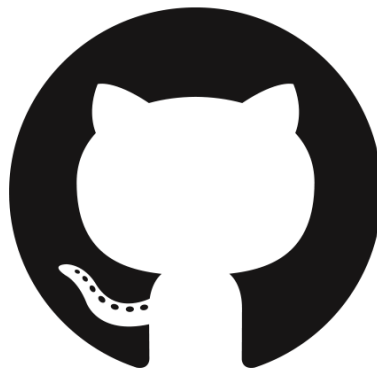


# TUTORIAL GIT & GITHUB (BASIC)

Pemrograman Berbasis WEB



## TARGET TUTORIAL :

Mahasiswa dapat mengupload **file repository** Latihan mingguan ke github, dan dapat mulai menggunakan git sebagai **sistem version control**. Pada tutorial ini belum menggunakan **GIT GUI**.

# GIT & GITHUB GUIDES

Source :

- <https://guides.github.com/>
- <https://guides.github.com/activities/hello-world/>
- <https://git-scm.com/>
- <https://git-scm.com/book/en/v2>
- [help.github.com](https://help.github.com)

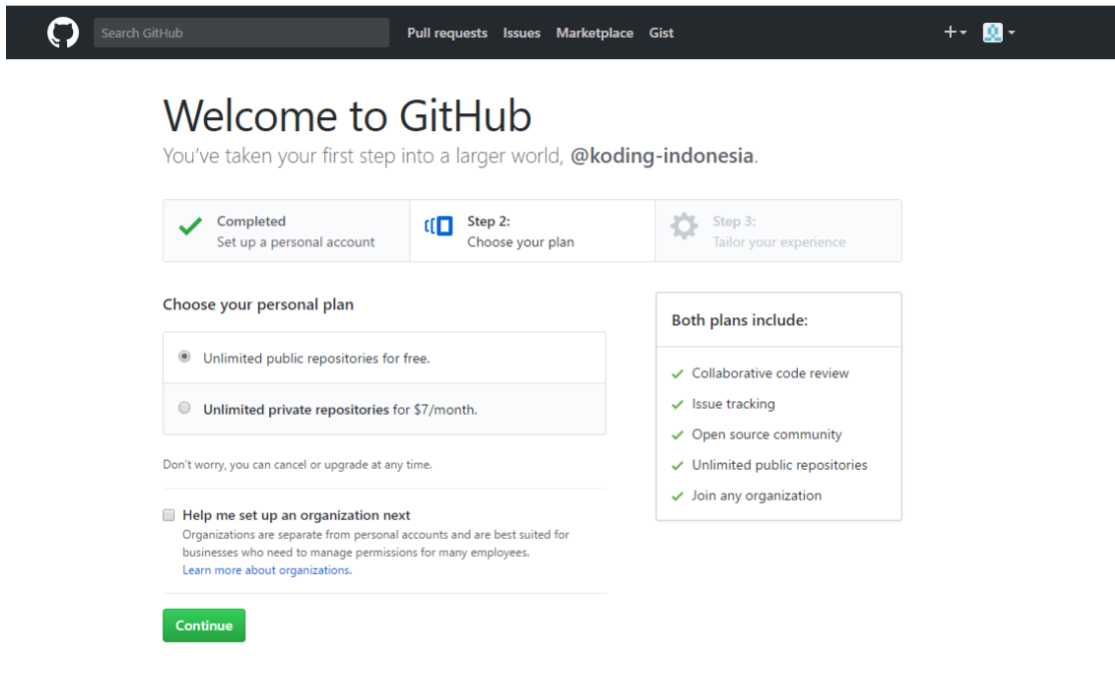
## Point yang dipelajari :

- **Registrasi** akun github
- Membuat **repository & Branch** pada github
- **Instalasi** Git
- **Perintah dasar** GIT
- **Remote** Github

