# Git Dan Github

VCS Digunakan untuk mengelola perubahan sebuah dokumen. Digunakan untuk mengkombinasi dokumen dll. Memungknkan kembali ke keadaan sebelum perubahan (checkout)

## Git

Sebuah vcs untuj mengeloka file didalam folder. Folder (**repository**)

Riwayat perubahan disimpan menggunakan **commit**

Setiap melakukan commit ada hast (penenda perubahan)

Melakukan percabangan untuk menghindari commit utama(**brach**)

Jika menanam **branch**  ke commit utama **branch**

Proses pengambilan file dari vcs adalah **pull - >**commitnya

Proses mengirimkannya **push -> commitnya**

Sebelum dapat menggunakan push dan pull

**Github :** system cloud VCS secara online



# Git Dan Github

Untuk memantau pergerakan penyimpanan commit akan menambah

# Branch

Digunakan untuk melakukan percabangan dari jalur utama (banch main), biasanya digunakan untuk menambahkan pengerjaan pada project yang akan dilakukan oleh orang lain, lalu pengabungannya melalui 2 tahap yaitu **pull** dan **merge**

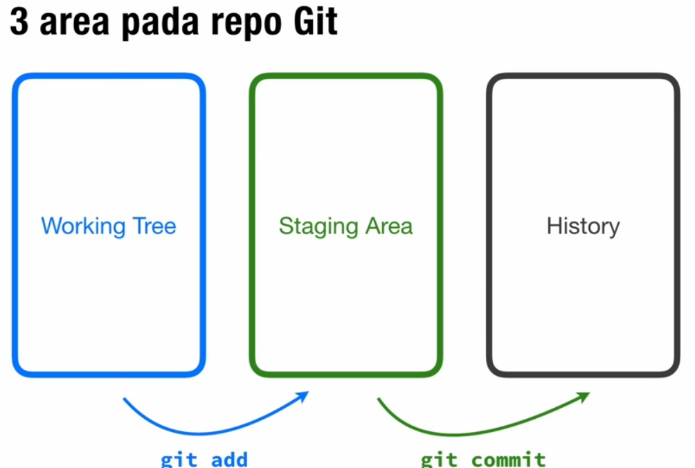
# Fork

Digunakan untuk mengduplikat repository eserta dengan history, bias untuk mengusulkan perubahan ke pemilik asli (maka kita sebagai orang yang melakukan fork akan melakukan pull request terhadap repo aslinya), **catatan form tidak sama dengan clone** karena fork digunakan untuk menduplikat repo orang lain ke kita, clone diduplikat ke computer local kita. Kita juga dapat melakukan pull request ke pemilik request aslinya.

# Bekerja dengan git

Kita akan berinteraksi denan gitbub secara local menggunakan computer kita menggunakan **git (**software**).**

**Mengistal** git-scm.com

** **

Kalau ada perubahan akan ke staging area menggunakan **git add**  maka perubahan akan disimpan di staging area, dari staging area kalau sudah oke, bias lakukan commit (**git commit**)

Pwd : letak git

Ls : menampilkan semua file

Cd : pindah directory

Init : akan muncul **master,**  dengan begini git akan membaca folder sebagai repository, dan akan dibuatkan folder .git

Ctt : ketikan akan melakukan commit perlu login dengangit menggunakan akun gitub

## Membuat folder menjadi repo git

Melakukan **git innit** agar folder menjadi repo

**Git add <nama file>** : untuk 1 file

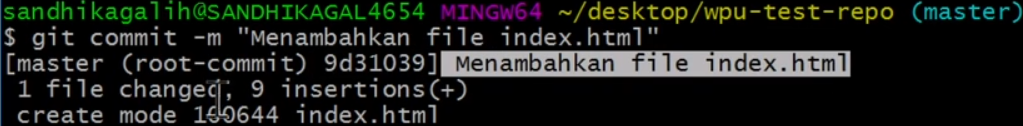
**Git add .** : Menambahkan banyak file secara otomatis



