**GIT**

**GIT Command(Local)**

**-**git init

-git add <file(s)>

-git status

-git commit

-git config

-git branch

-git help

-……

**3 Area pada repo GIT**

***-*Working tree**->Folder tempat kita bekerja

-S**taging area** ->kasih tau git kalau kita melakukan perubahan,dan belum dicommit

->dengan code git add

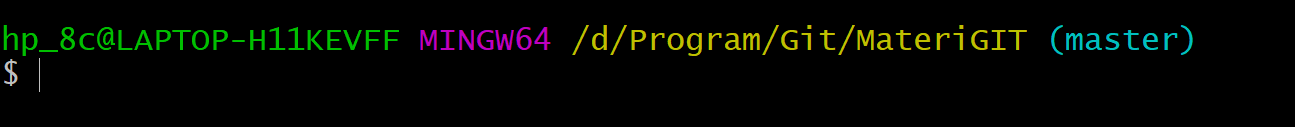
-**History**->jika sudah di commit

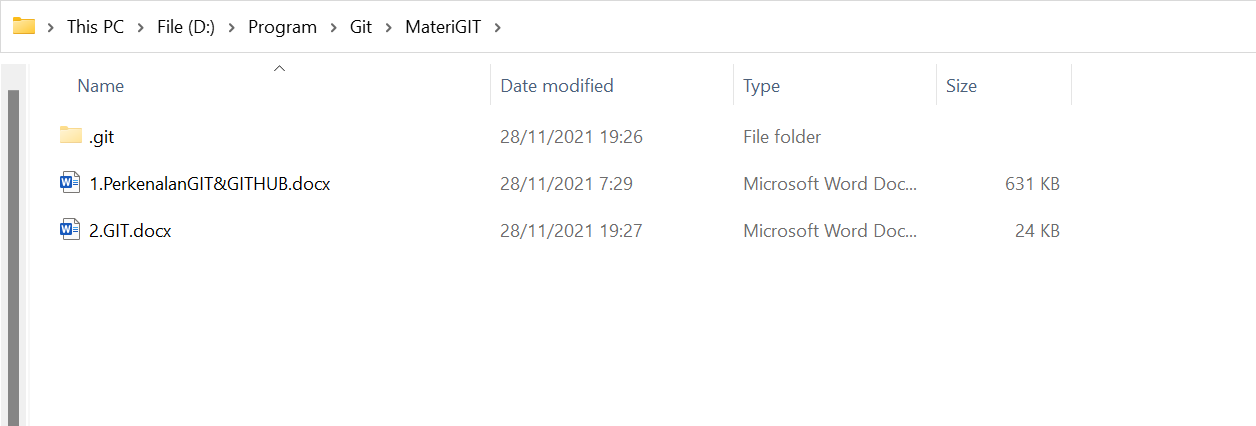
->dengan code git commit

->Staging dan history akan tersimpan di folder .git

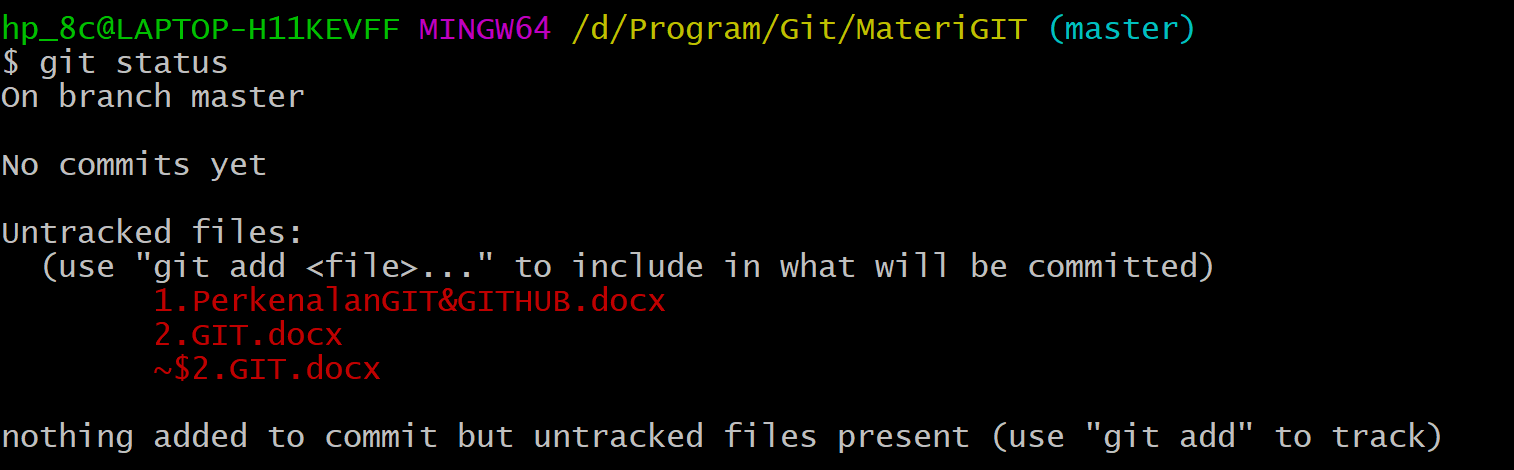
Langkah :

1.Pilih folder yg ingin dijadikan repo dan lakukan **git init**

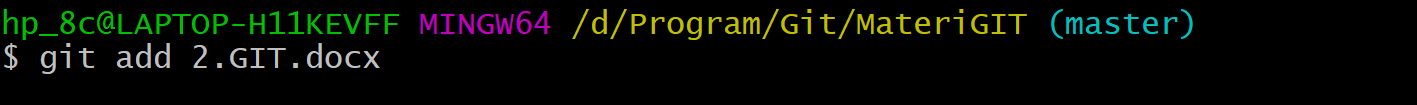




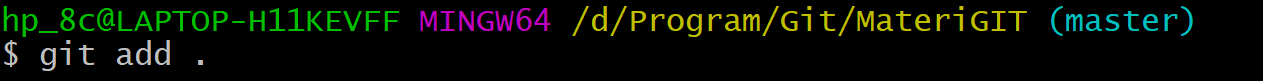
2.cek status -> git status



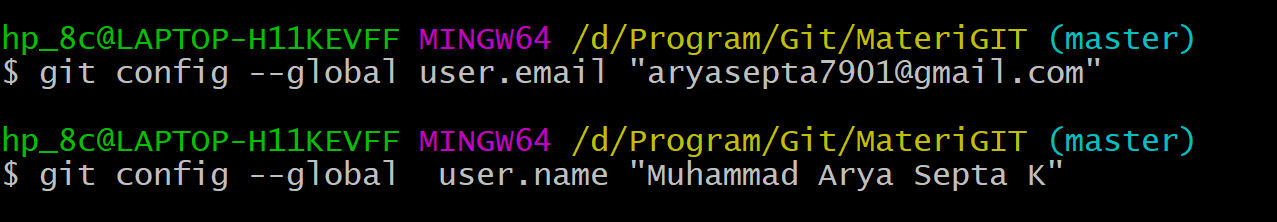
3. tambah file tersebut dengan git add -all



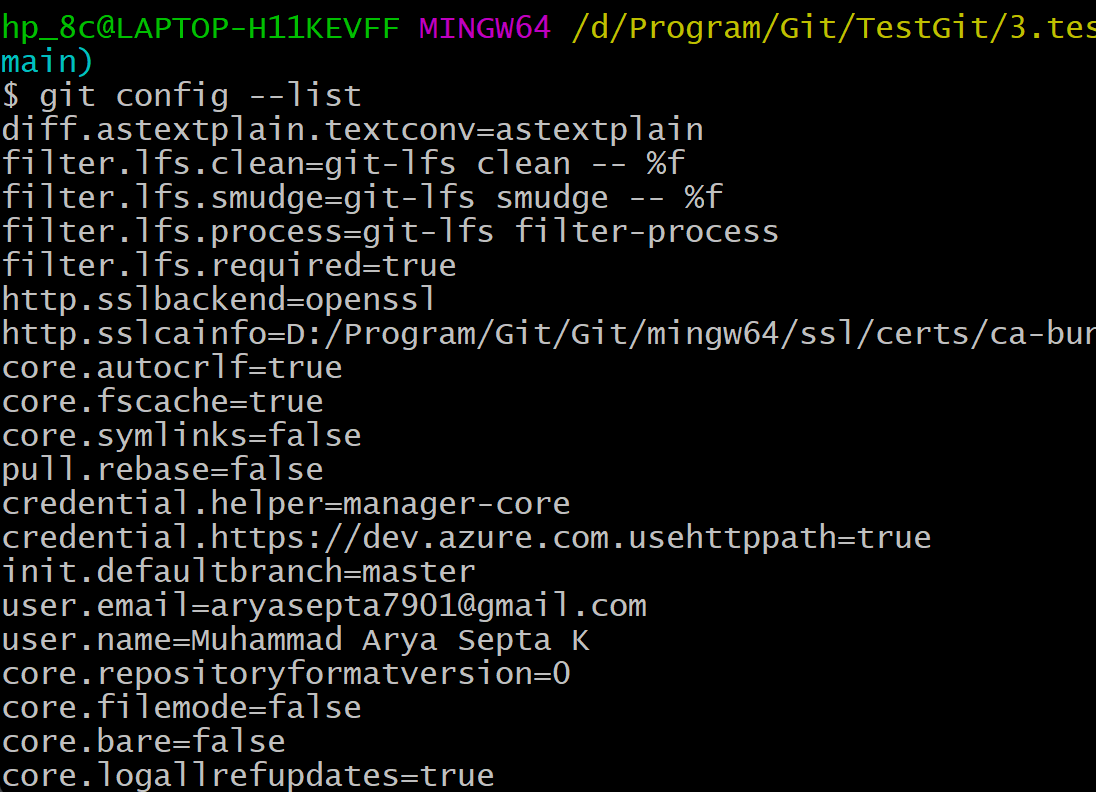
Untuk add semua



4.Tambahkan git commit,tapi sebelum itu kita tambah informasi data pribadi



Bisa pakai git config -- local -> untuk folder tertentu ,dan ceknya git config -- list



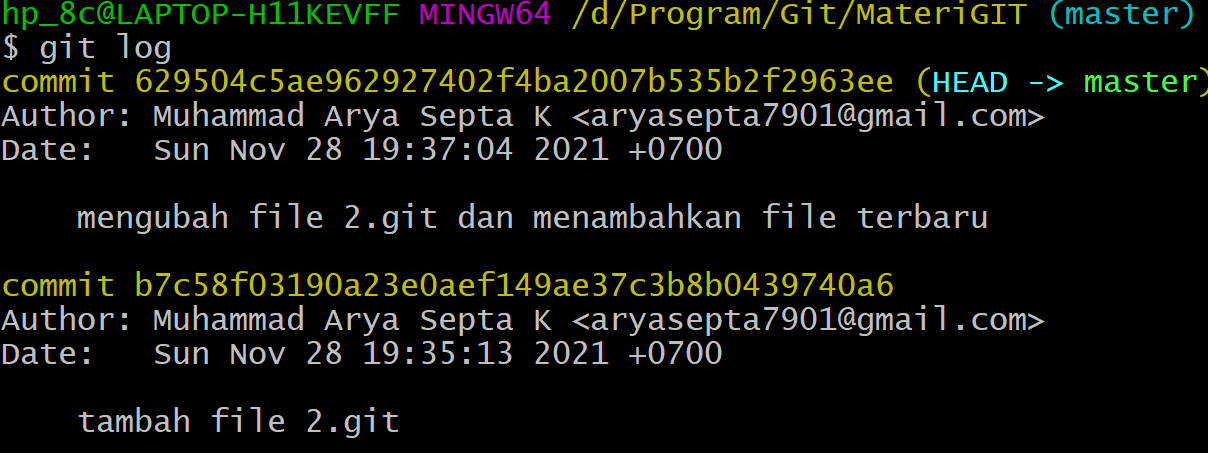
5.Lakukan commit

->git commit -m “menambahkan file indeks.html”



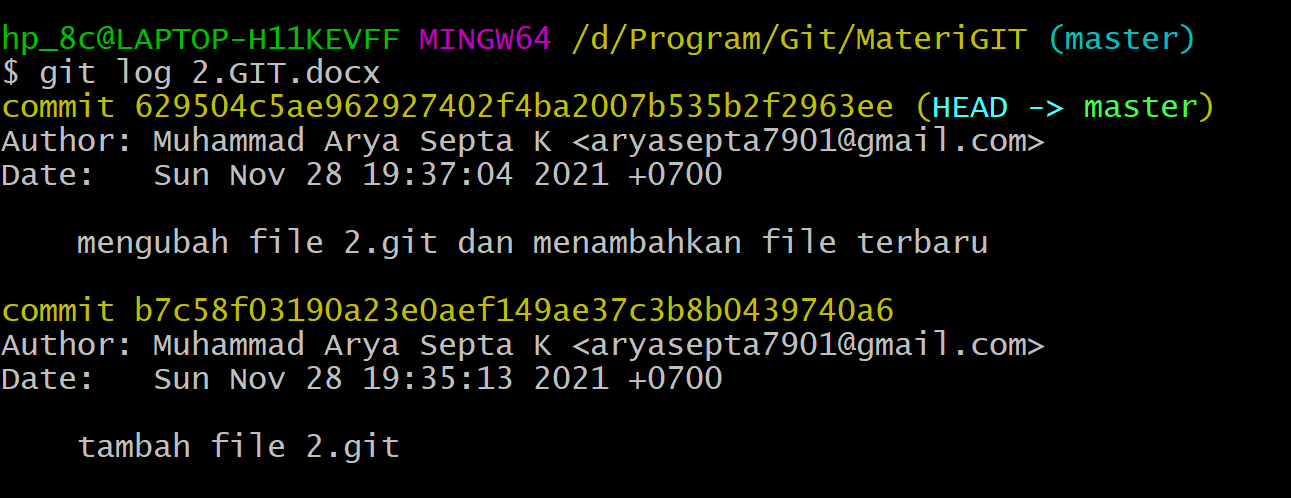
6. cek semua commit

->git log

-> git log pada kondisi tertentu git log satu dan dua



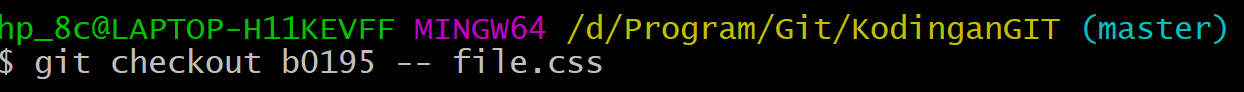
->git log pada file tertentu



7.mengembalikan file yang hilang

->git checkout <5angka awal commit> -- nama\_file

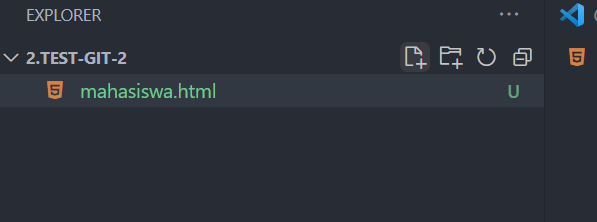
Note: directory dimana file tersebut belum dihapus

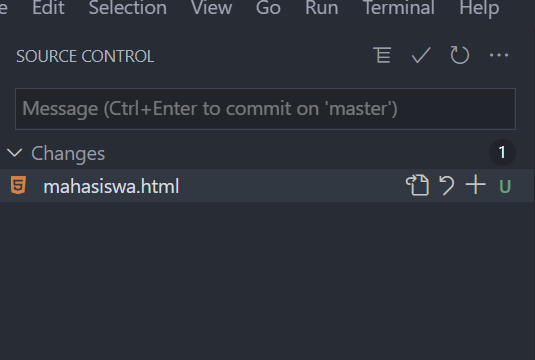


**GIT Branch dan Merge**

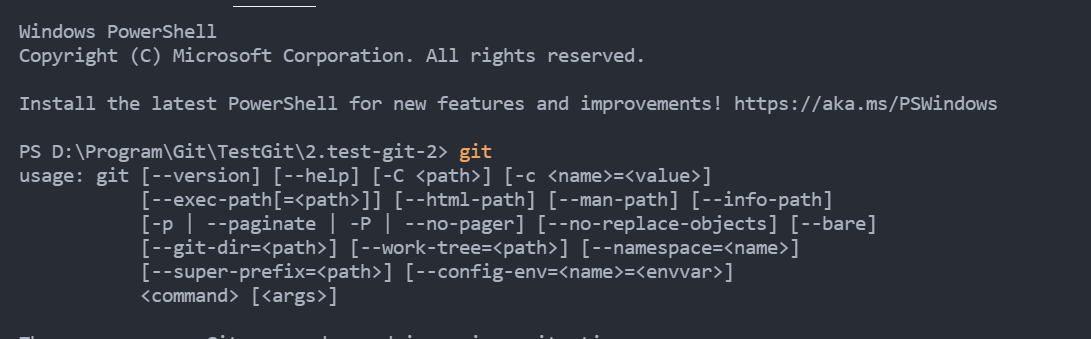
1.Git Init

2.File masi Untrack.



