**NAMA : FRADILA NUR HASANAH**

**KELAS : XI RPL 2**

**NO ABSEN : 22**

**RESUME GIT**

**Membuat Repository Baru Dalam Proyek**

git init proyek – 01 : Perintah tersebut akan membuat direktori bernama proyek-01.

git init . : Perintah berikut ini akan membuat repositori pada direktori saat ini (working directory).

**Simpan Perubahan Revisi Dengan Git Commit**

**Tiga Kelompok Kondisi File dalam GIT**

1. Modified adalah kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di version control.

2. Staged adalah kondisi dimana revisi sudah ditandai, tetapi belum disimpan di version control.

3. Commited adalah kondisi dimana revisi sudah disimpan di version control.

**Membuat Revisi**

* git add index.html
* git add about.html
* git add contact.html
* git add index.html about.html contect.html
* git add \*.hmtl
* git add .

git status : Git tersebut digunakan untuk mengecek status repository

git commit –m “commit pertama” : untuk menyimpan perubahan yang dilakukan, tetapi tidak ada perubahan pada remote repository.

**Melihat Catatan Log Revisi**

git log : untuk melihat catatan log perubahan pada repository

git log –online : untuk menampilkan log yang lebih pendek, kita bisa menambahkan argument –oneline

git log (nomer commit) : untuk melihat log pada nomer revisi / commit

git log index.html : untuk melihat revisi pada file tertentu, dengan cara memasukkan nama filenya

git log –-autor=’Fradila’ : digunakan jika repository dikerjakan oleh banyak orang. Maka kita dapat melihat revisi apa saja yang dilakukan oleh orang tertentu

**Melihat Perbandingan Revisi Dengan Git Diff**

git diff (nomer commit): digunakan untuk melihat perubahan yang dilakukan pada revisi tertentu

+ <p> : kode yang ditambahkan

-<i> : kode yang dihapus

-<span> : kode sebelum diubah

+<span> : kode sesudah diubah

git diff : membandingkan perubahan file yang baru saja dilakukan dengan revisi / commit terakhir

git diff <nomer comit> <nomer commit> : untuk membandingkan perubahan pada revisi yang lain

git diff <nama cabang> <nama cabang> : untuk melihat perbandingan perubahan antar cabang

**Perintah Untuk Membatalkan Revisi**

git checkout nama\_file.html : digunakan untuk mengembalikan seperti keadaan sebelumnya saat terjadi kesalahan / untuk membatalkan perubahan

git add nama\_file.html : untuk mengubah kondisi file menjadi staged

git reset nama\_file.html : untuk mengubah dari staged menjadi kondisi modified

* kalau filenya sudah dalam kondisi commited dan kita ingin mengembalikannya kita harus mengetahui nomer commit, kemudian mengembalikan perubahannya seperti pada nomer commit tersebut.
* Cara melihat nomer commit :

git log

git checkout (nomer comit) : untuk mengembalikan kondisi file.html, seperti pada commit sebelumnya

git reset nama\_file.html : untuk mengembalikan kondisi dari staged ke kondisi modified

git checkout HEAD~3 nama\_file.html : untuk mengembalikan ke 3 commit sebelumnya

git revert –n <nomer comit> : untuk mengembalikan semua file ke suatu commit

**Menggunakan Percabangan Untuk Mencegah Konflik**

git branch nama\_cabang: untuk membuat cabang baru

git branch : untuk melihat cabang apa saja yang ada di repository

git checkout nama\_cabang : digunakan untuk pindah ke cabang yang baru saja di buat

git checkout master : untuk pindah ke cabang master, biasanya dilakukan saat ingin menggabungkan cabang

git merge nama\_cabang : untuk menggabungkan cabang

git checkout nama\_cabang : untuk memindahkan bentrokan

git branch –d nama\_cabang : untuk menghapus cabang

**Perbedaan Git Chckout, Git Reset, dan Git Revert**

**Git Checkout**

mengembalikan file dalam kondisi sebelumnya, tapi bersifat sementara.

