# GitHub

**GitHub adalah platform berbasis web yang digunakan untuk pengelolaan kode sumber dan kolaborasi dalam pengembangan perangkat lunak. GitHub memungkinkan pengembang untuk menyimpan proyek mereka, berbagi kode, dan berkolaborasi dengan tim atau komunitas di seluruh dunia.**

# Penggunaan GitHub

1. LOGIN

- kita masuk ke browser dan ketik github di kolom pencarian lalu klik github dan masuk ke menu sign up pada pojok kanan atas.
- 2. Menambah Repository
- Klik pada tanda + diatas lalu pilih new repository, lalu untuk repository name kita kasih nama sesuai keinginan masing-masing, dan jangan lupa pilih opsi public supaya bisa diakses sama teman atau tim, dan tambah centang pada Add a README file, supaya sudah ada file yang di buat langsung oleh git nya, lalu scroll ke bawah pilih creat repository untuk menambahkan repository nya.
- 3. Menambahkan file pada Repository
- Klik tanda + pilih creat new file, lalu untuk name your file kita isi sesuai keinginan masing-masing, lalu edit file yang mau di edit, lalu pilih commit changes, untuk Commit message kita bisa ketik “ menambahkan file tes.txt” untuk descripsi bebas mau nambahnya atau tidak, kalau sudah kita tekan commit changes.
-

# GitHub Branch/cabang

- **Branching**
- - Membuat Git Baranch
- - Membuat snapshot tanpa mengganggu jalur utama (Master Branch)
- - Fitu exferemental
- - 2 orang yang mengerjakan repo yang sama

# GitHub Fork/Forking

- **Fork**
- 1. Membuat dan mengcopy/duplikat dari Repo orang lain (beserta history nya).
- 2. Jembatan antara Repo original dan duplikat nya.
- 3. Mempermudahkan modifikasi terhadap Repo original nya.
- 4. Berkontribusi pada Repo orang lain.

# Git

- Git adalah sistem kontrol versi (version control system) yang digunakan untuk melacak perubahan pada file dan mengelola proyek pengembangan perangkat lunak, terutama yang melibatkan kolaborasi antar pengembang. Git diciptakan oleh Linus Torvalds pada tahun 2005 untuk mendukung pengembangan kernel Linux.
- Fitur Utama Git:
- 1. Distributed Version Control System (DVCS)
- 2. Melacak Perubahan
- 3. Branching dan Merging
- 4. Kecepatan dan Efisiensi
- 5. Kolaborasi
-

# Instalasi Git pada (linux)

- 1. Masuk ke browser kita arahkan ke url (https://git-scm.com)
- Lihat kesebelah kanan yang ada gambar monitor komputer, dan monitor komputer tersebut akan otomatis mendeteksi perangkat yang kita gunakan baik itu Linux, Windows, MAC. Lalu klik monitor tersebut dan ikut langkah nya sebagai berikut.
- -$ sudo apt-get install git
- Untuk mengecek versi berapa git anda dan sistem operasi apa yang anda pakai yang anda download, bisa cek di terminal anda dengan mengetik perintah
- -$ git –version
- Dan untuk meninisialisasi Repo Git di komputer kita bisa mengetik perintah
- $ git init
- Dan untuk menambahkan file ke suatu area yang di sebut dengan Staging Area bisa menggunakan perintah
- $ git add nama_file
-

# Perintah-Perintah Git pada(Linux)

- Dan untuk mengecek file apakah sudah di tracked atau belum bisa ketik perintah .
- $ git status
- Untuk file berwarna merah berarti belum di tracked, supaya file nya bisa di commit harus disimpan dulu ke Staging Area. Untuk memasukkan atau meng commit file yang masih berwarna merah bisa ketik perintah
- $ git add nama_file
- Atau yang lebih simple untuk meng commit nya bisa ketik perintah
- $ git add .
- Untuk ‘git add . ‘ Semua file yang berubah mau itu nambah, hilang, atau di edit semua akan pindah ke Staging Area.
- untuk membuat commit di Git, yaitu menyimpan snapshot dari perubahan yang ada di staging area ke repositori lokal. Dengan menggunakan perintah : $ git commit -m "Menambahkan fitur login pada aplikasi"
- File sudah berada dalam tahapan Staging Area, dalam Staging Area supaya bisa di commit kita harus menambah nama,dan email(bebas), bisa menggunakan perintah:
- $ git config –global user.name “ Ilmi Yanur”
- untuk email $ git config –global user.email “ilmiyanur@mendek.com”