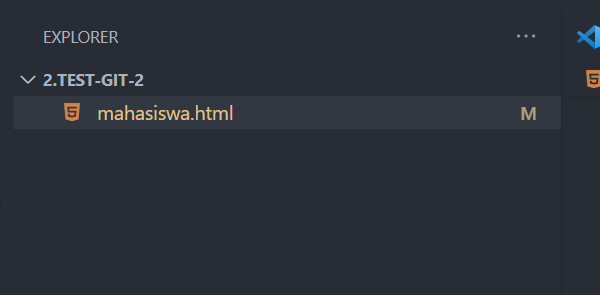


3. di VSC ada terminal dan bisa dijalankan disana

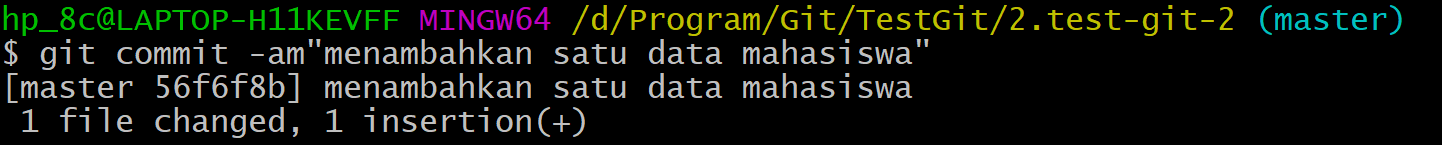


4.git add . ,dan git commit -m”…..”

5.Ketika file sudah di commit,dan kita ubah,maka dia modified

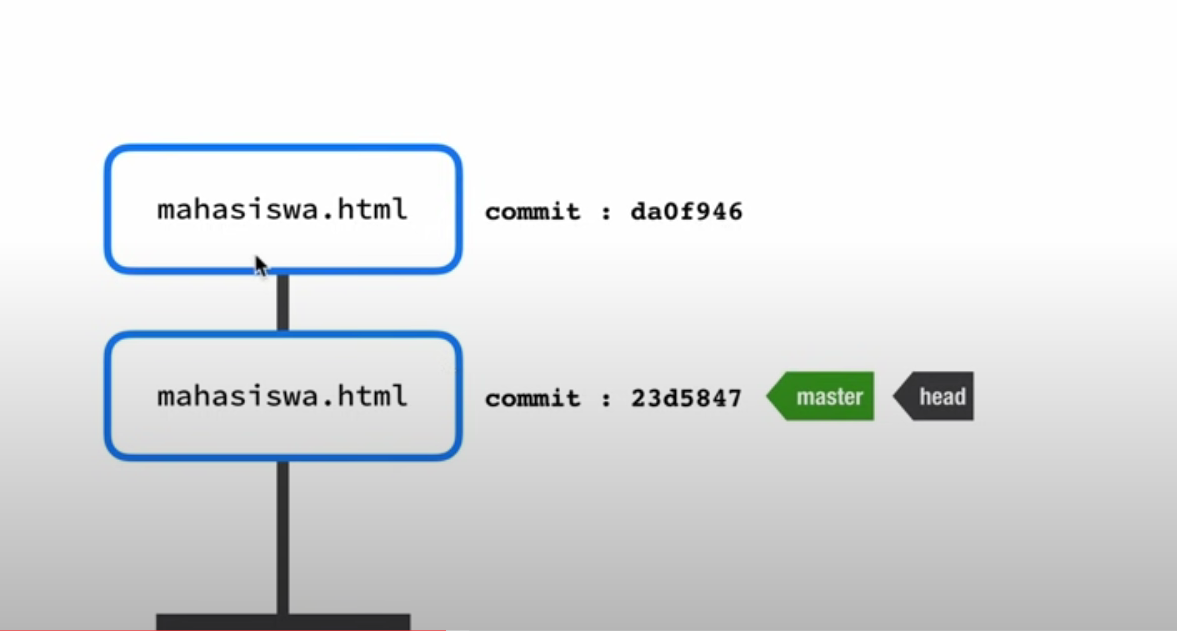


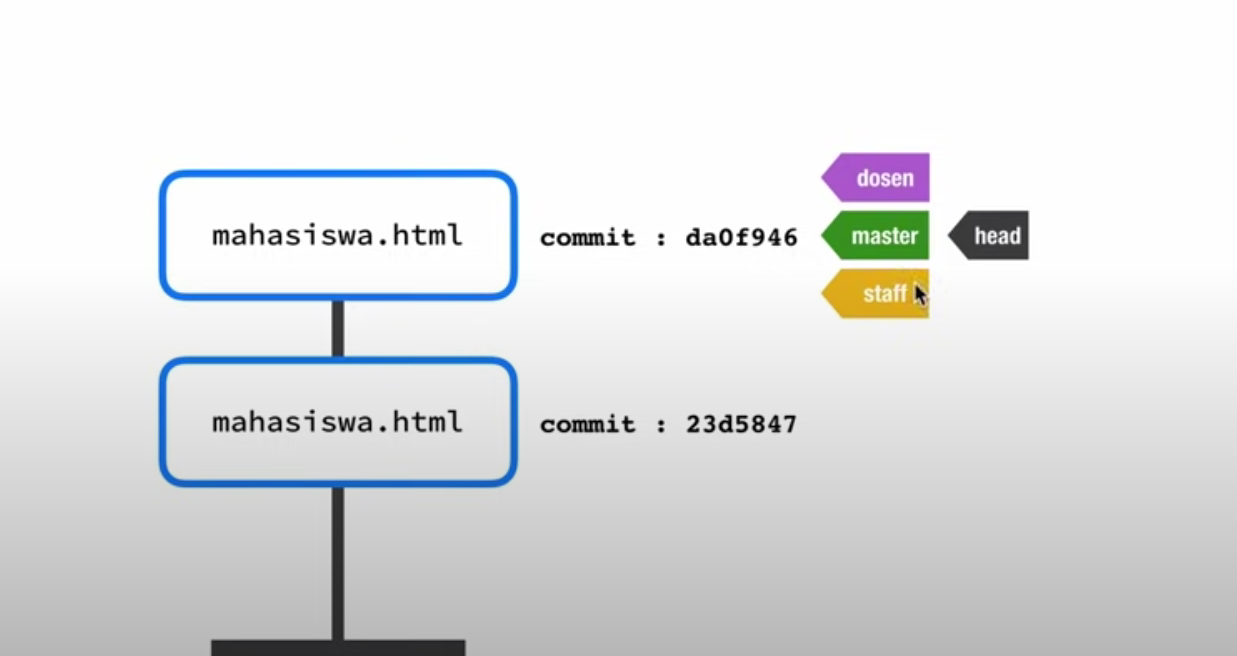
6.bisa pakai cara cepat : -> git commit -am”…..”



**Implementasi Branch :**

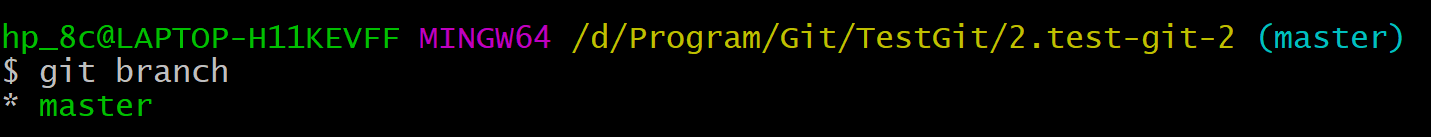
**->**Ketika kita melakukan commit,maka git akan mengarah ke branch **master**,dan untuk mengetahui kita dibranch mana maka paka konsep “**head”:**

****

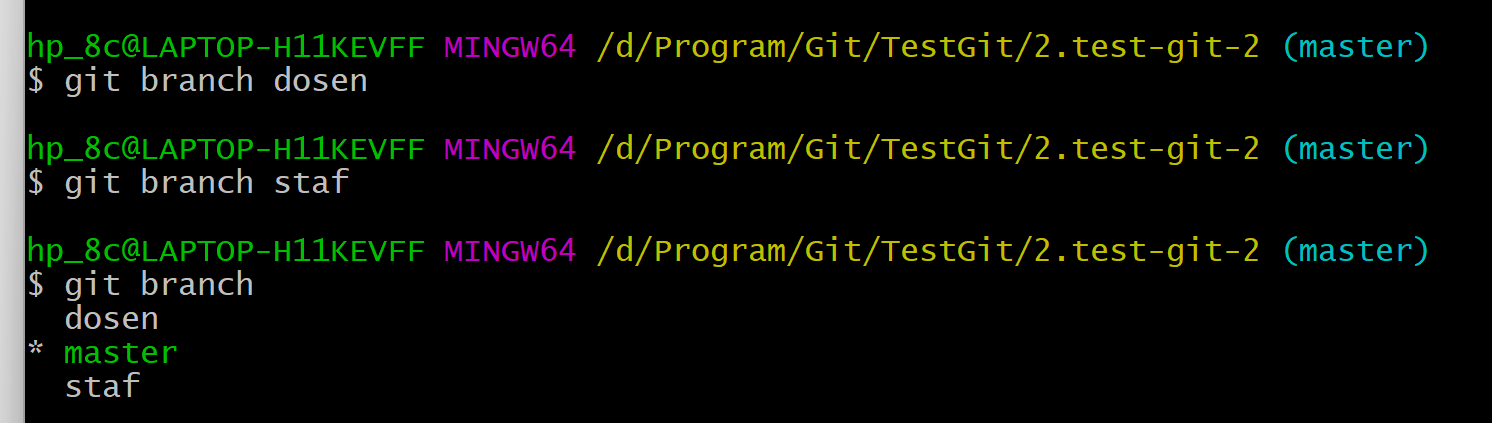
****

**Langkah2:**

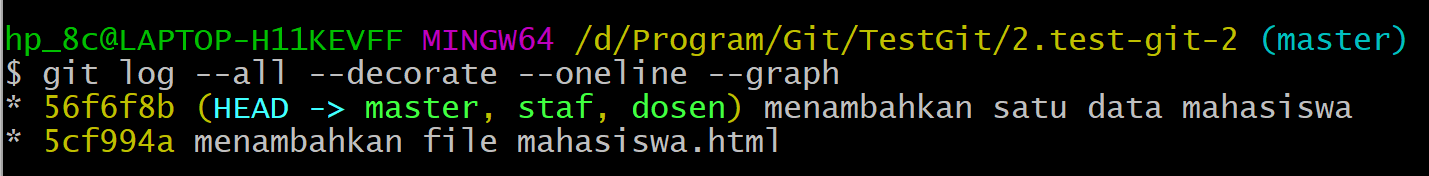
**1.**Menampilkan branch ->git branch



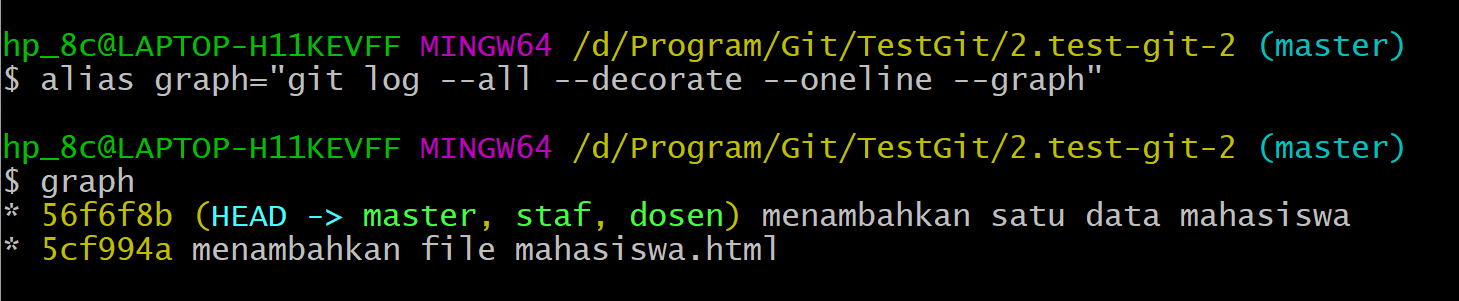
2.Cara buat branch



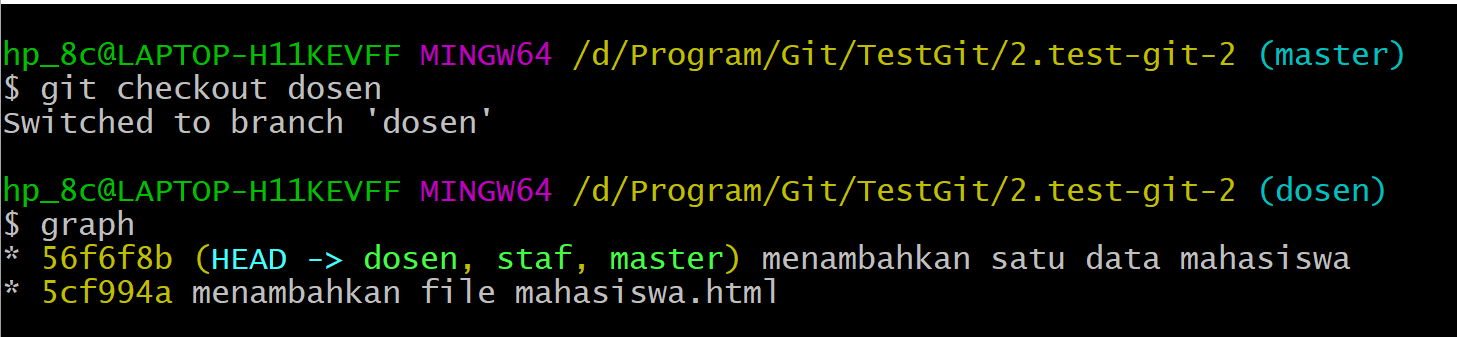
3.Cara melihat perjalanan branch dengan graf