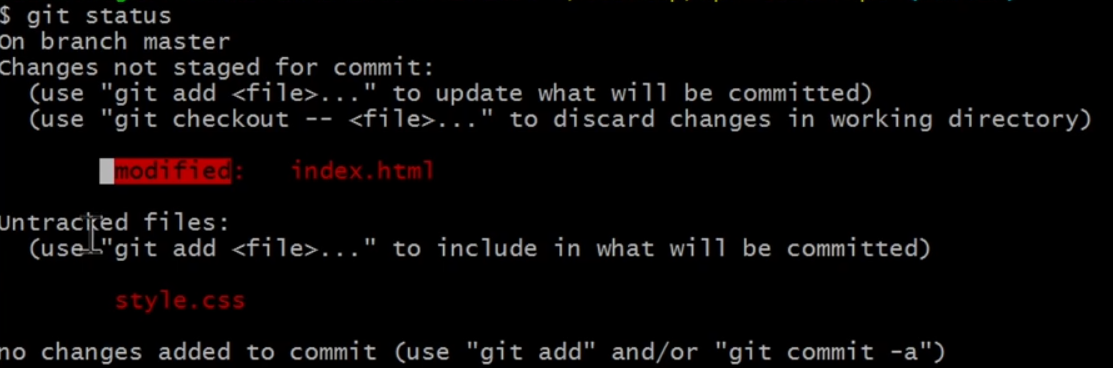
Code diatas untuk log8in ke github

Lalu commit

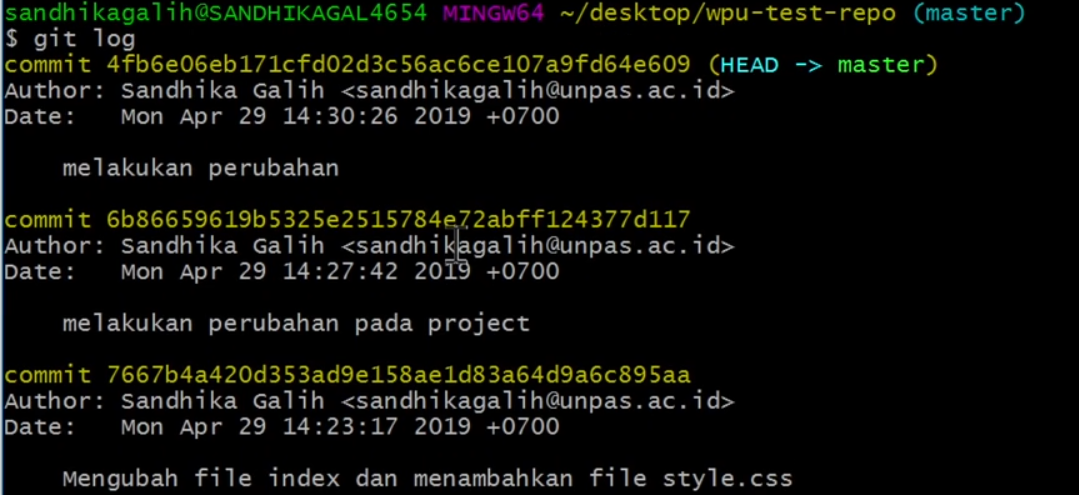
## Commit dengan pesan



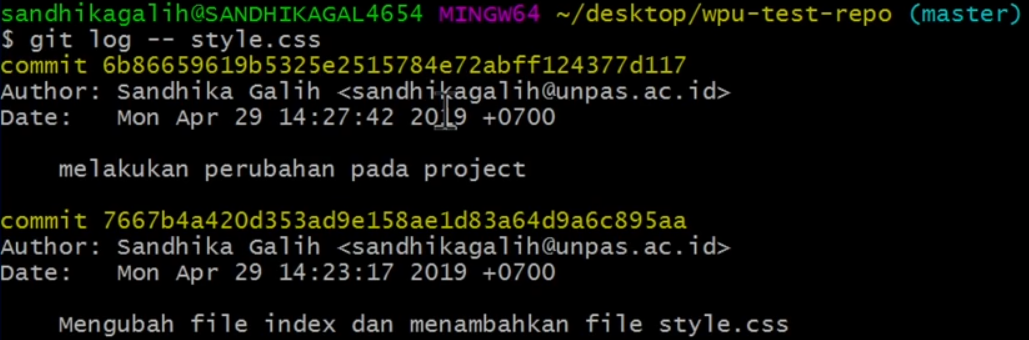
## Mengetahui status dalam repo



## Mengetahui comit commit apa saja yang sudah dilakukan

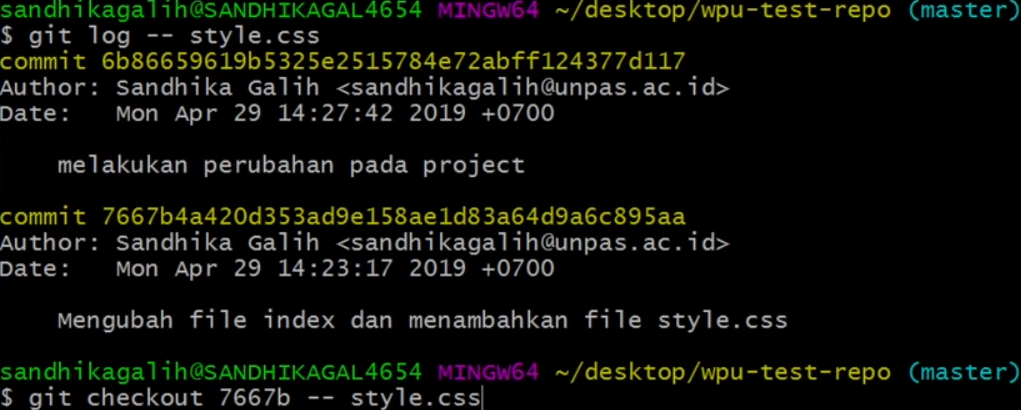


## Mengetahui commit pada file tertentu



## Mengembalikan keadaan (Cekout)

Misalkan sebelumnya telah menghapus file namun ingin mengembalikannya dengan menuliskan hashnya contoh:



Meski begitu kita perllu commit lagi untuk menyimpan cekout terhadap file style.css

**Git commit –m “cekout file syle.css”**

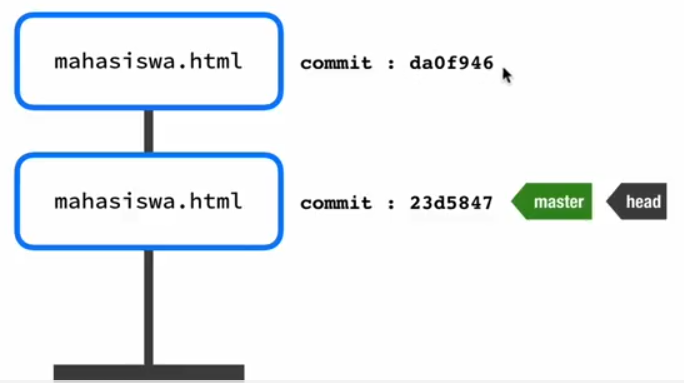
