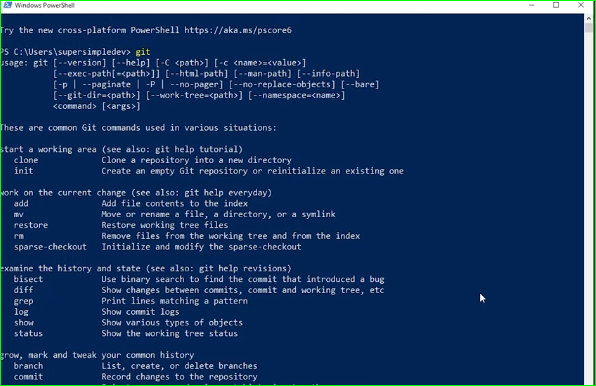
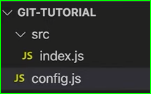
1. Install Git 🡺 <https://git-scm.com/downloads> and another cheatsheet for all command Git/Github <https://supersimpledev.github.io/references/git-github-reference.pdf>
2. Buat folder(destinasi) Coba taruh di desktop
3. Windows instalasion  
   kita perlu check di windows powershell kalo Git terisntall dengan baik  
   buka powershell 🡺 ketik git 🡺 enter  
     
   hasilnya  
   
4. Setup Git

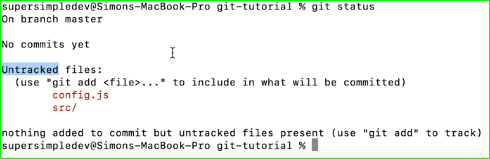
Setup Git means kita setup directory yg ingin kita tuju

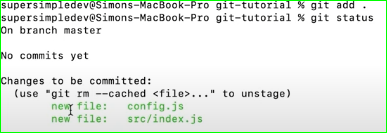
folder

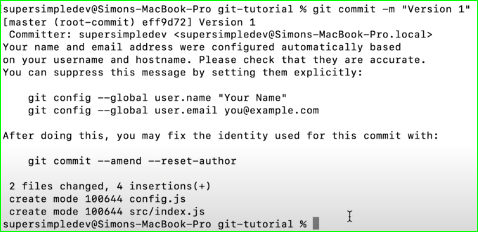
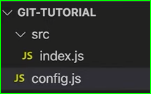


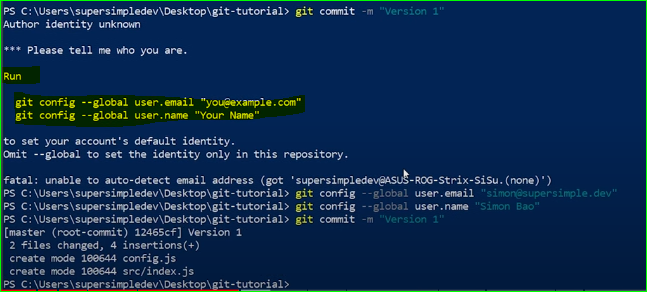
1. kemudian pada powers shell 🡺 ketik ls   
    ~ = Home folder/Root folder  
   ls untuk list all files and foilder pada folder sekarang  
     
   lalu tugas kita unutk **mengubah folder tujuan** kita
2. pada powershell gunakan | cd ~\dekstop\git-tutorial(targert foldernya)  
     
   CD means Change directory  
     
     
     
   masuk ke folder git-tutorial  
     
     
   done

Untuk catatn ketika masuk kembali dari awal maka harus melakukan langkah2 tersebut

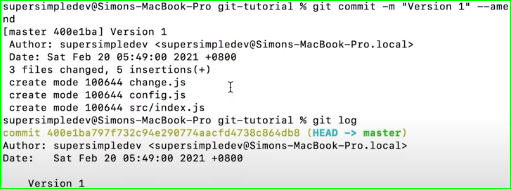
1. Creating version
   1. Create first version (untuk yg benar2 belum ada versinya)  
      ketik git init  
      fungsinya untuk tracking semua file /folder didalam folder ini untuk dirubah  
      
   2. Check status (mengecek perubahan apa saja yg sudah kita lakukan)  
      ketik git status  
        
      karena belum ada version yg kita buat maka akan tampil No commits yet  
      dan aakan tampil file maupun folder di dalam fodler yg kita tergetkan sebelumya
   3. Memilih file/directory yang akan kita kenai version setelahnya

Ketik git add config.js (maka akan memilih config.js saja) atau config.js bias diganti dengan src (jika memilih src saja)  
  
untuk memilih semua yg ada di folder saat ini kita menggunakan git add .  
  
