



Git Cheat Sheet

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