# Git Branch dan Merge

Kita akan mencoba buat file

* Git init
* Git add .
* Git commit –m “menambahkan file index.html”
* Memodifikasi index.html (ini akan membuat file berstatus modified dpt dilihat pada git status atau apada vc code langsung commit namun menggunakan kode) **git commit –a –m “<PESAN>”**

## Implementasi Branch

Materi : Setiap menggunakn git pasti aka nada branchnya secara default bernama **master.** dan untuk mengetahui branh mana yang sedang digunakan akan ada penanda **head** yang menunjukkan ada dibranch manakah kita.



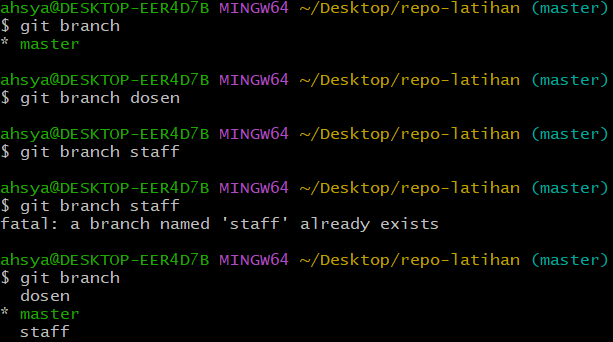


Disamping adalah ketika ada branch lain(dosen dan staff).