4.Agar tidak ketik ulang,bikin alias untuk graf

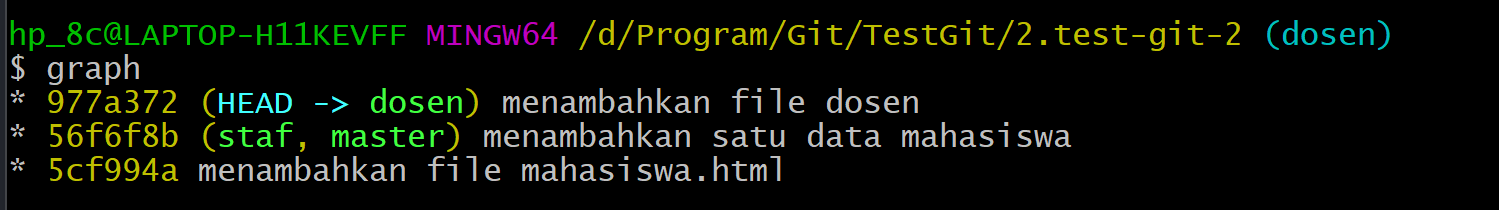


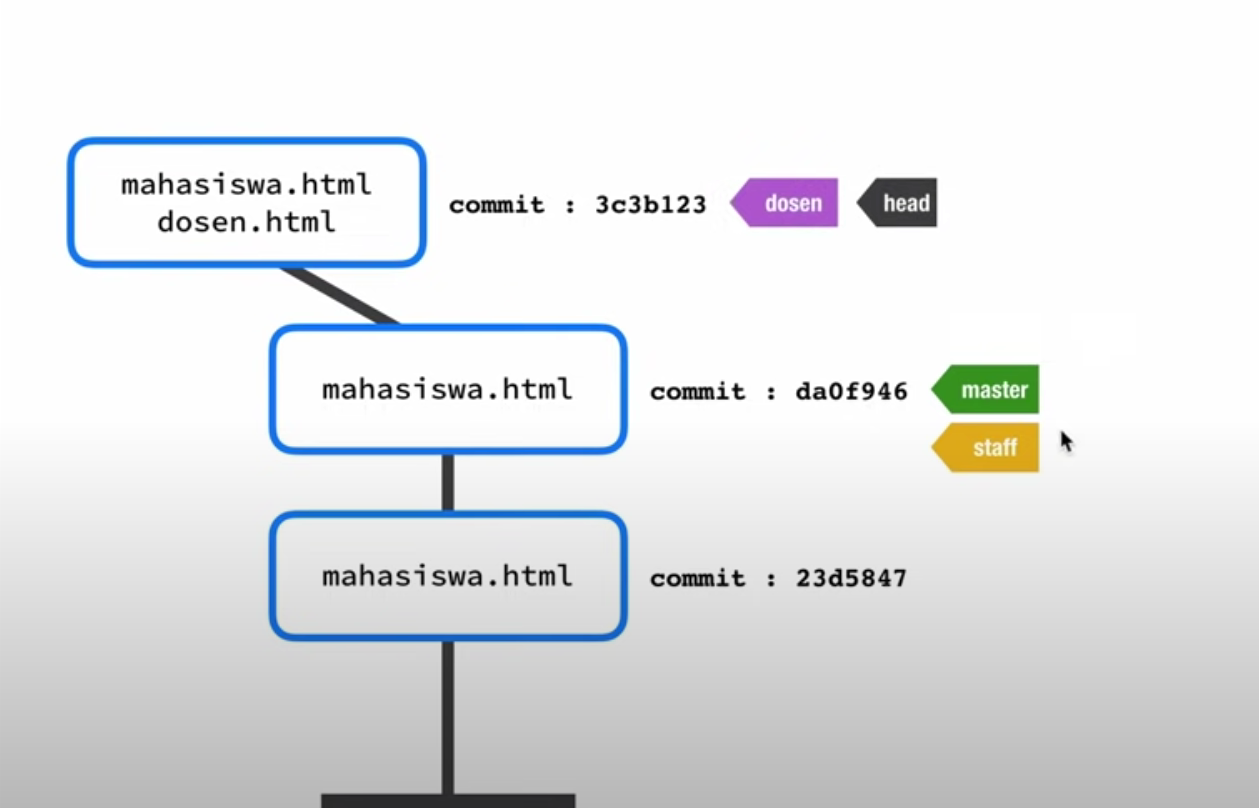
5.Cara pindah branch- > git checkout <nama\_branch> cth: git checkout dosen



6. lakukan commit pada data baru,maka akan membentuk branch seperti ini :

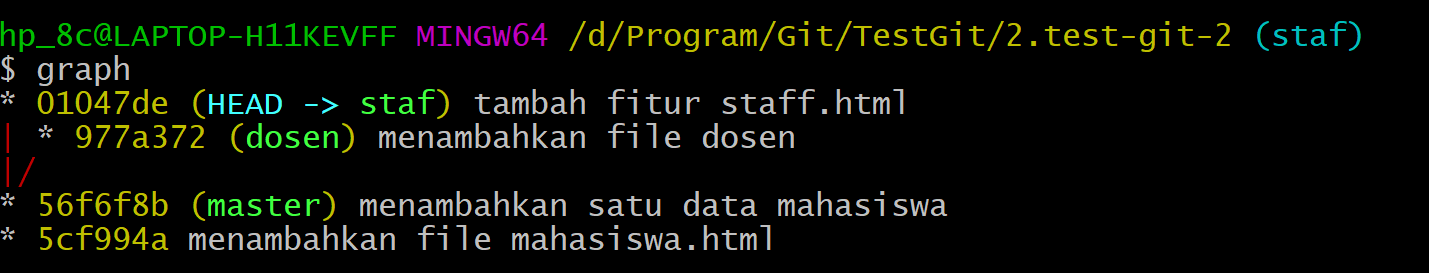
Note:dan yang hanya berubah di branch dosen

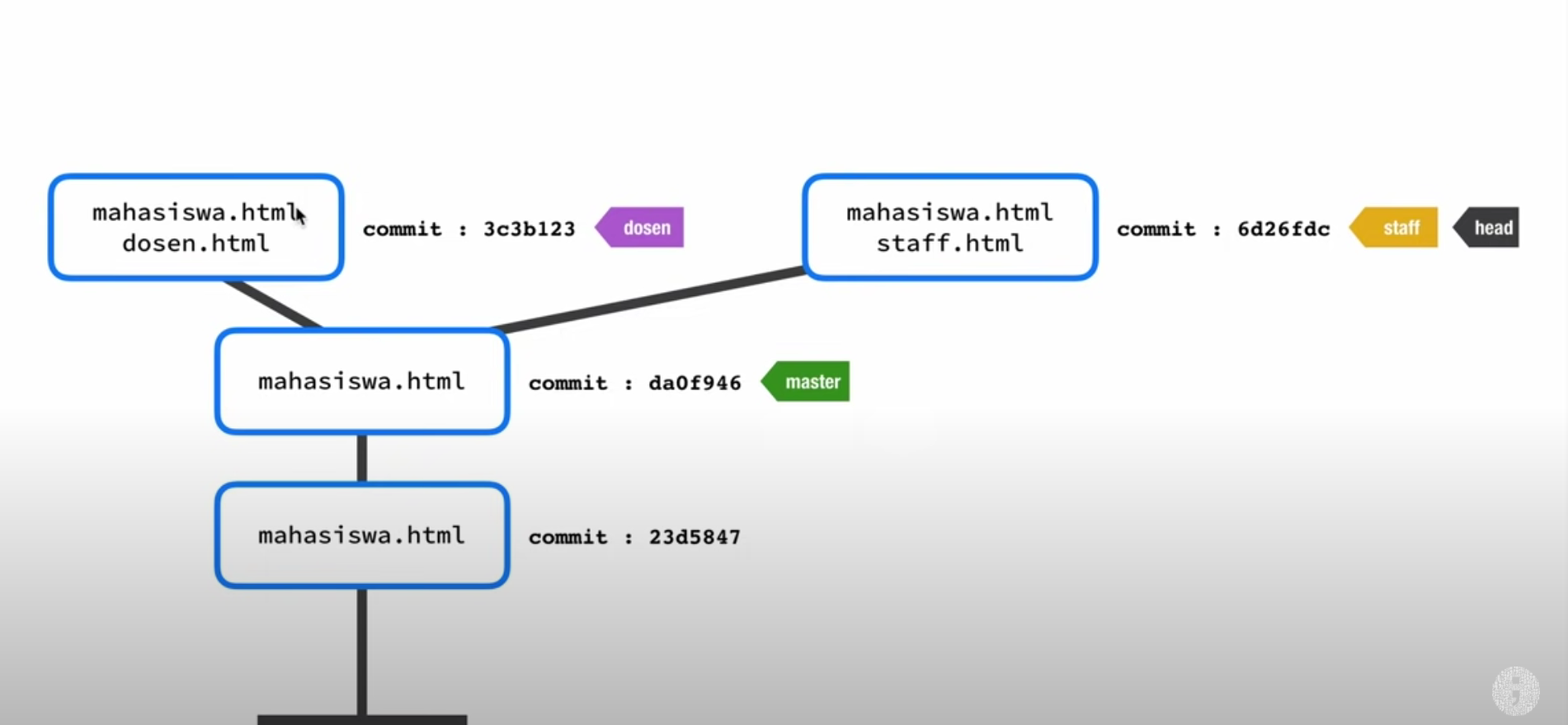




7.tambahkan juga branch pada staff ->git checkout staff

8.add data update baru,dengann git add . dan git commit -m”…” . Ilustrasinya



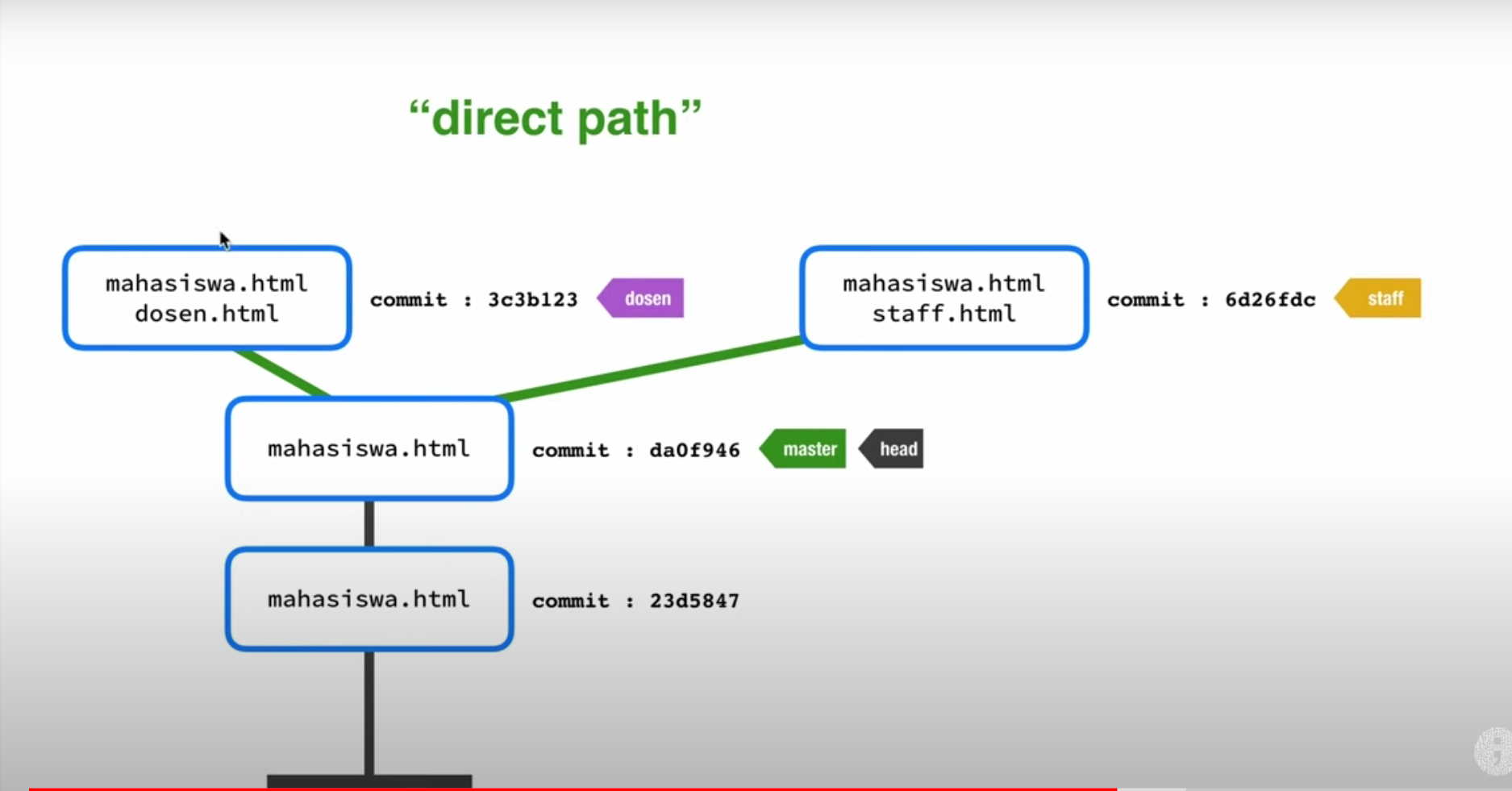


9.Merge antara branch

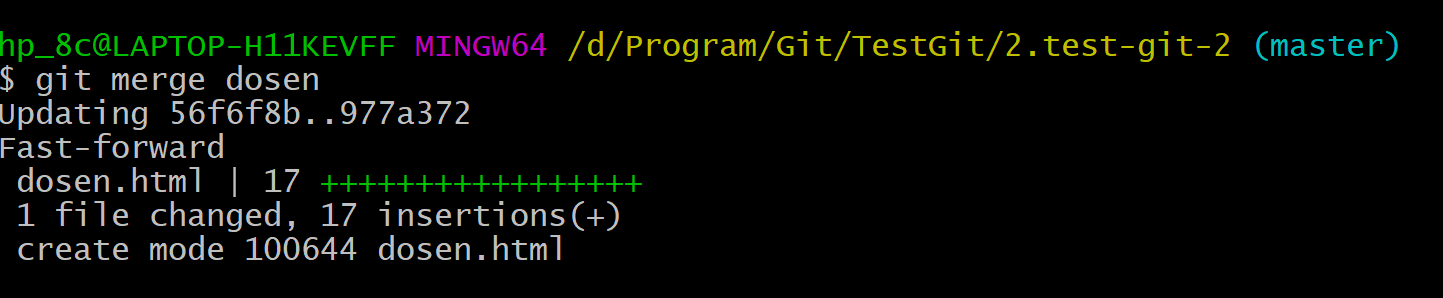
Ada dua jenis merge

-Fast Forward

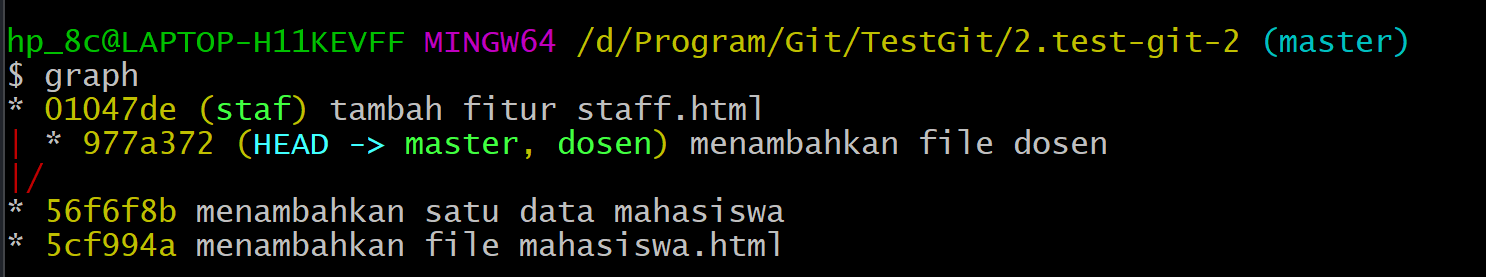
->Terjadi Ketika,branch yang kita gabungkan pada jalur langsung/direct



