihat option '-l' di atas, mengenai penggunaan option ini pada CVS. Revisi ini melakukan unlock revisi dengan nomor REV. Jika ada branch, unlock dilakukan pada revisi terakhir pada branch tersebut. Jika REV tidak diisi, maka akan menghapus lock yang dipakai oleh caller-nya. Biasanya hanya orang yang melakukan lock yang boleh meng-unlock sebuah revisi, jika di-unlock oleh orang lain maka lock akan terbuka, dan orang yang mengunci (locker) sebenarnya akan mendapatkan pemberitahuan 'commit'. Tidak boleh ada spasi antara '-u' dengan argumennya.

'-VN'

Pada versi CVS sebelumnya, option ini dimaksudkan untuk menulis (write) sebuah file RCS yang bisa diterima oleh RCS versi N, tapi option itu sekarang sudah kuno dan menggunakannya akan menyebabkan terjadi error.

'-xSUFFIXES'

Pada versi CVS sebelumnya, ini digunakan untuk menentukan nama dari file-file RCS. Namun pada CVS sekarang, option ini tidak digunakan karena file CVS secara otomatis akan memberikan akhiran ',v' pada file tersebut.

7. checkout

* Sinopsis : `checkout [option] modul...`

* Kebutuhan : repository

* Perubahan pada : direktori kerja

* Sinonim : `co`, `get`

Membuat atau meng-update direktori kerja berisi salinan file source sesuai yang ditentukan `MODULES`. Anda harus menjalankan `'checkout'` sebelum menggunakan perintah CVS lainnya, karena kebanyakan perintah tersebut menggunakan direktori kerja.

`MODULES` dapat berupa nama simbolik untuk beberapa direktori atau file source, atau path ke direktori atau file di repository. Nama simbolik ditentukan pada file `'modules'`.

Tergantung pada modul yang anda tentukan, `'checkout'` akan membuat direktori-direktori serta file-file yang sesuai. Kemudian baru anda bisa melakukan pengeditan dan operasi lainnya.

File yang dibuat oleh `'checkout'` adalah read-write, kecuali anda menggunakan option `'-r'`, menggunakan `'CVSREAD'`, atau file tersebut dipengaruhi watch.

7.1. Option untuk checkout

`'-D DATE'`

Menggunakan revisi terbaru yang tidak lebih baru dari `DATE`. Option ini bersifat sticky.

`'-f'`

Hanya berfungsi bersama flag `'-D DATE'` atau `'-r TAG'`. Jika revisi yang

dicari tidak ada, maka digunakan revisi yang paling baru.

'-k KFLAG'

Memproses keyword sesuai dengan KFLAG. Option ini bersifat sticky, berarti bahwa update file ini pada direktori kerjanya akan memakai KFLAG yang sama. Perintah 'status' bisa dipakai untuk melihat option sticky ini.

'-l'

lokal ; hanya berjalan pada direktori kerja yang sedang digunakan.

'-n'

Tidak menjalankan program checkout manapun.

'-P'

Memangkas (prune) direktori-direktori yang kosong.

'-p'

Pipe file-file ke output standar.

'-R'

Checkout direktori secara rekursif. Option ini aktif secara default.

'-r TAG'

Menggunakan revisi TAG. Option ini bersifat sticky.

Selain itu dari itu, sebagai tambahan anda juga dapat menggunakan option perintah khusus berikut :

'-A'

Reset semua sticky tag, tanggal (date), atau option '-k'.

'-c'

Salin (copy) file modul secara terurut ke output standar, dan tidak membuat atau memodifikasi file atau direktori yang ada di direktori kerja anda.



'-d DIR'

Membuat direktori bernama DIR untuk file-file kerja, bukan menggunakan nama modul. Secara umum, menggunakan flag ini sama artinya dengan penggunaan 'mkdir DIR; cd DIR' diikuti perintah checkout tanpa flag '-d'. Flag tersebut membuat CVS dapat memperpendek nama path untuk menghindari adanya direktori yang kosong.

Contoh yang cukup penting adalah, misalnya diberikan sebuah modul 'foo' yang berisi file 'bar.c', perintah 'cvs co -d dir foo' akan membuat direktori 'dir' dan menempatkan 'bar.c' di dalamnya. Contoh lain, jika diberikan modul 'bar' yang memiliki subdirektori 'baz' yang didalamnya terdapat file 'quux.c', perintah 'cvs co -d dir bar/baz' akan membuat direktori 'dir' dan menempatkan file 'quux.c' di dalamnya.

Penggunaan flag '-N' dapat mengalahkan perilaku di atas. Misalnya digunakan definisi modul yang sama seperti di atas, 'cvs co -N -d dir foo' akan menciptakan direktori-direktori 'dir/foo' dan menempatkan 'bar.c' di dalamnya, sedangkan perintah 'cvs co -N -d dir bar/baz' akan menciptakan direktori-direktori 'dir/bar/baz' menempatkan 'quux.c' di dalamnya.

'-j TAG'

Dengan menggunakan dua option '-j', akan dilakukan penggabungan (merge) perubahan dari revisi yang ditentukan oleh '-j' pertama ke revisi yang ditentukan oleh 'j' yang kedua, ke dalam direktori kerja.

Dengan satu option `'-j'`, akan menggabungkan perubahan dari revisi terdahulu (ancestor) ke revisi yang ditentukan oleh `'-j'`, ke dalam direktori kerja. Revisi ancestor adalah ancestor umum untuk revisi yang sedang dikerjakan dalam direktori kerja, dan untuk revisi yang ditentukan oleh `'-j'`. Sebagai tambahan, setiap option `'-j'` bisa berisi spesifikasi tanggal (date) opsional yang jika digunakan bersama branch dapat membatasi jumlah revisi yang dipilih menjadi satu dalam tanggal tertentu. Tanggal opsional tersebut ditentukan dengan menambahkan `':'` pada tag :

```
'-jSYMBOLIC_TAG:DATE_SPECIFIER'
```

`'-N'`

Hanya berguna jika dipakai bersama `'-d DIR'`. Dengan option ini, CVS tidak akan memperpendek path modul di direktori kerja anda saat anda melakukan check out sebuah modul.

`'-s'`

Sama seperti `'-c'`, tetapi memiliki status dari semua modul, dan mengurutkannya lewat status string.

7.2. Contoh checkout

Mengambil salinan modul `'tc'` :

```
$ cvs checkout tc
```

Mengambil salinan modul `'tc'` sebagaimana keadaannya satu hari lalu :

```
$ cvs checkout -D yesterday tc
```

8. commit

* Sinopsis : `commit [-lnrf] [-m 'log message' | -F file] [-r revision] [files...]`

* Kebutuhan : direktori kerja, repository

* Perubahan pada : repository

* Sinonim : `ci`

Gunakan `'commit'` untuk menerapkan perubahan dari file kerja anda ke dalam repository.