Contoh :

git checkout (nomer commit) : untuk mengembalikan kondisi file proyek seperti waktu yang di tuju

git checkout –b nama\_cabang <nomer\_commit> : untuk membuat cabang baru berdasarkan kondisi kode di masa lalu

**Git Reset**

akan mengembalikan file ke kondisi sebelumnya, kemudian menghapus catatan sejarah commit beikutnya..

Perintah ini memiliki tiga argumen atau opsi utama, yaitu --soft, --mixed, dan --hard.

• --soft akan mengebalikan dengan kondisi file dalam keadaan staged

• --mixed akan mengebalikan dengan kondisi file dalam keadaan modified

• --hard akan mengebalikan dengan kondisi file dalam keadaan commited

Contoh :

git reset –soft (nomer-comit)

**Git Revert**

mengembalikan file dengan tidak menghapus sejarah commit

contoh : git revert (nomer\_comit)

**Bekerja Dengan Remote Repository**

**Membuat Repository di Github**

1. Create a new repository : mempunyai repository kosong di github
2. Menambahkan dan Menghapus Remote : Sebelum kita bisa upload semua revisi yang ada di repository lokal, kita harus menambahkan remote-nya terlebih dahulu.

**Remote dapat di tambahkan dengan perintah seperti ini:**

* [**https://github.com/Fradila/belajar\_git.git**](https://github.com/Fradila/belajar_git.git)
* [**git@github.com:Fradila/belajar\_git.git**](mailto:git@github.com:Fradila/belajar_git.git)

**Perbedaan :** Kalau kita menggunakan HTTPS, maka kita akan diminta password setiap kali melakukan push. Sedangkan yang menggunakan SSH, kita tidak akan diminta password

git remote add github [git@github.com:Fradila/belajar\_git.git](mailto:git@github.com:Fradila/belajar_git.git) : untuk menambahkan remotenya

git remote –v : untuk melihat remote apa saja yang sudah ditambahkan

git remote rename github kantor : untuk menghapus dan mengubah nama remote

git remote remove github : untuk menghapus remote

1. Mengirim revisi ke remote repository

git push github master : untuk mengirim revisi ke remote repositrory

git remote : untuk mengecek repository lokal

**Mengambil Revisi dari Remote Repositorinya**

Ada dua perintah untuk mengambil revisi dari repository remote:

1. git fetch [nama remote] [nama cabang] : Perintah git fetch hanya akan mengambil revisi (commit) saja dan tidak langsung melakukan penggabungan (merge) terhadap repository lokal

2. git pull [nama remote] [nama cabang] : git pull akan mengambil revisi (commit) dan langsung melakukan penggabungan (merge) terhadap repository lokal.

**Mengambil Revisi dengan git fetch**

Silahkan buka github, dan tambahkan file README.md melalui Github. Klik tombol add README. Setelah itu, isilah file RAEDME.md dengan apapun yang kamu inginkan. Setelah selesai, simpan perubahan dengan melakukan commit langsung dari Github.

Git diff master belajar\_git/master : untuk mengecek apa saja perbedaan antara repository local dan remote, coba gunakan perintah git diff.

Git merge master belajar\_git/master : menggabungkan commit dari repository remote dengan lokal

**Mengambil Revisi dengan git pull**

Lakukan hal yang sama seperti tadi. Kali ini kita akan membuat file baru bernama register.html melalui Github. Berikan nama file dengan register.html dan isi dengan apa saja. Simpan revisi dan tambahkan pesan commit. Sekarang ada perubahan baru di repository remote dan kita akan mengambilnya dengan perintah git pull.

Git pull belajar\_git master

**Clone Remote Repository**

Clone repository bisa kita bilang seperti copy repository/download dari remote ke lokal. Perintah untuk melakukan clone adalah git clone.

git clone https :// github.com/Fradila/belajar\_git.git (nama dir)

Keterangan:

• https://... adalah URL repository remote, kita juga bisa menggunakan SSH.

• [nama dir] (opsional) adalah nama direktory yang akan dibuat di lokal. Jika kita tidak berikan nama direktori, maka akan otomatis menggunakan nama repository.

Cd -/Desktop : pindah ke desktop