Ketika kita merge dosen.html,maka branch master pindah ke arah kiri,dan branch dosen masih ada

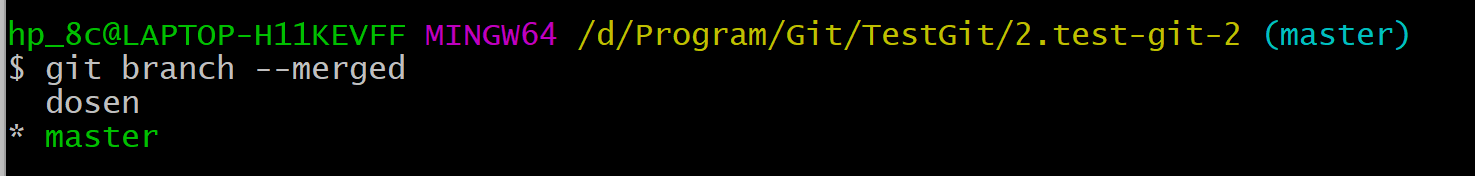




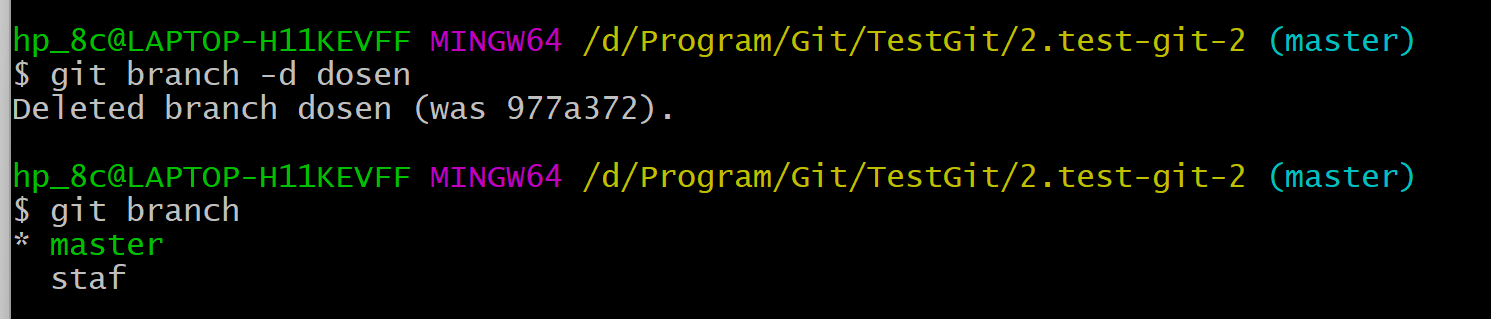


Karena berhasil,maka branch dosen tidak dibutuhkan lagi :

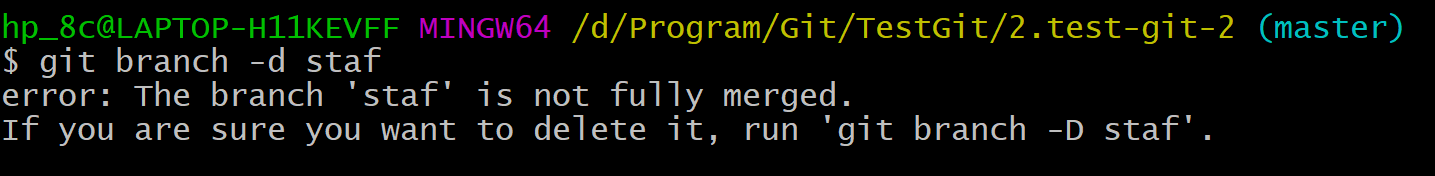
1.Cek yang sudah di merged



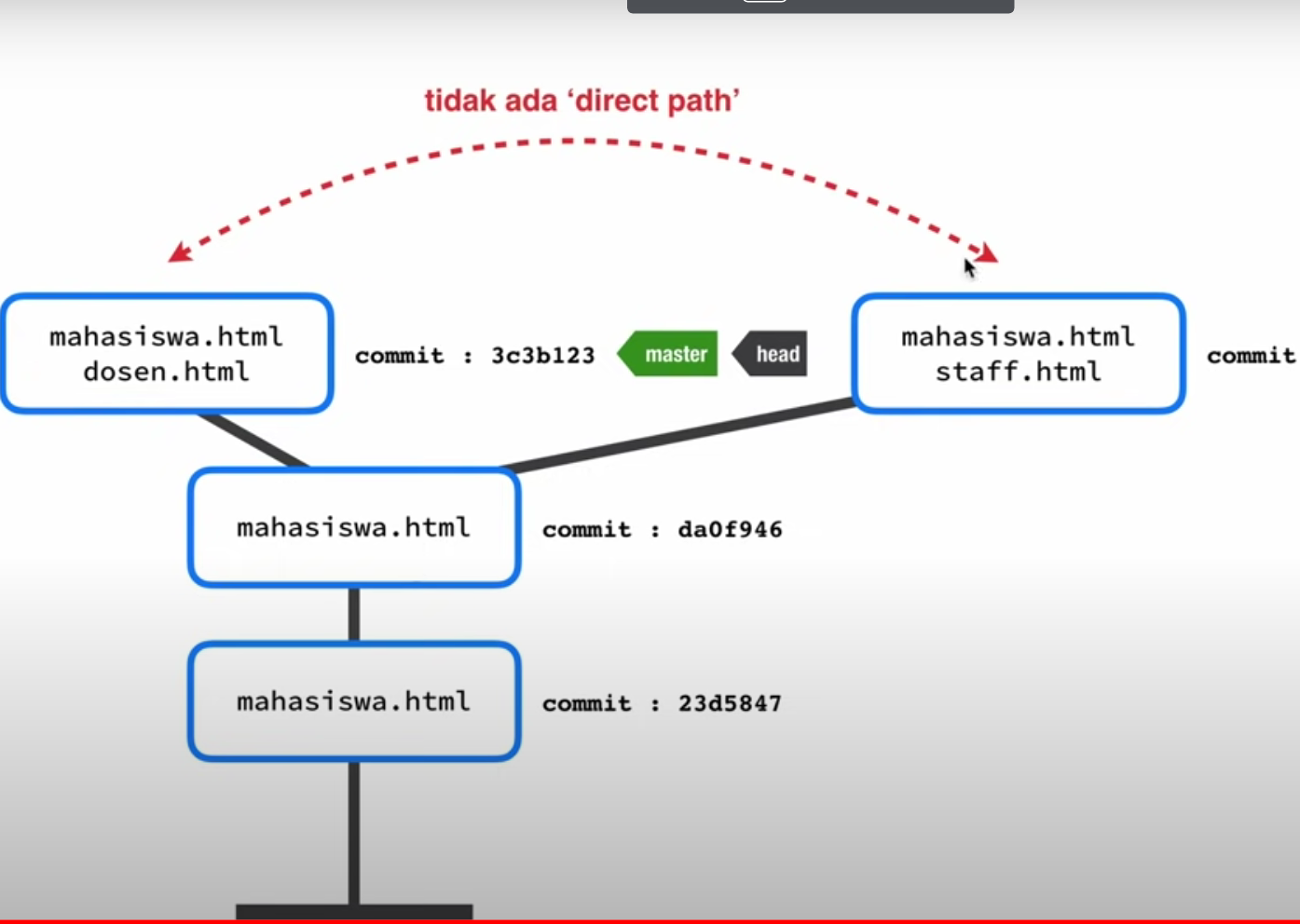
2.Delete



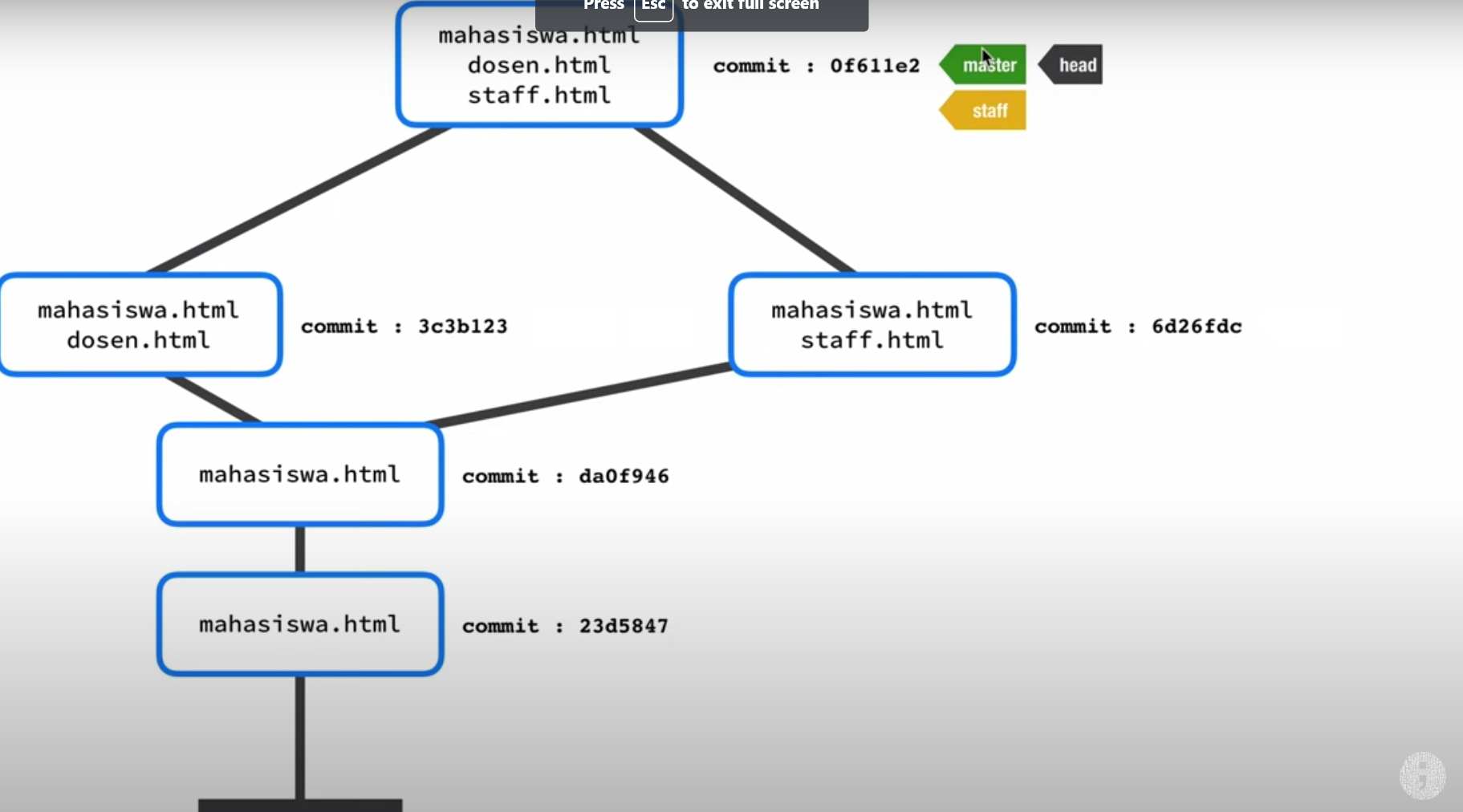
3.Jika kita delete fitur yang belum dimerged dia tidak bisa pakai syntax biasa :pakai git branch -D …



-Three-way Merge

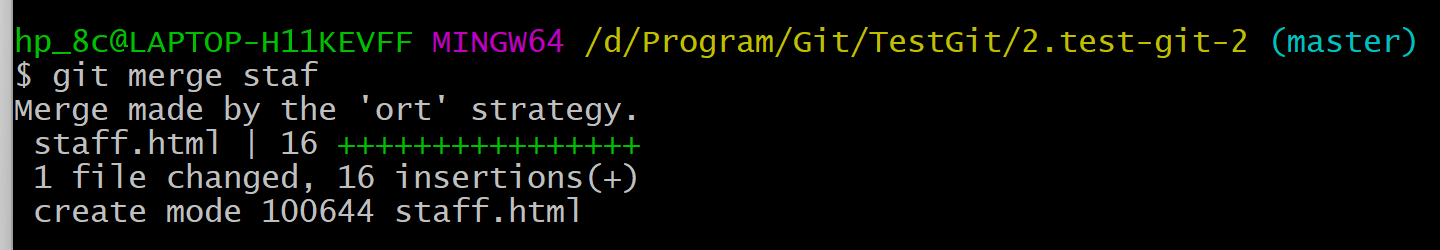


Maka akan seperti ini :

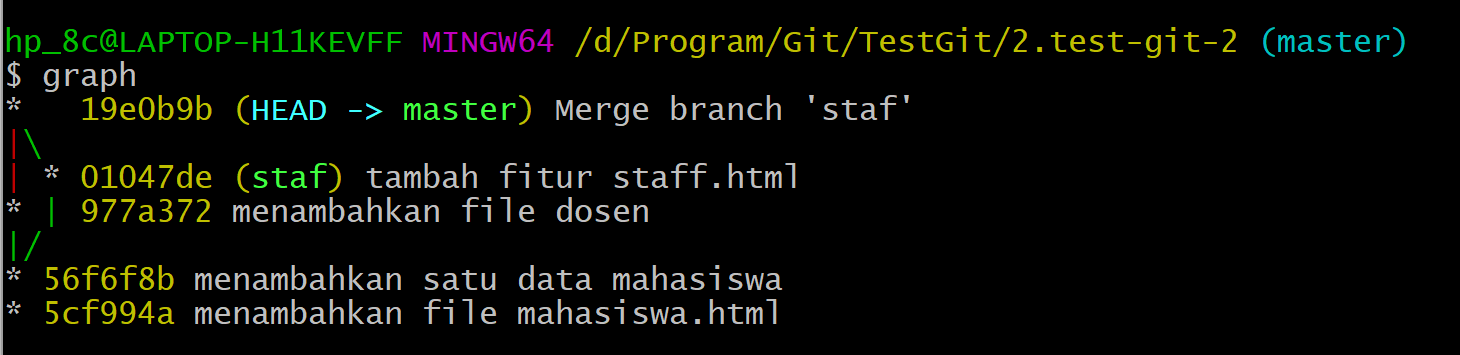


Langkah2 merge:

1.git merge staff…..