Jika anda tidak menentukan file mana yang akan di-commit, maka semua file dalam direktori kerja akan di-commit. `'commit'` akan mengubah isi repository hanya pada file yang telah anda ubah saja.

Jika file yang anda commit memiliki versi lebih lama daripada yang ada di repository, anda akan mendapat pemberitahuan dan proses commit akan dibatalkan. Anda harus menjalankan `'update'` secara manual.

8.1. Option untuk commit

* Option dasar untuk perintah `'commit'` :

`'-l'`

lokal ; hanya bekerja pada direktori kerja lokal.

`'-n'`

Jangan menjalankan program modul apapun.

`'-R'`

Commit direktori secara rekursif. Aktif secara default.

`'-r REVISION'`

Commit ke `REVISION`. `REVISION` harus berupa sebuah branch, atau revisi pada trunk utama yang lebih tinggi dari nomor revisi yang lain. Anda

tidak dapat commit pada revisi tertentu pada sebuah branch.



* Commit juga mendukung penggunaan option berikut ini :

'-F FILE'

Baca pesan log dari FILE, tanpa menggunakan editor.

'-f'

Memaksa CVS untuk commit revisi baru meskipun anda tidak melakukan perubahan pada file manapun. Jika revisi FILE adalah 1.7, maka dua perintah berikut adalah sama :

```
$ cvs commit -f FILE
$ cvs commit -r 1.8 FILE
```

Option '-f' mematikan kemampuan rekursi. Untuk memaksa CVS melakukan commit sebuah revisi baru untuk semua file pada semua subdirektori, anda harus menggunakan '-f -R'

'-m MESSAGE'

Gunakan MESSAGE sebagai pesan log, tanpa menggunakan editor.

8.2. Contoh Commit

8.2.1. Commit pada Sebuah Branch

Anda dapat commit pada sebuah revisi branch (yang memiliki jumlah titik '.' yang sama) menggunakan option '-r'. Untuk menciptakan sebuah revisi branch, gunakan option '-b' pada perintah 'rtag' atau 'tag'.

Kemudian, 'checkout' atau 'update' bisa digunakan untuk mebuat source anda berdasarkan pada branch yang baru dibuat. Dari sini, semua 'commit' pada source kerja ini akan secara otomatis ditambahkan pada revisi branch, sehingga tidak mengganggu main-

line development. Contohnya, jika anda telah menciptakan sebuah patch ke versi 1.2 sebuah produk, meskipun versi 2.0 sudah ada dalam pengembangan, anda dapat melakukan :

```
-- ---- - -r FCSI_2 FCSI_2_Patch product_module
$ cvs checkout -r FCSI_2_Patch product_module
$ cd product_module
[[ lakukan perubahan ]]
$ cvs commit
```

Ini akan bekerja secara otomatis karena '-r' bersifat sticky.

8.2.2. Menciptakan Branch Setelah Melakukan Pengeditan

Misal anda bekerja pada sebuah software, berdasarkan revisi yang anda checkout minggu lalu. Jika orang lain ingin bergabuang mengerjakan software bersama anda tanpa mengganggu main-line development, anda dapat melakukan commit perubahan anda pada branch baru. Kemudian orang lain bisa checkout file pada branch yang anda buat tersebut. rangkaian perintahnya sebagai berikut :

```
[[ sudah ada source yang sudah anda ubah ]]
$ cvs tag -b EXPR1
$ cvs update -r EXPR1
$ cvs commit
```

Perintah 'update' akan membuat option '-r EXPR1' menjadi sticky pada semua file. Semua perubahan yang anda lakukan pada file tidak akan dihapus oleh perintah 'update'. Perintah 'commit' akan secara otomatis commit pada branch yang benar, karena '-r' adalah sticky. Anda juga bisa menjalankan perintah berikut :

```
[[ sudah ada source yang sudah anda ubah ]]
$ cvs tag -b EXPR1
$ cvs commit -r EXPR1
```

Namun, hanya file yang pernah anda ubah yang akan meiliki sticky flag '-r EXPR1'. Jika anda melakukan modifikasi kemudian

commit tanpa memakai flag '-r EXPR1', beberapa file mungkin akan masuk ke trunk utama.

Pada perubahan yang anda lakukan, orang lain akan melakukan :

```
$ cvs checkout -r EXPR1 whatever_module
```

9. diff

* Sinopsis : `diff [-lR] [-k kflag] [format_options] [[-r rev1 | -D date1] [-r rev2 | -D date2]] [files..]`

* Kebutuhan : direktori kerja, repository

* Perubahan pada : tidak ada

Perintah 'diff' digunakan untuk membandingkan perbedaan antara beberapa revisi file. Pembandingan dilakukan pada file kerja anda dengan revisi sumbernya, dan melaporkan setiap perbedaan yang ada.

Jika diberikan nama file, maka hanya file tersebut yang akan dibandingkan. Jika diberikan nama direktori, semua file dalam direktori tersebut akan dibandingkan.

Exit status untuk perintah diff berbeda dengan perintah CVS yang lainnya.

9.1. Option untuk diff

* Option standar yang didukung oleh 'diff' antara lain :

'-D DATE'

Gunakan revisi paling akhir sebelum tanggal DATE.

'-k KFLAG'

Proses keyword sesuai dengan KFLAG

'-l'

lokal, hanya berjalan pada direktori kerja yang sedang dipakai.

'-R'

Periksa direktori secara rekursif

'-r TAG'

Option berikut menentukan format outputnya. Beberapa option memiliki dua bentuk yang ekuivalen yaitu satu huruf yang diawali '-' dan bentuk lain adalah nama panjang yang diawali '--'.

'-LINES'

Tampilkan context sebanyak LINES (integer) baris. Harus digunakan bersama '-c' atau '-u' agar bisa berjalan.

'-a'

Perlakukan semua file sebagai text dan membandingkannya secara line-by-line (perbaris), meskipun file tersebut bukan file text.

'-b'

Abaikan semua spasi kosong.

'-B'

Abaikan perubahan yang hanya memasukkan atau menghapus baris kosong.

'--binary'

Baca dan tulis file dalam format biner.

'--brief'

Hanya laporkan apakah file tersebut berbeda, bukan detail dari perbedaannya.

'-c'

Gunakan format output secara context.

'-C LINES'

'--context[=LINES]'

Gunakan format output secara context, sebanyak LINES baris context, atau tiga baris jika LINES tidak diberikan.

'--changed-group-format=FORMAT'

Gunakan FORMAT untuk menampilkan sebuah line group yang berisi perbedaan barisdari kedua file dalam format if-then-else.

'-d'

Ganti algoritma supaya bisa menemukan lebih ssedikit perbedaan. Ini membuat 'diff' berjalan lebih lambat.

'-e'

'--ed'

Membuat output yang merupakan sebuah script 'ed' yang valid.

