**PERTEMUAN-5**

**SCM, Revision Control / Source Code Management, dan Git**

**5.1 Tujuan**

1. Mahasiswa memahami pengertian dari Software Configuration Management.
2. Mahasiswa memahami keterkaitan antara SCM, Revision Control System, ource Code Management, dan Git
3. Mahasiswa memahami dan mampu mengaplikasikan dasar-dasar pemakaian Git untuk developer individual
4. Mahasiswa memahami dan mampu mengaplikasikan dasar-dasar pemakaian Git untuk berkolaborasi antar developer.

**5.2 Teori Singkat**

**Pengantar**

Ada dua kepanjangan dari SCM dan keduanya saling berkaitan. SCM pertama adalah Software Configuration Management. SCM kedua adalah Source Code Management. Untuk menghilangkan kebingungan, Source Code Management sering disebut juga Revision Control. Meskipun demikian, masih sering ditemui istilah di dunia komputasi yang menggunakan SCM untuk menunjukkan Software Configuration Management. Penjelasan ini semata-mata untuk menghilangkan kebingungan jika suatu saat membaca buku teks ataupun artikel yang menggunakan singkatan SCM. Untuk keperluan selanjutnya, SCM akan merujuk pada Software Configuration Management. SCM merupakan aktivitas tracking dan dan pengendalian perubahan pada software. SCM biasanya berisi dua aktivitas utama yaitu Revision Control dan penetapan Baselines. Suatu baseline merupakan suatu produk atau sekumpulan produk yang menjadi dasar perbandingan logis. Revision Control itu sendiri merupakan suatu pengelolaan terhadap berbagai perubahan yang terjadi terhadap berbagai artifak pada proses software development. Dalam software development dengan anggota tim pengembang yang terdiri atas lebih dari satu orang, sering terjadi kasus satu artifak (dokumen, source code, dll) diubah oleh lebih dari satu orang. Revision Control merupakan aktivitas untuk mengelola hal tersebut. Untuk mempermudah melakukan Revision Control, biasanya digunakan suatu software yang termasuk dalam kategori Version Control System.

**Software untuk Version Control System**

Banyak software yang dibuat untuk keperluan VCS ini. Pada dasarnya, VCS bisa dikategorikan ke dalam berbagai kelompok berikut ini:

* Lokal / Tersentralisasi, semua developer harus berada pada sistem komputer yang sama. Contoh software ini antara lain adalah RCS (Remote Control System) dan SCCS (Source Code Control System).
* Terdistribusi (DRCS / Distributed Revision Control System), menggunakan pendekatan Peer-to-Peer. Sinkronisasi dilakukan dengan mempertukarkan patches-set antar peer. Contoh software ini antara lain adalah Aegis, Bazaar, Git, Darcs, GNU Arch, Mercurial, Monotone.
* Client-server, semua developer melakukan sinkronisasi pada suatu server terpusat. Contoh dari software ini antara lain adalah CVS dan Subversion.

**Tentang Git**

Git adalah DRCS yang pada awalnya dirancang dan dibuat oleh Linus Torvalds untuk keperluan pengembangan kernel Linux setelah BitKeeper yang sebelumnya digunakan untuk keperluan pengembangan kernel berubah menjadi software proprietary. Git merupakan software bebas dan bisa diperoleh di http://www.git-scm.org.

**Source Code Hosting dengan Fasilitas Git**

Beberapa vendor menyediakan akses bebas untuk hosting source code, terutama untuk proyek-proyek opensource atau proyek-proyek software bebas. Beberapa diantaranya adalah:

1. GitHub (http://www.github.com)
2. Gitorious (http://www.gitorius.org)
3. GNU Savannah (http://savannah.gnu.org)
4. Project Kenai (http://www.kenai.com)
5. Sourceforge (http://sourceforge.net)

**Proyek-proyek Pengguna Git**

Git digunakan oleh banyak proyek pengembangan software, diantaranya adalah Linux kernel, Arch Linux, Merb, Ruby on Rails, Samba, GNOME, GTK, Fedora, dan lain-lain. KDE saat ini sedang mengevaluasi kemungkinan untuk menggunakan Git. Proyek dibawah KDE yang sudah berimigrasi ke Git adalah Amarok. Migrasi ini kemungkinan akan segera diikuti oleh berbagai proyek KDE lainnya.

**5.3 Praktik**

**Git untuk Individu**

**Untuk praktik ini, copy-kan seluruh file yang telan anda kerjakan pada suatu direktori baru. Kita akan melakukan pengelolaan source code pada posisi di dalam direktori baru tersebut.**

***Penting: Direktori dan file-file yang tertulis pada praktik ini lain dengan yang anda kerjakan. Tidak perlu memperhatikan perbedaan itu, yang perlu anda pahami adalah perintah yang anda kerjakan dan hasil pada komputer Anda.***

**Praktik**

Bagian ini digunakan jika kita hanya berurusan dengan diri kita sendiri. Source code tidak akan dikolaborasikan dengan developer lain. Pertama kita inisialisasi user dengan perintah berikut:

$ git config --global user.name "Bambang Purnomosid D. P."