# Perintah Git pada(Linux)

Apabila sudah di tahapan Staging Area, lanjut ke tahapan berikut nya yaitu tahapan History / melihat hasil dari perubahan edit file, bisa mengguanakan perintah :

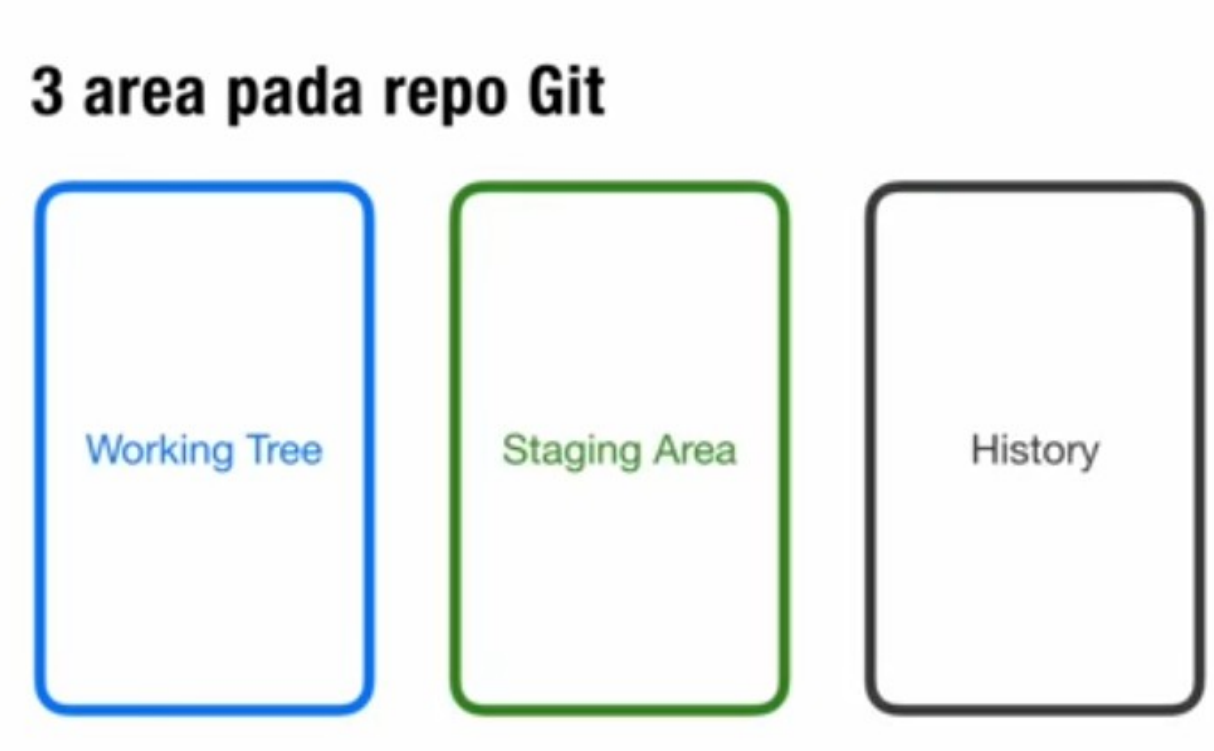
- $ git log
- Atau mau nampilkan sedikit dengan 3 file terakhir, bisa menggunakan perintah :
- $ git log -3
- Kalau file nya sudah di modifikasi, bisa langsung saja menggunakan perintah:
- $ git commit -am “menambahkan data1 mahasiswa”
- Commit yang dilankukan akan punya sebuah identifikasi berupa has(23d5847) . Untuk mengetahui Branch uang aktif itu yang mana git mempunyai namanya Head sebagai pointer yang mengarah ke Branch yang aktif. Dan apabila ada perubahan tambahan file setelah commit maka ada has yang baru dan otomatis juga pointer dari master dan head juga akan ikut berpindah , jadi sekarang commit yang aktifnya ada di has yang baru, dai merupakan Branch master.Dan Branch master adalah Branch yang aktif.
-

# Branch

Jika mau membuat branch dari suatu file dengan cara ketik $ git branch nama_branch