'--expand-tabs'

Perluas tab ke spasi, untuk mepertahankan alignment tab pada file input.

'-f'

Membuat output yang nampak seperti sebuah script 'ed' tapi memiliki perubahan sehingga dapat muncul dalam file.

'-F REGEXP'

Dalam format context dan unified,untuk setiap kumpulan perbedaan, tampilkan beberapa baris terakhir yang cocok dengan REGEXP.

'--forward-ed'

Membuat output yang nampak seperti sebuah script 'ed', tapi memiliki perubahan sehingga dapat muncul dalam file.



'-H'

Gunakan cara heuristik untuk mempercepat penanganan file besar yang memiliki banyak perubahan kecil.

'--horizon-lines=LINES'

Jangan membatalkan sejumlah LINES baris terakhir dari prefix umum dan sejumlah LINES baris pertama pada suffix umum.

'-i'

Abaikan perubahan terhadap case, tidak memmbedakan antara penulisan huruf besar dan kecil.

'-I REGEXP'

Abaikan perubahan yang hanya memasukkan atau menghapus baris yang sesuai dengan REGEXP.

'--ifdef=NAME'

Membuat gabungan output if-then-else yang menggunakan NAME.

'--ignore-all-space'

Abaikan spasi kosong saat membandingkan baris.

'--ignore-blank-lines'

Abaikan perubahan yang hanya memasukkan atau menghapus baris kosong.

'--ignore-case'

Abaikan perubahan terhadap case, tidak memmbedakan antara penulisan huruf besar dan kecil.

'--ignore-matching-lines=REGEXP'

Abaikan perubahan yang hanya memasukkan atau menghapus baris

yang sesuai dengan REGEXP.

'--ignore-space-change'

Abaikan spasi kosong saat membandingkan baris.

'--initial-tab'

Tampilkan tab bukan spasi sebelum baris teks pada format normal atau context. Sehingga alignment tab pada baris terlihat normal.

'-L LABEL'

Gunakan LABEL dan bukan menggunakan nama file pada format context dan pada format unified.

'--label=LABEL'

Gunakan LABEL dan bukan menggunakan nama file pada format context dan pada format unified.

'--left-column'

Cetak hanya kolom kiri dari dua baris umum pada format side-by-side.

'--line-format=FORMAT'

Gunakan FORMAT untuk menampilkan semua baris input pada format if-then-else.

'--minimal'

Ganti algoritma supaya bisa menemukan lebih sedikit perbedaan. Ini membuat 'diff' berjalan lebih lambat.

'-n'

Tampilkan diff dalam format RCS-format, sama seperti '-f' kecuali bahwa setiap perintah yang menentukan jumlah baris akan terpengaruh.

'-N'

'--new-file'

Pada pembandingan direktori, jika sebuah file hanya ditemukan pada

satu direktori saja, maka dianggap ada pada direktori lainnya tetapi file tersebut kosong.

`'--new-group-format=FORMAT'`

Gunakan FORMAT untuk menampilkan satu grup baris yang diambil dari file kedua pada format if-then-else.

`'--new-line-format=FORMAT'`

Gunakan FORMAT untuk menampilkan satu baris yang diambil dari file kedua pada format if-then-else.

`'--old-group-format=FORMAT'`

Gunakan FORMAT untuk menampilkan satu grup baris yang diambil dari file pertama pada format if-then-else.

`'--old-line-format=FORMAT'`

Gunakan FORMAT untuk menampilkan satu baris yang diambil dari file pertama pada format if-then-else.

`'-p'`

Tampilkan di function C mana perubahan dilakukan.

`'--rcs'`

Tampilkan diff dalam format RCS-format, sama seperti `'-f'` kecuali bahwa setiap perintah yang menentukan jumlah baris akan terpengaruh.

`'--report-identical-files'`

`'-s'`

Laporkan jika dua file tidak berbeda.

`'--show-c-function'`

Tampilkan di function C mana perubahan dilakukan.

`'--show-function-line=REGEXP'`

Dalam format context dan unified, untuk setiap kumpulan perbedaan,

tampilkan beberapa baris terakhir yang cocok dengan REGEXP.

'--side-by-side'

Gunakan format output side-by-side.

'--speed-large-files'

Gunakan cara heuristik untuk mempercepat penanganan file besar yang memiliki banyak perubahan kecil.

'--suppress-common-files'

Jangan tampilkan baris umum dalam format side-by-side.

'-t'

Perluas tab ke spasi, untuk mepertahankan alignment tab pada file input.

'-T'

Tampilkan tab bukan spasi sebelum baris teks pada format normal atau context. Sehingga alignment tab pada baris terlihat normal.

'--text'

Perlakukan semua file sebagai text dan membandingkannya secara line-by-line (perbaris), meskipun file tersebut bukan file text.

'-u'

Gunakan format output unified.

'--unchanged-group-format=FORMAT'

Gunakan FORMAT untuk menampilkan satu grup baris yang diambil dari kedua buah file dalam format if-then-else.

'--unchanged-line-format=FORMAT'

Gunakan FORMAT untuk menampilkan satu baris yang umum untuk kedua buah file dalam format if-then-else.

'-U LINES'

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

'--unified[=LINES]'

Gunakan format output unified, yang menampilkan sebanyak LINES baris context, atau tiga baris jika LINES tidak diisi. Pada operasi biasa, 'patch' biasanya memrlukan minimal dua baris context.

'-w'

Abaikan spasi kosong saat membandingkan baris-baris.

'-W COLUMNS'

'--width=COLUMNS'

Gunakan output dalam format tampilan side-by-side selebar COLUMN.

'-y'

Gunakan format output side by side.

9.1.1 Line Group Format

Line group format memungkinkan anda untuk menentukan format yang cocok untuk banyak aplikasi yang memungkinkan input if-then-else, termasuk bahasa pemrograman dan bahasa text formatting.

Sebagai contoh, perintah-perintah berikut ini membandingkan file TeX 'myfile' dengan versi aslinya di repository, dan outputnya adalah file yang telah di-merge di mana region yang lama di batasi oleh baris-baris 'em'-'em', dan region baru dibatasi oleh baris-baris 'bf'-'bf'.

```
cvs diff \
   --old-group-format='em
%<em
' \
   --new-group-format='bf
%>bf
' \
   myfile
```

Peintah selanjutnya ekivalen dengan di atas, namun lebih rumit karena membaca line group format defaultnya.

```
    --old-group-format='em
' }
%>bf
    --unchanged-group-format='%=' }
%<em
%>bf
    myfile
```

Berikut adalah contoh lebih lanjut, yang outputnya berupa sebuah diff listing dengan header yang berisi nomor baris dalam gaya "plain english".

```
  --unchanged-group-format='' \i0
%df:
  --new-group-format='-------- %dN line%(N=1?:s) added after
%>' }
at %df:
%>' }
  myfile
```

Untuk menentukan sebuah line group format, gunakan salah satu option di bawah ini. Anda bisa memakai sampai empat line group format, satu untuk setiap masing-masing jenis line group. Anda harus mengutip FORMAT, karena biasanya berisi kumpulan metakarakter shell.