2.Dan lakukan :wq!

->

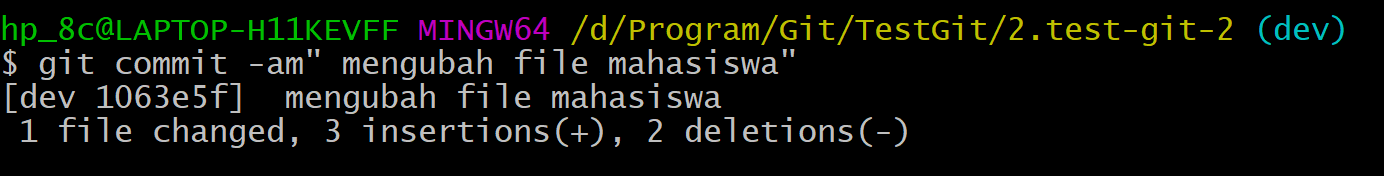
**GIT Merge Conflict**

**1.**tambahkan branch baru ->git branch <nama\_branch>

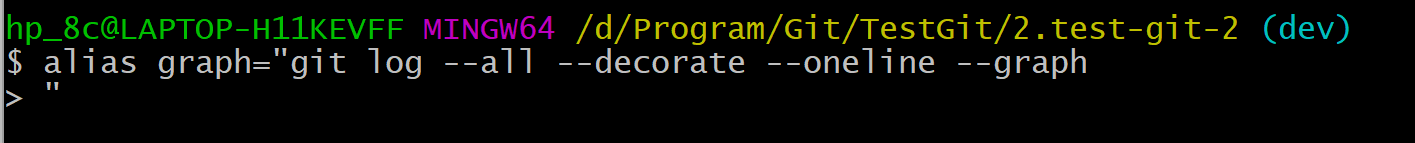
2.Jika ingin pindah ke branch baru -> git checkout <nama\_branch>

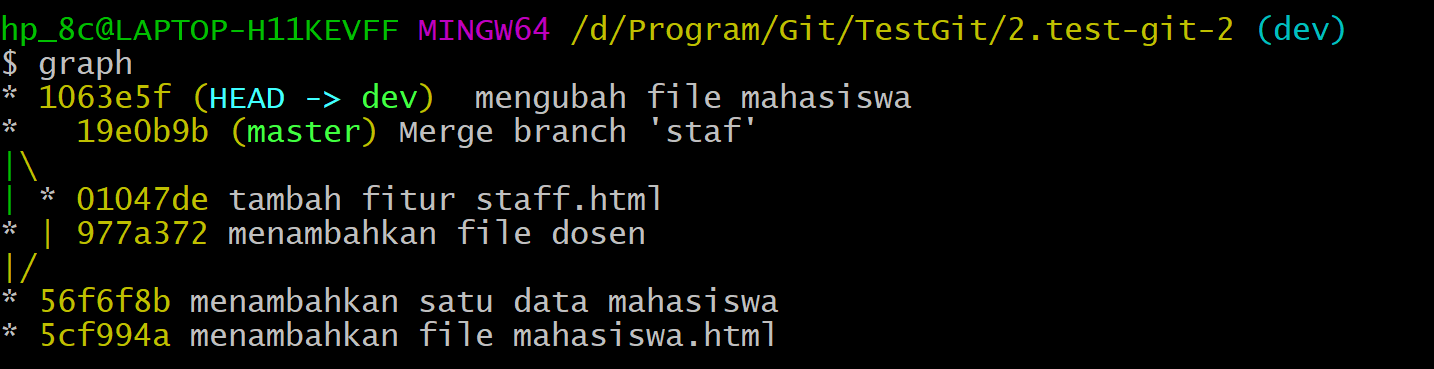
3.lakukan git add . , dan git commit -m”…” ->ini jika untrack

4.Lakukan git commit -am”…” -> jika modified

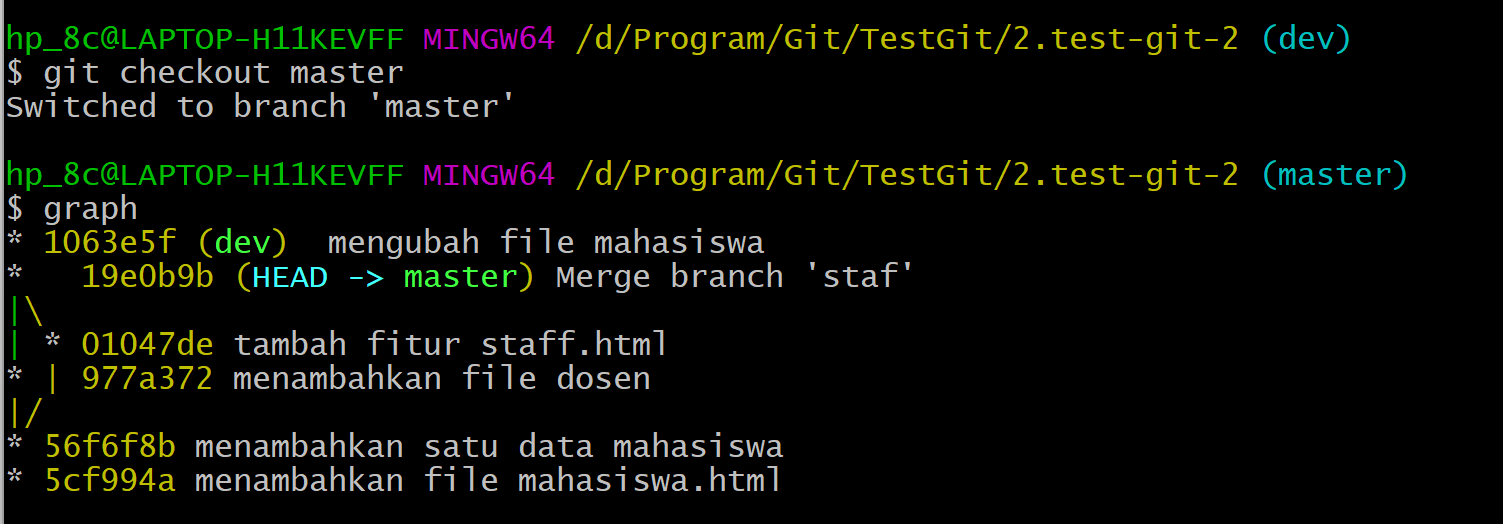


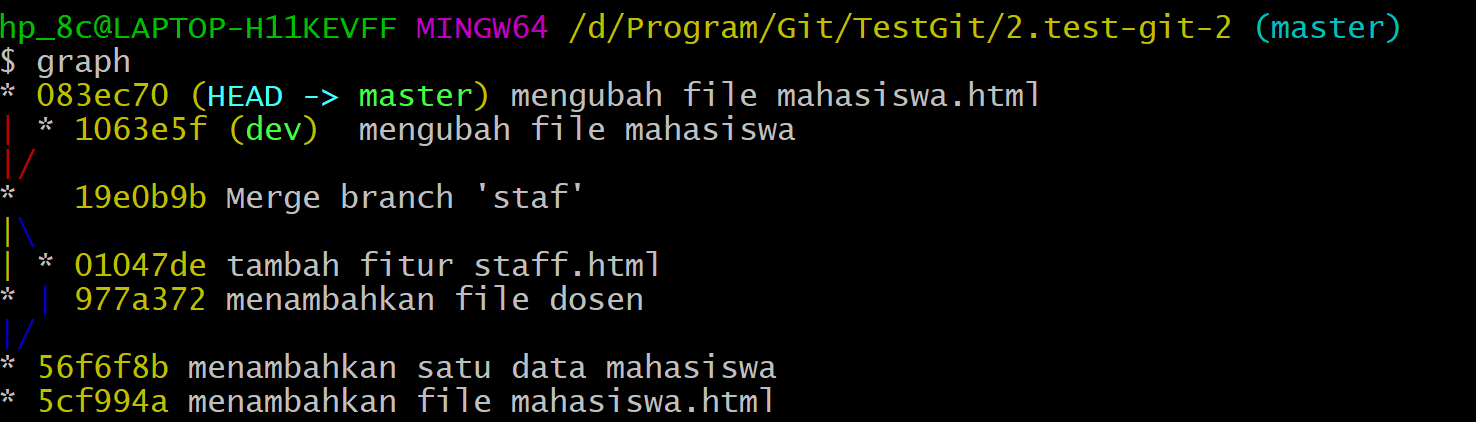
5.Bikin alias graph





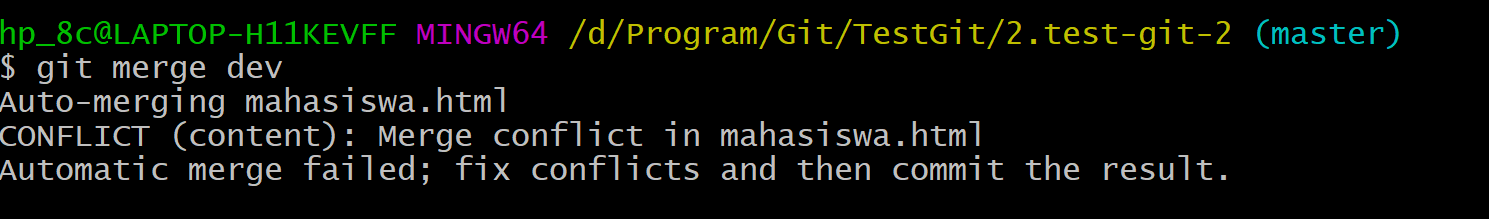
6.Kembali ke branch master,dan lakukan perubahan yang berbeda dengan dev



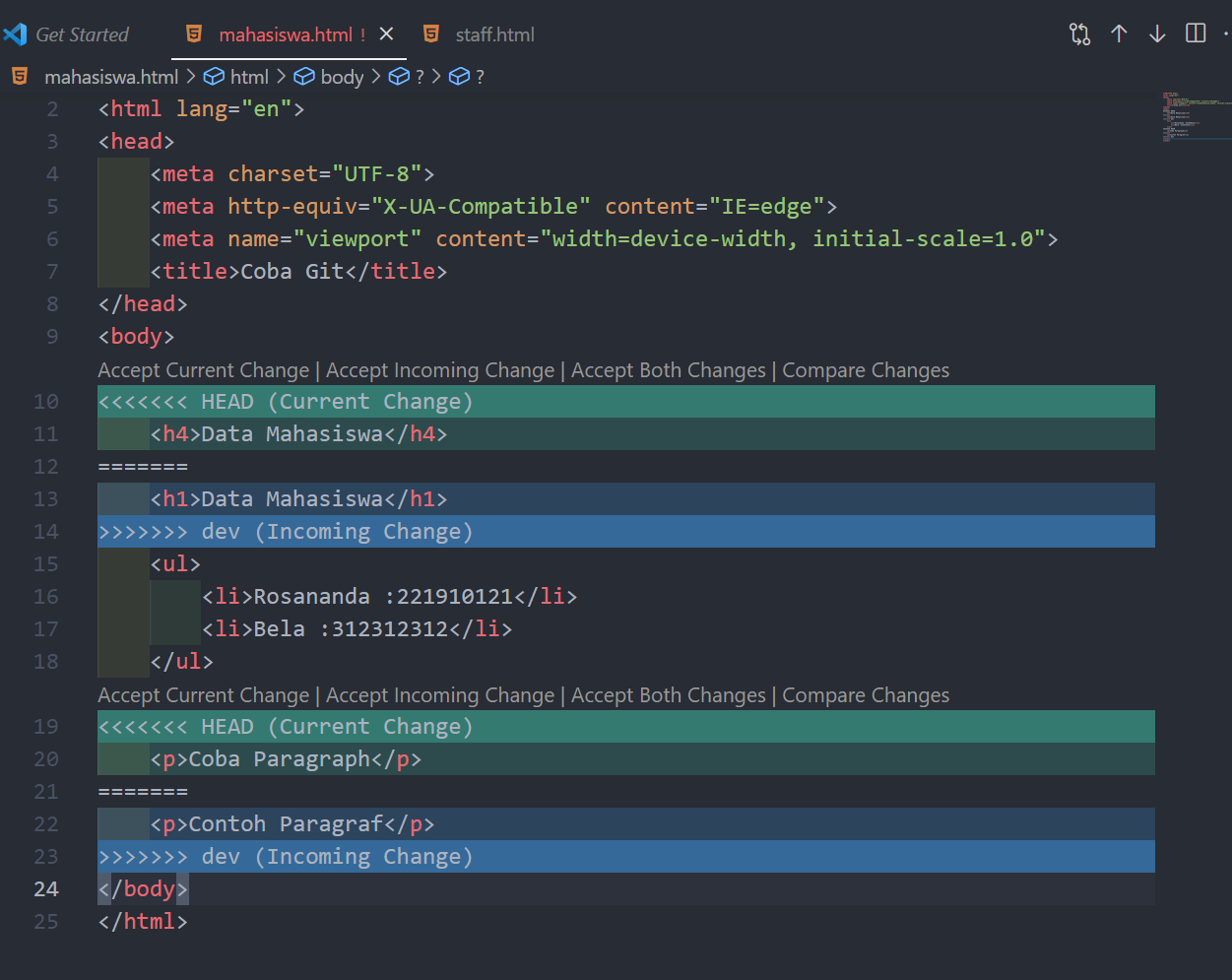


Note : ini adalah three way merge

7.Merge dilakukan di master

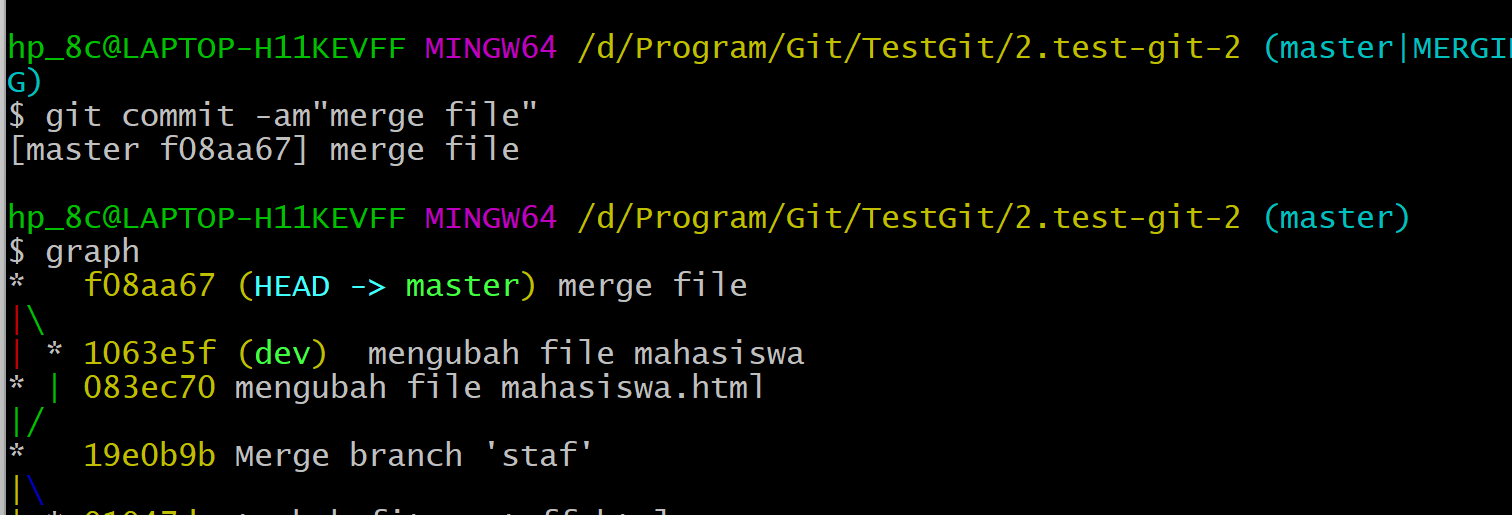


Maka akan terjadi conflct ,dan code editor akan terbuka:



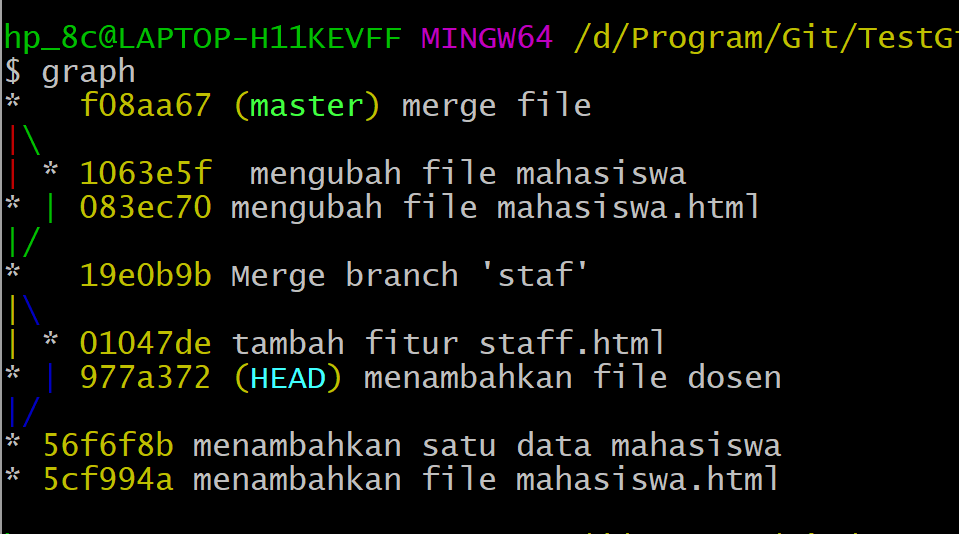
Dan pilih yang mana yang kita inginkan

8.Lakukan merge dengan -> git commit -am”merged file”



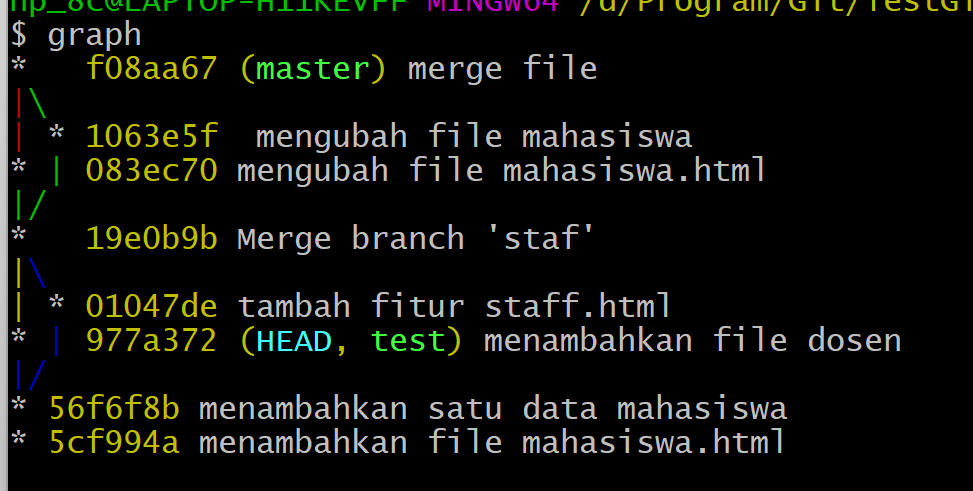
**Bagaimana Jika kita ingin Kembali ke keadaan awal?**

**1.**git checkout <5code commit>

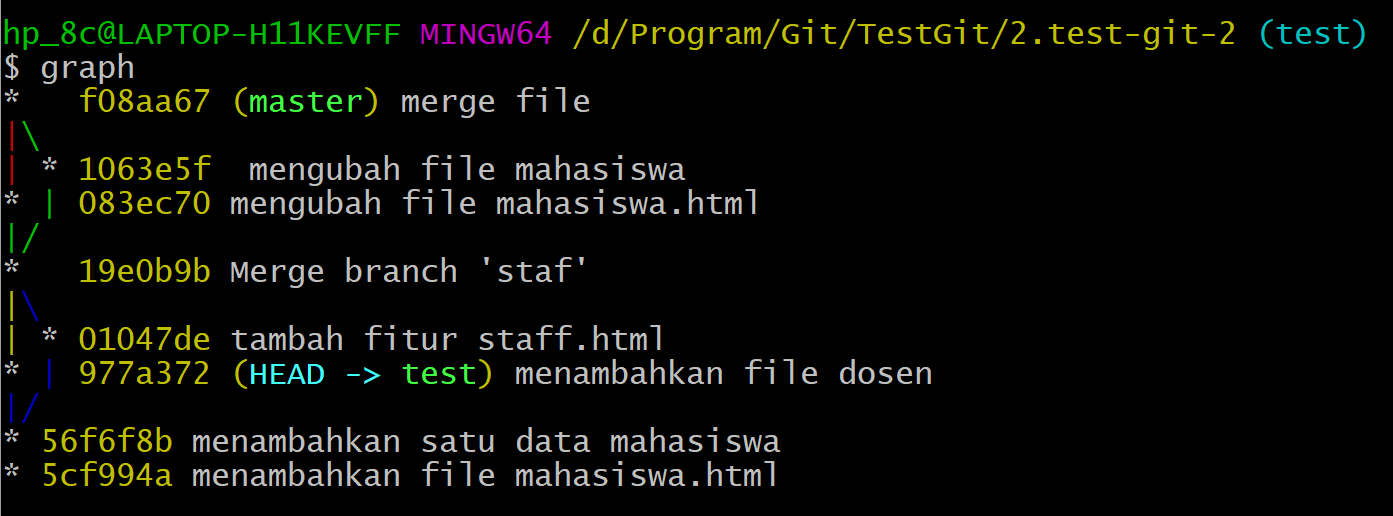


Maka Kita akan berada di file dosen

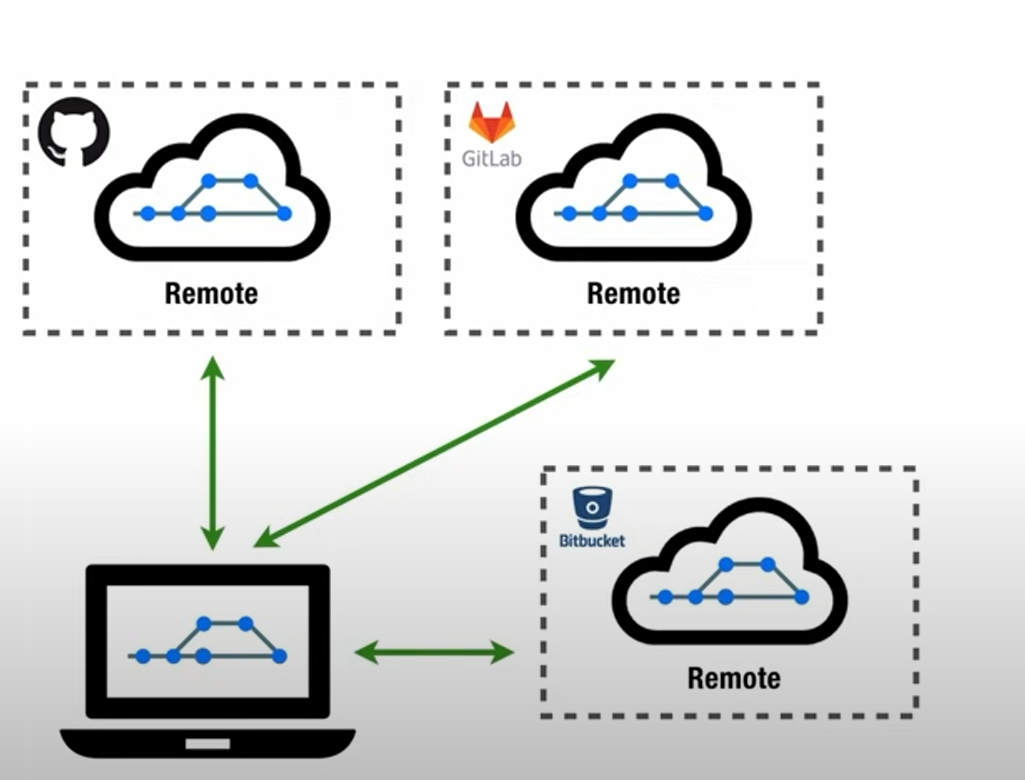
->buat branch baru



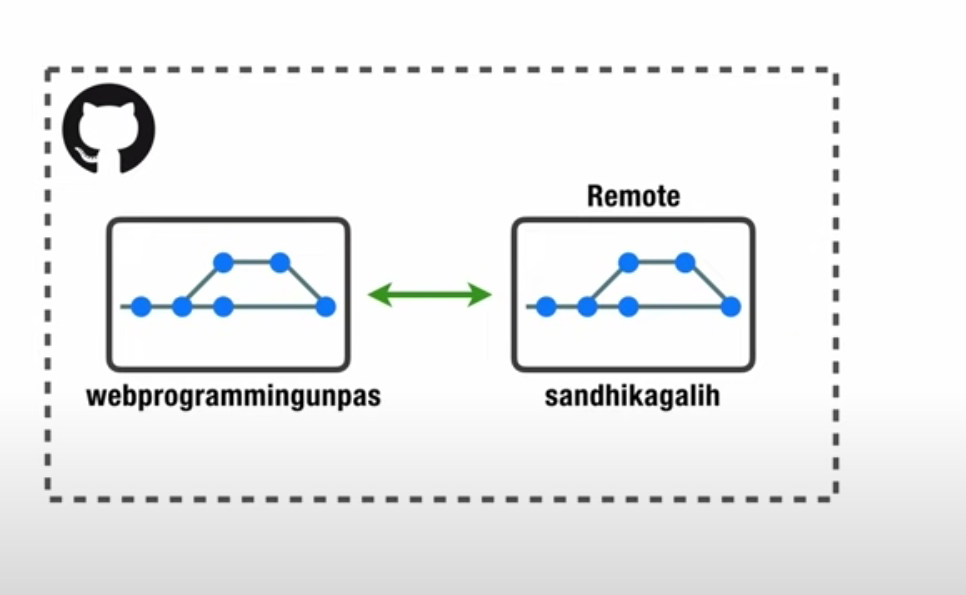
->agar normal,tambahkan : git checkout ….



**GIT REMOTE**



Kita nge forke/copy repo seseorang

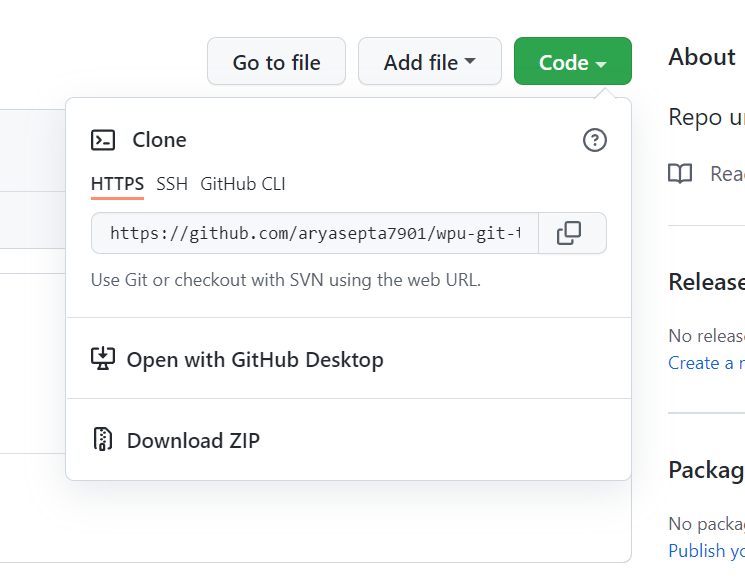


**Remote**: Duplikat dari repo yang kita punya

Langkah2 :