# REGISTRASI AKUN GITHUB

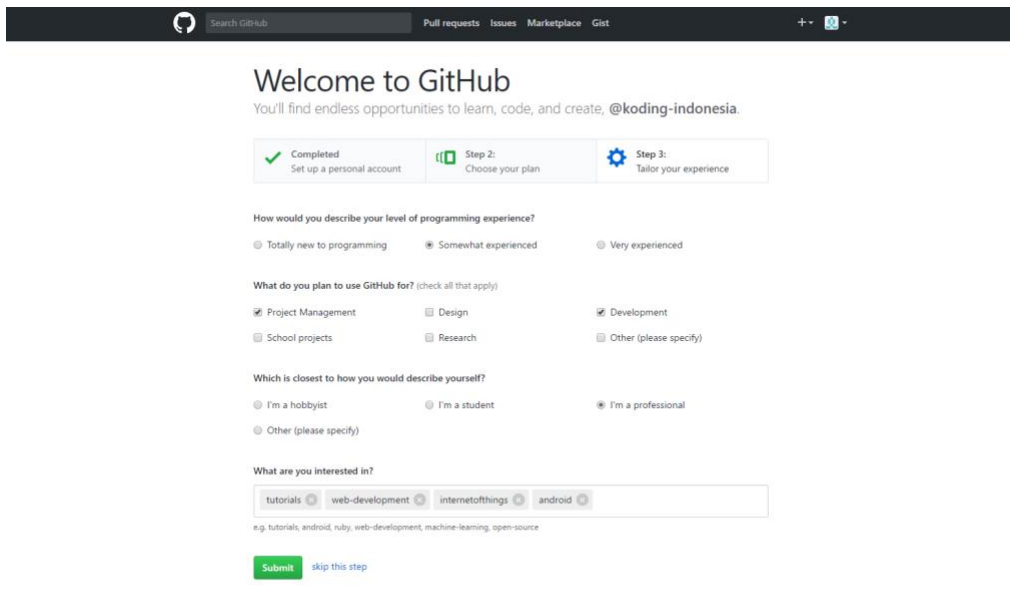
**GitHub** adalah wadah untuk proyek pengembangan perangkat lunak yang menggunakan git version control yang dapat berkolaborasi dengan berbagai developer. **GitHub** bisa disebut juga sebagai **hosting untuk penyimpanan file**.

- o Langkah pertama **Sign Up** di <https://github.com/>
- o Lengkapi form sign up dengan data yang valid, lalu klik tombol Create an account
- o Pilih Free Unlimited public repositories for free dan klik tombol continue



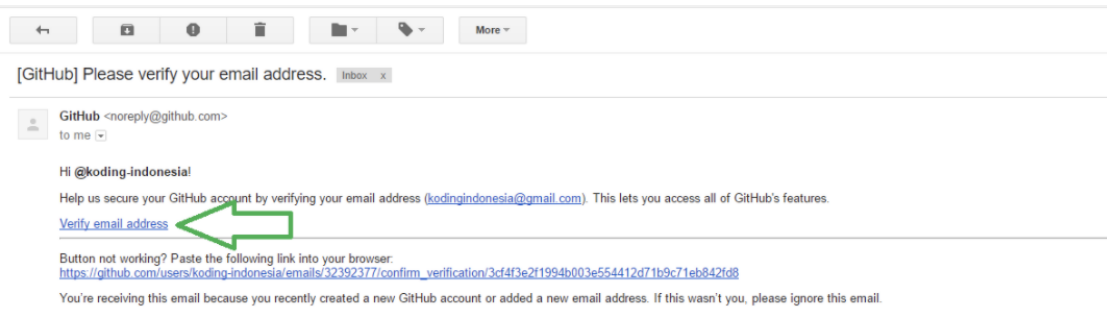
The screenshot shows the GitHub sign-up process. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Gist. Below this is a 'Welcome to GitHub' message with the user's email '@koding-indonesia'. A progress bar indicates three steps: Step 1 (Completed: Set up a personal account), Step 2 (Current: Choose your plan), and Step 3 (Tailor your experience). Under 'Choose your personal plan', there are two radio button options: 'Unlimited public repositories for free.' (selected) and 'Unlimited private repositories for \$7/month.'. A note states 'Don't worry, you can cancel or upgrade at any time.' Below this is a checkbox for 'Help me set up an organization next' with a link to 'Learn more about organizations.' A green 'Continue' button is at the bottom.