  
  
terlihat new file: merupakan pemilihan file yg akan dipilih untuk pemilihan pembuatan version, karena sebelumnya belum ada/belum dipilih maka akan di deteksi menjadi new file

* 1. Membuat new version  
     ketik git commit -m “pesan yg ingin ditaruh di version baru”  
       
     commit = version (kalo di git version menggunakan Commit)  
     -m = menaruh/attach message diikuti dengan “”(petik dua)  
       
        
     2 files changed , 4 insertion maksudnya = kalo dilihat brarti ada 2 file yg dirubah/ditambah version  
       
     dan karena masing2 memiliki 2 baris maka total baris yg ditambah adlah 4

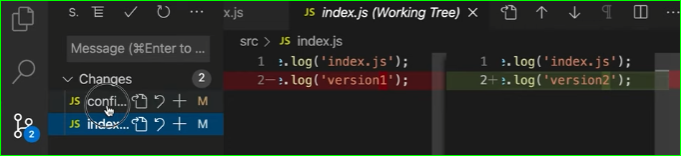
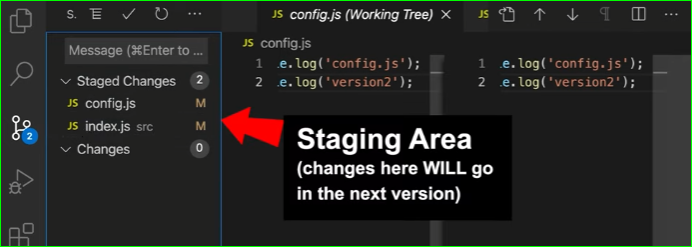
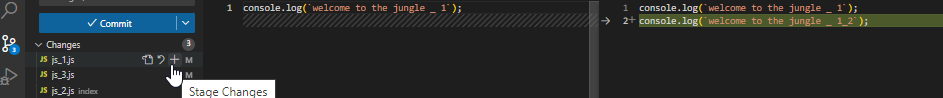
Note : Kalo dapet error seperti ini maka tinggal isi saja user email dan name saja, error ini hanya warning buat kita untuki configure user saja  


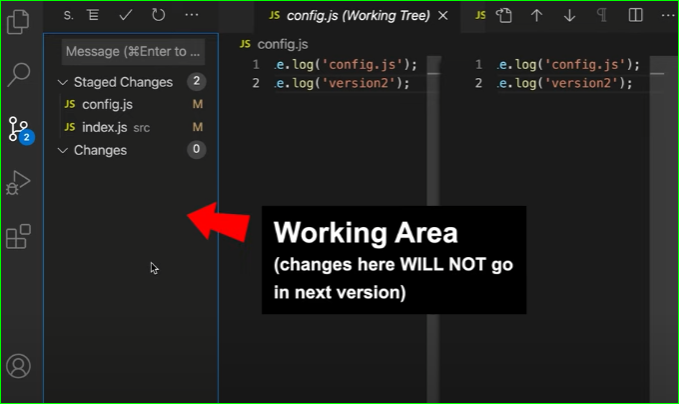
* 1. Check History commit atau version  
       
     ketik git log  
       
       
       
     Berisi tentang info2 yg kita cantumkan diatas

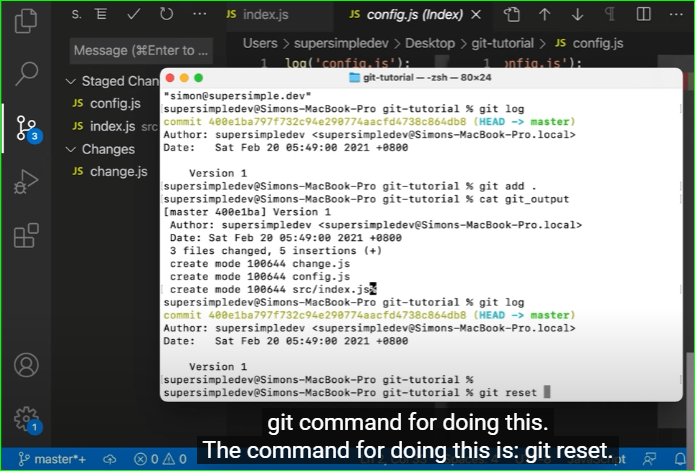
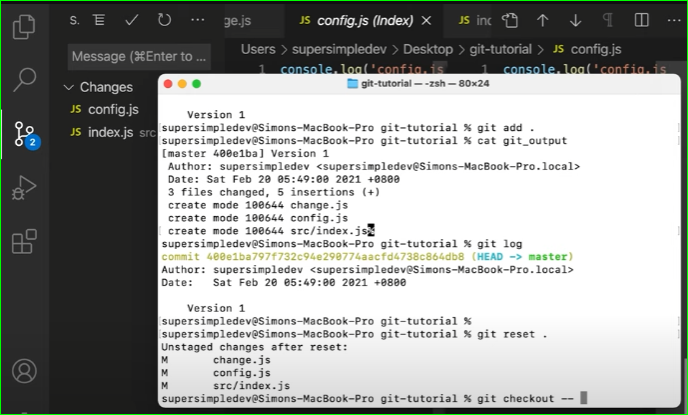
Note : jadi ketika kita ingin menumpuk commit (ketika terjadi kesalahan pada commit sebelumnya) kita bias menggunakan –amend  
  