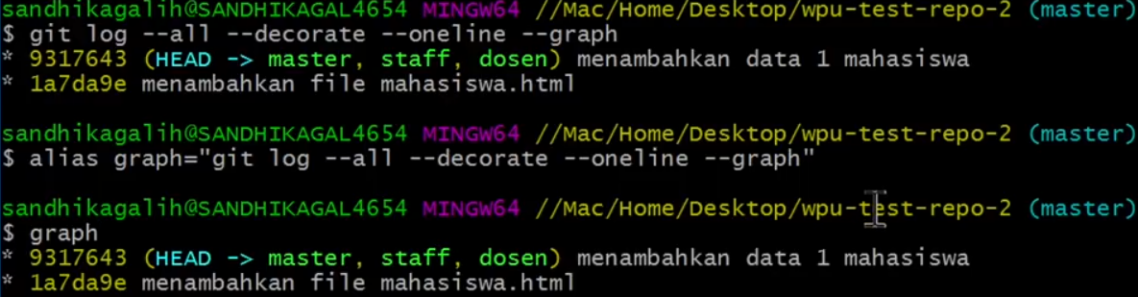
## Membuat branch

Git branch (mengetahui branch ada didmana)

Git branch <nama branch> (membuat branch)



## Print graph (history)

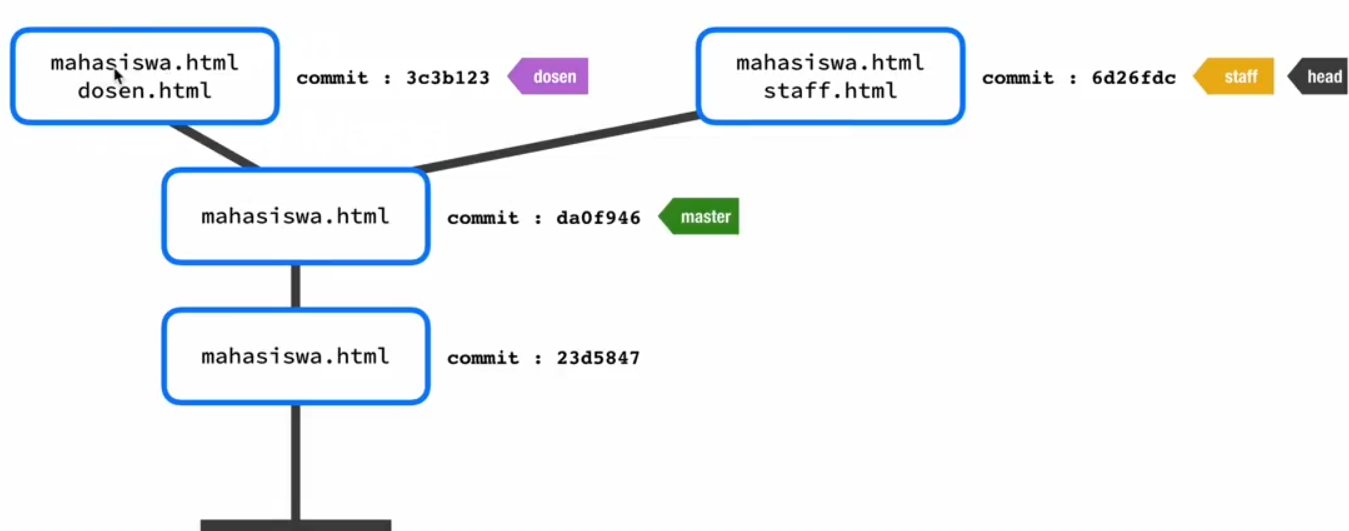


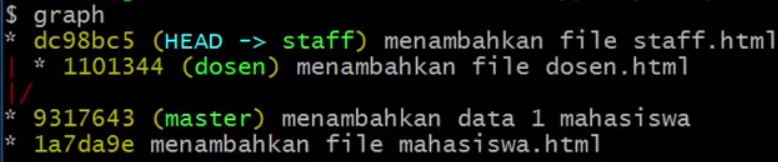
Diatas juga menambahkan command alias untuk mempermudah, pemangilan graph

