

# Modul 2 Programming Extra

## *Git dan Github*

### 1. Version Control System

Git merupakan salah satu Version Control System (VCS). VCS adalah sistem yang merekam perubahan yang terjadi pada *source code*. Selain itu, VCS memudahkan user dalam bekerja berkolaborasi. Dengan VCS, kita dapat mengetahui informasi siapa dan kapan perubahan terjadi. Contoh VCS lain selain Git adalah Subversion, Mercurial, dan CVS.

### 2. Git

Pada modul ini, kita fokus ke Git. Git dibuat oleh Linus Torvalds di 2005 dan open-source. Git ini sendiri berbeda dari Github. Git adalah VCS, Github adalah tempat menyimpan data-nya untuk mengelola project/repo git.

Pertama, lakukan instalasi Git dengan mengikuti link berikut :

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

Mari kita pelajari istilah-istilah yang akan sering didengar di Git:

- **Repo/Repository** adalah folder project dimana git akan merekam semua perubahan yang terjadi. Biasanya setiap kita membuat project baru, kita membuat repository baru.
- **Commit** adalah rekaman perubahan yang terjadi pada waktu tertentu. Misal kita melakukan perubahan terhadap suatu file di repository. Kita dapat melakukan commit untuk merekam perubahan itu.

### Inisialisasi Git

Dalam Git, kita pertama perlu melakukan inisialisasi Git. Inisialisasi Git dapat dilakukan dengan melakukan command 'git init' pada folder yang ingin dijadikan repo.

Buka terminal dan pergi menuju direktori folder yang ingin diinisialisasi menjadi repo Git. Lakukan perpindahan direktori menggunakan "cd"

```
$ cd <path-ke-folder>
```

```
$ git init
```

Setelah kita melakukan "git init", akan terbentuk file .git pada folder tersebut. File .git inilah tempat Git menyimpan perubahan-perubahan pada Staging area dan History.

## Commit dan Git Area

Saat kita memakai Git, terdapat 3 area pada repo.

1. **Working Tree**, folder kerja biasa dimana kita dapat menambahkan atau memodifikasi file
2. **Staging Area**, tempat Git menentukan file-file apa saja yang ditambahkan atau dimodifikasi yang akan direkam perubahannya saat kita melakukan commit.
3. **History**, setelah melakukan commit, perubahan tersebut direkam pada History.

Untuk melihat file apa saja yang ada di staging area, gunakan command “git status”.

```
$ cd <path-ke-folder> #Pastikan kalian berada di folder git
$ git status
```

Disitu, kita akan melihat file-file yang ada di staging area dan tidak. Untuk memasukan file dari Working Tree ke Staging Area, lakukan “git add”

```
$ git add .
# Atau
$ git add <nama-file>
```

Jika kita lakukan “git status” lagi, kita akan melihat bahwa file yang kita add sudah berada di Staging Area.

Langkah berikutnya adalah memasukan file dari Staging Area ke History. Hal ini bisa dilakukan dengan git commit.

```
$ git commit -m "<pesan-commit>"
```

Notice bahwa setiap commit ada pesan. Pesan ini bebas dan biasanya berisi perubahan apa yang terjadi pada commit tersebut.

Untuk melihat History commit yang telah kita lakukan, gunakan “git log”