  
  
jadi daripada kita membuat commit/version baru lagi kita bias menggunakan –amend untuk menumpuk file

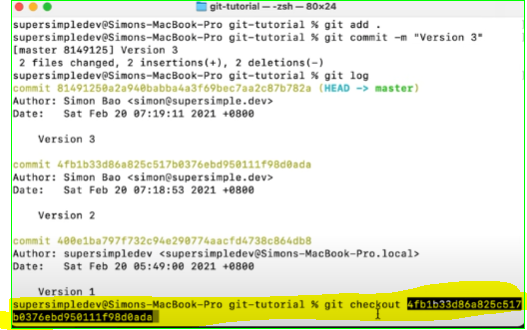
Mengvisualisasikan Perubahan pada Git using Git integration | menggunakan editor Visual studio code

Graphical user interface, application

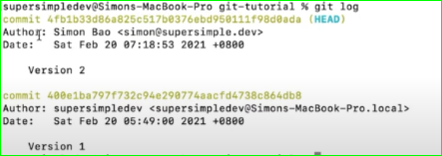
Description automatically generated  
setelah menyimpan file maka kita dapat melihat perubahan apay g terjadi pada menu source control di VS code,   
  
  
  
  
  
  
  
pada tahap ini seperti kita melakukan **git status**  
  
  
  
Brarti stageding area adalah area sebelum di edit(sebelum work area)/atau pemilihan file yg akan di edit ke version setlhnya  
setelah di commit maka akan muncul working Area(atau space yg bis akita edit)  
  
  
  
Tombol “+” digunakan untuk add atau menambah / doing git add (file yg dituju/ . untuk semua)

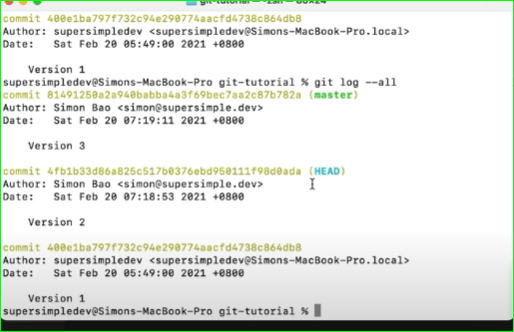
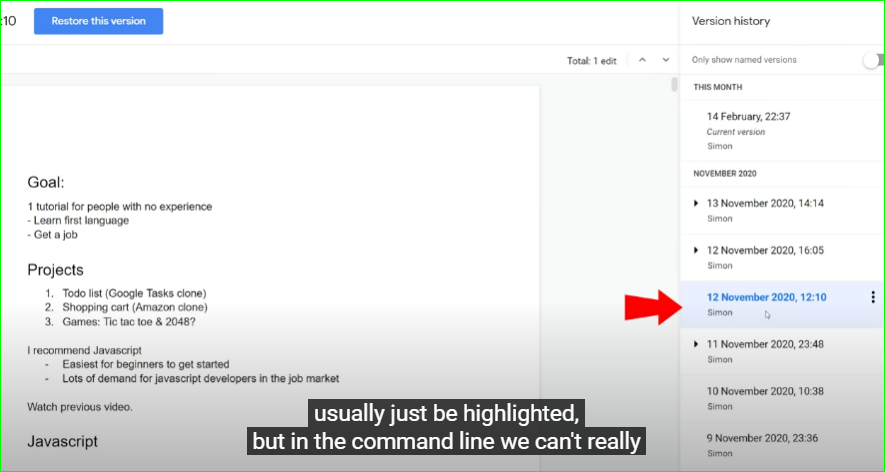
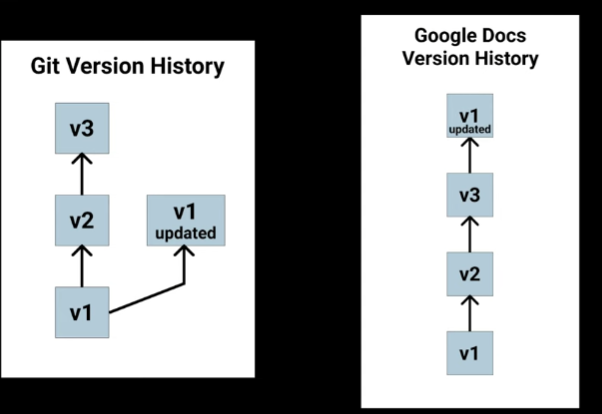
Semua perubahan di letakkan pada bagian Working area dan akan dilanjutkan ke next version/disubmit dengan menggunakan git add atau symbol “+”  