- o Lengkapi pilihan seperti gambar di bawah ini atau sesuai dengan kebutuhan.

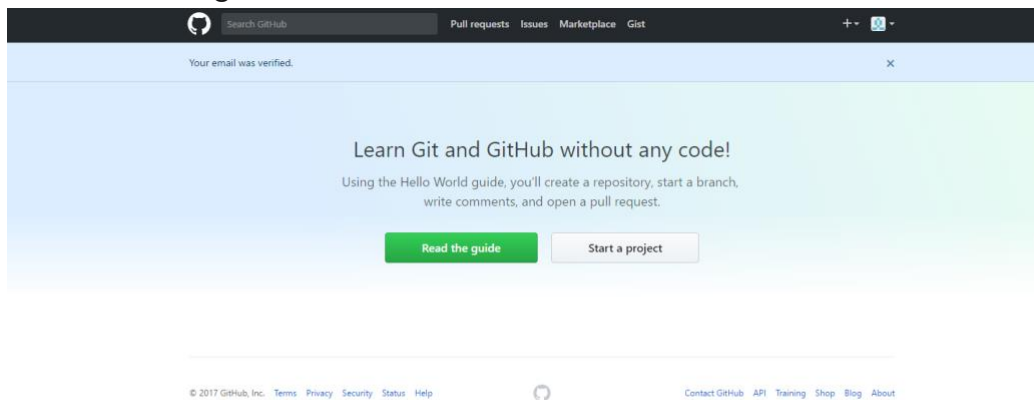


The screenshot shows the 'Tailor your experience' step of the GitHub sign-up process. It features the same navigation bar and welcome message as the previous step. The progress bar shows Step 1 (Completed), Step 2 (Choose your plan), and Step 3 (Current: Tailor your experience). The main content area asks 'How would you describe your level of programming experience?' with three radio button options: 'Totally new to programming', 'Somewhat experienced' (selected), and 'Very experienced'. Below this, it asks 'What do you plan to use GitHub for? (check all that apply)' with six checkboxes: 'Project Management', 'Design', 'Development' (checked), 'School projects', 'Research', and 'Other (please specify)'. The next question is 'Which is closest to how you would describe yourself?' with three radio button options: 'I'm a hobbyist', 'I'm a student', and 'I'm a professional' (selected). The final question is 'What are you interested in?' with a text input field containing 'tutorials', 'web-development', 'internetofthings', and 'android'. A small note below the input field says 'e.g. tutorials, android, ruby, web-development, machine-learning, open-source'. At the bottom are 'Submit' and 'skip this step' buttons.

- o Buka email dan klik link verifikasi untuk mengaktifkan akun.



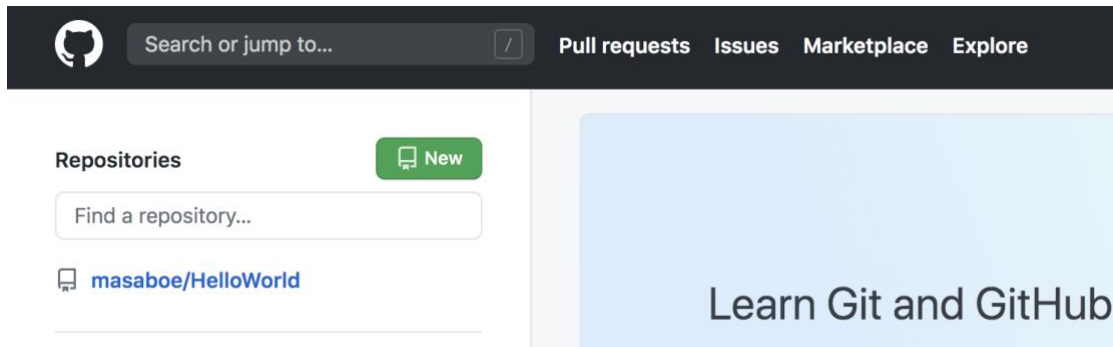
- o Selamat akun github telah berhasil dibuat