## Berpindah branch

Kita akan menggunakan **checkout**. Sebelumnya kita menggunakan checkout untuk berpindah pada suatu kondisi  **commit** namun sekarang kita gunakan untuk berpindah branch:: **git checkout <nama branch>**

## Print graph





## Merge

Git brance – merge :: utk mengetahui branch yang sudah di merge

### Fast Forward Merge

### Threeway Merge

# Merge Problem

1. Melakukan merge dengan branch lalu, jika error, lakukan editing dan **git add**

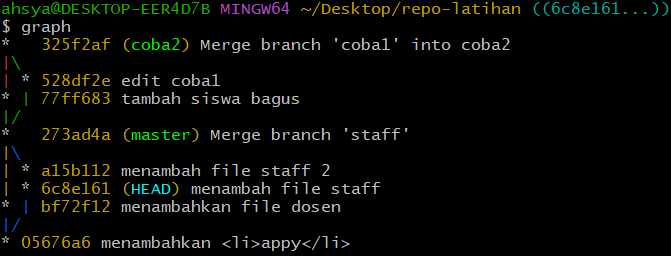
## Hal yang perlyu diperhadikan saat perpindah dengan commiyt

Kita akan melakukan pemindahan ke suatu commiyt

* Git log (lihat history commit)
* Git checkout <7 digit head>

### Mengatasi detached head

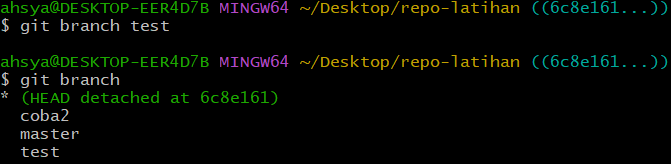
Sebelumnya kita telah berpindah kecommit masalahnya akan terjadi **detached head** yang menyebabkan head tidak pada suatu branch



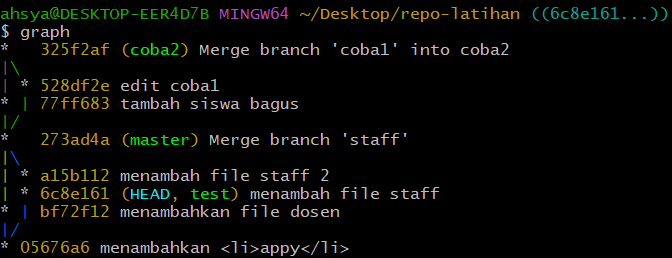
### Mengatasi detech head

Saat kita berpindah ke suatu commit menggunakan command <**git checkout** <7 digit commit>>

