

# #1 Git dan Github

## A Git

- J Secara sederhana, git merupakan sesuatu yang disebut dengan version control system (VCS)
- J VCS adalah version control system atau disebut juga source code management, artinya sistem yang mengelola perubahan dari sebuah dokumen, program komputer, website dan kumpulan informasi lain

## -J kenapa menggunakan VCS

- a. dengan menggunakan VCS, kita tidak perlu mengelola versi dari file kita secara manual karena pengelolaannya dilakukan cukup baik menggunakan VCS
- b. proses penggabungan atau kerja sama disebut dengan proses kolaborasi jika tanpa VCS, hal ini akan cukup merepotkan karena harus manual
- c. bisa melacak versi atau history dari perubahan yang terjadi pada software kita

## -J Apa bedanya dengan aplikasi seperti dropbox & gdrive

- a. tidak seflexibel ketika menggunakan VCS yang khusus untuk source code karena aplikasi gdrive dll didesain untuk mengerjakan untuk mengerjakan file yang bukan source code

-) Version control system

- a. sebuah sistem yang menyimpan 'rekam / snapshot' perubahan pada source code
- b. memungkinkan bekerja berkolaborasi dengan lebih baik
- c. mengetahui siapa yang melakukan dan kapan sebuah perubahan terjadi
- d. memungkinkan kita untuk kembali ke keadaan sebelum perubahan (checkout)

VCS ada :

- git

- mercurial

- subversion

- CVS

-) Git

Git adalah sebuah VCS terdistribusi untuk menge-  
lola perubahan file dalam folder. Sebuah software  
untuk mengelola perubahan file di dalam folder.

a. Foldernya disebut (repository / repo)

b. commit = riwayat perubahan file disimpan menggunakan serangkalan commit

Gambaran

Tanpa git

☒ revisi 1

☒ revisi 2

☒ revisi 3

☒ revisi 4

☒ revisi 5

⇒ nama filenya banyak

Dengan git

commit

- memulai mengejakan babi
- tambah latar belakang
- memperbaiki analisis
- hapus teori
- ☒ skripsi

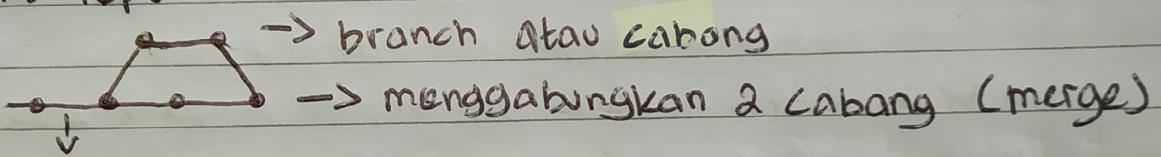
⇒ menyimpan setiap perubahan.

-> Isi commit

- angka panjang = hash (penanda setiap commit)
- yang melakukan edit
- kapan dilakukan commit
- pesan

jadi dengan git lebih terorganisir

-> Git Repo



ada sebuah repo yang sudah dilakukan commit  
bisa melakukan cabang agar tidak mengganggu  
jahr commit utama

B. Github

-> adalah layanan cloud untuk menyimpan & mengelola project / repo

-> di github juga bisa membuat repo, melakukan commit, membuat branch, dan merge

-> bedanya ini dilakukan secara online di cloud lewat Internet

-> atau istilahnya github adalah sebuah website yang di dalamnya menggunakan git

-> Push : mengirimkan source code / project kita ke github

-> Pull : mengambil source code / project , atau mengambil repo yang ada di github ke komputer kita

yang di push & pull adalah commit

syarat :

a. membuat github menjadi remote : sumber dari repo

-> Layanan Selain github

a. bitbucket

b. gitlab

-> check out : berpindah ke sebuah commit

-> clone : mengambil repo dari remote

## #2 Bekerja dengan Github

### A Membuat akun github

- a. Kunjungi website "github.com"
- b. sign up
- c. verifikasi email
- d. login

### B Membuat repository

- a. Klik "start a project"
- b. Klik (on (t) di pojok kanan atas. Lalu pilih "new repository"
- c. Membuat file text sederhana, isi kolom repository name dengan "lpu resolusi"
- d. tulis deskripsi
- e. Pilih private / public
- f. Ceklis yang dibutuhkan
- g. Klik Create repository

