

PRAKTIKUM 11

GIT

A. TUJUAN PEMBELAJARAN

1. Mahasiswa memahami fungsi, instalasi, konfigurasi dan membuat repositori.
2. Mahasiswa membuat cek log, cek revisi dengan Git Diff, dan menggunakan percabangan.

B. DASAR TEORI

a. Git

GIT adalah software yang dikembangkan oleh pengembang UNIX, Linus Torvalds, pada tahun 2005. GIT kini telah digunakan oleh jutaan perusahaan untuk melakukan manajemen efisiensi kode dan kontrol versi untuk project mereka. Software ini bersifat open source dan bisa di-download untuk Linux, Windows, Solaris dan Mac.

Seiring berkembangnya teknologi terjadi pergeseran penggunaan git, yang tadinya hanya digunakan oleh software developer dan programmer. Saat ini git dimanfaatkan untuk membuat cadangan berkas seperti sebuah static site.

b. Perintah Dasar Git

- **Git config**

Salah satu perintah git yang paling banyak digunakan adalah **git config**, yang bisa digunakan untuk mengatur konfigurasi tertentu sesuai keinginan pengguna, seperti email, algoritma untuk diff, username, format file, dll. Contohnya, perintah berikut bisa digunakan untuk mengatur email:

```
git config --global user.email sam@google.com
```

- **Git init**

Perintah ini digunakan untuk membuat repositori baru. Caranya:
`git init`

- **Git add**

Perintah git add bisa digunakan untuk menambahkan file ke index. Contohnya, perintah berikut akan menambahkan file bernama temp.txt yang ada di direktori lokal ke index:

```
git add temp.txt
```

- **Git clone**

Perintah git clone digunakan untuk checkout repositori. Jia repositori berada di remote server, gunakan:

```
git clone alex@93.188.160.58:/path/to/repository
```

Jika salinan repositori lokal ingin dibuat, gunakan:

```
git clone /path/to/repository
```

- **Git commit**

Perintah git commit digunakan untuk melakukan commit pada perubahan ke head. Ingat bahwa perubahan apapun yang di-commit tidak akan langsung ke remote repository. Gunakan:

```
git commit -m "Isi dengan keterangan untuk commit"
```

- **Git status**

Perintah git status menampilkan daftar file yang berubah bersama dengan file yang ingin di tambahkan atau di-commit. Gunakan:

```
git status
```

- **Git push**

git push adalah perintah git dasar lainnya. Push akan mengirimkan perubahan ke master branch dari remote repository yang berhubungan dengan direktori kerja Anda. Misalnya:

```
git push origin master
```

- **Git Checkout**

Perintah git checkout bisa digunakan untuk membuat branch atau untuk berpindah diantaranya. Misalnya, perintah berikut ini akan membuat branch baru dan berpindah ke dalamnya:

```
command git checkout -b <nama-branch>
```

Untuk berpindah dari branch satu ke lainnya, gunakan:

```
git checkout <branch-name>
```

- **Git remote**

Perintah git remote akan membuat user terhubung ke remote repository. Perintah berikut ini akan menampilkan repository yang sedang dikonfigurasi:

```
git remote -v
```

Perintah ini membuat user bisa menghubungkan repository lokal ke remote server:

```
git remote add origin <93.188.160.58>
```

- **Git branch**

Perintah git branch bisa digunakan untuk me-list, membuat atau menghapus branch. Untuk menampilkan semua branch yang ada di repository, gunakan:

```
git branch
```

Untuk menghapus branch:

```
git branch -d <branch-name>
```

- **Git pull**

Untuk menggabungkan semua perubahan yang ada di remote repository ke direktori lokal, gunakan perintah pull:

```
git pull
```

- **Git merge**

Perintah merge digunakan untuk menggabungkan sebuah branch ke branch aktif. Gunakan:

```
git merge <nama-branch>
```

- **Git diff**

Perintah git diff digunakan untuk menampilkan conflicts. Untuk melihat conflicts dengan file dasar, gunakan:

```
git diff --base <nama-file>
```

Perintah berikut digunakan untuk menampilkan conflicts diantara branch yang akan di-merge:

```
git diff <source-branch> <target-branch>
```

Untuk menampilkan semua conflict yang ada, gunakan:

`git diff`

- **Git tag**

Tagging digunakan untuk menandai commits tertentu. Contohnya:

`git tag 1.1.0 <insert-commitID-here>`

- **Git log**

Dengan menjalankan perintah ini akan menampilkan daftar commits yang ada di branch beserta detail-nya. Contoh outputnya adalah:

```
commit 15f4b6c44b3c8344caasdac9e4be13246e21sadm Author:
Alex Hunter <alexh@gmail.com> Date: Mon Oct 1 12:56:29
2016 -0600
```

