

Nama : Ibnu Rusdianto

Kelas : Full Stack 1

Tugas : GIT - Sesi 5

1. Sebutkan fungsi git ?

Jawaban : Mencatat perubahan seluruh file atau repository dari suatu project baik besar maupun kecil

2. Apa yang dimaksud version control sistem (VCS) ?

Jawaban : Version control adalah sebuah sistem yang merekam perubahan-perubahan dari sebuah berkas atau sekumpulan berkas dari waktu ke waktu sehingga dapat menilik kembali versi khusus suatu saat nanti. Sebagai contoh, akan menggunakan sumber kode perangkat lunak sebagai berkas-berkas yang direkam dengan version control, walau pada kenyataannya kita dapat melakukan ini dengan hampir semua jenis berkas pada komputer.

3. Apa Perbedaan dari git dengan github/gitlab/bitbucket ?

Jawaban : **Git** adalah sebuah sistem pengendalian versi (Version Control System, VCS) yang dirancang untuk mengelola perubahan pada sumber kode atau setiap kumpulan file, git dikenal juga perangkat lunak open source yang dapat diinstal dan dijalankan di hampir semua sistem operasi dan memiliki konsep commit, branch, merge, dan remote.

Sementara untuk **Github** berbasis cloud untuk proyek-proyek yang menggunakan Git sebagai VCS, selain itu fitur tambahan di github yaitu bug tracking, manajemen tugas, dan fitur kolaborasi.

Gitlab sebenarnya Mirip dengan GitHub, **GitLab** juga merupakan platform web yang menyediakan tempat untuk mengelola dan melacak pekerjaan pada proyek berbasis Git, point dari gitlab yaitu sering dipilih oleh organisasi yang ingin memiliki kontrol lebih atas data dan infrastruktur mereka.

Bitbucket, seperti GitHub dan GitLab, adalah platform hosting web untuk proyek berbasis Git. Dan pada intinya yaitu Git adalah alat yang digunakan untuk mengelola versi kode sumber, sedangkan GitHub, GitLab, dan

Bitbucket adalah platform hosting dan kolaborasi yang memungkinkan proyek berbasis Git untuk dikerjakan oleh banyak pengembang dari berbagai lokasi.

4. Sebutkan Aplikasi GUI untuk GIT

Jawaban : Setau saya git GUI ada beberapa jenis seperti **GitKraken**, **Sourcetree**, **SmartGit**, **GitForce** tetapi yang populer yaitu GitKraken (setau saya)



Link GitKraken : <https://www.gitkraken.com/>

5. Jelaskan alur pembuatan fitur baru dengan branching sampai di merge kembali dengan branch Master ?

Jawaban : untuk nomer 5 ini berarti kita harus membuat repository terlebih dahulu, lalu buat folder (untuk kita inisialisasi dan push ke github), setelah kita bikin folder, lalu bikin sebuah file html/php/etc (terserah) dan simpan dalam folder tersebut. Setelah sudah menyimpan satu atau lebih file kita buka CMD / Git Bash dan ketik perintah sebagai berikut

“Git init” <

“Git add . / nama_file (jika pake titik maka banyak file, jika pake nama_file berarti hanya file tertentu)” <

“Git Commit -m “pesan message” <

“Git remote add origin (url repo)” <

“Git Push -u origin master”

Kita sudah selesai push awal di repo master, selanjutnya dari pertanyaan nomer 5 yaitu kita harus melakukan update di file tapi dengan branch baru lalu di merge kembali ke master (default branch)

“Git branch (Masukan nama branch baru, contoh kelompok5)”<

“Git Checkout kelompok5 (nama dari branch baru)”<

Catatan : kita sudah berada di branch kelompok5 nah ketika kita sudah berada di kelompok5 ini kita boleh melakukan update atau menambahkan file di dalam folder repo kita (sebagai contoh disini kita sudah menambahkan file baru, lalu lanjut ke tahap terakhir yaitu push dan merge ke master)

“Git add . (posisi masih di branch kelompok5)”<

“Git commit “Message” (posisi di branch kelompok5) <

“Git push origin kelompok5 (karena kita ada di branch kelompok5) <

Jika sudah di push, yang terakhir yaitu kita pindah ke branch master lalu merge branch dari kelompok5

“Git checkout master (pindah ke master)”