- Misalnya mau membuat dua buah fitur, satu fitur untuk mengelola data dosen, dan satu lagi untuk mengelola data staff, jadi setelah menulis $ git branch dosen , maka nanti akan ada branch baru namanya dosen di commit yang sama. Lalu tambah lagi $ git branch staff , maka akan nambah branch staff, yang pointer nya mengarah yang sama.
- Untuk menampilkan Branch apa saja yang ada di dalam nya bisa ketik perintah: $ git branch , jikalau branch berwarna hijau berarti branch nya yang sedang aktif. Untuk berpindah branch kita bisa menggunakan perintah :
- $ git branch dosen , maka git nya akan membuat duplikat/ snapshot tapi masih di commit yang sama yaitu di branch master. Kita bisa melihat perjalanan dari branch kita menggunakan perintah graph:
- $ git - -all - -decorate - -oneline - -graph , untuk mempersingkat perintah bisa menggunakan alias:
- $ alias graph=”git - -all - -decorate - -oneline - -graph” , untuk penggunaan nya $ graph
- Jikalau mau berpisah branch bisa ketik $ git checkout nama_branch , kalau mau mengubah apapun di file mahasiswa.html dan melakukan commit maka yang bergeser adalah branch dan pointer nya.

# Ilustrasi Git



# Git MERGE

- Ada dua jenis Merge di dalam git
- 1. Fast Forward, terjadi ketika branch yang ingin digabungkan itu benda dalam jalur langsung/direat path. Merge ke dosen dengan cara $ git merge dosen , maka yang akan terjadi branch dan pointer nya akan berpindah ke commit di branch yang akan di merge, maka akan pindah ke file mahasiswa.html dan dosen.html $ git merge dosen
- Untuk menghapus branch dosen yang tidak dibutuhkan lagi bisa ketik perintah : $ git branch -d dosen , untuk mengetahui branch mana yang di merge bisa ketik  perintah: $ git branch - -merged
- 2.  Three-way Merge, jika ingin kembali ke file dosen, caranya kita bisa lakukan checkout ke commit 7 digit awalan nya, contoh: $ git checkout f923bc9 , pada saat checkout kesebuah commit kita ada di keadaan namanya’ detached Head’ artinya poin headnya lepas dari branch.

# Git Merge Conflig

- Git merge conflict adalah situasi yang terjadi ketika Git tidak dapat secara otomatis menggabungkan perubahan dari dua cabang karena adanya konflik dalam bagian yang sama pada file yang diubah. Konflik ini perlu diselesaikan secara manual oleh pengguna sebelum proses merge dapat diselesaikan.

# Git Remote

## SSH

- Ssh adalah singkatan dari Secure Shell. Ssh merupakan protokol jaringan untuk komunikasi jaringan yang aman dan terenskripsi. Pengguna sistem operasi Linuxatau MAC biasanya sudah sangat familiar dengan SSH. SSH merupakan aplikasi berbasis terminal. Di linux dan MAC, ssh sudah terinstall otomatis,sedangkan di windows, ketika kita menginstal Git, secara otomatis Git akan menginstall ssh juga. Git sendiri memiliki beberapa mekanisme untuk berkomunikasi dengan Git server, seperti http dan ssh.

## SSH KEY

- Hal yang pertama kita lakukan ketika menggunakan ssh adalah, membuat ssh key. Ssh key merupakan kunci yang digunakan untuk autentikasi ke ssh server. Untuk membuat ssh key, kita bisa menggunakan perintah ssh-keygen diterminal. Setelah selesai maka, secara otomatis akan terdapat 2 key di local kita, yaitu private key dan public key. Kita bisa melihat nya di dalam folder.Ssh di home direktory kita. File id_rsa adalah private key, dan id_rsa.pub adlah public key. SSH-keygen akan otomatis menghasilkan public/private rsa key pair dan dia akan disimpan ke /home/ilmi dan ada folder ssh yang dibuat otomatis di /id_rsa. Lalu klik enter langsung sampai selesai. Dan dia akan membuat private ke ydi /home/ilmi/.ssh/id_ed25519. Dan /home/ilmi/.ssh/id_ed25519.pub. Lalu masuk ke dalam direktori .ssh nya di /home/ilmi/ ketik perintah $ cd /home/ilmi/.ssh

- Lalu ketik $ ls -l untuk melihat isi file di ssh public ke Github.