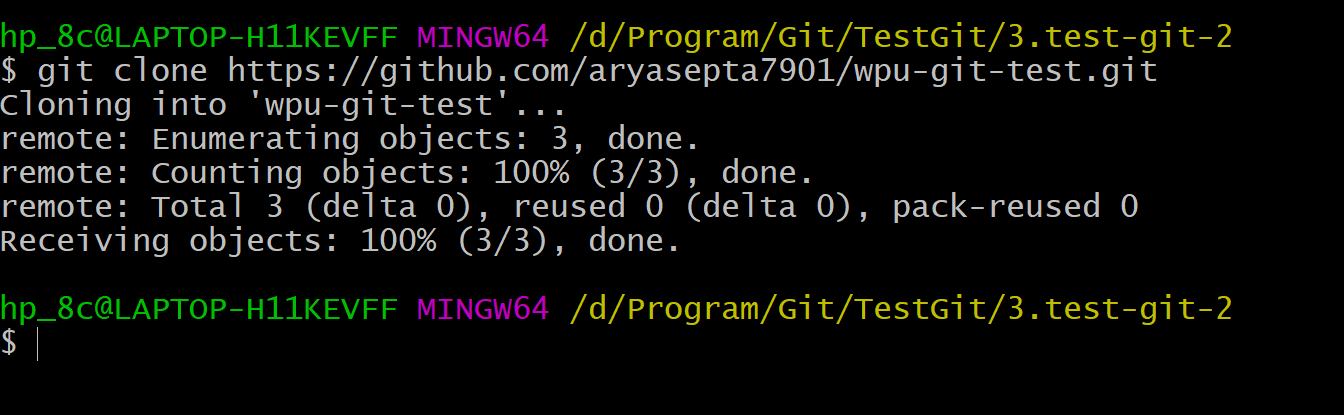
1.Buat Repo baru di github

2.Klik Code u/ melakukan clone dari github



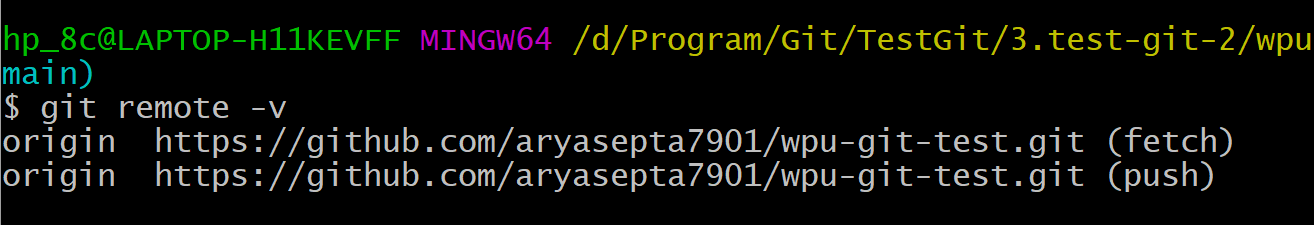
Ada 2 cara :

**1.Pakai HTTPS**

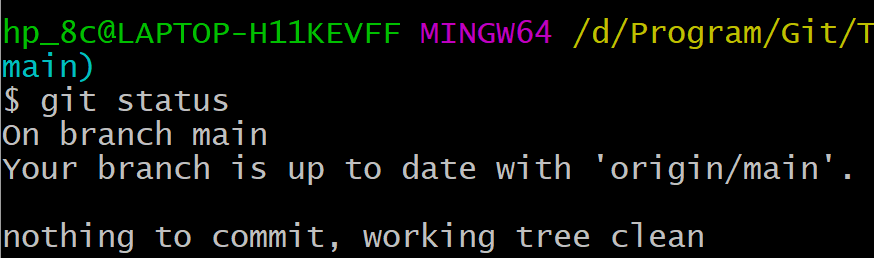


Cek nama default remote nya



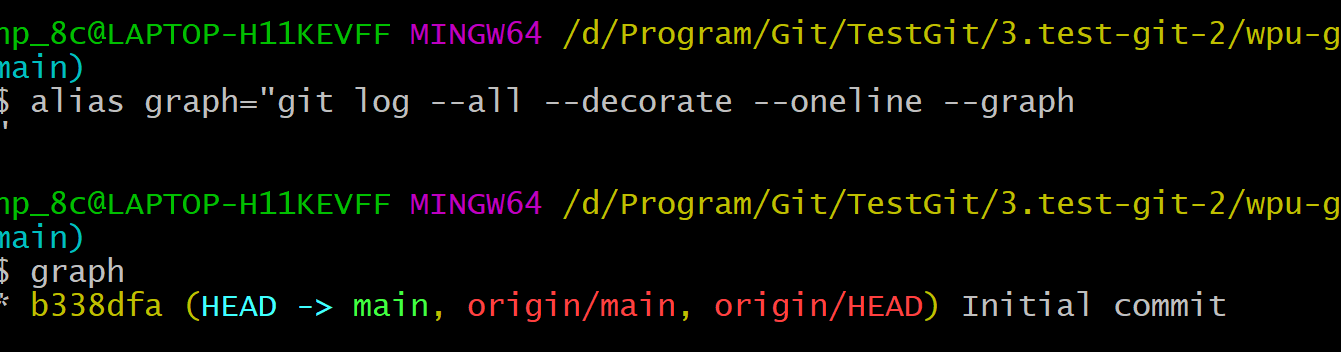


Cek status :

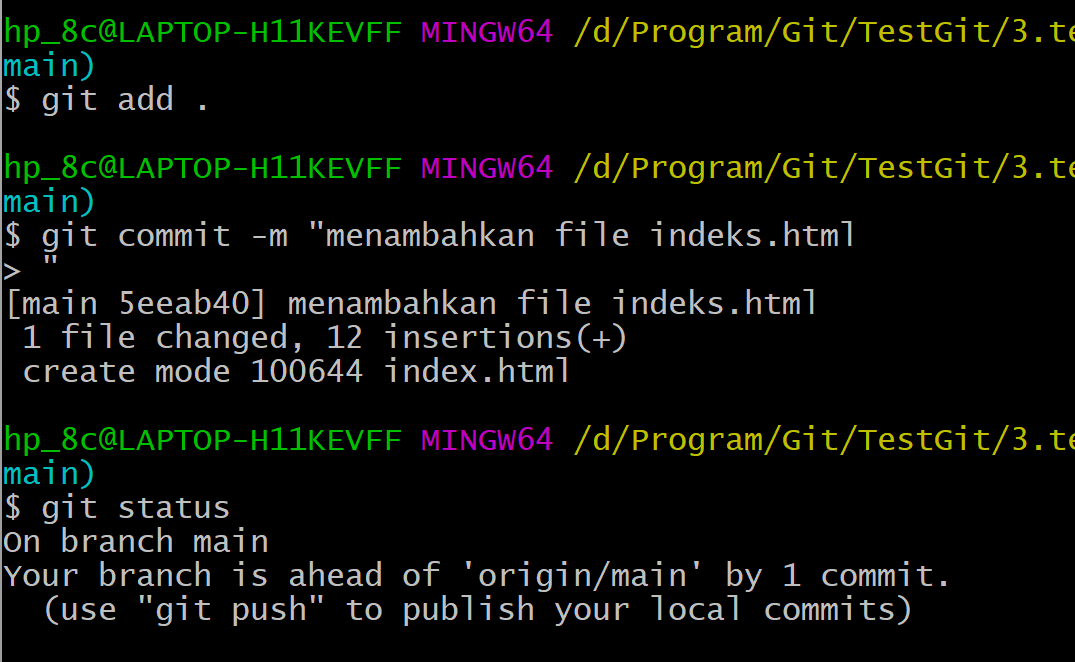


Note: Branch pada web sama local sudah up to date

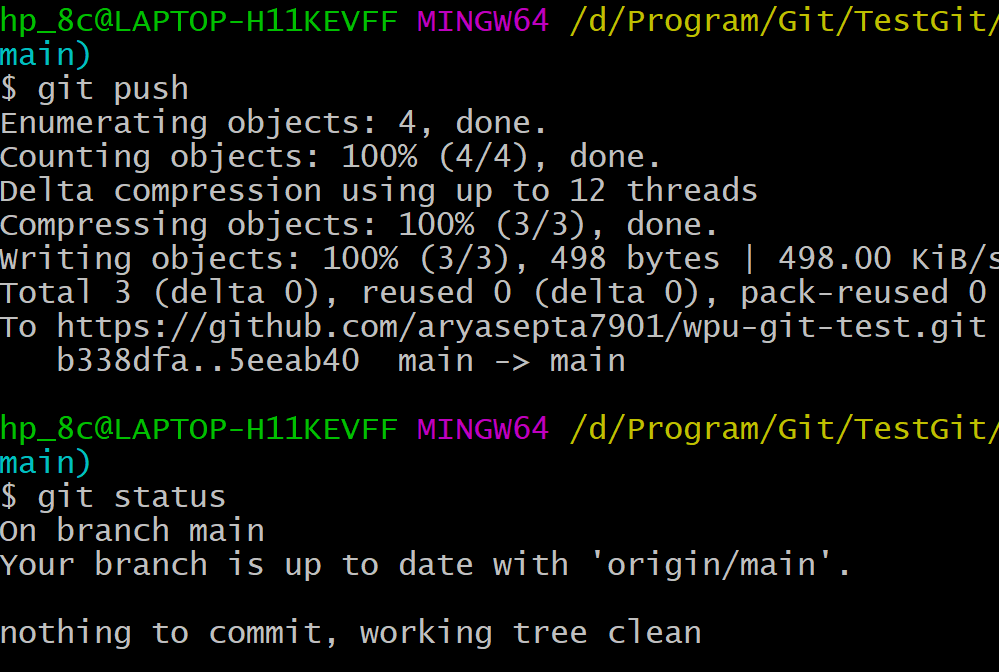
Buat graph dengan inisial



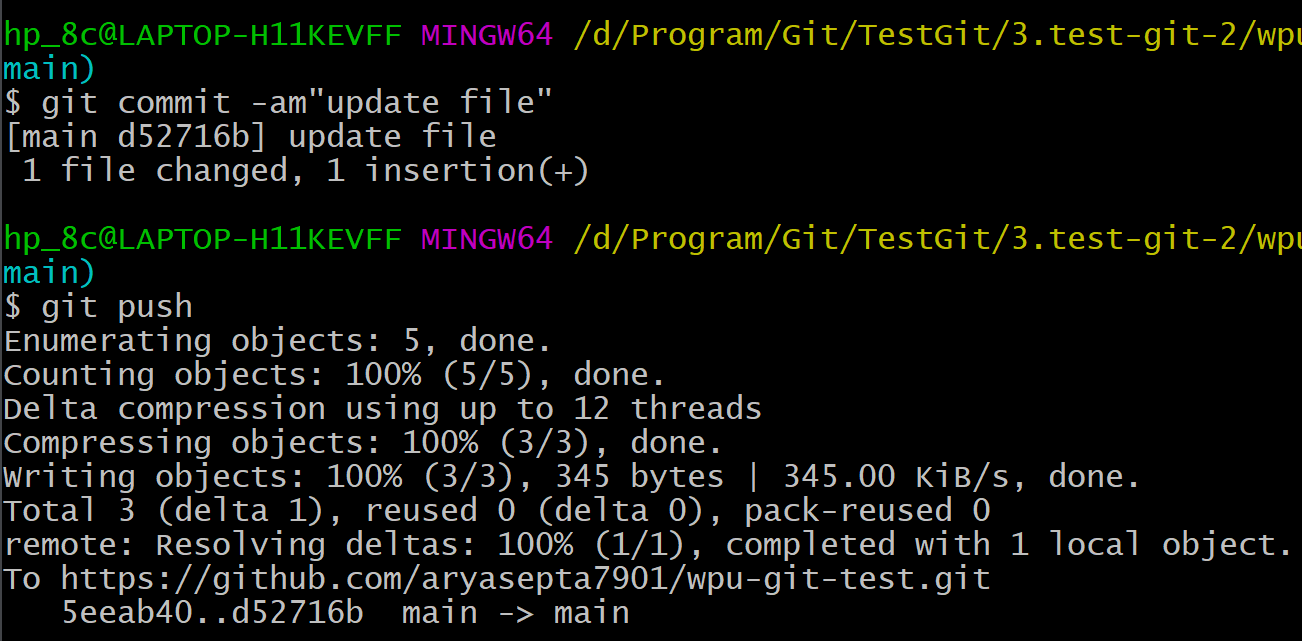
Lakukan penambahan file dan add file tersebut dan commit



Kita lakukan Push agar web sm local sama



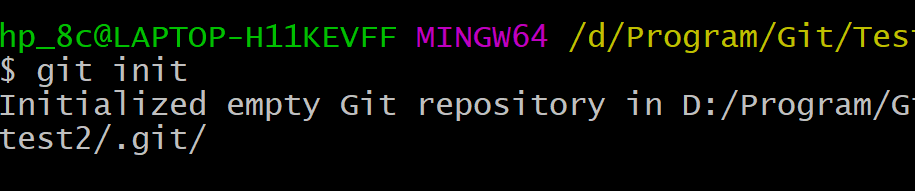
Jika file sudah di modified kita bisa langsung pakai code git commit -am” ..” dan lakukan git push

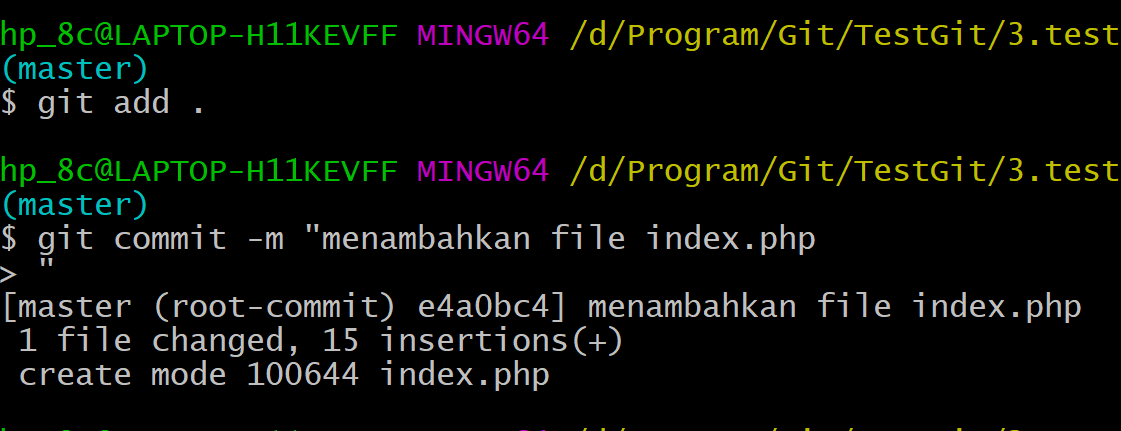


2.Pakai cara SSH

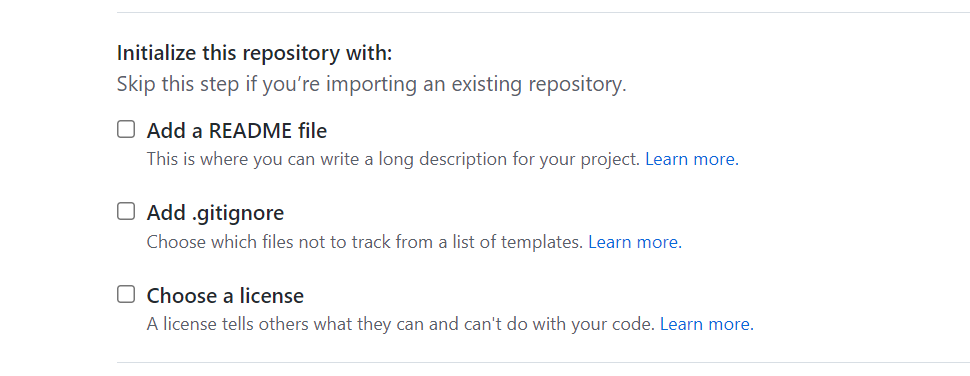
Langkah2 :