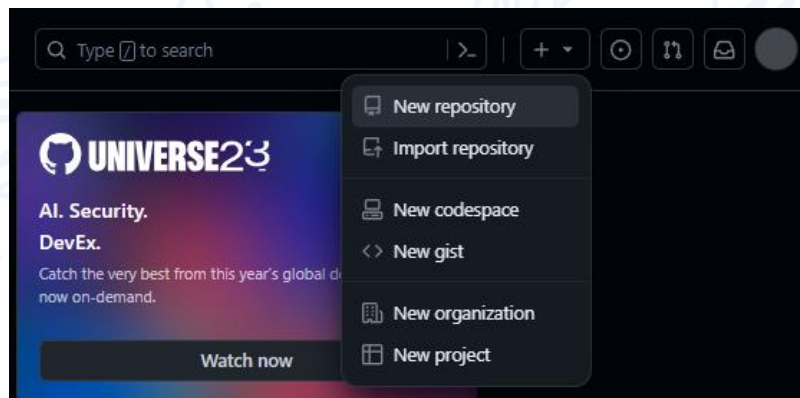
```
$ git log
```

## Github

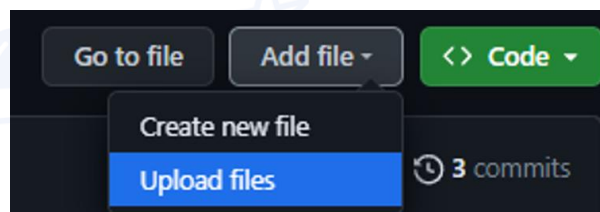
Github berbeda dengan git. Github ini semacam Google Drive (Cloud Service) yang dikhususkan untuk mengelola repo Git. Kita dapat membuat repo, melakukan commit, dan mengintegrasikan Git yang berada pada folder local kita dan pada Github.

Ada 2 cara untuk menambahkan file pada repo Github (kita sebut repo remote) yaitu dengan cara manual di web, dan dengan melakukan push di terminal.

Pada web, kita dapat membuat repo dengan klik simbol + pada ujung kanan dashboard dan New Repository.

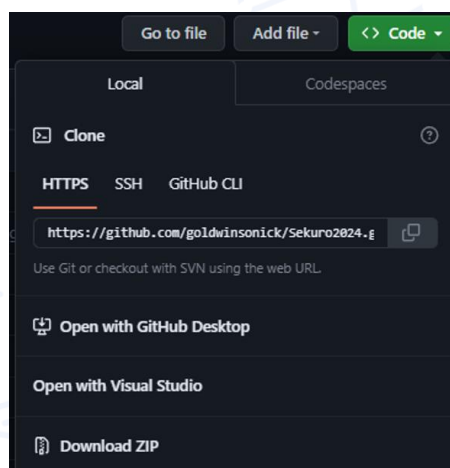


Setelah dibuat repo, kita dapat menambahkan file dengan web yaitu dengan memilih Upload File pada selector Add file.



Cara kedua adalah dengan push repo local ke repo remote.

Git clone adalah command untuk melakukan 'clone' terhadap repo remote ke repo local. Misal kita ada repo bernama "Sekuro2024" pada Github. Kita dapat melakukan git clone sehingga repo tersebut ada di folder pada PC kita.



Pertama, copy link https:/ yang ada di repo remote kita.

Kemudian, lakukan git clone pada folder yang ingin kita masukan repo.

```
$ git clone <link-https-repo-remote>
```

Saat kita melakukan git clone, Git akan otomatis melakukan "git init", "git pull", dan mengatur origin.

Jika kita sudah clone, kita dapat membuat file baru, memodifikasi file, dan lain-lain pada folder yang telah kita clone.

Setelah selesai menambahkan perubahan pada source code dan setelah kita commit ke History local kita. Kita ingin mengirimkan perubahan-perubahan pada history local ke remote.

Caranya adalah dengan “git push”

```
$ git push
```

Dengan melakukan git push, history git akan dikirim ke Github yang kita miliki.

## Git Origin

Itu tadi kita beranggapan bahwa kita membuat repo remote terlebih dahulu tanpa membuat repo local (dengan git init). Bagaimana jika kita telah melakukan git init pada repo local dan ingin membuat repo di Github berdasar repo local tersebut.

Caranya adalah dengan membuat repo remote seperti sebelumnya. Untuk memberi tahu kepada Git local mengenai kemana push dilakukan, kita harus melakukan set origin. Hal ini dilakukan dengan command berikut:

```
$ git remote set-url origin "<repo-url>"
```

<repo-url> itu yang link https tadi.

Dengan ini, sekarang jika kita melakukan push pada Git local kita, dia akan push ke repo remote kita.

Itu tadi merupakan tutorial atau materi Git yang diperlukan untuk kalian dapat menggunakan Git secara dasar. Banyak sekali konsep Git lainnya yang dapat dipelajari seperti Branching, Merging, Fork, Pull Request, dsb. Untuk mempelajari hal tersebut, Anda dapat mencarinya di internet dan mencobanya. Cara terbaik untuk mempelajari git adalah dengan menggunakannya sehingga kita dapat terbiasa dengan menggunakan git.