- o URL **github** adalah <http://github.com/username>

# MEMBUAT REPOSITORY & BRANCH PADA GITHUB

- Pilih New Repository




- Isikan Form repository dengan isian kurang lebih seperti berikut :

## Create a new repository

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

Owner \*

 masaboe ▾

Repository name \*

/ webdas ✓

Great repository names are short and memorable. Need inspiration? How about **stunning-lamp**?

Description (optional)

☒  **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:

Skip this step if you're importing an existing repository.

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**

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

☐ **Choose a license**

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

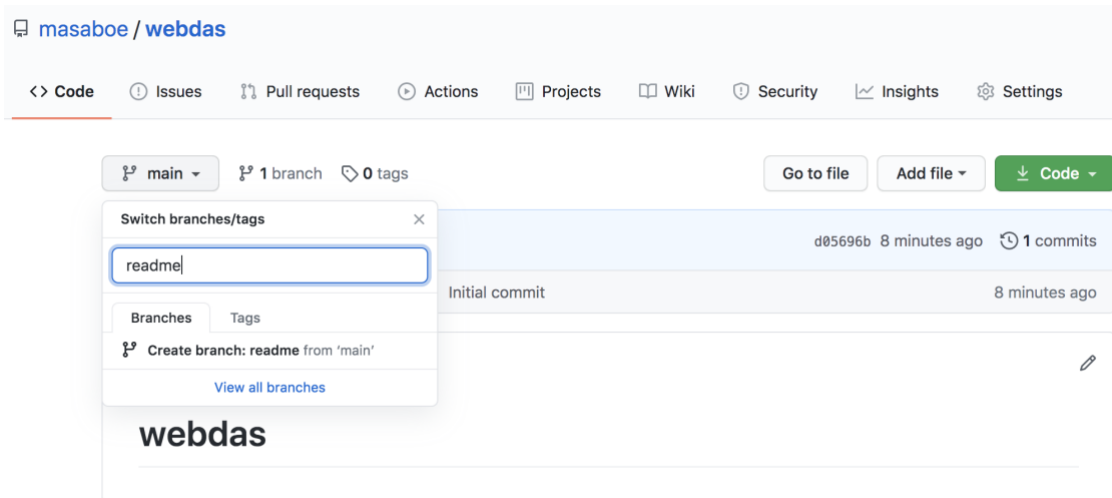
This will set  **main** as the default branch. Change the default name in your [settings](#).

Create repository

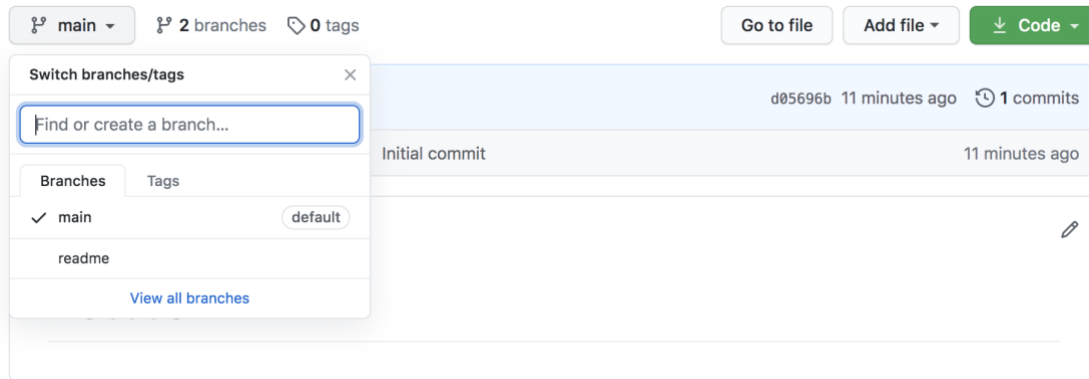
- Lalu **Klik** Create Repository
- Membuat Branch (*Optional*) : Satu repository bisa berisi banyak branch.