- **git reset**

Untuk me-reset index dan bekerja dengan kondisi commit paling baru, gunakan perintah git reset:

`git reset --hard HEAD`

- **git rm**

Gunakan perintah ini untuk menghapus file dari index dan direktori kerja. Contohnya:

`git rm filename.txt`

- **git stash**

Mungkin inilah salah satu perintah dasar git yang jarang digunakan orang, yang bisa membantu menyimpan perubahan yang tidak langsung di-commit, namun hanya sementara. Contoh:

`git stash`

- **git show**

Untuk menampilkan informasi tentang object pada git, gunakan git show:

`git show`

- **git fetch**

Perintah ini digunakan untuk menampilkan semua object dari remote repository yang tidak berada di direktori kerja lokal. Contohnya:

`git fetch origin`

- **git ls-tree**
Untuk menampilkan susunan object berdasarkan nama dan mode setiap item, dan nilai blob SHA-1, gunakan perintah git ls-tree. Contohnya:
`git ls-tree HEAD`
- **git cat-file**
Menggunakan nilai SHA-1, menampilkan tipe object dengan menggunakan perintah git cat-file. Contohnya:
`git cat-file -p
d670460b4b4aece5915caf5c68d12f560a9fe3e4`
- **git grep**
git grep mengizinkan pengguna mencari frase dan/atau kata yang berada di dalam direktori. Contohnya, untuk mencari `www.hostinger.co.id` di dalam semua file, gunakan:
`git grep "www.hostinger.co.id"`
- **gitk**
gitk adalah tampilan grafis dari repository lokal yang bisa dipanggil dengan menjalankan perintah:
`gitk`
- **git instaweb**
Dengan perintah git instaweb, web server bisa dijalan berdampingan dengan repository lokal. Nantinya web browser akan langsung diarahkan ke server tersebut. Contohnya:
`git instaweb -httpd=webrick`
- **git gc**
Untuk mengoptimasi repository melalui garbage collection, yang akan membersihkan file yang tidak dibutuhkan dan mengoptimasinya, gunakan:
`git gc`
- **git archive**
Perintah git archive memungkinkan user membuat file zip atau tar yang mengandung susunan repository. Contohnya:
`git archive --format=tar master`

- **git prune**

Melalui perintah git prune, object yang tidak memiliki incoming pointers akan dihapus. Gunakan:

```
git prune
```

- **git fsck**

Untuk membuat pengecekan keseluruhan dari file system git, gunakan perintah git fsck. Object yang corrupt akan dikenali:

```
git fsck
```

- **git rebase**

Perintah ini digunakan untuk menerapkan ulang commit di branch yang lain. Contohnya:

```
git rebase master
```

C. TUGAS PENDAHULUAN

1. Lakukan sign up untuk yang belum memiliki akun github atau sign in untuk yang sudah memiliki akun github.
2. Install git di Mac OS dan Windows

- **Mac Os**

Install Git Dengan Brew

Buka terminal, kemudian ketikkan perintah di bawah ini :

```
brew install git
```

Tunggu proses instalasi sampai selesai, kemudian cek versi git apakah sudah berhasil terinstall

```
git --version
```

```
[yesta@Yestas-MacBook-Air ~ % git version  
git version 2.38.1  
yesta@Yestas-MacBook-Air ~ % █
```

- **Windows**

Untuk di windows kamu didownload di <https://git-scm.com/download/win>

Kemudian double click dan ikuti langkah-langiknya dengan menekan **Next** sampai **Finish**.



Tunggu proses instalasi sampai selesai, kemudian cek versi git kamu apakah sudah berhasil terinstall

`git --version`

D. PERCOBAAN

Percobaan 1 : Membuat Project Git

Langkah Percobaan:

1. Membuat folder baru di xampp/htdocs, dengan nama web_git
2. Buka terminal Mac OS atau windows dan arahkan ke direktori folder web_git

```
yesta@Yestas-MacBook-Air Workshop Desain Web % cd web_git/
```

3. Kemudian di terminal ketikkan `git init`, jika berhasil akan muncul tampilan dibawah:

```

yesta@Yestas-MacBook-Air web_git % git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /Users/yesta/Documents/PENS/Workshop Desain
Web/web_git/.git/
yesta@Yestas-MacBook-Air web_git %