# Git Remote

- Selanjutnya kita harus menambahkan ssh public key kita ke Github, setelah kita membuat ssh key, selanjutnya kita perlu meregistrasikan ssh public key ke Github. Hal ini dilakukan, agar ketika nanti terkoneksi ke git server di Github, kita perlu melakukan autentikasi lagi. Lalu masuk ke server Git yaitu Github dan pilih menu setting dan pilih lagi SSH and GPG keys.dan untuk title nya bebas, dan untuk key nya adalah public key nya yang ada di terminal tadi.lalu masuk lagi ke file .ssh tadi bisa dibuka file id_ed25519.pub dengan perintah: $ cat id_ed25519.pub , lalu copy semuanya dan pindahkan ke key yang ada di server tadi, lalu klik ADD SSH key. Dan selanjutnya kita melakukan proses pengetesan nya, jadi apakah kita udah bisa terkoneksi ke server Githubnya menggunakan ssh, jadi caranya kita bisa gunakan perintah: $ ssh -T git@github.com , dan apabila muncul seperti “ Hi yanurmendek! Successfully authenicated, butGithub does not provide shell acces” Artinya sudah sukses proses SSH nya.
- **REMOTE REPOSITORY**
- Ketika kita membuat git project, secara default, git tidak tahu tentang ramote repository, kita perlu memberi tah git project yang sudah kita buat tentang lokasi git repository. Untuk menambah remote repository, kita bisa gunakan perintah:
- $ git remote add origin , salah satu kebiasan di git, biasanya memberi: nama untuk remote repository dengan nama origin
- Yang pertama kita akan bikin direktori project nya dengan printah: $ cd Documents/  $ mkdir belajar-git-remote  , lalu
- $ git init , untuk membuat project nya lalu kita cek git branch nya : $ git branch  , lalu tambah file index.html : $ touch index.html
- Lalu masuk ke  file nya ketik perintah : $ nano index.html terus disimpan , lalu kita ketik : $ git add .

# Git Remote

- Lalu kita cek status : $ git status , kalau sudah ada kita coba git commit: $ git commit -m “ menambahkan file index.html”
- Lalu kita cek $ git branch , lepas tu baru kita tambahkan $ git remote add orogin git@github.com!ilmimendek/belajar-git-remote.git , Untuk melihat Remote Repository yang ada di git project , kita bisa gunakan perintah : $ git remote , untuk melihat URL detail remote repository, kita bisa gunakan perintah $ git remote get-url origin , utnuk menghapus Remote Repository , kita bisa menggunakan perintah : $ git remote rm name
- **PUSH**
- Walaupun kita sudah menyimpan perubahan di git project di local, tapi tidak secara otomatis akan di sync dengan Remote Repository. Hal ini karena sejak awal Git di deain sebagai distributed version control, artinya kita bisa melakukan perubahan dimanapun dan kapanpun, tanpa harus terkoneksi ke Git server. Oleh karna itu, jika ingin mengirim perubahan yang terjadi di git project di local kita, kita perlu mengirimnya secara manual ke git server untuk mengirim perubahan di local ke git server, kita bisa gunakan perintah yang bernama push.
- **PUSH BRANCH**
- Untuk mengirim perubahan branch ke remote repository dengan namabranch sama gunakan perintah: $ git push namaremote localbranch , untuk mengirim perubahan branch ke remote repository dengan nama branch yang berbeda-beda kita bisa gunakan perintah: $ git push namaremote localbranch:remotebranch.

# PUSH SEMUA BRANCH

- Jika kita ingin mengirim semua perubahan di semua branch ke remote repository kita bisa gunakan perintah :
- $ git push  - -all  , untuk menghapus branch yang ada di remote repository, kita bisa gunakan perintah : $ git push - -delete namaremote namabranch , perlu di ingat, menghapus remote branch bukan berarti menghapus branch di local, jadi jika kita ingin menghapus di local, kita harus lakukan secara manual.
- **CLONE**
- Apa yang harus dilakukan jika misal kita ingin download project git yang ada di sever ke komputer baru, Hal ini dinamakan perintah clone, dengan perintah clone, kita bisa download project di remote repository ke local dan secara otomatis di download sebagai git project. Melakukan clone , untuk melakukan clonem kita bisa gunanakan perintah : $ git clone urlremoterepository , secara default, clone akan membuat project dengan nama foldersama dengan nama project remote repository, jika kita ingin melakukan clone dengan nama folder yang berbeda dengan nama project remote repository kita bisa gunakan perintah;
- $ git clone urlremoterepository namafolder
- Default hasil clone , default clone akan berisi remote repository origin ke git remote repository yang kita clone, default clone akan berisikan branch utama di remote repository.