$ git config --global user.email "bpurnomo@akakom.ac.id"

$

Perintah ini cukup dilakukan sekali saja. Perintah ini akan menyimpan konfigurasi di direktori $HOME/.gitconfig berikut ini:

$ cat /home/bpdp/.gitconfig

[user]

name = Bambang Purnomosid D. P.

email = bpurnomo@akakom.ac.id

$

Setelah itu inisialisasi direktori proyek tempat source code berada:

[bpdp@bpdp-arch bab2]$ git init

Initialized empty Git repository in /home/bpdp/kerjaan/modul-praktikum/sqa/bab2/.git/

[bpdp@bpdp-arch bab2]$ ls -la

total 28

drwxr-xr-x 6 bpdp users 4096 2009-10-09 05:12 .

drwxr-xr-x 9 bpdp users 4096 2009-10-09 05:06 ..

drwxr-xr-x 7 bpdp users 4096 2009-10-09 05:12 .git

-rw-r--r-- 1 bpdp users 1785 2009-10-09 05:07 build.xml

drwxr-xr-x 4 bpdp users 4096 2009-10-09 05:07 checkstyle

drwxr-xr-x 2 bpdp users 4096 2009-10-09 05:07 classes

drwxr-xr-x 3 bpdp users 4096 2009-10-09 05:07 src

[bpdp@bpdp-arch bab2]$

Tambahkan semua source code ke dalam repository:

[bpdp@bpdp-arch bab2]$ git add .

[bpdp@bpdp-arch bab2]$ ls -la .git/

total 44

drwxr-xr-x  7 bpdp users 4096 2009-10-09 05:14 .

drwxr-xr-x  6 bpdp users 4096 2009-10-09 05:14 ..

-rw-r--r--  1 bpdp users   23 2009-10-09 05:14 HEAD

drwxr-xr-x  2 bpdp users 4096 2009-10-09 05:14 branches

-rw-r--r--  1 bpdp users   92 2009-10-09 05:14 config

-rw-r--r--  1 bpdp users   73 2009-10-09 05:14 description

drwxr-xr-x  2 bpdp users 4096 2009-10-09 05:14 hooks

-rw-r--r--  1 bpdp users 1088 2009-10-09 05:14 index

drwxr-xr-x  2 bpdp users 4096 2009-10-09 05:14 info

drwxr-xr-x 15 bpdp users 4096 2009-10-09 05:14 objects

drwxr-xr-x  4 bpdp users 4096 2009-10-09 05:14 refs

[bpdp@bpdp-arch bab2]$

Setelah itu lakukan commit. Commit adalah aktivitas untuk menuliskan atau menggabungkan perubahan-perubahan yang telah dilakukan ke dalam repository. Istilah lain untuk commit adalah checkin atau ci). “Initial commit for project” di bawah ini adalah pesan yang akan dituliskan pada commit ini.

Initial commit for project

# Please enter the commit message for your changes. Lines starting

# with '#' will be ignored, and an empty message aborts the commit.

# On branch master

#

# Initial commit

#

# Changes to be committed:

#   (use "git rm --cached <file>..." to unstage)

#

#       new file:   build.xml

#       new file:   checkstyle/checks/checkstyle\_checks.xml

#       new file:   checkstyle/checks/sun\_checks.xml

#       new file:   checkstyle/lib/antlr.jar

#       new file:   checkstyle/lib/checkstyle-5.0.jar

#       new file:   checkstyle/lib/checkstyle-all-5.0.jar

#       new file:   checkstyle/lib/commons-beanutils-core.jar

#       new file:   checkstyle/lib/commons-cli-1.1.jar

#       new file:   checkstyle/lib/commons-logging.jar

#       new file:   src/vehicles/Bicycle.java

#       new file:   src/vehicles/BicycleMain.java

#

~

~

~

~

~

~

~

~

".git/COMMIT\_EDITMSG" 22 lines, 758 characters

Setelah disimpan dengan menggunakan perintah pada editor yang bersangkutan, hasilnya adalah sebagai berikut:

[bpdp@bpdp-arch bab2]$ git commit

[master (root-commit) a533121] Initial commit for project

 11 files changed, 504 insertions(+), 0 deletions(-)

 create mode 100644 build.xml

 create mode 100644 checkstyle/checks/checkstyle\_checks.xml

 create mode 100644 checkstyle/checks/sun\_checks.xml

 create mode 100644 checkstyle/lib/antlr.jar

 create mode 100644 checkstyle/lib/checkstyle-5.0.jar

 create mode 100644 checkstyle/lib/checkstyle-all-5.0.jar

 create mode 100644 checkstyle/lib/commons-beanutils-core.jar

 create mode 100644 checkstyle/lib/commons-cli-1.1.jar

 create mode 100644 checkstyle/lib/commons-logging.jar

 create mode 100644 src/vehicles/Bicycle.java

 create mode 100644 src/vehicles/BicycleMain.java

[bpdp@bpdp-arch bab2]$

Perintah-perintah berikut ini akan banyak digunakan pada saat kita bekerja dengan source code.