## Sekilas mengenai Branch :

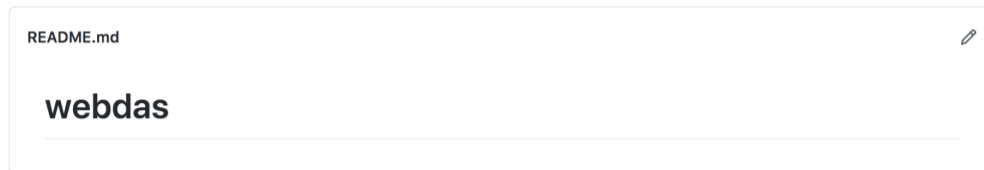
- **Repository** dapat memiliki beberapa branch. Secara default, branch yang kita gunakan adalah branch **master**. Setiap branch memiliki data commit masing-masing. Developer dapat menyunting kode di branch lain kemudian jika telah selesai bisa langsung menggabungkan (**merge**) branch-nya dengan branch **master**.
  - Apa kegunaan branch?  
Biasanya branch digunakan ketika kita ingin **menambah** suatu **fitur** atau memperbaiki **bug**. **Contohnya** ketika kita ingin menambah fitur **otentikasi**, maka kita dapat membuat branch **otentikasi** dan jika sudah selesai langsung gabungkan dengan **branch master**.
  - Kenapa harus branch baru?  
Tanpa branch baru tidak masalah, tapi kalau langsung edit di branch **master**, history commit-nya tercampur dengan yang lain. *Jika setiap fitur dibuat di branch masing-masing, pengembangan dan perbaikan fitur akan lebih mudah karena developer dapat melihat history commit fitur tersebut dari awal sampai ketika digabungkan dengan master.*
- o Pembuatan Branch, klik panah pada menu branch **main/master** (default branch) kemudian tuliskan nama branch baru dan tekan **enter**.



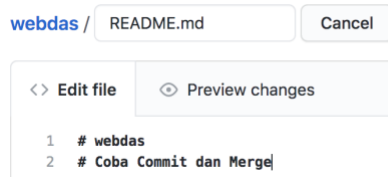
- Apabila kita klik ulang panah akan muncul 2 branch aktif (main dan readme) sebagai berikut :



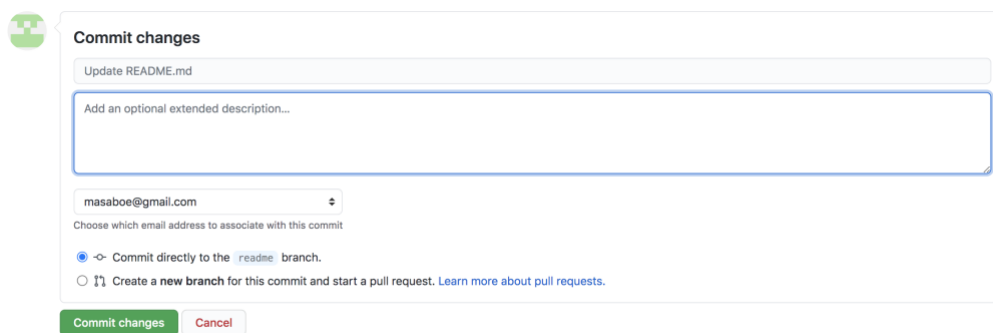
- **Proses commit**
  - Pertama Aktifkan branch readme kemudian buka dan ubah file README.md (klik icon edit)



- Tuliskan item perubahan



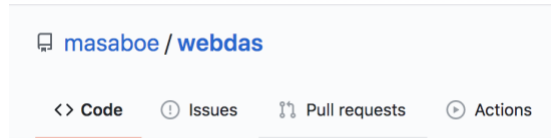
- Tambahkan komentar perubahan dan klik tombol "Commit Change".



