**Catatan Belajar Git WPU**

**Singkatan**

* Init : initialization
* am : add and give message
* m : message
* . : all
* v : verbose
* mkdir : make directory
* cd : change directory
* pwd : print working directory
* ls : list
* .. : back to 1 folder above

**#6 & #7 Branch & Merge**

Syntax git

1. git init : menginisialisasi folder untuk menjadi repository git
2. git add : menambahkan perubahan/penambahan file atau folder ke

dalam staging area

1. git commit : melakukan commit (merekam history)
2. git log : melihat history git
3. git checkout : kembali atau memilih keadaan commit tertentu
4. git commit -am : melakukan commit dan memberikan catatan terhadap file yang

sebelumnya sudah pernah ditambahkan ke staging area

(modified)

1. git branch : melihat semua branch dan branch aktif yang ada pada repo
2. git merge : melakukan merging
3. -m “[pesan]” : memberi pesan commit
4. git remote : mengecek apakah repo terhubung secara remote dengan repo

yang ada di suatu cloud

1. git remote -v : mengecek remote secara lebih detail
2. mkdir : membuat directory baru pada lokasi yang digunakan saat ini
3. cd : berpindah directory
4. pwd : cek directory saat ini
5. ls : cek isi directory
6. cd .. : kembali 1 folder diatas
7. git fetch : mengambil informasi dari github
8. git pull : menarik history repo dari git hub
9. git clone [alamat] : meng-clonning repo dari github
10. alias [nama alias] = ”command” : membuat alias untuk suatu perintah
11. git log --all --decorate --oneline –graph : menampilkan history commit dalam bentuk

visual

1. git branch -d [nama branch] : untuk menghapus branch yang telah di

merge

1. git branch -D [nama branch] : untuk menghapus branch yang telah atau

belum di merge

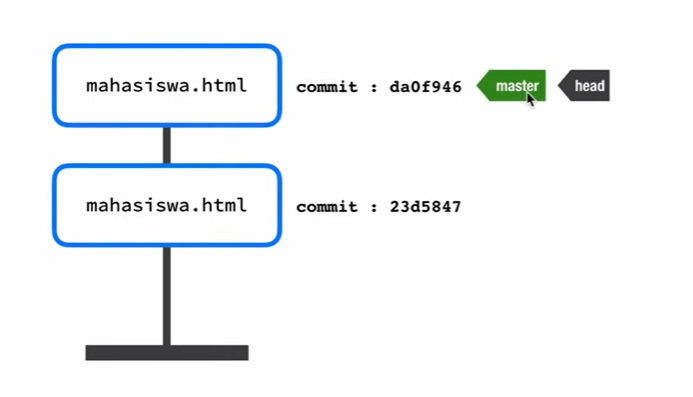
1. git checkout -b [nama branch] : Menambahkan branch lalu berpindah ke

branch yang baru saja ditambahkan

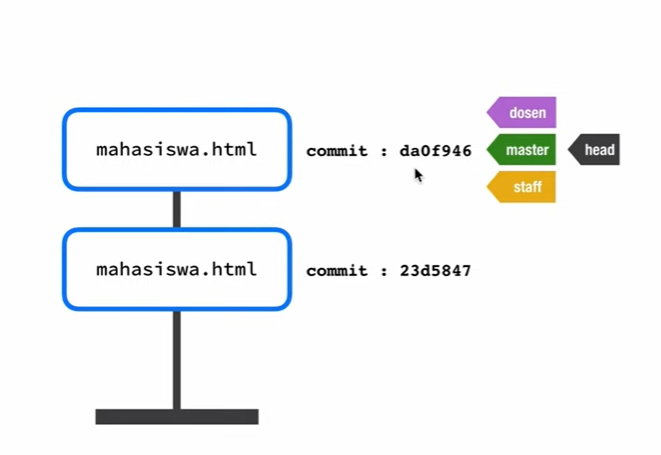
1. git add remote [alamat] : melakukan remote terhadap remote yang