Mengeluarkan file dari staging area ke working area / mengundo file (untuk di edit lagi sebelum final masuk ke staging Area)  
using **git reset (file yg dipilih/dot)**  
Mengembalikan Kembali file seperti semua **(menggunakan git checkout – (file/ . semua)** **menggunakan tanda putar balik**

**Unutk memilih version yg kita mau | git checkout (commit ex 027639c45d03763161f7b24aae45dee4f01d42e4)** **contoh memilih version 2 | tergantung version mana kemudian kita masukkan commitnya**

**lalu check dengan git log  
maka yg tersisa hanya versiom 1 dan 2 saja**

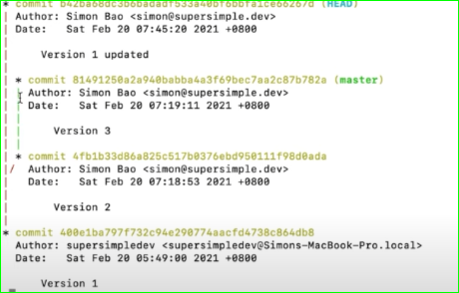


Untuk check semua version yg telah dibuat gunakakn git log –all  
  
  
  
like this one  
  
  
jadi pneegunaan git dapat membuat branch jadi V1 update branch daripada v1  
  
  
  
  
  
  
  
  
  
to check graph/pola per version bisa menggunakan – graph pada git log –all –graph

UnBranching

Branching

Contoh branch pada git



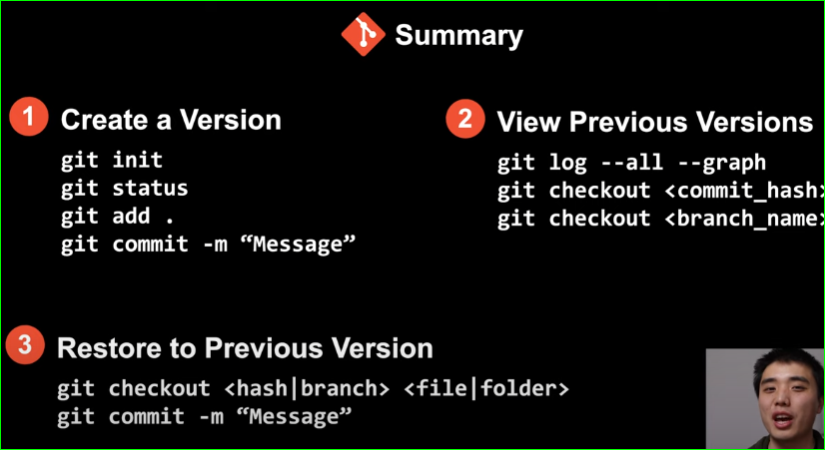
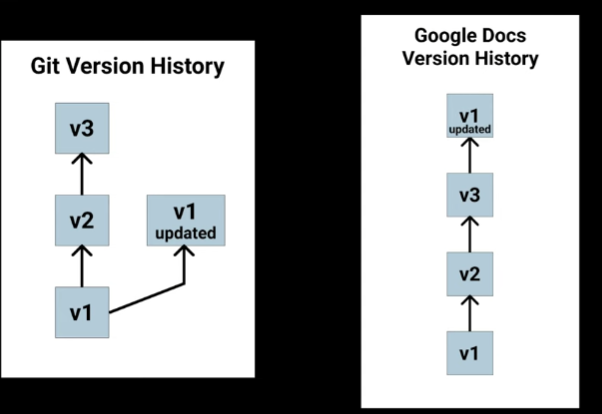
Jadi penggunaan Unbranch dapat memanfaatkan posisi master

Lalu untuk penambahan tanpa menggunakan branch  
1. Pindah ke main/master terlebih dahulu  
checkout master/main

2. edit file => add => commit message

3. file akan langsung berada diatas file utama tanpa melewati branching  
  
atau   
git checkout hash(id unique dari tiap version) file yg akan di kenai(bisa opsi semua(`.`))  
  
2. edit file => add => commit message

3. file akan langsung berada diatas file utama tanpa melewati branching  
  
maka tidak membuat branching  
  
Check <https://www.youtube.com/watch?v=hrTQipWp6co&t=1408s> at 28:00  
  
NB :  
HEAD | posisi dimana kita membuka sekarang  
q | exit

Next practice  
  
  
dan buat 2 jenis version diatas

Bisa membuat Shortcut untuk command  