* Buat cabang baru dari commit **git branch test**



* Sekarang ada 2 branch pada commit namun head tidak mengarah ke test

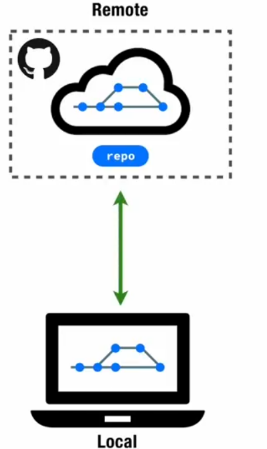


* Maka kita hanya perlu pndahkan dengan

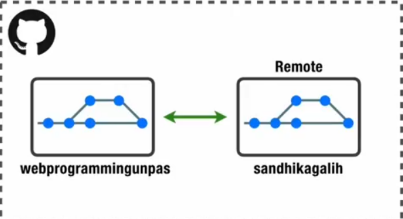
**Git checkout test**

sehingga seperti membuat branch baru seperti biasa

# Git Remote

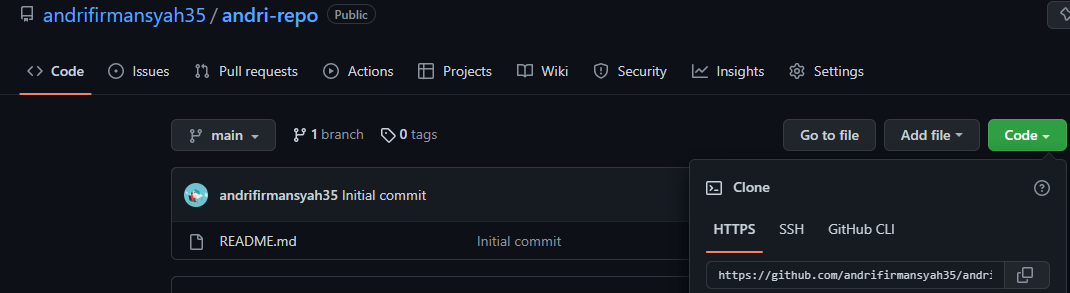
Menghubungkan git local keg it hub, jika ada yang mengedit atau commit keg it hub, kita bias menatrik perubahannya dengan **patch atau pull**

* Dengan git remote, local dapat terhubung dengan banyak remote(githib,gitlab,dll)
* Bias juga git hub ke github dengan melakukan **fork** bias menjadi remot untuk repo kita

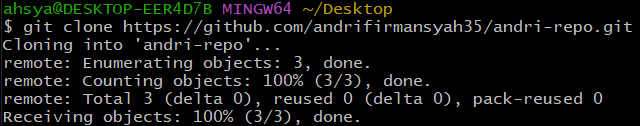


## Mulai remote

1. gunakan link https pada github-repo



1. buka git pada desktop, lalu lakukan



1. maka anda akan mendapatkan folder sama dengan repo yang telah diclone
2. lalu kita masuk ke folder **repo** dengan git, lalau jalankan **git remote**

## nama dari remote (origin)

origin merupakan default nama dari remote, namun kita bias saja membuat dengan nama lain



### Detail remote

Menunjukkan link yang dapat digunakan untuh fetch dan push pada remote **origin**



### Branch pada remote

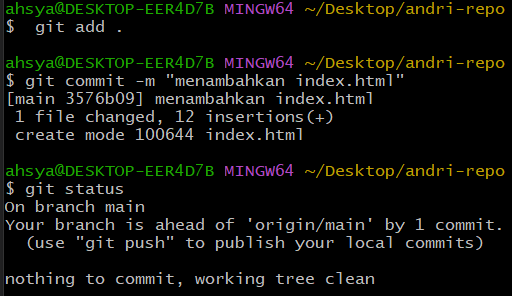


Diatas merupakan satu commit dan tanda **merah**  merupakaan branch dari remote **origin** pada pointer yang sama

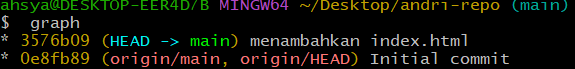
## Melakukan push

Coba buat suatu file kemudian commit file tersebut

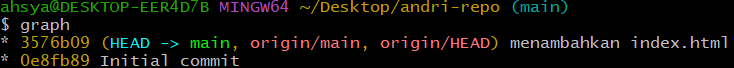
Lalu coba untuk melihat status giit maka



Branch yang kita punya mendahului branch yang ada pada remote dengan satu vcommit , jjika kita lihat pada graph maka graph pada remote tertinggal satu commit dari branch yang kita gunakan sekarang



Agar sama kita perlu mengirimkan perubahannya melalui remote kita dengan perintah **push, cukup**  git push

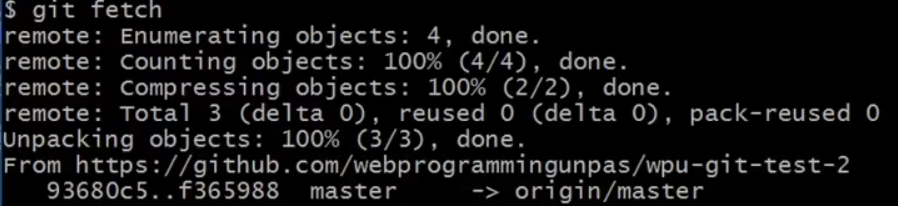


## Git remote ke local ke github

1. Membuat repo – isikan sesuatu
2. Buat repo digithub lalu jangan ada readme supaya benar benar kosong baru,
3. 
4. Ctt : pastikan repo local dengan github memiliki nama yang sama