'--old-group-format=FORMAT'

Isinya hanya baris-baris dari file pertama. Ini menetukan old group format.

'--new-group-format=FORMAT'

Isinya hanya baris-baris dari file kedua. Ini menetukan new group format.

'--changed-group-format=FORMAT'

Isinya adalah baris-baris dari kedua file.

'--unchanged-group-format=FORMAT'

Isinya hanya baris-baris yang umum untuk kedua file.

Dalam sebuah line group format, biasanya karakter mewakili dirinya sendiri, spesifikasi konversi diawali dengan '%' dan memiliki salah satu bentuk berikut.

'%<'

Berarti baris-baris dari file pertama, termasuk baris baru yang mengikutinya. Setiap baris diformat sesuai dengan old line format.

'%>'

Berarti baris-baris dari file kedua, termasuk baris baru yang mengikutinya. Setiap baris diformat sesuai dengan new line format.

'%='

Berarti baris-baris yang umum untuk kedua file, termasuk baris baru yang mengikutinya. Setiap baris diformat sesuai dengan unchanged line format.

'%%'

Berarti '%'.

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

`'%c'C''`

di mana `C` adalah karakter tunggal, berarti C. `C` tidak boleh sebuah backslashi atau sebuah apostrophe. Contoh, `'%c':''` berarti sebuah colon (:), meskipun di dalam bagian then pada sebuah format if-then-else, yang biasanya colon akan dihilangkan.

`'%c'''`

di mana O adalah sebuah string berisi 1,2, atau 3 digit oktal, berarti karakter dengan kode oktal O. Contoh, `'%c'''` berarti sebuah karakter null.

`'FN'`

di mana `F` adalah sebuah spesifikasi konversi `'printf'` dan `N` adalah salah satu dari huru-huruf berikut ini, yang berarti value dari `N` dalam format `F`.

`'e'`

nomor baris dari baris tepat sebelum group dalam file lama.

`'f'`

nomor baris dari baris pertama di group di dalam file lama, sama dengan `E+1`.

`'l'`

nomor baris dari baris terakhir di group dalam file lama.

`'m'`

nomor baris dari baris tepat setelah group dalam file lama, sama dengan `L+1`.

`'n'`

jumlah baris di group dalam file lama, sama dengan `L-F+1`

'E, F, L, M, N'

sama seperti keterangan di atas.

Spesifikasi konversi 'printf' dapat berupa'%d', '%o', '%x', atau '%X', menetukan output bentuk desimal, oktal, lower case heksadesimal, atau upper case heksadesimal. Setelah '%' juga mungkin muncul option lain, '-' menandakan left-justification, sebuah integer menetukan lebar minimal dari field, dan sebuah titik (.) diikuti integer opsional menandakan jumlah digit minimum. Contoh, '%5dN' akan menampilkan jumlah baris baru dalam sebuah field dengan lebar 5 karakter, menggunakan format 'printf' yaitu '''%5d"'.

'(A=B?T:E)'

If A sama dengan B then T else E. A dan B bisa berupa konstanta desimal ataupun karakter seperti di atas.

Contoh, '%(N=0?no:%dN) line%(N=1?:s)' adalah ekivalen dengan 'no lines' jika N (jumlah baris di group dalam file baru) adalah 0, ke '1 line' jika N adalah 1, dan ke '%dN lines' jika tidak 1.

9.1.2 Line Format

Line format mengontrol bagaimana setiap baris diambil dari file input adalah output sebagai bagian dari sebuah line group dengan format if-then-else.

Sebagai contoh, perintah berikut menampilkan text dengan sebuah indikator perubahan one-column ke bagian kiri text. Kolom

pertama adalah '-' untuk baris yang di-delete, '|' untuk baris yang di-add, dan spasi untuk baris yang tidak berubah.

```
$ diff \
   --old-line-format='-%l
   --new-line-format='|%l
   --unchanged-line-format=' %l
   myfile
```

Untuk menentukan sebuah line format, gunakan salah satu option berikut. Anda harus mengutip FORMAT karena biasanya berisi metakarakter shell.

'--old-line-format=FORMAT'

format baris sama seperti file pertama.

'--new-line-format=FORMAT'

format baris sama seperti file kedua.

'--unchanged-line-format=FORMAT'

format baris yang umum untuk kedua file

'--line-format=FORMAT'

format semua baris, ini berarti sama dengan menjalankan ketiga option di atas.

Dalam sebuah line format, biasanya karakter mewakili dirinya sendiri, spesifikasi konversi diwala dengan '%' dan memiliki salah satu bentuk berikut.

'%l'

berarti isi dari baris, tidak termasuk baris baru yang mengikutinya. Format ini mengabaikan apakah baris tersebut

tidak sempurna.

'%L'

berarti isi dari baris, termasuk baris baru yang mengikutinya. Jika sebuah baris tidak sempurna, format ini akan menyimpan masalah tersebut.

'%%'

berarti '%'.

'%c'C''

di mana c adalah karakter tunggal, berarti c. c tidak boleh sebuah backslashi atau sebuah apostrophe. Contoh, '%c':'' berarti sebuah colon (:).

'%c'''

di mana O adalah sebuah string berisi 1,2, atau 3 digit oktal, berarti karakter dengan kode oktal O. Contoh, '%c''' berarti sebuah karakter null.

'Fn'

di mana F adalah sebuah spesifikasi konversi 'printf', berarti nomor baris yang berformat F. Contoh, '%.5dn' akan menampilkan nomor baris yang menggunakan 'printf' format '"%.5d"'.

Line format defaultnya adalah '%l' diikuti karakter baris baru.

Jika input berisi karakter tab, anda harus yakin ada '%l' or '%L' dalam sebuah line format tepat setelah sebuah tab, atau anda harus menggunakan option '-t' atau '--expand-tabs'.

