

Tutorial Git

Tutorial ini mengasumsikan penggunaan pada *Windows*, tanpa melarang penggunaan OS lain, silakan menyesuaikan.

1. Installing Git

- Unduh *installer* di <https://git-scm.com/>
- Jalankan installer, ikuti petunjuk yang diberikan (tips: cukup menggunakan *default setting*).

2. Git Setup

- Jalankan aplikasi Git Bash.
- Pengaturan nama, jalankan perintah:

```
$ git config --global user.name "Ganti Dengan Nama Sendiri"
```

- Pengaturan email, jalankan perintah:

```
$ git config --global user.email gantidengan@email.sendiri
```

- Memeriksa *setting*, jalankan perintah:

```
$ git config --list
```

- Anda dapat memeriksa atau mengganti *setting* menggunakan *key* spesifik, misalnya:

```
$ git config user.name
```

```
$ git config user.email
```

- Tips: untuk menampilkan bantuan dapat menggunakan perintah:

```
$ git help gantidengannamaperintah
```

3. Local Repository

- Buat *directory* lokal dengan nama tutorial-01, jalankan perintah:

```
$ mkdir tutorial-01
```

- Buat Git *repository*, jalankan perintah:

```
$ git init tutorial-01
```

Perintah ini membuat *subdirectory* baru bernama `.git` yang berisi *files* terkait Git *repository*.

Q1: Apa yang terjadi jika perintah dijalankan tanpa menyantumkan nama *(path) directory*?

- Buatlah *file* baru (biodata.txt) di dalam *repository* yang anda buat, isi *file* tersebut dengan:

NO.REG : 18XXXXXX (No. Reg. Masing-Masing)
 Nama : Ganti Dengan Nama Sendiri

- Tampilkan status *repository* (*working tree*) anda, jalankan perintah:

```
$ git status
```

- Mulai *version-controlling files* yang ada (tidak hanya *directory* kosong). Terdapat dua perintah utama yang digunakan: *add* dan *commit*.
- Tambahkan biodata.txt kedalam *index working tree* (*staging*), jalankan perintah:

```
$ git add biodata.txt
```

Q2: Jelaskan perbedaan status *working tree* sebelum dan setelah menjalankan perintah *add*?

- Catat perubahan, pada konteks ini perintah *add*, kedalam *repository*, jalankan perintah:

```
$ git commit -m 'initial project version'
```

Q3: Apakah maksud dari -m dan 'initial project version' pada perintah diatas? Apa yang terjadi jika perintah dilakukan tanpa menambahkan -m?

- Menampilkan *log* dari *commit*, jalankan perintah:

```
$ git log
```

Q4: Informasi apa saja yang ditampilkan pada perintah *log*?

- Ubah biodata.txt dengan menambahkan informasi berikut:

Jurusan : Isi Dengan Program Studi Anda

- Periksa status dengan menggunakan perintah yang sesuai.

Q5: Informasi apa yang anda peroleh dari status saat ini?

- Lakukan *staging* terhadap perubahan yang terjadi menggunakan perintah yang sesuai.
- Periksa status dengan menggunakan perintah yang sesuai.

Q6: Informasi apa yang anda peroleh dari status saat ini?

- Lakukan *commit* menggunakan perintah yang sesuai.
- Temukan id *commit* pertama yang anda lakukan pada *log*. Gunakan perintah yang sesuai.

Q7: Berapakah id *commit* pertama anda?

- Menampilkan perbedaan antar *commit* pertama dengan *commit* terakhir (HEAD), jalankan perintah:

```
$ git diff gantidenganidcommitpertamaanda
```

4. Remote Repository Part 1 – push

- Buatlah *account* pada <https://github.com/>.
- Buatlah *repository* baru. Silakan gunakan npm masing-masing.
- Jalankan perintah untuk *push repository* yang sudah ada (dari *local* ke *remote*). Cek pada GitHub masing-masing.

Q8: Apakah maksud dari -u, origin, dan master pada perintah dalam kotak merah?

- Ubah biodata.txt dengan menambahkan informasi berikut:

Angkatan : Isi Dengan Tahun Angkatan Anda

Q9: Apa perbedaan antara *local* dan *remote repository* pada titik ini?

- Lakukan *staging* dan *commit* menggunakan perintah yang sesuai.
- Perbarui *remote*, jalankan perintah:

```
$ git push
```

Q10: Perubahan apa yang terjadi pada *remote repository*?

5. Remote Repository Part 2 – branch, merge

- Buat *branch* dengan nama testing, jalankan perintah:

```
$ git branch testing
```

- Pindah ke *branch* testing, jalankan perintah:

```
$ git checkout testing
```

- Buatlah file baru (branch.txt) pada *branch* testing, lalu lakukan *commit* menggunakan perintah yang sesuai.
- Lakukan *push* menggunakan perintah yang sesuai. Jika diminta, lakukan *set upstream*, jalankan perintah: (setelahnya, pastikan dengan melakukan *push* kembali)

```
$ git push --set-upstream origin testing
```

- Kembali ke *branch* master, jalankan perintah:

```
$ git checkout master
```

- Lakukan *merge branch* testing ke master, jalankan perintah:

```
$ git merge testing
```

- Lakukan *push* menggunakan perintah yang sesuai.

6. Remote Repository Part 3 – clone

- Lakukan clone repository yang diminta, jalankan perintah:

```
$ git clone https://github.com/[NAMA_AKUN]/3sc3_2019.git
```

- Tambahkan *file* biodata.txt kedalam apap *repository*, ubah nama file dari biodata.txt menjadi npmanda.txt lalu lakukan *commit* dan *push* perubahan yang telah anda lakukan.

Q11: Apa itu *conflict* dalam konteks Git. Jika pada langkah ini anda menemukan *conflict*, tambahkan pada jawaban anda bagaimana anda menyelesaikannya.

Pengumpulan

1. *Screen capture* dari setiap langkah yang dilakukan beserta deskripsi/*caption*-nya, singkat saja.
2. Jawaban dari setiap pertanyaan yang ditandai dengan warna hijau.

Tuliskan dalam satu *file* dengan format noreg_nama-lengkap-anda.pdf dan unggah ke email faculty.