**Perintah dasar GIT**

* **git config**

Salah satu perintah git yang paling banyak digunakan adalah **git config**, yang bisa digunakan untuk mengatur konfigurasi tertentu sesuai keinginan pengguna, seperti email, algoritma untuk diff, username, format file, dll. Contohnya, perintah berikut bisa digunakan untuk mengatur email:

git config --global user.email [sam@google.com](mailto:sam@google.com)

* **git config --global user.name "[nama]"**

Mengatur nama yang ingin ditautkan pada transaksi commit Anda

* **git init**

Perintah ini digunakan untuk membuat repositori baru. Caranya:

git init

* **git add**

Perintah git add bisa digunakan untuk menambahkan file ke index. Contohnya, perintah berikut ii akan menambahkan file bernama temp.txt yang ada di direktori lokal ke index:

git add temp.txt

* **git clone**

Perintah git clone digunakan untuk checkout repositori. Jia repositori berada di remove server, gunakan:

git clone alex@93.188.160.58:/path/to/repository

Jika salinan repositori lokal ingin dibuat, gunakan:

git clone /path/to/repository

* **git commit**

Perintah git commit digunakan untuk melakukan commit pada perubahan ke head. Ingat bahwa perubahan apapun yang di-commit tidak akan langsung ke remote repository. Gunakan:

git commit –m “Isi dengan keterangan untuk commit”

* **git branch -d [nama-cabang]**

Hapus cabang tertentu

* **git status**

Perintah git status menampilkan daftar file yang berubah bersama dengan file yang ingin di tambahkan atau di-commit. Gunakan:

git status

* **git push**

git push adalah perintah git dasar lainnya. Push akan mengirimkan perubahan ke master branch dari remote repository yang berhubungan dengan direktori kerja Anda. Misalnya:

git push origin master

* **git checkout**

Perintah git checkout bisa digunakan untuk membuat branch atau untuk berpindah diantaranya. Misalnya, perintah berikut ini akan membuat branch baru dan berpindah ke dalamnya:

command git checkout -b <nama-branch>

Untuk berpindah dari branch satu ke lainnya, gunakan:

git checkout <branch-name>

* **git remote**

Perintah git remote akan membuat user terhubung ke remote repository. Perintah berikut ini akan menampilkan repository yang sedang dikonfigurasi:

git rmote -v

Perintah ini membuat user bisa menghubungkan repository lokal ke remote server:

git remote add origin <93.188.160.58>

* **git remote rm *remote-name***

Memutus koneksi repo lokal dari CodeCommit repositori, dengan menggunakan nama julukan yang ditentukan yang dimiliki oleh repo lokal untuk CodeCommit repositori

* **git branch**

Perintah git branch bisa digunakan untuk me-list, membuat atau menghapus branch. Untuk menampilkan semua branch yang ada di repository, gunakan:

git branch

Untuk menghapus branch:

git branch -d <branch-name>

* **git pull**

Untuk menggabungkan semua perubahan yang ada di remote repository ke direktori lokal, gunakan perintah pull:

git pull

* **git merge**

Perintah merge digunakan untuk menggabungkan sebuah branch ke branch aktif. Gunakan:

git merge <nama-branch>

* **git diff**

Perintah git diff digunakan untuk menampilkan conflicts. Untuk melihat conflicts dengan file dasar, gunakan:

**git diff --base <nama-file>**

Perintah berikut digunakan untuk menampilkan conflicts diantara branch yang akan di-merge:

**git diff <source-branch> <target-branch>**

Untuk menampilkan semau conflict yang ada, gunakan:

git diff

**git diff --staged**

Menunjukkan perbedaan berkas hasil revisi dengan versi berkas terakhir yang terdaftar

* **git tag**

Tagging digunakan untuk menandai commits tertentu. Contohnya:

git tag 1.1.0 <insert-commitID-here>

* **git log**

Dengan menjalankan peritah ini akan menampilkan daftar commits yang ada di branch beserta detail-nya. Contoh outputnya adalah:

* **git log --follow [berkas]**

Daftar riwayat revisi untuk sebuah berkas, termasuk pergantian namanya

* commit 15f4b6c44b3c8344caasdac9e4be13246e21sadw

Author: Alex Hunter <alexh@gmail.com>

Date: Mon Oct 1 12:56:29 2016 -0600

* **git reset**

Untuk me-reset index dan bekerja dengan kondisi commit paling baru, gunakan perintah git reset:

git reset --hard HEAD

* **git reset --hard [commit]**

Membuang semua riwayat dan perubahan sampai di titik yang ditentukan oleh commit

* **git rm**

Gunakan perintah ini untuk menghapus file dari index dan direktori kerja. Contohnya:

git rm filename.txt

* + **git rm --cached [berkas]**

Menghapus berkas dari riwayat revisi dengan tetap mempertahankan berkas local

* + **git mv [berkas-asli] [berkas-baru]**

Mengganti nama berkas dan mempersiapkan berkas untuk pendaftaran commit

* **git stash**

Mungkin inilah salah satu perintah dasar git yang jarang digunakan orang, yang bisa membantu menyimpan perubahan yang tidak langsung di-commit, namun hanya sementara. Contoh:

git stash

* **git stash pop**

Mengembalikan berkas yang paling baru disimpan

* **git stash list**

Daftar semua koleksi perubahan yang tersimpan

* **git stash drop**

Membuang koleksi perubahan yang paling baru disimpan

* **git show**

Untuk menampilkan informasi tentang object pada git, gunakan git show:

git show

* **git fetch**

Perintah ini digunakan untuk menampilkan semua object dari remote repository yang tidak berada di direktori kerja lokal. Contohnya:

git fetch origin

* **git ls-tree**

Untuk menampilkan susunan object berdasarkan nama dan mode setiap item, dan nilai blob SHA-1, gunakan perintah git ls-tree. Contohnya:

git ls-tree HEAD

* **git cat-file**

Menggunakan nilai SHA-1, menampilkan tipe object dengan menggunakan perintah git cat-file. Contohnya:

git cat-file –p d670460b4b4aece5915caf5c68d12f560a9fe3e4

* **git prep**

git prep mengizinkan pengguna mencari frase dan/atau kata yang berada di dalam direktori. Contohnya, untuk mencari www.hostinger.co.id di dalam semua file, gunakan:

git grep [www.hostinger.co.id](http://www.hostinger.co.id)

* **gitk**  
  gitk adalah tampilan grafis dari repository lokal yang bisa dipanggil dengan menjalankan perintah:

gitk

* **git instaweb**

Dengan perintah git instaweb, web server bisa dijalan berdampingan dengan repository lokal. Nantinya web browser akan langsung diarahkan ke server tersebut. Contohnya:

git instaweb –httpd=webrick

* **git gc**

Untuk mengoptimasi repository melalui garbage collection, yang akan membersihkan file yang tidak dibutuhkan dan mengoptimasinya, gunakan:

git gc

* **git archive**

Perintah git archive memungkinkan user membuat file zip atau tar yang mengandung susunan repository. Contohnya:

git archive --format=tar master

* **git prune**

Melalui perintah git prune, object yang tidak memiliki incoming pointers akan dihapus. Gunakan:

git prune

* **git fsck**

Untuk membuat pengecekan keseluruhan dari file system git, gunakan perintah git fsck. Object yang corrupt akan dikenali:

git fsck

* **git rebase**

Perintah ini digunakan untuk menerapkan ulang commit di branch yang lain. Contohnya:

git rebase master