Line format dan line group format memungkinkan anda memakai banyak format. Misalnya, perintah berikut menggunakan sebuah format yang sama dengan fomat normal pada `'diff'`.

```
    --old-line-format='< %l

    --new-line-format='> %l

    --old-group-format='%df%(f=l?:,%dl)d%dE

    --new-group-format='%dea%dF%(F=L?:,%dL)

    --changed-group-format='%df%(f=l?:,%dl)c%dF%(F=L?:,%dL)

%>' }

    myfile
```

9.2. Contoh diff

Baris berikut menghasilkan sebuah Unidiff (flag `'-u'`) antara revisi 1.14 dan 1.19 dari file `'backend.c'`. Dengan flag `'-kk'` tidak ada keyword yang disubstitusi, sehingga perbedaan yang hanya bergantung pada keyword substitution akan diabaikan.

```
$ cvs diff -kk -u -r 1.14 -r 1.19 backend.c
```

Misal ada branch `EXPR1` berdasarkan pada sekumpulan file yang di-tag `RELEASE_1_0`. Untuk mengetahui apa yang telah terjadi pada branch tersebut, gunakan :

```
$ cvs diff -r RELEASE_1_0 -r EXPR1
```

Perintah seperti berikut bisa digunakan untuk menghasilkan sebuah context diff antara dua rilis :

```
$ cvs diff -c -r RELEASE_1_0 -r RELEASE_1_1 > diffs
```

Jika anda mempertahankan ChangeLog, perintah berikut ini sebelum anda melakukan commit dapat membantu anda menuliskan entry untuk

ChangeLog. Semua modifikasi lokal yang belum anda commit akan ditampilkan :

```
$ cvs diff -u | less
```

10. export

* Sinopsis : `export [-flNnR] [-r rev|-D date] [-k subst] [-d dir] module...`

* Kebutuhan : repository

* Perubahan pada : direktori yang sedang dipakai.

Perintah ini adalah jenis lain dari `'checkout'`, gunakan saat anda memerlukan sebuah salianan dari source untuk modul tanpa memakai direktori administratif dari CVS.

Menggunakan `'-kv'` pada perintah `'cvs export'` akan menyebakan semua keyword akan di-expand seperti pada perintah import. Option ini tidak bisa menangani export file biner secara benar. Anda juga menjadi tidak dapat lagi menggunakan perintah `'ident'`.

Option untuk export

Berikut ini adalah option standar untuk `'export'` :

`'-D DATE'`

Gunakan revisi terakhir sebelum tanggal `DATE`.

`'-f'`

Jika tidak ditemukan revisi yang cocok, retrieve revisi paling baru.

`'-l'`

Lokal, hanya bekerja di direktori kerja yang sedang digunakan.

`'-n'`

Jangan menjalankan program checkout apapun.



'-R'

Export direktori secara rekursif.

'-r TAG'

Gunakan revisi TAG.

Sebagai tambahan, option berikut juga dapat digunakan (umum untuk 'checkout' dan 'export') :

'-d DIR'

Membuat sebuah direktori benama DIR untuk file kerja, bukan menggunakan nama modulnya.

'-k SUBST'

Menggunakan mode keyword expansion.

'-N'

Hanya berguna jika digunakan bersama '-d DIR'.

11. history

* Sinopsis : `history [-report] [-flags] [-options args] [files...]`

* Kebutuhan : file `$CVSROOT/CVSROOT/history'

* Perubahan pada : tidak ada

CVS bisa menyimpan sebuah file history untuk melacak setiap penggunaan perintah 'checkout', 'commit', 'rtag', 'update', dan 'release'. Gunakan perintah 'history' untuk menampilkan informasi.

Logging harus diaktifkan dengan cara membuat file :

'$CVSROOT/CVSROOT/history'

'history menggunakan '-f', '-l', '-n', dan '-p' saat terjadi conflict dengan penggunaan normal dalam CVS.

Option untuk history

Berikut ini beberapa option (di atas disebut `'-report'`) mengontrol jenis report apa yang akan di-generate :

`'-c'`

Melaporkan (report) setiap kali dilakukan commit (setiap kali repository dimodifikasi).

`'-e'`

Everything (semua tipe rekord). Ekivalen dengan penggunaan `'-x'` dengan semua tipe rekord. Jika anda menulis script yang hanya menangani tipe rekord tertentu, anda akan menggunakan `'-x'`.

`'-m MODULE'`

Report pada modul tertentu. (anda dapat menggunakan `'-m'` lebih dari satu kali pada command line).

`'-o'`

Report pada modul-mudul yang telah di-check out. Ini adalah tipe report secara default.

`'-T'`

Report pada semua tag

`'-x TYPE'`

Ekstrak sekumpulan tipe rekord `TYPE` dari history CVS. Tipennya diindikasikan dengan satu huruf, yang dapat anda kombinasikan.

Perintah-perintah berikut memiliki sebuah tipe rekord :

`'F'`

Release

'O'

checkout

'E'

export

'T'

rtag

Salah satu dari empat tipe rekord berikut dapat merupakan hasil dari sebuah update :

'C'

Sebuah merge telah dibuat tetapi terdeteksi adanya collision (memerlukan merging secara manual)

'G'

Sebuah merge telah dibuat dan sukses.

'U'

Sebuah file kerja telah disalin dari repository

'W'

Salinan kerja dari sebuah file telah di-delete saat proses update (karena hilang dari repository).

Salah satu dari tiga tipe rekord berikut adalah hasil dari commit :

'A'

Sebuah file telah ditambahkan (add) untuk pertama kalinya.

'M'

Sebuah file telah dimodifikasi.

'R'

Sebuah file telah di-remove.

Option yang ditunjukkan di atas sebagai `'-flags'` meng-expand laporan (report) tanpa memerlukan argumen option :

`'-a'`

Tampilkan data ke semua user (secara default hanya menampilkan data pada pengguna yang menjalankan `'history'`)

`'-l'`

Tampilkan hanya modifikasi terakhir.

`'-w'`

Tampilkan hanya rekord untuk modifikasi yang telah dilakukan dari direktori kerja yang sama di mana `'history'` dieksekusi.

Option yang ditunjukkan di atas sebagai `'-options ARGS'` menampilkan report berdasarkan sebuah argumen :

`'-b STR'`

Tampilkan data kembali ke sebuah rekord yang berisi string `STR` dalam bentuk nama modul, nama file, atau repository path.

`'-D DATE'`

Tampilkan data semenjak `DATE`. Di sini berbeda dengan penggunaan `'-D DATE'` yang biasanya untuk menampilkan revisi terbaru yang lebih lama dari `DATE`.

`'-f FILE'`

Tampilkan data untuk file tertentu (anda dapat memberikan banyak option `'-f'` pada satu command line).

`'-n MODULE'`

Tampilkan data untuk modul tertentu (anda dapat memberikan banyak option `'-n'` pada satu command line).

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

`'-p REPOSITORY'`

Tampilkan data untuk source repository tertentu (anda dapat memberikan banyak option `'-p'` pada satu command line).

`'-r REV'`

Tampilkan rekord yang mewakili revisi semenjak revisi atau tag bernama `REV` muncul dalam file RCS. Setiap file RCS akan dicari revisi atau tag-nya.

`'-t TAG'`

Tampilkan rekord semenjak tag `TAG` terakhir di-add ke file history. Ini berbeda dengan flag `'-r'` di atas yang hanya membaca file history saja, bukan file RCS.

`'-u NAME'`

Tampilkan rekord untuk user `NAME`.