ada di github

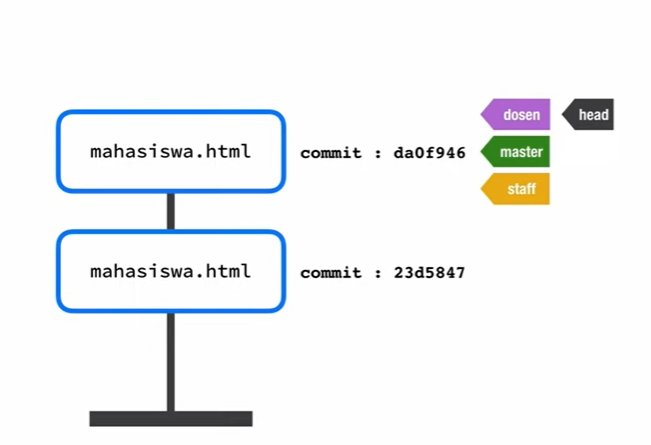
* **Ilustrasi**

****

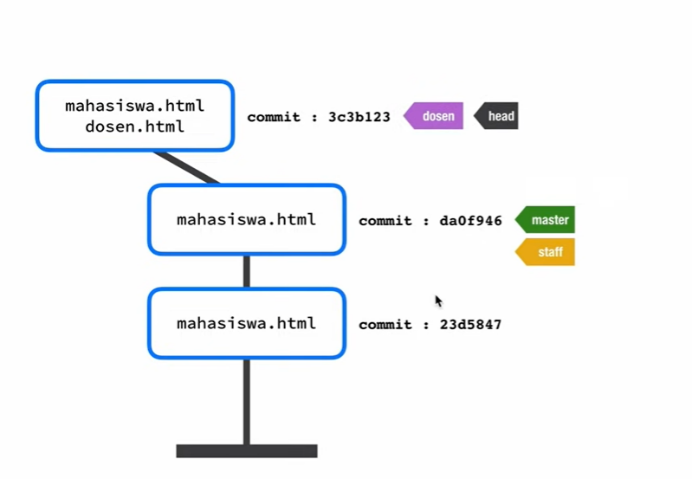
Gambar 1. Ilustasi Commit

****

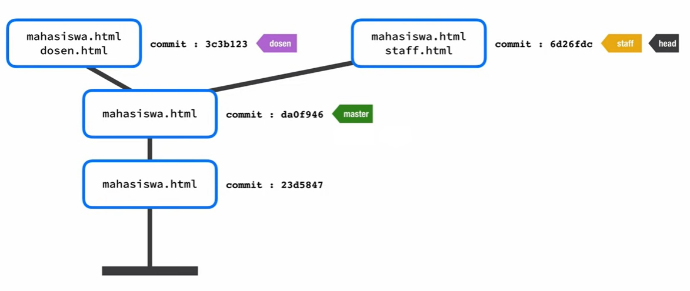
Gambar 2. Membuat branch



Gambar 3. Pindah branch

****

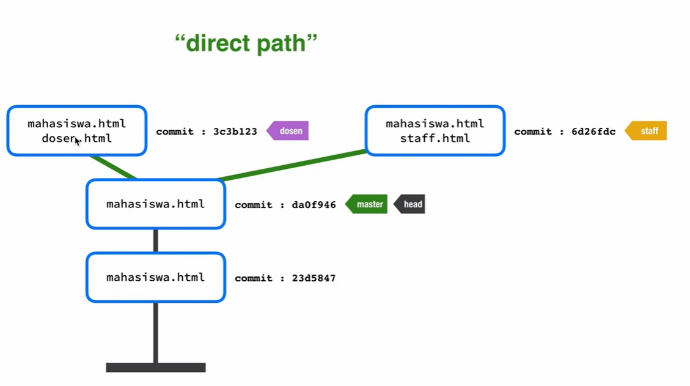
Gambar 4. Commit pada branch

****

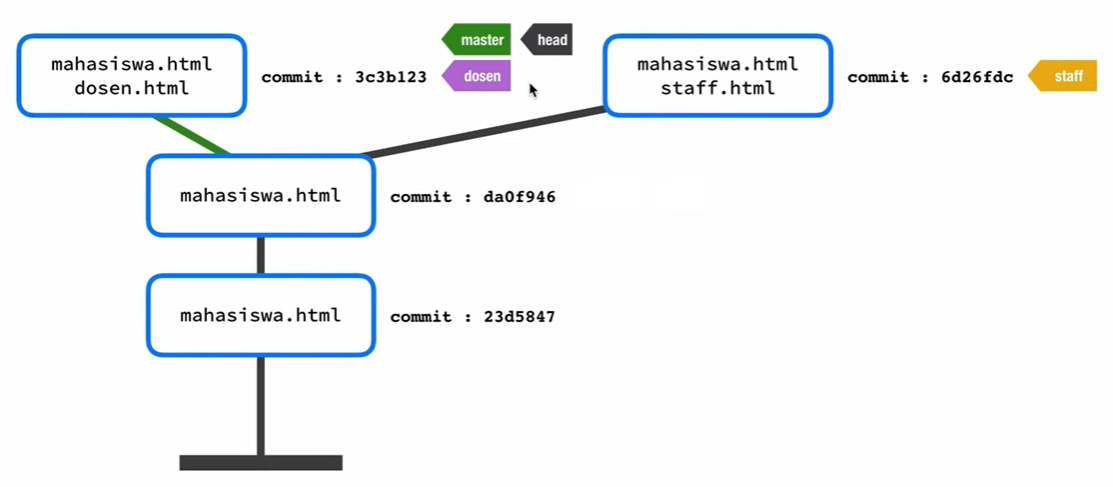
Gambar 5. Commit pada branch lain

* **Jenis Merge**
* Fast Forward

Ketika kedua branch yang akan di merge terhubung secara langsung

****

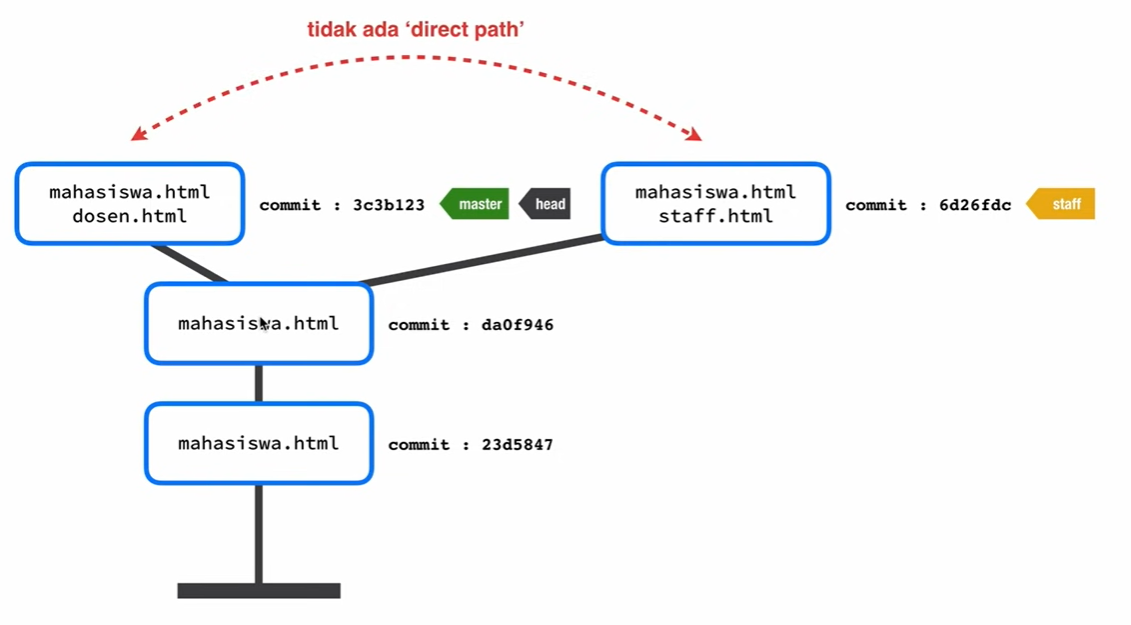
Gambar 6. Direct path



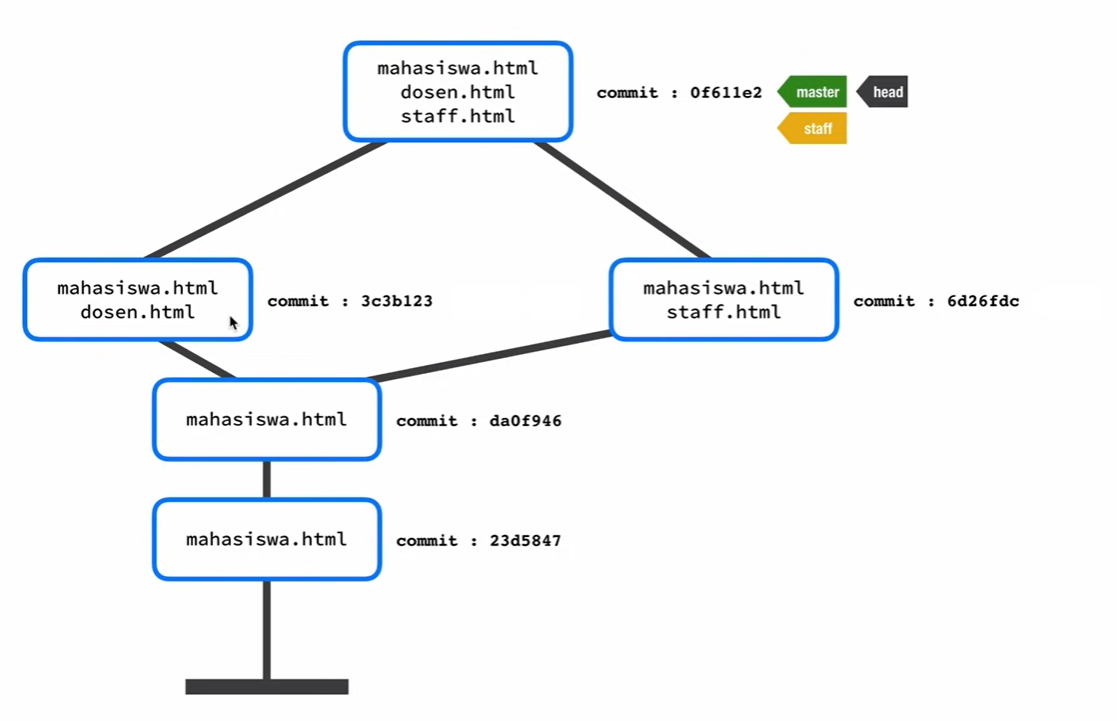
Gambar 7. Hasil fast forward merge

* Three-way Merge (Merge-Commit)

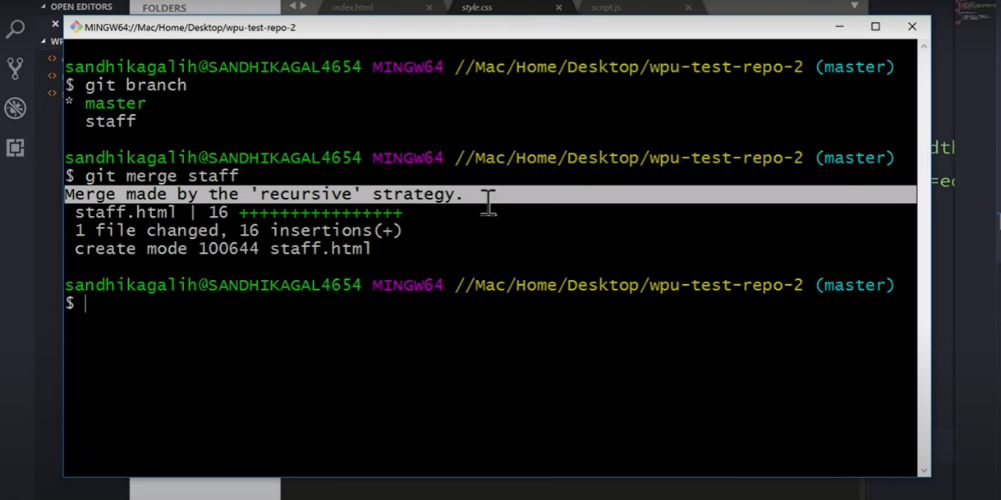
Dilakukan ketika tidak ada direct path antara dua commit branch yang akan d merge

****

Gambar 8. Tidak ada jalur langsung