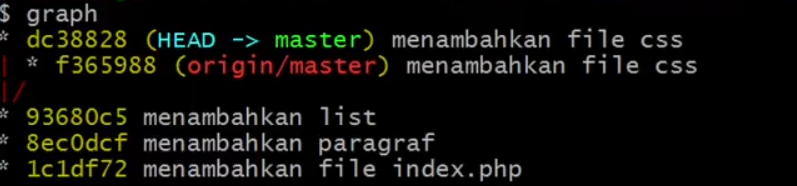
## Git fetch

Cek commit remote

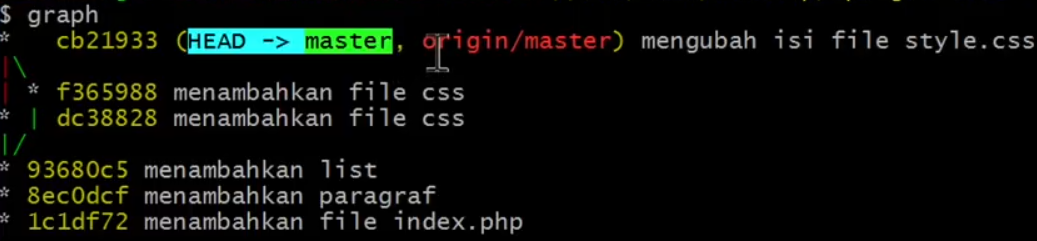


## Gitt pull

Jika kita ada config ada 2 commit satu dari github dan local berbeda, dan jika sebelumnya gagal di push , local membuat commit dan rremote membuat commit (sehingga membuat cabang)



* Lakukan git pull
* Edit problem
* Git add., git commit, git push



# Git Hub Pages

Menampilkan penjelasan repo/halaman github kita, kebanykan github pages digunakan sebagai webhosting. Sehingga kita dapat memanfaatkan halaman gthub sebagai halaman pages kita.

Setting git hub pages

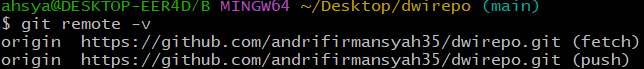
# Multiple Remote

Misalkan kita tertarik terhadap repo github orang lain

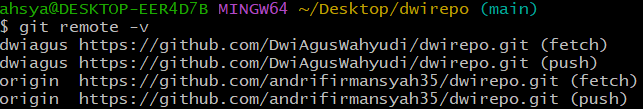
Kita dapat melakukan fork pada repo tersebut<untuk memiliki cloning kea kun github kita>

Kita juga inggin mencopy juga ke local

**Sehingga kita inggin ketiganya selalu sinkron**

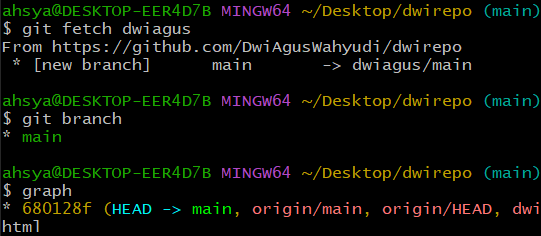
* **Melakukan form** pada repo milik dwi agus repo yg digunakan **andrifirmansyah35**
* Lakukan clone pada repo andri dengan https
* **Cek remote**
* ****
* **Sehingg** kita pada saat menggunakan remote, yg berubah hanya github **andrifirmansyah35**

**Agar sinksorn ketiga repo**  kitap perlu menambah remote baru

* Giit remote add <nama remote> <url>
* Git remote add dwiagus https://github.com/DwiAgusWahyudi/dwirepo.git
* 

## Melakukan fetch agar kedua remote ada

Fetch agar commit diambil



Sudah terhubung dengan kedua remote, dengan kondisi commit yang sama (ada andri,local,dan asa(dwiagus))

