# **VCS dan GIT**

Dibuat oleh: Erwin (Nindya 8G – IT Development Manager)

Sabtu, 7 Mei 2016

VCS adalah kepanjangan dari Version Control System.

VCS merupakan sebuah sistem yang menyimpan perubahan pada satu atau lebih file pada waktu tertentu yang nantinya kita bisa sebut dengan suatu versi.

VCS inilah yang mengatur software berada pada versi apa (atau berapa) setelah kita membuat satu atau lebih fitur software.

Contoh:

- Windows 7, Windows 8, Win 10

- Game versi Beta lalu akhirnya rilis secara official

- update "patch" game Dota 2 yang terus menerus, seperti patch 6.77 dan 6.78b.

- Nindyamaya v 1.0 dan v 2.0

- Binusmaya v.3 u/ kemanggisan, v.5 u/ alsut

Contoh-contoh system VCS:

* Git (github.com, Gitlab.com, Bitbucket.org, dll)
* SVN (Subversion)
* Mercurial
* Dan lainnya (browsing sendiri hehe :P )

Istilah-istilah yang sering digunakan pada VCS:

* Repository: suatu folder projek
* Clone: meng-copy folder project
* Branch: Cabang projek yg nantinya akan digabungkan menjadi hasil akhir projek.
* Branch master: Cabang utama yg merupakan hasil akhir projek.
* Commit: membuat checkpoint project
* Push: mengirim data/file-file project ke server
* Pull: mengambil data dari server
* Conflict: Adanya tabrakan pada perubahan file project secara keseluruhan, biasa terjadi saat pull/push.

Berikut adalah sebagian *command-command* yang ada pada git. No 1-6 wajib dikuasai. No 7 dst command support tambahan, biasanya hanya untuk Project Manager.

1. git clone [link repository git]

Fungsi: untuk meng-copy repository

Penggunaan:

git clone <https://github.com/ITDevelopmentBSLC/latihan_git_bslc.git>

1. git init

Fungsi: untuk meng-inisialisasi suatu folder kosong untuk dijadikan repository git

Penggunaan:

git init

1. git add [namaFile]

Fungsi: untuk menambahkan suatufile yang sudah dibuat/diubah sebelum dimasukkan pada *checkpoint commit.*

Penggunaan:

Untuk menambahkan **semua** file, gunakan

git add .

Untuk menambahkan 1 file saja gunakan

git add namafile.ext

1. git commit –m “isi pesan”

Fungsi: untuk menyimpan perubahan repository menjadi suatu checkpoint.

“isi pesan” disarankan merupakan pesan yang jelas mengenai perubahan apa yang telah dibuat.

Sebelum melakukan commit, pastikan tidak ada error pada repository.

Penggunaan:

git commit –m “fitur login cavis done”

1. git push origin namaBranch

Fungsi: mengirimkan hasil perubahan pada projek ke server. Push hanya bisa dilakukan setelah commit / tidak ada perubahan pada projek.

Penggunaan:

git push origin master //push projek ke master

1. git pull origin namaBranch

Fungsi: meng-update projek dari server ke folder projek di tempat kita. Sama seperti Push, Pull hanya bisa dilakukan setelah commit / tidak ada perubahan pada projek.

Penggunaan:

git pull origin master //update projek kita dari branch master

1. git checkout namaBranchTujuan

Fungsi: Pindah menuju branch tujuan. Hanya bisa dilakukan apabila tidak ada perubahan yg perlu di-commit pada branch awal.

Penggunaan:

git checkout branch-pertama

1. git merge namaBranch

Fungsi: Menggabungkan 2 branch.

Penggunaan:

//asumsi kita berada pada branch master, mau menggabungkan branch-pertama dengan master

git merge branch-pertama

1. git status

Fungsi: menampilkan status keadaan projek sekarang dengan commit terakhir.

Penggunaan

git status

1. git log

Fungsi: menampilkan log atau catatan mengenai commit apa saja yang sudah dilakukkan.

Penggunaan

git log