Gambar 9. Iluatrasi hasil three-way merge



Gambar 10. Hasil three-way merge pada tampilan git bash

**#8 Git Remote**

1. Melakukan clone ketika git ada di github (server) dan ingin mengclone di git (local)

* Buat repo pada github
* Tentukan lokasi repo akan disimpan
* Lakukan clone pada git-bash dengan mengetikkan perintah git clone alamat HTTPS

1. Melakukan remote pada github (server) dengan repository yang sudah ada di git (local)

* Buat repo pada github dengan nama yang sama dengan dengan repo yang ada di git (tidak harus sama) agar memudahkan pendataan
* Kosongkan checklist untuk membuat file README
* Ketikkan perintah berintah berikut pada git bash: git remote add [nama remote (secara default namanya adalah origin)] [alamat HTTPS repo pada github]
* Buat git branch dengan perintah: git branch -M main
* Setidaknya melakukan 1x commit pada git local
* Lakukan push upstream dengan perintah: git push -u [nama remote] [nama branch]

1. Menyelesaikan atau mengatasi conflict ketika file yang sama pada github dan git dirubah pada baris yang sama

* Pull terlebih dahulu file yang ada di github ke git
* Git akan mendeteksi adanya conflict dan secara otomatis akan membuka text editor untuk menyelesaikan conflict
* Selesaikan conflict pada text editor
* Lakukan merge-commit
* Push perubahan ke github

**#9 GitHub Pages**

Link mengupload web static ke internet: <https://youtu.be/rgDDWhQe-ow>

**#10 Multiple Remotes**

Multiple remote digunakan untuk menghubungkan git local dengan lebih dari satu remote yang ada di github atau penyedia layanan vcs berbasis cloud lainnya. Hal ini dilakukan ketika seorang programmer mengerjakan suatu project secara tim sehingga repo dari setiap tim saling terhubung.