## Story

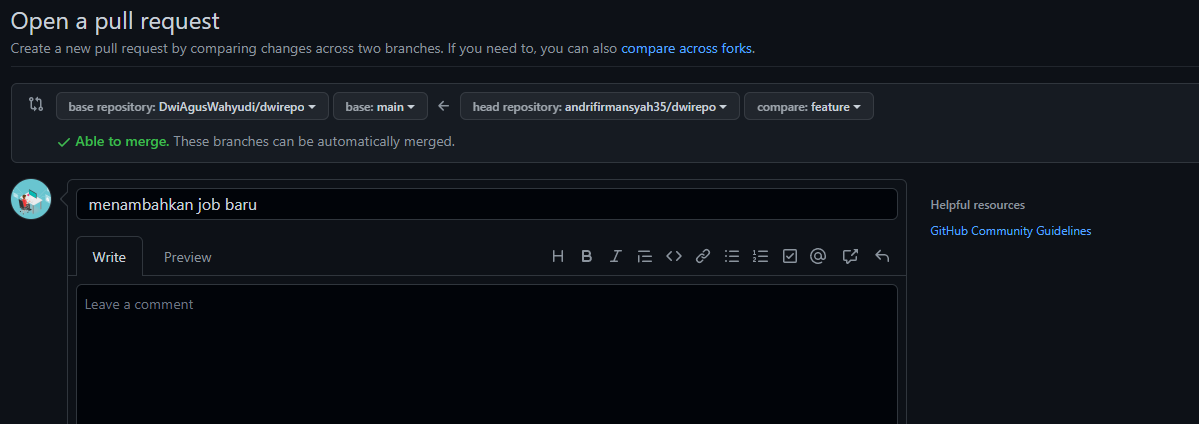
Kita akan melakukan perubahan pada repo asli(dwiagus) lalu apa yang akan dilakukan supaya andri Firmansyah(akun yang melakukan **fork**) dapat mendapatkan perubahan yang terjadi pada repo asli (dwiagusWahyudi/dwiagus)

1. Melakukan **git fetch** untuk mengetahui bahwa commit akun asli
2. Jika sudah ketahuan kita hanya perlu merge branch pada akun kita, karena masih pada local maka sekarang kita hanya perlu melakukan push pada akun kita

# Remote Branch

Kita akan melakukan melakukan pull request dimana perubahan agar dapat diterima oleh repo pemilik aslinya.

Lakukan perubahan dengan branch baru lalu push branch tersebut, lalu kea kun lakukan pull



## Akun asal repo accept pull request

Setelahmelakukan nya,akun fork perlu memperbaiki branch utamanya, maka perlu melakukan fetch untuk mengetahui perkembangan commit repo asal , melakukan merge dengan remote repoasal lalu push ke repo

* Git push origin –delete feature : push dan delete branch

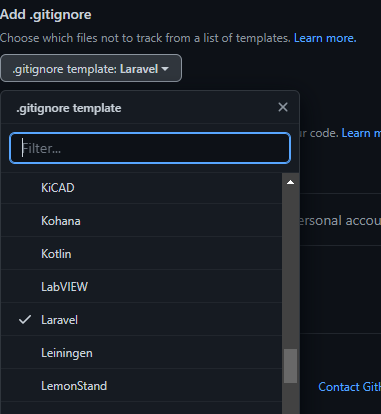
# Gitignore

(.gitignore) ini merupakan sebuah file yang dapat kita simpan pada git saat kita melakukan add dan commit. Misalkan pada os ada file hidden yang ketika kita add akan ikut tersimpan dan pada windows ada file x yg ikut tersimpan, itu semua bias diatur menggunakan git ignore.

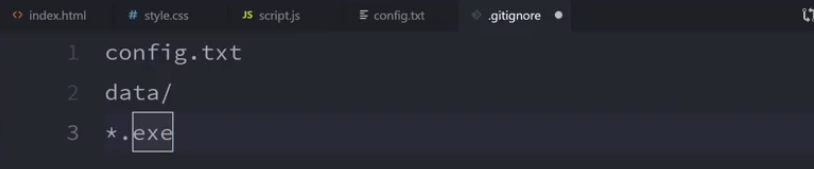
## Penggunaan gitognore-

Membuat repo <coba git ignore>

## Membuat git ignore saat memulai git



## Git ignore pada local(membuat file dengan nama .gitignore)



1. Mengignore file(config.txt)
2. Meng-ignore file dalam flder(data)
3. Mengignore semua file dengan extensi (.exe)

## Referensi ignore

Github.com/github/gitignore

Gitignore.io