```

4. Menambahkan File

Setelah itu buat dokumen index.html pada folder web_git dengan kode dibawah ini:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Git Project Tutorial</title>
5  </head>
6  <body>
7
8  <h1> Git Project Pertamaku</h1>
9  <h2> Membuat file project baru </h2>
10 <p> ini adalah git project pertamaku dengan membuat file baru</p>
11
12 </body>
13 </html>

```

5. Melihat status dari dokumen html yang dibuat di git

git status

6. Maka hasilnya sebagai berikut :

```

yesta@Yestas-MacBook-Air web_git % git status
On branch master

No commits yet

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

nothing added to commit but untracked files present (use "git add" to track)
yesta@Yestas-MacBook-Air web_git %

```


7. Melakukan commit terhadap file yang dibuat :

```
git add index.html
git status
```

```
yesta@Yestas-MacBook-Air web_git % git add index.html
yesta@Yestas-MacBook-Air web_git % git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   index.html

yesta@Yestas-MacBook-Air web_git %
```

8. Buat File index.css dan README.md

Pada dokumen index.css tambahkan script sebagai berikut :

```
1  body {
2      color: blue;
3  }
4
5  h1 {
6      color: green;
7  }
8
9  h2 {
10     text-indent: 50px;
11     color: blueviolet;
12 }
13
14 p {
15     font-family: "Times New Roman", Times, serif;
16 }
```

Pada README.md tambahkan :

```
1  # Belajar GIT
2  Belajar GIT adalah project untuk saya belajar GIT
```

9. Menambahkan semua file di folder web_git dan melakukan commit :

```
yesta@Yestas-MacBook-Air web_git % git add .
yesta@Yestas-MacBook-Air web_git % git commit -m 'create css style, readme.md and link style css with index.html'
[master (root-commit) c96d41d] create css style, readme.md and link style css with index.html
3 files changed, 31 insertions(+)
create mode 100644 README.md
create mode 100644 index.css
create mode 100644 index.html
```

10. Mengecek status dari project yang ditambahkan :

```
yesta@Yestas-MacBook-Air web_git % git status
On branch master
nothing to commit, working tree clean
yesta@Yestas-MacBook-Air web_git %
```

Percobaan 2 : Membuat Branch Git Baru

Langkah Percobaan:

1. Masuk ke direktori yang sudah dibuat sebelumnya web_git melalui terminal dengan perintah `cd web_git`
2. Membuat branch baru dengan format sebagai berikut :
 - Menggunakan nama developer
 - Menggunakan slash atau underscore untuk pemisah
 - Nama features atau perubahan yang akan dilakukan

```
nothing to commit, working tree clean
yesta@Yestas-MacBook-Air web_git % git checkout -b wdw/web
Switched to a new branch 'wdw/web'
yesta@Yestas-MacBook-Air web_git %
```

-b = singkatan dari nama branch yang akan dibuat

3. Selanjutnya cek daftar branch yang telah dibuat di lokal git masing-masing dengan perintah : `git branch`

