 **Git Cheat Sheet**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Ikhtisar**  Ketika pertama kali setup Git, yaitu mengatur username dan alamat email sehingga commit pertama anda merekamnya dengan benar.  git config --global user.name “Nama Lengkap”  git config --global user.mail “pengguna@email.com”  **Tentang Git**  Git adalah free dan open source, sistem pengontrol revisi terdistribusi yang dirancang untuk menangani proyek kecil sampai sangat besar dengan menekankan pada kecepatan dan efisiensi | | | **Baranch dan Merge**  Bekerja dengan Git percabangan dan stash | | |
| git branch | List cabang anda, tanda \* akan muncul di sebelah cabang yang aktif. | |
| git branch [nama-baranch] | Membuat cabang baru pada saat commit. | |
| git checkout [branch] | Beralih ke cabang lain, memeriksanya ke working direktory anda | |
| git checkout –b [branch] | Membuat cabang baru dan langsung beralih ke cabang tersebut | |
| git merge [branch] | Menggabung cabaing lain menjadi satu ke cabang yang sedang aktif dan merekam merge sebagai commit. | |
| **Contoh Alur Kerja Dasar Git**  Untuk mengawali git repositori baru, kemudian stage semua file ke dalam direktori dan akhirnya commit snapshot awal.  $ git init  $ git add .  $ git commit –m ‘commit awal’  Membuat cabang baru bernama FeatureA, kemudian melakukan checkout untuk mengaktifkan cabang, kemudian lakukan perubahan dan stage beberapa file dan terakhir commit snapshot baru.  $ git branch featureA  $ git checkout featureA  $ (edit files)  $ git add (files)  $ git commit –m ‘menambahkan feature A’  Beralih kembali ke branch master, mengembalikan perubahan featureA yang baru saja dibuat, kemudian mengedit beberapa file dan commit perubahan baru anda secara langsung dalam kontek cabang master.  $ git checkout master  $ (edit files)  $ git commit –a –m ‘perubahan files’  Menggabungkan perubahan featureA kedalam kontek cabang master, menggabung semua pekerjaan anda, terakhir hapus cabang featureA.  $ git merge featureA  $ git branch –d featureA | | | git log | Menampilkan log commit. | |
| git stash | Menyimpan di sisi lain modifikasi saat ini tanpa commit dalam working directory sementara. | |
| git stash apply | Mendaftar ulang perubahan yang distash terakhir. | |
| **Share dan Update**  Mengambil, menggabung dan bekerja dengan update ke repositori lain. | | |
| git remote add [alias] [url] | | Menambahkan git URL sebagai alias |
| git fetch | | Mengambil semua cabang dari git remote |
| git merge [alias]/[branch] | | menggabungkan cabang pada server ke dalam cabang Anda yang sedang aktif untuk medapatkan pembaharuan |
| git push [alias] [branch] | | push pekerjaan di cabang Anda untuk memperbarui cabang di remote git repositori. |
| git pull | | Mengambil dari URL yang terlacak oleh cabang saat ini dengan segera mencoba untuk menggabung dari cabang yang terlacak. |
| **Setup dan Init**  Git konigurasi, inisialisasi dan cloning Git | | |
| **Inspect dan compare**  Meneliti log, diffs dan informasi objek. | | |
| git config [key] [value] | | Menetapkan nilai config di repositori |
| git config --global [key] [value] | | Menetapkan nilai konfigurasi global untuk user |
| git log | | menunjukkan sejarah commit untuk cabang yang sedang aktif |
| git init | | Menginisialisasi direktori yang ada sebagai repositori git |
| git log branchB..branchA | | menunjukkan komit pada branchA yang tidak pada branchB |
| git clone [url] | | Clone repositori git dari URL |
| git help [command] | | mendapatkan bantuan di setiap perintah git | git log --follow [file] | | menunjukkan commit dari perubahan file, bahkan mengganti nama. |
| **Stage dan Snapshot**  Bekerja dengan snapshot dan stage area git | | |
| Git status | menunjukkan status apa yang di stage untuk selanjutnya melakukan commit dan apa yang diubah dalam direktori kerja Anda. | | git diff branchB...branchA | | menunjukkan diff dari apa yang ada di branchA yang tidak branchB |
| git add [file] | menambahkan file seperti yang terlihat sekarang untuk berikutnya di commit (stage) | |
| Git reset [file] | reset staging area untuk sebuah file sehingga perubahan tidak dalam commit Anda berikutnya (unstage) | | Git show [SHA] | | menunjukkan objek dalam Git dalam format yang dapat terbaca oleh manusia. |
| git diff | diff dari apa yang berubah tapi tidak pada stage | |
| git diff --staged | diff dari apa yang distaged tetapi belum dicommit | |
| git commit | commit konten stage Anda sebagai commit snapshot baru | | gitx | | tcl Program / ​​tk untuk menampilkan log commit dalam GUI. |
| git rm [file] | Menghapus file dari working directory anda dan unstage. | |
| git gui | tcl / tk program GUI untuk membuat semua perintah sederhana | |

|  |  |
| --- | --- |
| **M. Saiful Mukharom – Komunitas CAH UNP Kediri** | **http://git-scm.com** |