“Git merge kelompok5” < Selesai

Link live video menyelesaikan tugas nomer 5 : <https://youtu.be/ujJjR-gmJqg>

6. Apa penyebab dari error tersebut ?

```
LENOVO@LAPTOP-JARM1174 MINGW64 ~/Music/WEB2 (master)
$ git push -u origin master
To https://github.com/fhrdn24/WEB2.git
! [rejected]        master -> master (fetch first)
error: failed to push some refs to 'https://github.com/fhrdn24/WEB2.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Jawaban : Terjadi ketika push perubahan-perubahan lokal ke GitHub, sementara repositori lokal (repo) belum diperbarui dengan perubahan apa pun yang telah dibuat di repositori jarak jauh (remote repo). Git memberi tahu untuk memperbarui repo lokal dengan perubahan-perubahan terkini dari repo jarak jauh sebelum mendorong perubahan kita sendiri. Ini diperlukan agar tidak menimpa perubahan yang telah dibuat oleh orang lain.

Untuk memperbaikinya cukup command “git pull --rebase”

```

gme@ubuntu:~/gitlab-made-easy$ git branch -all
error: did you mean '--all' (with two dashes)?
gme@ubuntu:~/gitlab-made-easy$ git branch --all
* develop
  remotes/origin/develop
gme@ubuntu:~/gitlab-made-easy$ git checkout master
error: pathspec 'master' did not match any file(s) known to git
gme@ubuntu:~/gitlab-made-easy$

```

7.

Jawaban : untuk nomer 7 ini sudah jelas bahwa branch master itu tidak ada, coba perhatikan sesudah memasukkan command “git branch –all” nah disitu hanya memiliki branch yaitu “develop, remote/origin develop” dan master tidak ada, nah jadi disini sudah jelas bahwa branch master kemungkin sudah dihapus atau diganti branch nya. Lalu bagaimana cara kita memperbaikinya? **Kalo menurut saya pribadi bikin branch baru yaitu “master”**

```

Untracked files:
  (use "git add <file>..." to include in what will be committed)

src/js,
loading.svg

no changes added to commit (use "git add" and/or "git commit -a")
root@test:/opt/odoo/custom# git push origin master
Username for 'https://github.com': ashish-webkul
Password for 'https://ashish-webkul@github.com':
remote: Invalid username or password.
fatal: Authentication failed for '
root@test:/opt/odoo/custom# tailf /var/log/odoo/odoo-server.log

```

8.

Jawaban : Kalo menurut saya pribadi itu sudah jelas bahwa errornya yaitu “Invalid Username or password” berarti kemungkinan developernya **salah memasukkan Kredensial username dan password**, tapi bisa juga bermasalah dibagian **Token Akses**, Untuk menghadapi masalah ini kalian cukup cek kembali Kredensial (reset lebih bagus) atau bisa menggunakan command “git credential reject” > membersihkan kredensial lama

```
Surjar2SNDAR MINGW64 / (Running Project/FirstAndG-1) (master)
$ git push origin master
error: src refspec master does not match any
error: failed to push some refs to 'origin' Issue
$
Surjar2SNDAR MINGW64 / (Running Project/FirstAndG-1) (master)
$
Surjar2SNDAR MINGW64 / (Running Project/FirstAndG-1) (master)
$
```

9.

cabang 'master' tidak ada. Mungkin kita melakukan meng-klon sebuah repositori baru dan cabang defaultnya adalah 'main', sehingga tidak ada cabang 'master' ketika kita mencoba untuk melakukan push ke cabang tersebut, namun bisa juga karena belum kita commit, untuk memperbaikinya coba menggunakan “**git push -u origin master**”

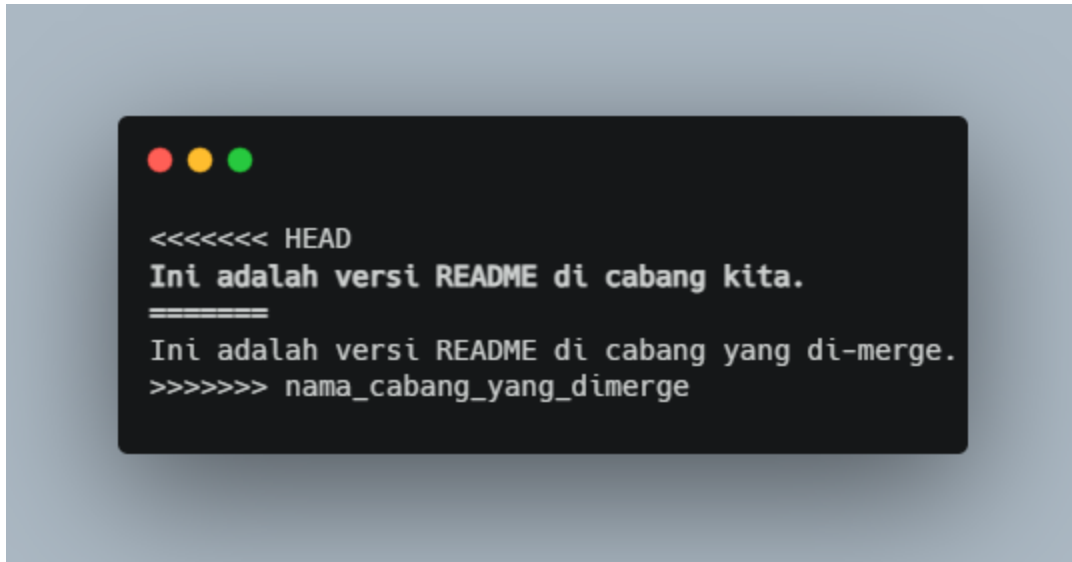
10.

