

GIT & GITHUB

▪ alias	2
▪ pwd.....	2
▪ ls.....	2
▪ cd.....	2
▪ mkdir <nama_folder>	2
▪ git init	2
▪ git add <file(s)>	2
▪ git rm –cached <file>	2
▪ git status.....	2
▪ git commit -m “<pesan>”	2
▪ git config.....	2
▪ git log	2
▪ git checkout	2
▪ git branch.....	3
▪ git merge <branch_tujuan>	3
▪ git help.....	3
▪ git clone <link>	3
▪ git remote –v.....	3
▪ git push	3
▪ git fetch.....	3
▪ git pull	3
▪ tahapan repo biasa.....	4
▪ tahapan mengambil repo dari github (di local tidak, github ada)	4
▪ tahapan menyimpan repo ke github (di local ada, github tidak)	4
▪ tahapan menyimpan repo konflik ke github (di local ada, github ada)	4
▪ multiple remote.....	4
▪ melakukan pull request.....	4

- **alias**
membuat snippets : alias <nama>="<kode>"
contoh :
 - pembuatan : alias graph="git log --all --decorate --oneline --graph"
 - pemanggilan : \$ graph
- **pwd**
cek lokasi direktori
- **ls**
cek ada folder apa saja
- **cd**
pindah direktori
- **mkdir <nama_folder>**
membuat folder
- **git init**
Menginisialisasi repo git di komputer
- **git add <file(s)>**
untuk menambahkan file ke staging area
- **git rm --cached <file>**
menghapus file dari staging area
- **git status**
untuk mengetahui status repo
- **git commit -m "<pesan>"**
save perubahan
 - git commit -m "<pesan>" : save perubahan
 - git commit -am "<pesan>" : save perubahan tanpa git add
jika ada repo berstatus modified (tidak berlaku jika ada repo baru)
- **git config**
untuk konfigurasi
 - git config --global user.name "<username>"
 - git config --global user.email <email>
- **git log**
melihat riwayat commit
 - git log -3 : menampilkan sesuai jumlah yg diinginkan (3)
 - git log -- style.css : menampilkan sesuai nama folder yg diinginkan (style.css)
 - git log --all --decorate --oneline --graph : tampilan seperti github
- **git checkout**
mengembalikan perubahan file
 - git checkout <kodeTerdepan> : mengembalikan file ke file yg dipilih
 - git checkout <nama_branch> : pindah branch

- `git checkout <7kodeTerdepan>` : pindah ke commit yg dituju
- **git branch**
memeriksa sedang di branch mana.
 - `git branch <nama_branch>` : membuat branch
 - `git branch -d <nama_branch>` : menghapus branch
 - `git branch -D <nama_branch>` : menghapus branch yg blm di merge
 - `git branch --merged` : untuk melihat branch mana yg sudah di merge
- **git merge <branch_tujuan>**
menggabungkan branch
- **git help**
membuka bantuan
- **git clone <link>**
untuk menyalin repo github ke local
- **git remote -v**
periksa daftar remote
- **git push**
mengirim/menyimpan repo dari local ke github
- **git fetch**
memeriksa, repo yg ada di remote (github) sudah sejauh mana commitnya
- **git pull**
menyalin repo dari github ke local

■ tahapan repo biasa

- git init
- git add
- git commit -m "pesan"
- git commit -am "pesan" : jika ada repo berstatus modified dan belum di add

■ tahapan mengambil repo dari github (di local tidak, github ada)

- git clone <link> : untuk menyalin repo dari github ke local
 - git push : mengirim/menyimpan repo dari local ke github
- catatan : sebelum push, add dan commit terlebih dahulu

■ tahapan menyimpan repo ke github (di local ada, github tidak)

- git init
 - git remote add origin <link> : menambahkan remote
 - git add .
 - git commit -m <pesan>
 - git push -u origin master : push pertama kali
 - git push : push kedua dan seterusnya
- catatan : sebelum push, add dan commit terlebih dahulu

■ tahapan menyimpan repo konflik ke github (di local ada, github ada)

contoh kasus : A menambah style.css, B juga menambah style.css

- git fetch
- git graph
- git pull : menyalin repo dari github ke local

■ multiple remote

- fork repo terlebih dahulu di github
- git clone <link> : untuk menyalin repo dari github ke local
- git remote add <link> : menambah remote pemilik repo
- git fetch <remote_pemilik> : menambahkan commit pemilik repo

■ melakukan pull request

Mengirim Pull Request (Aktor: Pengirim Pull)

- git checkout -b <nama_branch> : membuat branch dan langsung pindah ke branch tsb
- melakukan perubahan file (ngoding)
- add + commit perubahan yg dibuat
- git push origin <nama_branch_dibuat> : push repo dari local ke github
- pull request di web github (compare & pull request)

Meng up to date repo sesuai repo yg menerima pull request (Aktor: Penerima Pull)

- git fetch <nama_remote>
- git graph (opsional) : memeriksa status local, github, dan pemilik
- git branch (opsional) : memeriksa saat ini berada di branch mana
- git checkout master : memindahkan branch ke master
- git merge <remote_pemilik>/master : menyalin repo pemilik ke local
- git push origin master : push repo local ke github

- `git branch -d <nama_branch_dibuat>` : menghapus branch local
- `git push origin --delete <nama_branch_dibuat>` : menghapus branch github