`'-z TIMEZONE'`

Tampilkan waktu dalam rekord terpilih menggunakan time zone yang diberikan, bukan menggunakan UTC.

12. import

* Sinopsis : `import [-options] repository vendortag releasetag..`

* Kebutuhan : repository, direktori source distribution

* Perubahan pada : repository

Gunakan `'import'` untuk memasukkan sebuah source distribution keseluruhan dari sebuah source luar (misalnya source vendor) ke dalam direktori repository source anda. Anda dapat menggunakan perintah ini saat pembuatan sebuah repository dan untuk update modul dari source luar.

Argumen REPOSITORY memberikan sebuah nama direktori (atau path yang menunjukkan direktori) di bawah direktori root CVS untuk repository, jika direktori tersebut tidak ada, import akan membuatnya.

Saat anda menggunakan import untuk melakukan update terhadap source yang telah dimodifikasi dalam source repository anda, anda akan diberitahu jika ada file yang conflict dalam dua branch pengembangan, gunakan 'checkout -j' untuk menyelesaikan perbedaannya.

Jika CVS mengabaikan (ignore) sebuah file, maka tidak akan di-import dan akan menampilkan 'I ' diikuti dengan nama file.

Jika terdapat file '$CVSROOT/CVSROOT/cvswrappers', file manapun yang memiliki nama yang cocok dengan spesifikasi dalam file tadi akan dianggap sebagai package dan proses filtering akan dilakukan pada file/direktori sebelum dilakukan import.

Source luar akan disimpan pada sebuah branch level pertama, secara default 1.1.1. Update adalah leaves (daun) dari branch ini. Contoh, file dari koleksi source yang pertama di-import akan menjadi revisi 1.1.1.1, kemudian file dari update yang pertama di-import akan menjadi revisi 1.1.1.2, dan seterusnya.

Setidaknya diperlukan tiga buah argumen. REPOSITORY dibutuhkan untuk mengidentifikasi koleksi source. VENDORTAG adalah sebuah tag untuk seluruh branch (untuk 1.1.1). Anda juga harus memakai setidaknya satu RELEASETAG untuk mengidentifikasi file-file pada leaves yang dibuat setiap kali anda menjalankan 'import'.

Import tidak akan mengubah dirtektori yang mendasarinya. Artinya, direktori tersebut tidak akan dijadikan sebagai sebuah direktori kerja CVS, jika anda ingin mengerjakan source tersebut, anda harus melakukan import terlebih dahulu kemudian check out ke direktori yang lain.

12.1. Option untuk import

Berikut ini adalah option standar yang didukung oleh `'import'` :

`'-m MESSAGE'`

Gunakan `MESSAGE` sebagai informasi log, tidak menggunakan sebuah editor.

Berikut ini adalah option khusus tambahan :

`'-b BRANCH'`

Keterangan tentang option ini lebih detil dibahas di Branch untuk banyak vendor.

`'-k SUBST'`

Menunjukkan mode keyword expansion yang diinginkan. Seting ini akan diterapkan ke seluruh file yang dibuat saat import, tapi tidak terhadap semua file yang telah ada di repository.

`'-I NAME'`

Menentukan nama file yang harus diabaikan saat import. Option ini bisa digunakan berulang-ulang. Untuk menghindari adanya file yang diabaikan, gunakan `'-I !'`

`NAME` dapat berupa pola nama file yang tipenya sama yang dapat anda tentukan dalam file `'.cvsignore'`.

`'-W SPEC'`

Menetukan nama file yang akan disaring saat import. Option ini bisa

digunakan berulang-ulang. `SPEC` dapat berupa pola nama file yang tipenya sama yang dapat anda tentukan dalam file `'.cvsignore'`.

12.2. Output dari import

`'import'` membuat anda tetap mengetahui kemajuannya dengan mencetak satu baris untuk setiap file, diawali satu karakter yang menunjukkan status file tersebut.

`'U FILE'`

File tersebut telah ada di repositoy dan belum dimodifikasi secara lokal, atau sebuah revisi telah dibuat.

`'N FILE'`

File tersebut adalah sebuah file baru yang telah dimasukkan (add) ke dalam repository.

`'C FILE'`

File tersebut telah ada di repositoy dan telah dimodifikasi secara lokal, anda harus merge perubahan tersebut.

`'I FILE'`

File tersebut diabaikan (ignore).

`'L FILE'`

File tersebut adalah sebuah symbolic link, `'cvs import'` mengabaikan symbolic link.

13. log

* Sinopsis : `log [options] [files...]`
* Kebutuhan : repository, direktori kerja
* Perubahan pada : tidak ada

Peintah ini menampilkan informasi log untuk file. `'log'` biasanya digunakan untuk memanggil utility RCS `'rlog'`. History ini menentukan format dari output dan option.

Outputnya meliputi lokasi dari file RCS, revisi "head" (revisi terbaru pada trunk), semua nama simbolik (tag), dan lain-lain. Untuk setiap revisi, nomor revisi, author, nomor baris yang di-add/delete dan log message semuanya akan dicetak (ditampilkan). Semua waktu akan ditampilkan dalam bentuk Coordinated Universal Time (UTC).

Perhatikan bahwa `'log'` menggunakan `'-R'` sehingga dapat menimbulkan conflict dengan penggunaan secar normal dalam CVS.

Option untuk log

Secara default, `'log'` akan mencetak semua informasi yang tersedia. Semua option lainnya digunakan untuk membatasi output tersebut.

`'-b'`

Cetak informasi tentang revisi pada default branch, biasanya branch tertinggi pada trunk.

`'-d DATES'`

Cetak informasi tentang revisi dengan sebuah tanggal/waktu check in pada range yang ditentukan oleh daftar tanggal yang dipisahkan oleh semikolon. Tanggal (date) bisa merupakan gabungan range seperti berikut :

`'D1<D2'`

`'D2>D1'`

Pilih revisi yang telah disimpan antara tanggal `D1` dan `D2`.

'<D'

'D>'

Pilih semua revisi yang tanggalnya lebih D atau yang lebih awal.

'D<'

'>D'

Pilih semua revisi yang tanggalnya lebih D atau yang lebih baru.

'D'

Pilih satu revisi terbaru pada tanggal D atau lebih awal.

Tanda '>' atau '<' dapat diikuti oleh tanda '='.

Perhatikan bahwa pemisahnya adalah semikolon (;).

'-h'

Hanya cetak nama file RCS, nama file yang ada di direktori kerja, head, default branch, daftar akses, lock, nama simbolik, dan suffix.

'-l'

Lokal, hanya berjalan di direktori kerja yang sedang digunakan.

'-N'

Jangan cetak daftar tag untuk file ini.

'-R'

Hanya cetak nama file RCS.