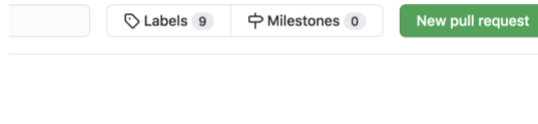
- **Pull Request** (untuk melakukan pull perubahan sebagai tahap awal merge / penggabungan perubahan file dari branch readme ke branch **main**)



- o Klik tab Pull requests



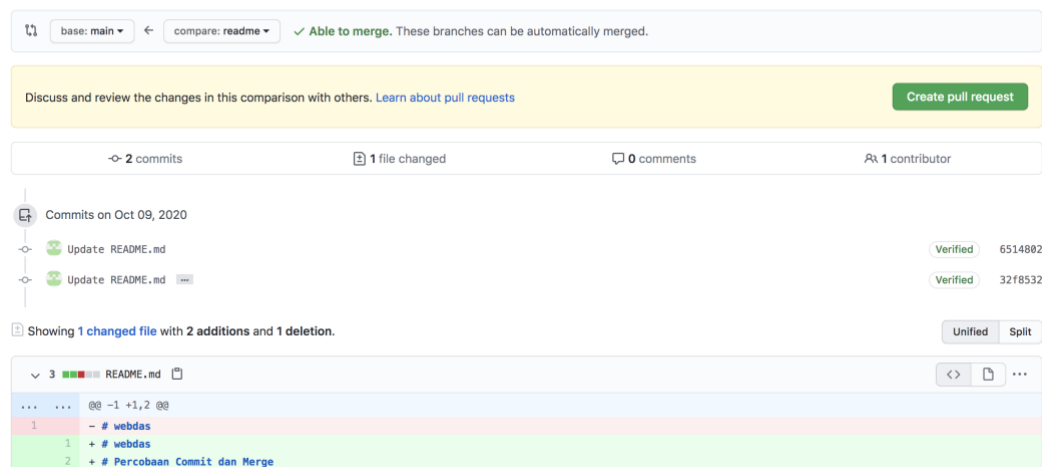
- o Klik Tombol New Pull Request



- o Klik Compare, pilih branch readme (compare) dan main (base) kemudian klik tombol Create Pull Request

#### Comparing changes

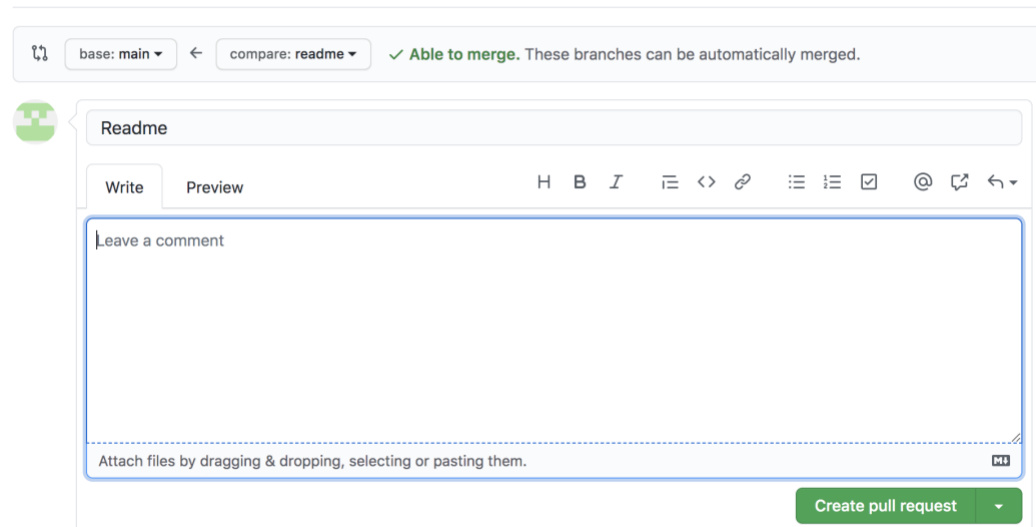
Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



