

PERINTAH YANG SERING DIGUNAKAN DI GIT!

Git adalah sebuah sistem version control yang memungkinkan anda untuk melacak perubahan pada file atau proyek anda, sehingga anda dapat bekerja secara kolaboratif dengan orang lain dan mengelola versi dari proyek anda. berikut adalah beberapa perintah dasar pada git dan fungsi yang mereka lakukan:

1. **git init:** untuk membuat repository git baru pada direktori/folder kerja anda. perintah ini akan membuat direktori .git di dalam direktori kerja anda, yang akan berisi semua data git yang diperlukan untuk melacak versi proyek anda.
2. **git add:** untuk menambahkan file yang sudah diubah atau file baru ke dalam index git, sehingga git dapat melacak perubahan pada file tersebut.
3. **git rm --cached (nama file):** untuk mengeluarkan kembali file yang berada pada staging index.
4. **git commit:** untuk menyimpan perubahan yang ada pada file ke dalam repository git. setiap commit akan memiliki pesan yang berisi deskripsi singkat tentang perubahan yang telah dilakukan pada file.
5. **git status:** untuk menampilkan status dari file yang ada dalam direktori kerja anda, termasuk file-file yang sudah di-add ke dalam index git, file-file yang sudah di-commit, file-file yang berubah, dan file-file yang belum di-add ke dalam index git.
6. **git log:** untuk menampilkan riwayat commit dari repository git. perintah ini akan menampilkan daftar semua commit yang telah dilakukan, beserta pesan commit dan detail lainnya.
7. **git branch (namabranchnya):** untuk membuat, menampilkan, atau menghapus branch pada repository git. branch adalah fitur pada git yang memungkinkan anda untuk bekerja pada bagian dari proyek tanpa mempengaruhi bagian yang lain.
8. **git merge (namabranchnya):** untuk menggabungkan dua branch yang berbeda ke dalam satu branch. dengan perintah ini, perubahan pada branch yang satu dapat digabungkan ke dalam branch yang lain, sehingga proyek menjadi lebih terintegrasi.
9. **git clone:** untuk membuat salinan dari repository git yang sudah ada pada mesin lain/komputer. dengan perintah ini, anda dapat mengunduh proyek dari mesin lain dan mulai bekerja pada proyek tersebut.
10. **git push:** untuk mengunggah perubahan yang telah dilakukan ke repository git yang terhubung ke mesin lain/akun github, sehingga perubahan tersebut dapat dilihat dan diakses oleh anggota tim yang lain.

11. **git pull**: untuk mengambil perubahan terbaru yang telah dilakukan pada repository git yang terhubung ke mesin lain, sehingga proyek anda selalu diperbarui dengan perubahan terbaru.
12. **git branch -d (nama branch)** : untuk mendelete branch yang sudah tidak digunakan/dipakai lagi
13. **git branch -D (nama branch)**: untuk mendelete branch yg sudah/ belum di merge. dalam kata lain tidak peduli sudah di merge atau belum
14. **git branch --merged**: untuk mengetahui bahwa branch tersebut sudah dimerged
15. **git checkout (nama branch)**: untuk berpindah branch
16. **alias graph="git log --all --decorate --oneline --graph"**: untuk membuat alias atau menambahkan graph secara visual
17. **git remote -v** : digunakan untuk menampilkan daftar repo yang terhubung dengan repo local saat ini beserta URL-nya.
18. **git remote add (namaremove) (URL)** : perintah ini untuk menambahkan remote repository baru kedalam repository local kita
19. **git remote -u** : untuk mengirim perubahan pada repository lokal ke repository jarak jauh (remote repository) pada branch yang ditentukan.
20. **git pull --rebase (namaremove) (namabranh)** : adalah perintah git pull yang digunakan untuk mengambil perubahan pada repository jarak jauh (remote repository) dan melakukan penggabungan (merge) pada branch lokal Anda dengan melakukan rebase pada perubahan Anda terhadap perubahan terakhir di repository jarak jauh.
21. **git mv nama_file nama_folder/** : untuk memindahkan beberapa file kedalam folder yang sudah dibuat.
22. **touch (nama_file)** : untuk membuat file baru pada git
23. **git rm (nama_file/folder)** : untuk menghapus file/folder
24. **git rm -r folder/** : untuk menghapus folder beserta direktorinya
25. **git rm --cached (file.txt)** : menghapus file.txt dari Git tetapi tidak dari direktori kerja
26. **git rm -r --cached (folder)**: menghapus folder beserta isinya dari Git tetapi tidak dari direktori kerja
27. **git clean -f (file)**: untuk menghapus file yg masih untracked secara paksa
28. **git clean -f -d (folder)**: untuk menghapus folder dan isi didalamnya yg masih untracked secara paksa
29. **git clean -n**: untuk menampilkan daftar file/folder yang akan di hapus tanpa benar-benar menghapusnya