'-rREVISIONS'

Cetak informasi tentang revisi-revisi dalam daftar REVISIONS yang dipisahkan koma dari revisi-revisi atau range. Berikut ini adalah format range yang dapat digunakan :

'REV1:REV2'

Revisi-revisi REV1 sampai REV2 (harus di branch yang sama).



'REV1::REV2'

Sama, tetapi tidak termasuk REV1.

':REV'

'::REV'

Revisi-revisi dari awal branch sampai REV dan termasuk REV.

'REV:'

Revisi-revisi dimulai dari REV sampai ke akhir branch yang di dalamnya terdapat REV.

'REV::'

Revisi-revisi dimulai tep[at setelah REV sampai ke akhir branch yang di dalamnya terdapat REV.

'BRANCH'

Semua revisi yang ada pada branch tersebut.

'BRANCH1:BRANCH2'

'BRANCH1::BRANCH2'

Sebuah range dari branch yang berarti semua revisi pada branch tersebut yang masuk dalam range.

'BRANCH.'

Revisi terbaru pada branch.

'-r' tanpa keterangan revisi berarti revisi terbaru pada default branch, biasanya trunk. Tidak boleh ada spasi antara option dengan argumennya.

'-S'

Lakukan suppress terhadap header jika tidak ada revisi yang dipilih.

'-s STATES'

Cetak informasi tentang revisi yang atribut statenya cocok dengan salah satu state yang ada di daftar STATES yang dipisahkan koma.

'-t'

Mencetak sama seperti '-h', ditambahkan text deskriptif.

'-wLOGINS'

Cetak informasi tentang revisi yang di-check in oleh user yang ada di daftar LOGINS yang dipisahkan koma. Jika LOGINS tidak diisi, maka akan digunakan login user. Tidak boleh ada spasi option '-w' dengan argumennya.

14. rdiff

* rdiff [-flags] [-V vn] [-r t|-D d [-r t2|-D d2]] modules...

* Kebutuhan : repository

* Perubahan pada : tidak ada

* Sinonim : patch

Membangun (build) sebuah file patch(1) berformat Larry Wall antara dua rilis, yang dapat diberikan langsung ke program 'patch' untuk memjadikan rilis lama up-to-date denga rilis yang baru. Output dari diff dikirimkan ke standard output device.

Dengan menggunakan option standar '-r' dan '-D' anda dapat menentukan kombinasi manapun dari satu atau dua revisi atau tanggal (date). Jika hanya satu revisi atau date yang diberikan, file patch akan menggambarkan perbedaan antara revisi atau date tersebut dan head revision yang sedang dipakai dalam file RCS.

Jika rilis software yang terpengaruh terdapat pada banyak direktori, maka perlu digunakan option '-p' pada perintah 'patch' saat melakukan

patching pada source lama, sehingga 'patch' dapat menemukan file yang terletak di direktori yang lain.

14.1. Option untuk rdiff

Berikut ini adalah daftar option standar yang didukung oleh perintah 'rdiff' :

'-D DATE'

Gunakan revisi paling baru sebelum tanggal DATE.

'-f'

Jika tidak ditemukan revisi yang cocok, retrieve revisi paling baru.

'-l'

Lokal, jangan menurunkan subdirektori.

'-R'

Periksa direktori secara rekursif. Option ini aktif secara default.

'-r TAG'

gunakan revisi TAG.

Selain option standar di atas, juga terdapat beberapa option tambahan yang tersedia yaitu :

'-c'

Gunakan format context diff. Ini adalah format default.

'-s'

Buat sebuah report ringkasan perubahan, bukan memakai patch. Ringkasan ini meliputi informasi tentang file-file yang telah diubah atau ditambahkan (add) antara rilis. Ini dikirimkan ke standard output device. Ini berguna untuk mencari, misalnya file mana saja yang telah diubah antara dua date atau revisi.

'-t'

Sebuah diff dari dua revisi teratas dikirimkan ke standard output device. Ini berguna untuk mengetahui perubahan apa yang dilakukan terakhir pada file.

'-u'

Gunakan format unidiff untuk context diff. Ingat bahwa versi lama dari program 'patch' tidak bisa menangani format unidiff, jadi jika anda ingin menyebarkan patch ini ke jaringan/internet sebaiknya jangan gunakan '-u'.

'-V VN'

Expand terhadap keyword berdasarkan aturan pada RCS versi VN (expansion format berubah dengan RCS versi 5). Option ini sudah tidak dapat diterima lagi. CVS akan selalu meng-expand keyword sesuai dengan RCS versi 5.

14.2. Contoh rdiff

Misal anda menerima surat dari foo@example.net yang meminta update dari rilis 1.2 ke 1.4 pada compiler tc. Anda tidak memiliki satupun patch-nya, tapi dengan CVS ini bisa diatasi menggunakan perintah berikut :

```
$ cvs rdiff -c -r FOO1_2 -r FOO1_4 tc | }
$$ Mail -s 'The patches you asked for' foo@example.net
```

Misal anda telah membuat rilis 1.3, dan sebuah branch bernama 'R_1_3fix' untuk perbaikan bug. 'R_1_3_1' berhubungan dengan rilis 1.3.1, yang telah dibuat beberapa waktu yang lalu. Sekarang anda ingin

mengetahui seberapa banyak pengembangan yang telah dilakukan pada branch tersebut. Perintah berikut dapat digunakan :

```
$ cvs patch -s -r R_1_3_1 -r R_1_3fix module-name
cvs rdiff: Diffing module-name
File ChangeLog,v changed from revision 1.52.2.5 to 1.52.2.6
File foo.c,v changed from revision 1.52.2.3 to 1.52.2.4
File bar.h,v changed from revision 1.29.2.1 to 1.2
```

15. release

* `release [-d] directories...`

* Kebutuhan : direktori kerja

* Perubahan pada : direktori kerja, history log.

Perintah ini dimaksudkan untuk secara aman membatalkan efek dari `'cvs checkout'`. Karena CVS tidak mengunci (lock) file, perintah ini tidak akan terlalu dibutuhkan. Anda dapat saja langsung menghapus direktori kerja anda, tapi anda akan kehilangan perubahan yang mungkin anda lupa, dan anda tidak meninggalkan jejak di file history CVS yang anda hentikan.

gunakan perintah `'cvs release'` untuk mengatasi masalah tersebut. Perintah ini memeriksa bahwa tidak ada perubahan yang belum di-commit. memeriksa bahwa mengeksekusinya dari atas sebuah direktori kerja CVS, dan memeriksa bahwa file anda yang terekam di repository sama dengan yang didefinisikan repository dalam database modul.

jika semua itu benar, `'cvs release'` akan meninggalkan sebuah rekord dari eksekusinyan dalam history log CVS.

15.1. Option untuk release

Perintah `'release'` mendukung satu option :

`'-d'`

Hapus (delete) salinan kerja dari file jika release berhasil. Jika flag ini tidak diberikan maka file anda akan tetap ada di direktori kerja anda.