- o Isikan komentar proses pull request, kemudian klik tombol Create pull request

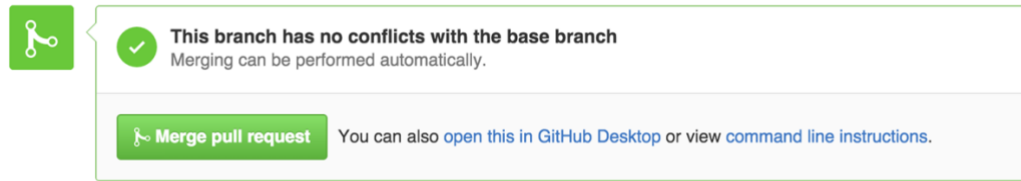
#### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

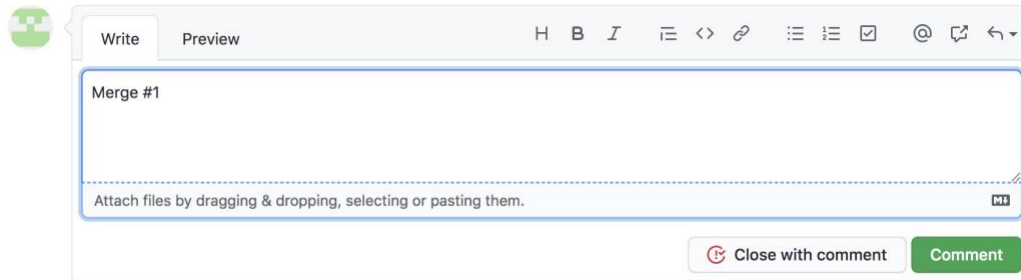


Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

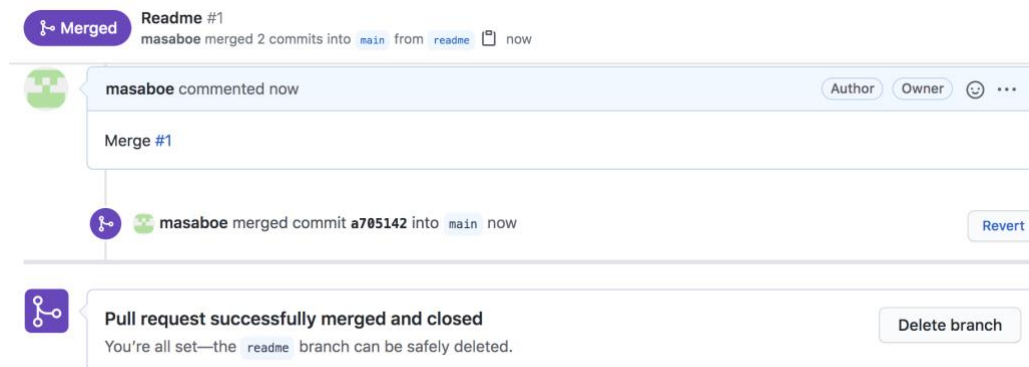
- Merge Pull Request (Proses Akhir Merge / Penggabungan perubahan file dari branch readme ke branch main)
- Klik Tombol Merge pull request



- Isikan Komentar (jika diperlukan)



- Klik Tombol Confirm merge dan branch **main** posisi sudah terupdate sesuai dengan perubahan terakhir di branch **readme**



# INSTALASI GIT

Git diciptakan oleh Linus Torvalds. Git digunakan untuk menyimpan perubahan source code, yang disebut juga sistem version control. Git dapat digunakan sendiri maupun untuk kolaborasi dengan team. Git bersifat terdistribusi dan individual, sehingga jika salah satu server mati, developer dapat menggunakan server lain dengan mudah. Jika developer tidak terhubung dengan internet, git masih dapat digunakan secara offline, bahkan developer bisa melihat history kode-kodenya tanpa perlu terhubung ke remote server.

