# **GIT AND GITHUB**

Git dan github adalah sesuatu yang berbeda. Git dan github bias dijalankan secara terpisah/bersamaan.

VSC : Version control system, disebut juga revision control system atau source code management adalah system yang mengelola perubahan dari sebuah document, program computer, website dan kumpulan informasi lain.

Git : Git adalah Version Control System (VCS) terdistribusi/software untuk mengelola perubahan file di dalam folder.git bias dilakukan secara local/ofline

Commit: Git akan menyimpan setiap rangkaian perubahan / history perubahan file yang kita kelola menggunakan commit

Repository / repo : adalah sebuah folder yang menyimpan file yang akan kita kelola

Branch:Memungkinkan kita melakukan perubahan pada git secara bercabang / tanpa harus mempengaruhi file utamanya

Merge: ialah cara / proses penggabungan 2 branch menjadi satu

Checkout : memungkinkan kita untuk kembali ke keadaan sebelum perubahan

GitHub: Adalah Sebuah website yang berfungsi untuk layanan cloud untuk menyimpan dan mengelola project / repo git, github dilakukan secara online (Github adalah sebuah website yang menggunakan git)

Jika Git dan github digabungkan : maka memungkinkan kita mengcloud/ meupload repo dilaptop kita kedalam github, dan memungkinkan kita mengambil repo yang berada di github menggunakan git yang terinstal di laptop kita. Syaratnya kita harus menjadikan github sebagai remote / sumber dari reponya lalu kemudian repo tersebut kita clone kedalam computer kita.

Push : adalah proses mengirimkan file/commit dari computer yang kita miliki ke github.

Pull: adalah proses menerima / mengambil file/commit yang berada di github melalui computer

Pull Request : meminta pemilik repo untuk mengabil perubahan yang telah dilakukan

Fork : memungkinkan kita menduplikat repo milik orang lain

# **GIT**

PENULISAN REPO TIDAK BOLEH MENGGUNAKAN SPASI

**GIT INIT = Inisialisasi repository**

Perintah untuk mengisi nama pada config :git config –global user.name “"

Perintah untuk mengisi email pada config :git config –global user.email “"

Perintah untuk mencek ada apa saja di dalam folder : ls

Melakukan pengelolaan di device menggunakan git

Untuk melakukan pengelolaan program dengan git langkah pertama yang harus dilakukan ialah menjadikan folder yang kita kelola menjadi repo/repository jika folder program yg kita kelola sudah dijadikan folder repo maka folder tersebut disebut (area working tree).

Perintah Untuk menjadikan polder sebagai folder repo : **git init**

**GIT COMMIT = MEMBUAT HISTORI PERUBAHAN PADA REPO**

**Untuk melakukan commit pada git (menghistory perubahan reponya) kita harus memasukan perubahan file pada reponya ke staging area, staging area adalah langkah yang harus dilakukan sebelum kita melakukan commit pada repository yg kita miliki pada git.**

**Perintah Untuk mencek perubahan pada repo : git status**

**Perintah untuk memindahkan perubahan repo ke staging area: git add <file> (ini untuk perintah jika file yg akan kita commit Cuma 1) / git add . (ini untuk perintah memindahkan semua file ke staging area)**

Jika perubahan sdh dimasukan ke dalam staging area maka perubahan tersebut dapat kita buat commitnya

Perintah untuk commit pada git : git commit –m “pesan/ deskripsi commit”

Checkout memungkinkan kita kembali ke commit sebelum kita melakukan perubahan

Perintah untuk mencek semua commit : git log

Perintah untuk mencek commit pada file tertentu : git log -- <file>

Perintah checkout pada git : git checkout “5 digit awal git hashnya” (checkout atau mengembalikan semua file ke keadaan komit sebelumnya).

git checkout “5 digit awal git hashnya” <filenya> (checkout atau mengembalikan satu file ke keadaan komit sebelumnya).

**GIT BRANCH = membuat jalur/cabang tanpa mengubah branch masternya**

Perintah untuk mencek config : git config –list

Perintah untuk mencek branch apa saja yg tersedia di repo : git branch

Perintah untuk membuat branch baru : git branch <file>

Perintah untuk beralih ke branch lain : git checkout <branch>

Perintah untuk meliat graph pada repo : git log –online –decorate –graph

Perintah untuk membuat alias pada code : alias <nama alias>=”<code yg ingin dialias>”

Perintah untuk mengahpus branch yg sudah dimerge : git branch –d <nama branch>

Perintah untuk mengahpus branch yg blm di merge : git branch –D <nama branch>

**GIT MERGE = menggabungkan branch lain ke branch master**

Git marge ada dua :

1. Merge Fast Forward = ketika branch yg ingin di marge berjalur langsung dengan branch master
2. Merge Three Way = ketika branch yg ingin digabungkan tidak berjalur langsung, jadi dihubungkan menggunakan commit baru

Perintah untuk melakukan merge:pergi ke branch master lalu git merge <branch lain yg ingin dimerge>

Perintah untuk mengetahui branch mana yang sudah di merge : git branch --merged

**GIT MERGE CONFLICK = terjadi karena 2 branch mengubah baris yang sama pada satu repo**

**Detec head = kondisi dimana kita kembali ke komit milik branch yang sudah terhapus, untuk menormalkan head ke branch yg masih ada maka kita harus membuat branch baru lalu kita checkout ke branch barunya**

**GIT REMOTE =**

Default untuk nama remote adalah origin

Perintah untuk mencek nama remote = git remote /-v

Perintah untuk mengclone repo yg berada pada github = git cone <link>

Perintah untuk mengirimkan perubahan repo local ke github / mengirim kommit local keg it hub : git push

Perintah untuk mencek perubahan pada repo (memastikan repo pda github sama dengan repo local kita): git fetch

Perintah untuk mensamakan keadaan repo dengan remote : git pull

Untuk menghindari conflik saat mem push langkah pertama

Lakukan git status

Lakukan git fetch

Lakukan pull (jika repo sudah tidak sama dengan repo yang ada di remote)

Lakukan kommit setelah melakukan pull

**Cara git remote dengan kasus remotenya dari git hub:**

1.buat repository github

2. lakukan clone https ke dalam git (git clone <link>)

**Cara git remote dengan kasus remotnya dari repo local (sudah ada dilocal tp blm ada di repo github) :**

1. Buat repo dengan nama yang sama dengan nama folder repo localnya lalu jangan centangkan menambahkan readme (repo githhubnya harus kosong)
2. Perintah selanjutnya ada di github

**GITHUB PAGES = memungkinkan web static yang berada di reponya bisa ditampilkan seperti halnya kita menyimpan file di hosting hanya saja berlaku untuk website static/tanpa server contoh (html,css)**

**Cara agar dapat github pages:**

1.beri nama repo kita username.git.io pastikan untuk menampilkan halaman utamanya dengan file index

2. pergi kesetting cari github pages