```
yesta@Yestas-MacBook-Air web_git % git branch
master
* wdw/web
yesta@Yestas-MacBook-Air web_git %
```

4. Saat berpindah dari branch git ke master git gunakan perintah: `git checkout master`

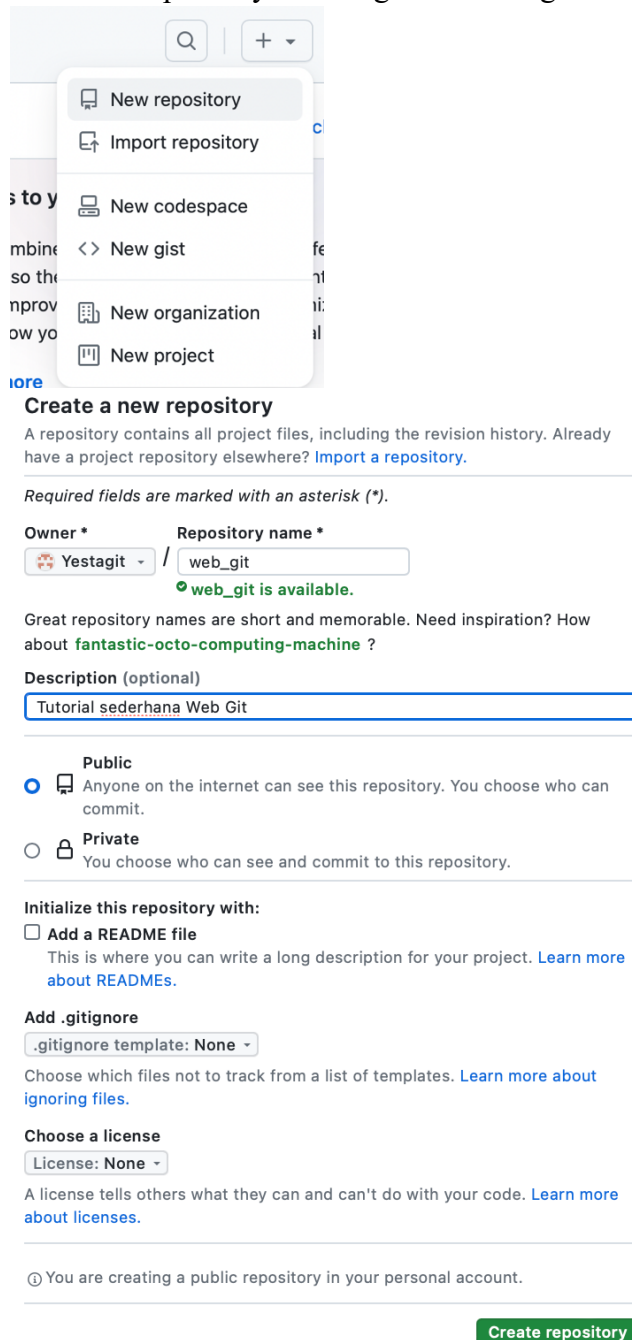
```
yesta@Yestas-MacBook-Air web_git % git checkout master
Switched to branch 'master'
yesta@Yestas-MacBook-Air web_git %
```

Percobaan 3 : Publish Project Git (Add, Push)

Langkah Percobaan:

1. Membuat Project Online GITHUB

Membuat repository baru di github masing-masing



Search +

- New repository
- Import repository
- New codespace
- <> New gist
- New organization
- New project

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * Yestagit / Repository name * web_git

✔ web_git is available.

Great repository names are short and memorable. Need inspiration? How about [fantastic-octo-computing-machine](#) ?

Description (optional)
Tutorial sederhana Web Git

☒ **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs](#).

Add .gitignore
.gitignore template: None

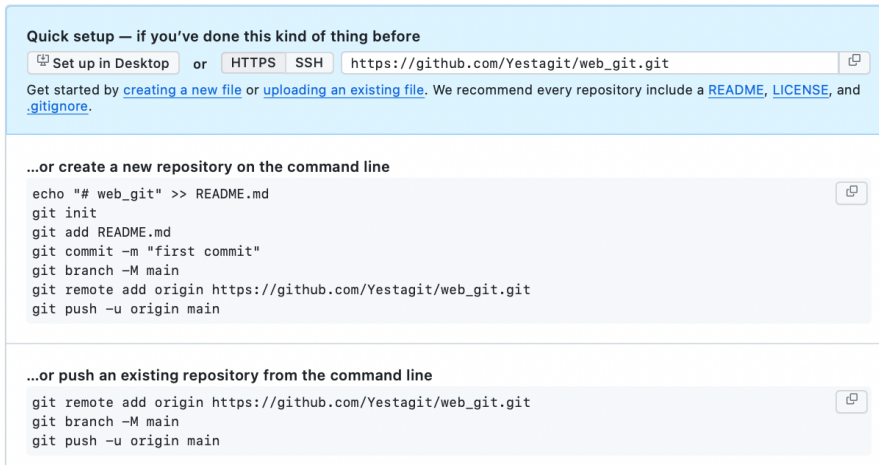
Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license
License: None

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

ⓘ You are creating a public repository in your personal account.

[Create repository](#)



2. Tambahkan URL (Menghubungkan dengan online GIT Local
Arahkan ke project GIT yang sebelumnya dibuat **web_git**

cd web_git

Hubungkan dengan Add

git remote add origin https://github.com/zidniryi/web_git.git

```
yesta@Yestas-MacBook-Air web_git % git remote add origin https://github.com/Yestagit/web_git.git
yesta@Yestas-MacBook-Air web_git %
```

#NOTE

Untuk push selanjutnya hanya perlu menggunakan perintah **git push**

3. Push Git Project (Upload Ke Github)
Selanjutnya lakukan upload Git local project ke repo online dengan syntax sederhana:
git push --set-upstream origin master

Jika berhasil maka pada terminal akan menampilkan gambar di bawah ini:

E. LATIHAN

Implementasikan git dengan project yang pernah dibuat. Lakukan push, pull, dan merge terhadap project tadi.

F. TUGAS

Buatlah repositori dari project yang akan dibuat, tambahkan user siapa saja yang bisa mengakses repo project tadi.

G. LAPORAN RESMI

Kumpulkan hasil Percobaan, Latihan dan Tugas di atas dan tambahkan analisa untuk tiap percobaan, latihan, dan tugas yang telah dibuat.

H. REFERENSI

- <https://github.com/Yestagit/cara-penggunaan-github>
- <https://konsepkoding.com>