### C Membuat satu file

- a. pilih "create new file"
- b. isi nama file
- c. Klik enter
- d. edit file
- e. tulis pesan commit (setiap perubahan sebaiknya kasih tau yang berubah itu apa)
- f. ada pilihan :
  - ] commit langsung ke master-branch
  - ] bikin branch baru
- g. Klik "commit new file"
- h. File akan tersimpan di repository

D Mengedit File

- a. Klik File yang akan diedit
- b. Klik icon "Edit" di sebelah kanan
- c. Edit File
- d. Commit Lagi & beri pesan perubahan

 E Jika kita ingin melihat perubahan

- a. Klik File
- b. Klik history

## #3 Branch

### A Branching

- a. membuat git branch
- b. membuat snapshot tanpa menganggu jalur utama (master-branch)
- c. Fitur experimental
- d. 2 orang memgerakkan repo yang sama

### B Membuat file baru sebagai branch baru

Cara #1 = - edit

- pilih branch default atau baru

# 2 = - klik "branch: master"

- cari branch atau buat baru branch

a. create "new file" untuk membuat file baru

b. tulis pesan commit

c. ketik "commit new file"

Jadi kita memiliki 2 branch

a. branch master

b. branch yang baru dibuat tadi

Jika kita bukan master-branch, file yang dibuat tadi tidak ada. (di jalur utama tidak ada konten) Ketika berpindah branch di git, ini disebut sebagai checkout

C Merging (menggabungkan 2 branch)

Cara : a. ketik "compare & pull request"

b. akan diarahkan ke halaman "open pull request"

c. isi pesan branch ke master

d. klik "create pull request"

e. confirm merge

\* Menghapus branch

a. masuk ke branches

b. pilih branch yang dihapus

c. branch sisa satu (master-branch)

D Menggabungkan 3 branch

a. buat branch lalu baru dari master terlebih dahulu

b. cara menggabungkannya harus satu-satu, mau menggabungkan master dengan yang mana dahulu pilih branch 1 atau 2

c. pilih branch 1, lalu klik "compare & pull request"

d. klik "create pull request"

e. cek lagi, kalau sudah klik "merge pull request"

f. Pilih branch 2. klik "compare & pull request".

terdapat conflict. Pull request nya masih bisa dilakukan, tapi tidak bisa langsung di merge

g. klik "pull request", ada conflict. Jadi sebagian pemilik master itu yang harus me resolve / menyelesaikan masalahnya, diselesaikan secara manual

h. klik "resolve conflict", nanti akan diarahkan ke code editor

i. pilih tetep mau ada struktur datanya / tidak

j. hapus yang tidak dunginkan

k. jika sudah, klik "mark as resolved".

L. klik "commit merge"

M. jika sudah tidak ada conflict, klik "merge pull request"

⇒ Branch = jalur development bebas dari sebuan commit

⇒ Checkout = berpindah ke branch / commit yang lain

⇒ Pull request = meminta pemilik repo untuk mengambil perubahan yang telah dilakukan

⇒ Merge = menggabungkan 2 branch

⇒ Merge conflict = baris yang sama diubah oleh 2 branch yang berbeda

## #4 Fork

### A Fork / Forking

- a. Membuat 'copy / duplikat' dari repo orang lain (beserta history-nya)
- b. jembatan antara repo original dan duplikatnya
- c. modifikasi terhadap repo original
- d. berkontribusi pada repo orang lain
- e. Fork != close

### Cara kerja Fork ddi 2

# 1 => Langsung edit filenya, klik edit, saat diarahkan ke tombol edit, tulisannya menjadi "Fork this project and edit the file"

# 2 => Langsung klik tombol Fork yang ada di kanan atas

- a. klik tombol Fork
- b. edit reponya
- c. Create new file
- d. isi file
- e. commit ke master branch