Link: <https://youtube.com/playlist?list=PLFIM0718LjIVknj6sgsSceMqlq242-jNf>

**#11 Remote Branch**

Remote branch adalah membuat branch baru untuk mengedit repo hasil fork dan mengedit repo asli pada remote branch yang kemudian melakukan pull request (PR) terhadap perubahan yang telah dibuat kemudian bila perubahan diterima oleh penulis kode asli maka dilakukan merge di lokal, dan melakukan push ke master branch dari hasil fork yang sebelumnya dilakukan agar baik github maupun git up to date dengan repo asli hasil forking.

Link: <https://youtube.com/playlist?list=PLFIM0718LjIVknj6sgsSceMqlq242-jNf>

**#12 Gitignore**

Gitignore adalah sebuah file yang disimpan di dalam repository dimana isi dari gitignore adalah list dari file2 atau folder2 yang tidak akan diikutsertakan pada saat memasukkan file2 atau folder2 ke *staging area* dengan perintah masukkan semua data yang berada pada working tree ke staging area **(git add .)**. Contoh file yang umum untuk tidak dimasukkan adalah file system seperti thumbs, flie konfigurasi lokal, dll.

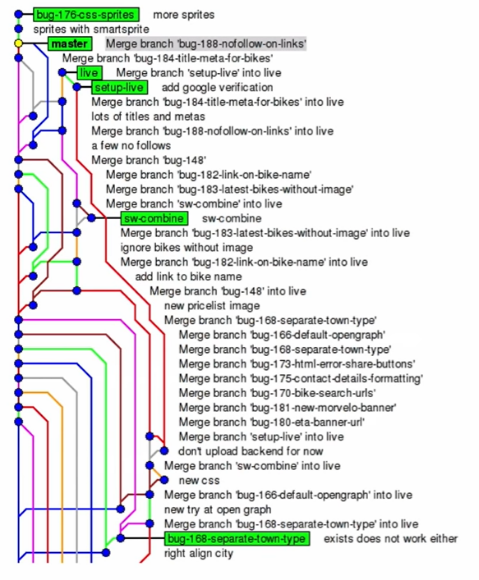
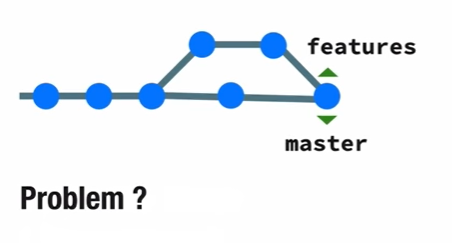
* Untuk memasukkan folder ke dalam .gitignore 🡪 [tulis nama folder]/
* Untuk memasukkan semua tipe file yang sama 🡪 \*.[format file]

#13 GitRebase

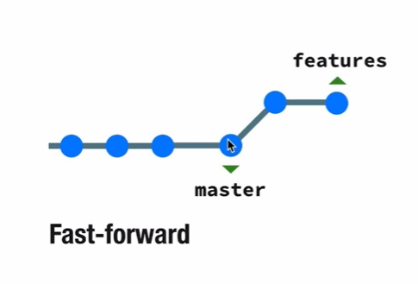
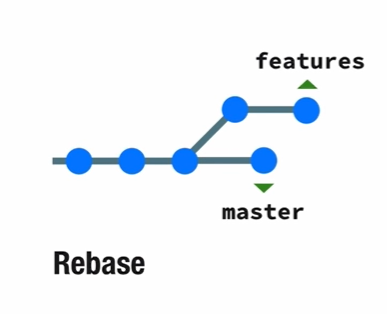
Git Rebase adalah salah satu workflow selain merge dimana rebase akan memajukan commit dari branch yang tidak memliliki direct path dengan branch lain yang akan dilakukan merge sehingga akan masuk ke jalur branch yang ingin dimasukkan kedalamnya.

* Git rebase memudahkan dalam melakukan tracking karena tidak melakukan proses three-way merge atau merge commit

**Ilustrasi GitRebase**



Gambar 11. Merge Workflow



Gambar 12 Rebase Workflow