30. **git clean -i**: untuk menghapus folder secara interaktif
31. **git clean -df**: untuk menghapus file/folder yang masih untracked beserta isinya
32. **git commit --amend -m (pesan)** : untuk merubah/membatalkan commit yang sudah di commit dan kita bisa memperbaiki commitnya (isi pesannya).
33. **git log --oneline** : untuk melihat commit dalam format yang ringkas.
34. **gitk** : untuk melihat commit log secara visual.
35. **git reset HEAD <nama_file>**: untuk menarik / membatalkan / sudah di add ke dalam staging index
36. **git restore --staged <nama_file>** untuk menarik / membatalkan / sudah di add ke dalam staging index
37. **git add /d/Project/file.txt** : untuk menambahkan file diluar direktori saat ini dan yang masih untracked. Contoh seperti diatar /d adalah merupakan direktori d. /Project adalah sebuah folder. Dan /file.txt merupakan sebuah filenya yang masih untracked.
38. **git branch -m <nama-branch-lama> <nama-branch-baru>** : untuk mengganti nama branch.
39. **git push origin -u <nama-branch-baru>**: untuk mengunggah perubahan nama branch baru agar up to date local maupun remote.
40. **git remote remove <nama_remote> / git remote rm <nama_remote>**: perintah ini untuk menghapus remote pada git
41. **git config --list**: untuk menampilkan daftar konfigurasi Git yang terkait dengan repositori Git yang sedang aktif atau konfigurasi global pada komputer Anda.
42. **git config --local user.name "helloworld"**: untuk merubah konfigurasi username/akun hanya untuk satu folder git
43. **git config --local user.email "helloworld@gmail.com"**: untuk merubah konfigurasi email/akun hanya untuk satu folder git
44. **git config --global user.name "helloworld"**: untuk merubah konfigurasi username/akun untuk semua folder git
45. **git config --global user.email "helloworld@gmail.com"**: untuk merubah konfigurasi email/akun untuk semua folder git
46. **git log -3/n**: untuk menampilkan commit terakhir (contohnya 3)
47. **ssh-keygen**: untuk membuat ssh key
48. **Git stash**: untuk memungkinkan anda menyimpan perubahan kerja sementara tanpa harus melakukan commit.

49. **Git stash list:** untuk menampilkan daftar semua stash yang telah dibuat.
50. **Git stash pop:** untuk menerapkan perubahan dari stash yang paling terbaru ke direktori kerja Anda dan menghapus stash tersebut dari daftar stash.
51. **git stash save "pesan deskriptif":** untuk menambahkan pesan deskriptif untuk stash Anda, sehingga lebih mudah diidentifikasi di kemudian hari.
52. **git stash apply:** untuk menerapkan perubahan dari stash yang paling terbaru ke direktori kerja Anda.
53. **git stash apply stash@{index}:** untuk menerapkan stash tertentu yang ditentukan oleh indeks (index).
54. **git stash drop stash@{index}:** untuk menerapkan menghapus stash tertentu dari daftar stash.
55. **git stash clear:** untuk menghapus semua stash yang ada.
56. **git config --global --list:** untuk melihat konfigurasi global.