**Perintah “git status” dan “git diff“**

Perintah “git status” digunakan untuk melihat status dari working copy. Jika tidak ada perubahan, hasilnya adalah sebagai berikut:

[bpdp@bpdp-arch bab2]$ git status

# On branch master

nothing to commit (working directory clean)

[bpdp@bpdp-arch bab2]$

Jika kita telah mengedit file kita, kita bisa melihat perubahan yang kita lakukan terhadap source code dengan perintah “git diff”. Berikut adalah hasilnya:

[bpdp@bpdp-arch bab2]$ git status

# On branch master

# Changed but not updated:

#   (use "git add <file>..." to update what will be committed)

#   (use "git checkout -- <file>..." to discard changes in working directory)

#

#    modified:   build.xml

no changes added to commit (use "git add" and/or "git commit -a")

[bpdp@bpdp-arch bab2]$ git diff

diff --git a/build.xml b/build.xml

index 5726443..68907c7 100644

--- a/build.xml

+++ b/build.xml

@@ -1,4 +1,4 @@

-<project name="sample" default="checkstyle" basedir=".">

+<project name="Checkstyle" default="checkstyle" basedir=".">

 <property name="classes.dir" value="classes" />

 <property name="src.dir" value="src" />

(END)

Setelah mengedit dan menyimpan perubahan tersebut, lakukan commit sebagai berikut (anda harus mengisikan commit message):

[bpdp@bpdp-arch bab2]$ git commit build.xml

[master 2b3b11d] Change project name

 1 files changed, 1 insertions(+), 1 deletions(-)

[bpdp@bpdp-arch bab2]$

**Perintah “git add” dan “git rm”**

Jika kita menambahkan file atau direktori ke proyek kita, kita bisa menggunakan “git add”. Berikut ini adalah snapshot dari proses penambahan tersebut:

[bpdp@bpdp-arch bab2]$ git status

# On branch master

# Untracked files:

#   (use "git add <file>..." to include in what will be committed)

#

#    README

nothing added to commit but untracked files present (use "git add" to track)

[bpdp@bpdp-arch bab2]$ git add README

[bpdp@bpdp-arch bab2]$ git status

# On branch master

# Changes to be committed:

#   (use "git reset HEAD <file>..." to unstage)

#

#    new file:   README

#

[bpdp@bpdp-arch bab2]$ git commit

[master 063f06a] Menambahkan file README

 1 files changed, 2 insertions(+), 0 deletions(-)

 create mode 100644 README

[bpdp@bpdp-arch bab2]$ git status

# On branch master

nothing to commit (working directory clean)

[bpdp@bpdp-arch bab2]$

Jika ingin me-remove file dari pelacakan git, gunakan “git rm”

**Perintah “git log”**

Perintah ini digunakan untuk menampilkan catatan-catatan (logs) terhadap apa yang telah dilakukan terhadap source code:

commit 063f06a0c22af5a72e50c1d5eca2d42480d58e33

Author: Bambang Purnomosid D. P <bpurnomo@akakom.ac.id>

Date:   Fri Oct 9 05:33:05 2009 +0700

 Menambahkan file README

commit 2b3b11d560b61c025f3756206fe7121fe1443fd3

Author: Bambang Purnomosid D. P <bpurnomo@akakom.ac.id>

Date:   Fri Oct 9 05:29:23 2009 +0700

 Change project name

commit a533121fadeff09c76569679516155a534e270e8

Author: Bambang Purnomosid D. P <bpurnomo@akakom.ac.id>

Date:   Fri Oct 9 05:16:05 2009 +0700

 Initial commit for project

(END)

Jika memerlukan lebih rinci lagi, bisa menggunakan “git log –stat –summary”.

**Git untuk Kolaborasi**

Kolaborasi akan melibatkan lebih dari satu developer. Pada dasarnya ada beberapa perintah yang sering digunakan pada saat berkolaborasi.