Perintah `'release'` akan menghapus direktori dan file secara rekursif.

15.2. Outpout dari release

Sebelum `'release'` melepaskan source anda akan dicetak satu baris pesan untuk semua file yang tidak up-to-date.

Direktori baru yang anda buat tapi belum dimasukkan ke dalam hirarki repository menggunakan perintah `'add'`, akan diabaikan meskipun terdapat file di dalamnya.

`'U FILE'`

`'P FILE'`

Terdapat sebuah revisi yang lebih baru di repository, dan anda belum memodifikasi salinan file lokal anda.

`'A FILE'`

File telah di-remove dari salinan source anda tapi belum di-commit ke repository. Jika anda delete salinan source anda maka ini akan hilang.

`'R FILE'`

File telah ditambahkan ke salinan pribadi source anda tapi belum di-commit ke repository.

`'M FILE'`

File ini telah dimodifikasi dalam direktori kerja anda. mungkin juga terdapat revisi yang lebih baru di repository.

```
'? FILE'
```

FILE ada dalam direktori kerja anda, tapi tidak terhubung dengan yang lain di source repository, dan tidak terdaftar pada file yang diabaikan CVS. Jika anda remove source kerja anda, file in akan hilang.

15.3. Contoh release

Release direktori 'tc', kemudian delete salinan lokal dari file milik anda :

```
$ cd ..       # You must stand immediately above the
              # sources when you issue `cvs release'.
$ cvs release -d tc
You have [0] altered files in this repository.
Are you sure you want to release (and delete) directory `tc': y
$
```

16. update

```
* update [-ACdflPpR] [-I name] [-j rev [-j rev]] [-k kflag]   [-r
tag|-D
 date] [-W spec] files...
```

* Kebutuhan : repository, direktori kerja

* Perubahan pada : direktori kerja

Setelah anda menjalankan 'checkout', anda dapat menggunakan perintah 'update' dari dalam direktori kerja anda untuk reconcile pekerjaan anda dengan revisi manapun yang diterapkan di source repository sejak update anda terakhir.

16.1. Option untuk update

Berikut ini adalah daftar option standar yang didukung oleh perintah 'update' :

UNIT PELAKSANA TEKNIS PERPUSTAKAAN UNIKOM

'-D date'

Gunakan revisi paling baru yang tidak lebih dari tanggal DATE. Option ini bersifat sticky dan menyatakan '-P'.

'-f'

Hanya berguna bersama flag '-D DATE' atau '-r TAG'. Jika tidak ditemukan revisi yang cocok, retrieve revisi paling baru.

'-k KFLAG'

Memproses keyword berdasarkan KFLAG. Option ini bersifat sticky .

'-l'

Lokal, hanya berjalan di direktori kerja yang sedang dipakai.

'-P'

Prune (pangkas) semua direktori kosong.

'-p'

Pipe file-file ke output standar.

'-R'

Update direktori secara rekursif.

'-r rev'

Retrieve revisi/tag REV. Option ini bersifat sticky .

Option-option khusus berikut ini juga tersedia untuk perintah 'update' :

'-A'

Reset semua sticky tag, date, atau option '-k'.

'-C'

Overwrite file yang telah dimodifikasi secara lokal dengan menggunakan salinan file dari repository. (File yang telah dimodifikasi ini sendiri disimpan dalam '.#FILE.REVISION')

'-d'

Buat semua direktori yang ada di repository jika hilang dari direktori kerja anda.

'-I NAME'

Ignore (abaikan) file yang namanya cocok dengan NAME (dalam direktori kerja anda). Gunakan '-I !' agar tidak ada file yang diabaikan.

'-WSPEC'

Menetukan nama file yang akan disaring (filter) saat update.

SPEC dapat berupa nama file atau pola tipe yang sama yang dapat anda tentukan dalam file '.cvswrappers'.

'-jREVISION'

Dengan menggunakan dua option '-j', merge perubahan dari revisi yang ditentukan oleh '-j' yang pertama ke revisi yang ditentukan oleh 'j' yang kedua, ke dalam direktori kerja anda.

Dengan menggunakan satu option '-j', merge perubahan dari revisi ancestor ke revisi yang anda tentukan dengan '-j', ke dalam direktori kerja anda.

Menggunakan option '-j TAGNAME' bukannya '-j BRANCHNAME' untuk melakukan merge dari sebuah branch sering kali tidak me-remove file yang telah di-remove dari branch tersebut.

Sebagai tambahan, setiap option '-j' dapat meliputi spesifikasi date opsional, yang jika digunakan bersama branch, dapat membatasi pilihan revisi menggunakan tanggal yang ditentukan. Date opsional

tersebut diberikan dengan menambahkan sebuah kolon (:) pada tag:

```
'-jSYMBOLIC_TAG:DATE_SPECIFIER'
```

16.2. Output dari update

Perintah `'update'` dan `'checkout'` menjaga anda agar tetap mengetahui kemajuannya dengan mencetak satu baris pesan untuk setiap file, diawali dengan satu karakter yang menandakan status dari file.

`'U FILE'`

File dijadikan up-to-date sesuai dengan repository, in dilakukan jika file ada di repository tapi tidak ada di source anda, dan untuk file yang belum anda ubah tapi bukan versi paling baru yang tersedia di repository.

`'P FILE'`

Seperti `'U'`, tapi CVS server mengirimkan sebuah patch bukan keseluruhan file. Hasilnya sama saja dengan `'U'` tapi menggunakan lebih sedikit bandwidth.

`'A FILE'`

File telah ditambahkan (add) ke salinan source pribadi anda, dan akan ditambahkan ke repository jika anda menjalankan commit, ini sebagai pengingat bahwa anda harus commit perubahan file anda.

`'R FILE'`

File telah di-remove ke salinan source pribadi anda, dan akan ditambahkan ke repository jika anda menjalankan commit, ini sebagai pengingat bahwa anda harus commit perubahan file anda.

'M FILE'

File dimodifikasi dalam direktori kerja anda.

'M' dapat menggambarkan salah satu dari dua state file yang anda kerjakan yaitu :

- tidak ada modifikasi pada file yang sama di repository,
- ada modifikasi pada file yang sama di repository sebagaimana di salinan anda, tapi berhasil di-merge, tanpa conflict, dalam direktori kerja anda.

CVS akan mencetak beberapa pesan jika merge berhasil, dan akan dibuat cadangan salinan file anda (seperti saat anda belum menjalankan update). Nama file tersebut sebenarnya akan dicetak saat 'update' berjalan.

'C FILE'

Terjadi conflict saat merge perubahan anda ke FILE yang telah berubah dalam repository.

'? FILE'

FILE ada dalam direktori kerja anda, tapi tidak terhubung dengan yang lain di source repository, dan tidak terdaftar pada file yang diabaikan CVS.