```
kb@kb:~/example$ git branch
conflicting_branch
* master
kb@kb:~/example$ git merge conflicting_branch
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.
```

Jawaban : Untuk masalah ini kalian baca2 terlebih dahulu mengenai conflict merge di link

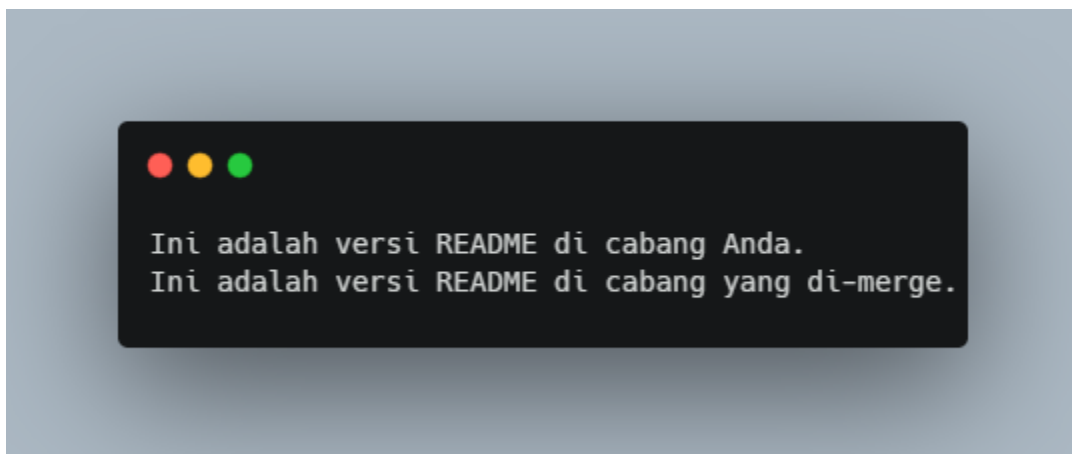
<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/addressing-merge-conflicts/resolving-a-merge-conflict-using-the-command-line> , Konflik merge dalam file README.md terjadi ketika Git tidak bisa

secara otomatis menyelesaikan perbedaan antara dua commit. Biasanya, ini berarti bahwa perubahan pada file di cabang yang sedang kita merge bertentangan dengan perubahan yang telah dilakukan di cabang kita saat ini.



```
<<<<<<< HEAD
Ini adalah versi README di cabang kita.
=====
Ini adalah versi README di cabang yang di-merge.
>>>>>>> nama_cabang_yang_dimerge
```

Contoh seperti gambar diatas : Konflik akan ditandai dengan sintaks khusus yang dimulai dengan <<<<<<< dan diakhiri dengan >>>>>>>., untuk menyelesaikan kasus seperti ini kita harus memilih antara versi di cabang kita atau versi dari cabang yang di-merge. kita juga bisa membuat kombinasi keduanya atau menambahkan sesuatu yang baru.



```
Ini adalah versi README di cabang Anda.
Ini adalah versi README di cabang yang di-merge.
```

Catatan : Hapus semua tanda <<<<<<<, =====, dan >>>>>>> setelah menyelesaikan konflik. Selanjutnya tinggal commit perubahan dan push.