**Perintah “git clone“**

Perintah ini digunakan untuk mengambil repository dari. Contoh jika akan mengambil dari remote host berikut ini adalah sebagai berikut:

[bpdp@bpdp-arch scm]$ git clone git://github.com/progit/progit.git

Initialized empty Git repository in /home/bpdp/buku/soft-dev/scm/progit/.git/

remote: Counting objects: 3652, done.

remote: Compressing objects: 100% (1160/1160), done.

remote: Total 3652 (delta 1895), reused 3511 (delta 1795)

Receiving objects: 100% (3652/3652), 5.49 MiB | 20 KiB/s, done.

Resolving deltas: 100% (1895/1895), done.

[bpdp@bpdp-arch scm]$

Hasilnya adalah sebagai berikut:

[bpdp@bpdp-arch scm]$ ls -la progit/

total 76

drwxr-xr-x 17 bpdp users 4096 2009-10-09 04:14 .

drwxr-xr-x  3 bpdp users 4096 2009-10-09 04:09 ..

drwxr-xr-x  8 bpdp users 4096 2009-10-09 04:14 .git

-rw-r--r--  1 bpdp users   97 2009-10-09 04:14 .gitignore

-rw-r--r--  1 bpdp users  821 2009-10-09 04:14 README

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 cs

drwxr-xr-x  9 bpdp users 4096 2009-10-09 04:14 de

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 en

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 es-es

drwxr-xr-x  3 bpdp users 4096 2009-10-09 04:14 es-mx

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 es-ni

drwxr-xr-x  2 bpdp users 4096 2009-10-09 04:14 figures

drwxr-xr-x  2 bpdp users 4096 2009-10-09 04:14 figures-source

drwxr-xr-x 10 bpdp users 4096 2009-10-09 04:14 ja

drwxr-xr-x  2 bpdp users 4096 2009-10-09 04:14 latex

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 nl

drwxr-xr-x 11 bpdp users 4096 2009-10-09 04:14 pt-br

drwxr-xr-x  7 bpdp users 4096 2009-10-09 04:14 ru

drwxr-xr-x  9 bpdp users 4096 2009-10-09 04:14 zh

[bpdp@bpdp-arch scm]$

Perintah git clone juga bisa dilakukan pada host yang sama seperti berikut ini:

SOFTWARE QUALITY ASSURANCE

[root@bpdp-arch ~]# git clone /home/bpdp/kerjaan/modul-praktikum/sqa/bab2

Initialized empty Git repository in /root/bab2/.git/

[root@bpdp-arch ~]# ls -la bab2/

total 28

drwxr-xr-x  5 root root 4096 2009-10-09 05:57 .

drwxr-x--- 16 root root 4096 2009-10-09 05:57 ..

drwxr-xr-x  8 root root 4096 2009-10-09 05:57 .git

-rw-r--r--  1 root root  123 2009-10-09 05:57 README

-rw-r--r--  1 root root 1789 2009-10-09 05:57 build.xml

drwxr-xr-x  4 root root 4096 2009-10-09 05:57 checkstyle

drwxr-xr-x  3 root root 4096 2009-10-09 05:57 src

[root@bpdp-arch ~]#

Perintah git clone juga bisa dilakukan menggunakan protokol ssh berikut ini:

[root@bpdp-arch ~]# git clone bpdp@192.168.1.248:/home/bpdp/kerjaan/modul-praktikum/sqa/bab2

Initialized empty Git repository in /root/bab2/.git/

The authenticity of host '192.168.1.248 (192.168.1.248)' can't be established.

RSA key fingerprint is d4:ce:34:11:64:61:a3:fc:07:6d:51:96:8b:6e:cd:ea.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added '192.168.1.248' (RSA) to the list of known hosts.

bpdp@192.168.1.248's password:

remote: Counting objects: 24, done.

remote: Compressing objects: 100% (23/23), done.

remote: Total 24 (delta 2), reused 0 (delta 0)

Receiving objects: 100% (24/24), 2.17 MiB, done.

Resolving deltas: 100% (2/2), done.

[root@bpdp-arch ~]#

Jika dua orang developer bekerja sama pada source code yang sama, maka bisa digunakan perintah “git pull” untuk mengintegrasikan perubahan-perubahan yang telah di-commit oleh masing-masing developer tersebut. Perintah “git pull” juga bisa menggunakan local copy, melalui ssh, atau protokol lain.

**5.4 Latihan**

Cobalah melakukan clone terhadap hasil pekerjaan teman anda dengan menggunakan protokol SSH. Letakkan hasilnya di suatu direktori tertentu yang lain dari direktori yang telah anda kerjakan.

**5.5 Tugas**

a. Buatlah proyek baru dan ulangi lagi menggunakan langkah-langkah praktik untuk developer individual di atas. Pahami fungsi dari masing-masing langkah tersebut dengan baik.

b. Jelaskan fungsi “git init”, *warning* yang muncul saat pelaksanaan praktikum, “git clone” !