1.Buat folder baru dan jadikan repo git



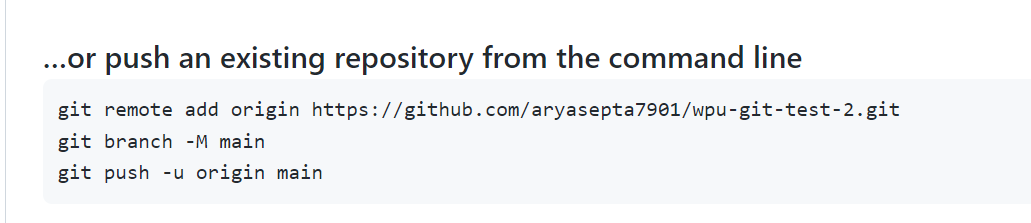
2. add file dan lakukan commit

3.Bikin Repo di github



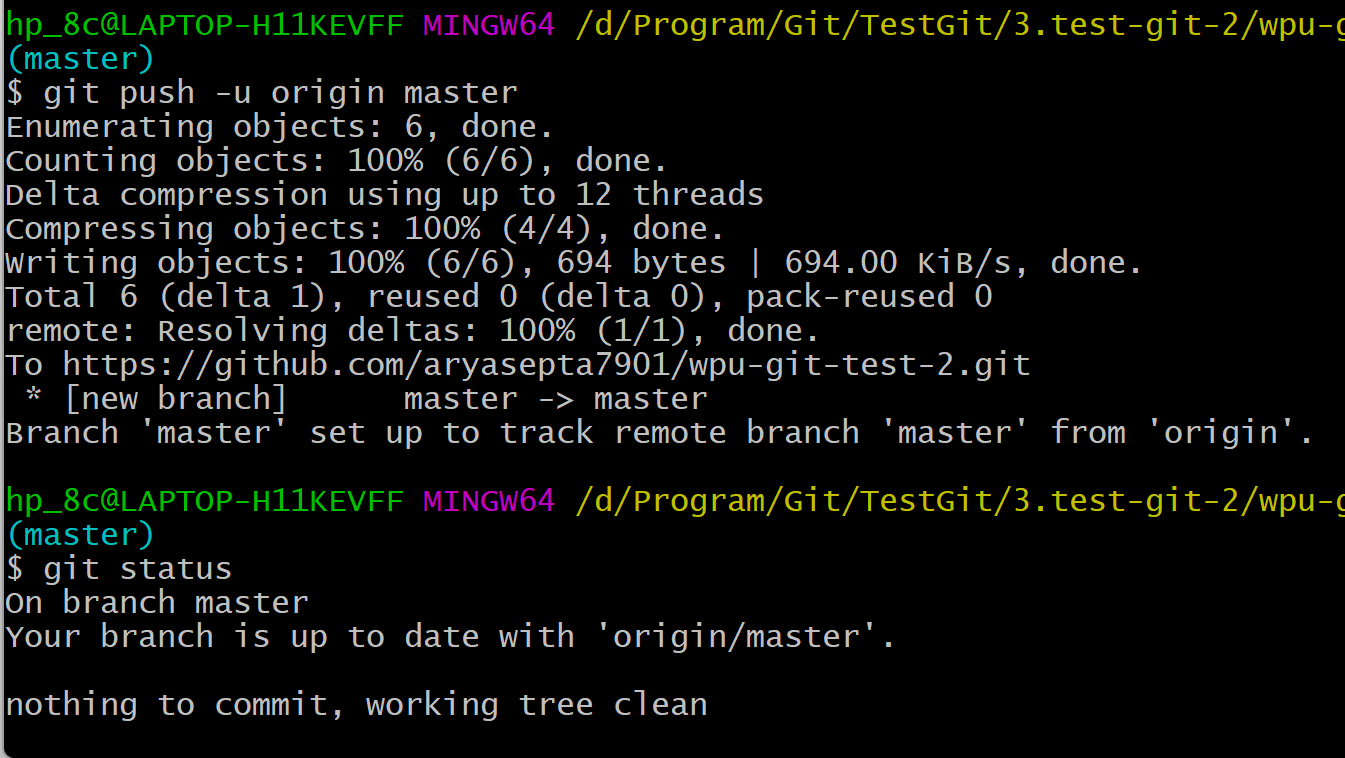
Jangan add a readme file..

4.Lakukan Ini



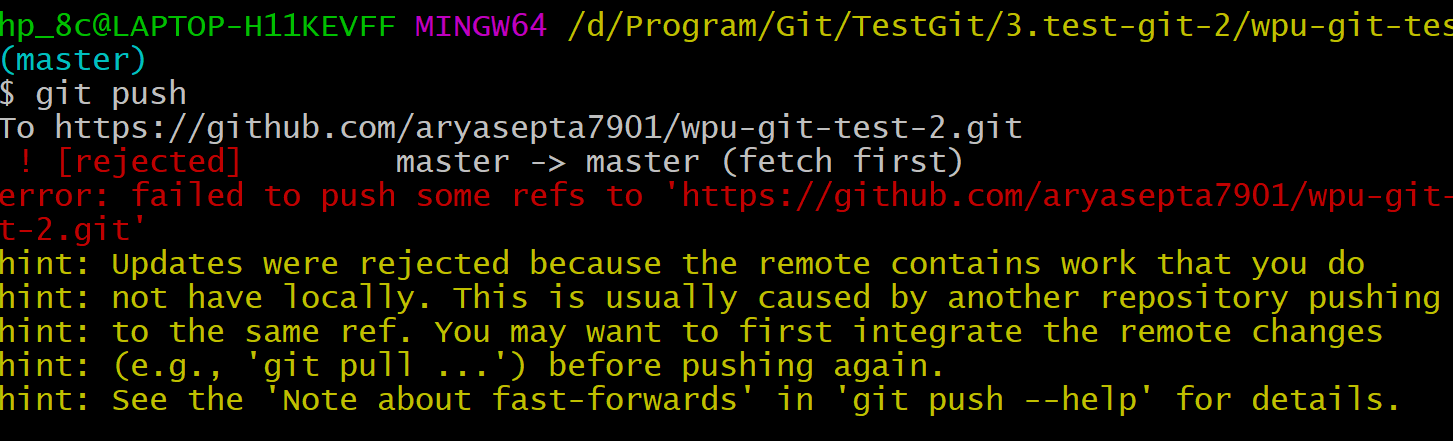


5.Lakukan Push



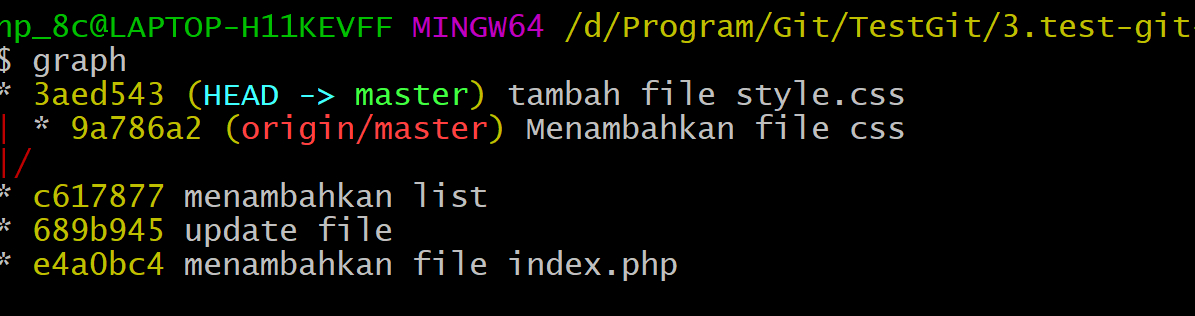
Note:pakai syntax git push -u origin master,jika kita ubah file lagi setelah di push cukup git push saja

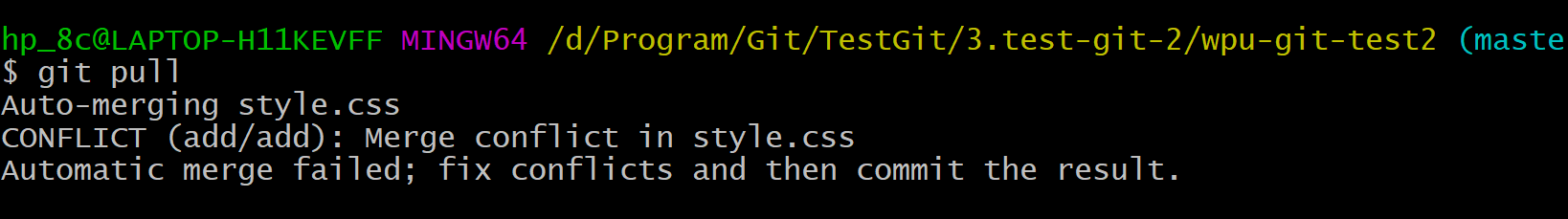
Terkadang Ketika kita push ditolak ,karena terjadi conflict file

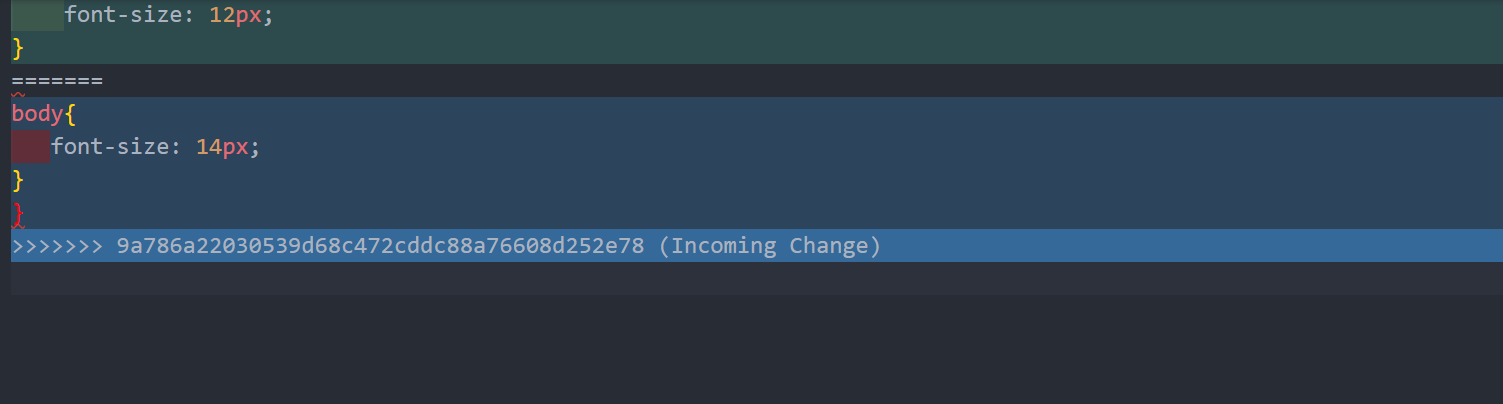


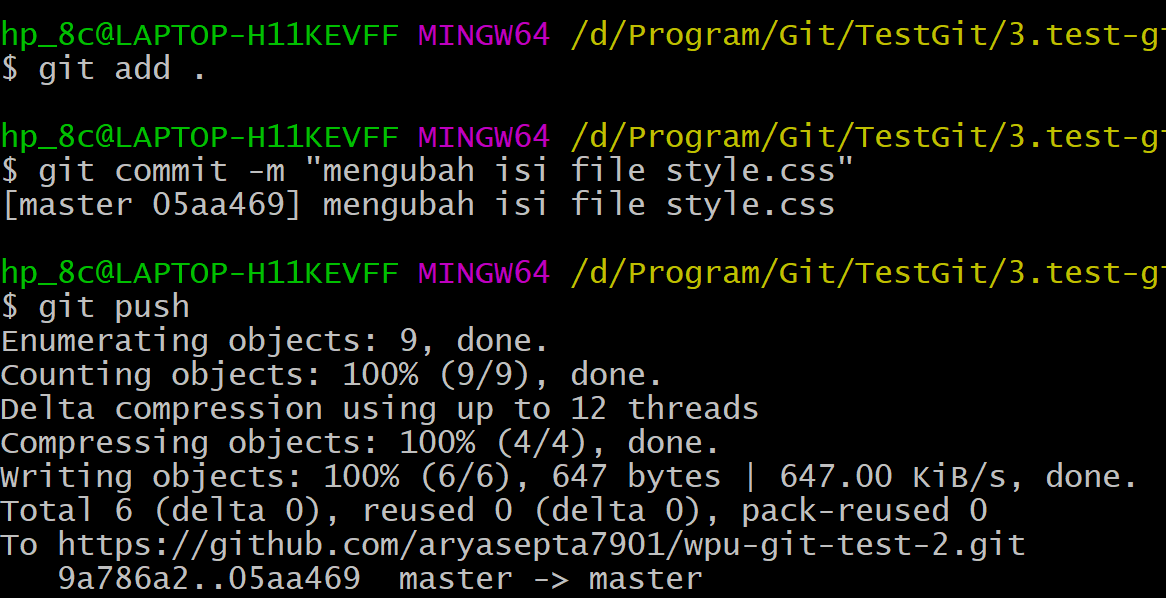
Solusi :

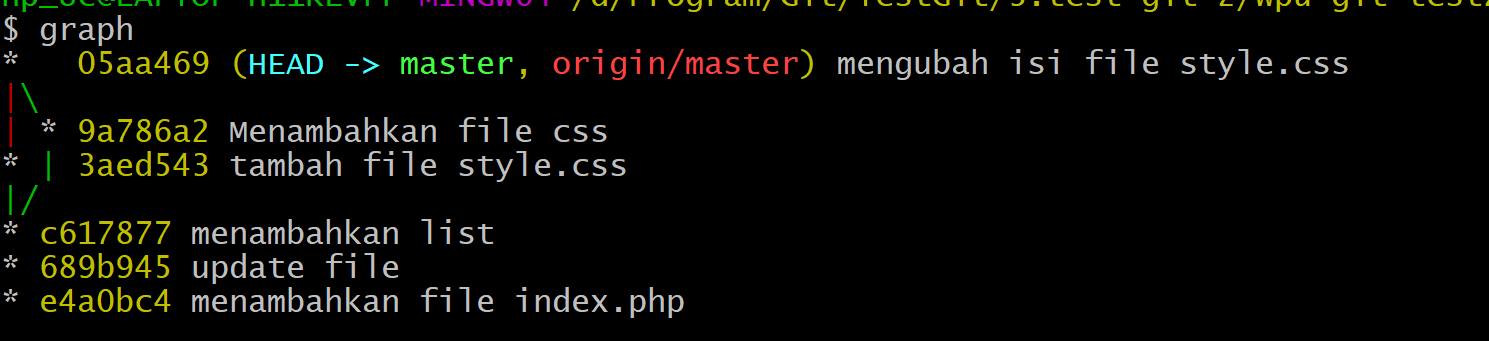




** us**

****

****

****