Instalasi GIT menyesuaikan Sistem Operasi yang dipakai, berikut gambaran untuk tahap instalasi GIT :

## WINDOWS

Download GIT untuk windows melalui url dibawah ini:

<https://git-scm.com/download/win>

setelah file selesai di download double click dan ikuti langkah-langkannya dengan klik **Next** sampai **Finish**, disarankan untuk klik **Next** sampai finisih agar standard GIT nya bagi pemula.



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

```
git -version
```

contoh :

```
[192:~ masaboe$ git --version  
git version 2.28.0
```

Jika berhasil akan muncul versi git sesuai dengan yang diinstall.

## MAC OS

Bagi pengguna Mac OS, ada beberapa acara install Git di Mac OS.

- **Install Xcode dan Git akan otomatis terinstall (sudah otomatis)**
- **Install Git Dengan Brew**

Buka terminal kemudian tuliskan perintah berikut :

```
brew install git
```

Tunggu proses instalasi sampai selesai, kemudian cek versi git apakah sudah berhasil terinstall sama dengan Langkah pada windows diatas.

## LINUX

Untuk instalasi pada linux tergantung distribusi linux yang digunakan, Pada Fedora dengan menggunakan yum, dapat menuliskan perintah dibawah ini :

```
$ yum install git
```

Jika menggunakan Ubuntu, dapat menggunakan apt-get dengan menuliskan perintah dibawah ini :

```
$ apt-get install git
```

Lebih lengkap dapat mengikuti dokumentasi perintah instalasi melalui url <https://git-scm.com/download/linux> .

Setelah selesai proses instalasi jalankan perintah `git - version` untuk memastikan pemasangan git berhasil.

## KONFIGURASI AWAL

Setelah Proses Instalasi GIT berhasil lakukan beberapa perintah dibawah sebagai tahap awal konfigurasi :

```
git config --global user.name "username"
```

```
git config --global user.email "email "
```

Karena kita akan mengkolaborasikan git dengan remote server github, maka username dan email isikan sesuai **username dan email akun github.com**

Untuk melihat konfigurasi pada GIT gunakan perintah :

`git config --list`

```
192:hellogit masaboe$ git config --list
credential.helper=osxkeychain
core.excludesfile=/Users/masaboe/.gitignore_global
difftool.sourcetree.cmd=opendiff "$LOCAL" "$REMOTE"
difftool.sourcetree.path=
mergetool.sourcetree.cmd=/Applications/Sourcetree.app/Contents/Resources/opendiff-w.sh "$LOCAL" "$REMOTE" -ancestor "$BASE" -merge "$MERGED"
mergetool.sourcetree.trustexitcode=true
user.name=masaboe
user.email=masaboe@gmail.com
commit.template=/Users/masaboe/.stCommitMsg
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
core.ignorecase=true
core.precomposeunicode=true
192:hellogit masaboe$
```

## SETUP TAMBAHAN

Masih di Git Bash, ketik perintah seperti di bawah ini lalu klik enter

`ssh-keygen`

### Penting:

Saat diminta menginputkan directory dan passphrase bisa langsung di **enter** saja.

Dan akan muncul seperti tampilan dibawah ini (contoh pada windows):

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/WebDev/.ssh/id_rsa):
Created directory '/c/Users/WebDev/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/WebDev/.ssh/id_rsa.
Your public key has been saved in /c/Users/WebDev/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:tDA0QqHdhQqIXCma4zUrTwCpY80umVHzAmaMHIXfRYI WebDev@WebDev-PC
The key's randomart image is:
+---[RSA 2048]---+
|X. *+o++       |
|*AE=oo         |
|B+Xo= o .      |
|*ooO + + .     |
|ooO + . 5      |
| * +           |
| o             |
|               |
+----[SHA256]-----+
```

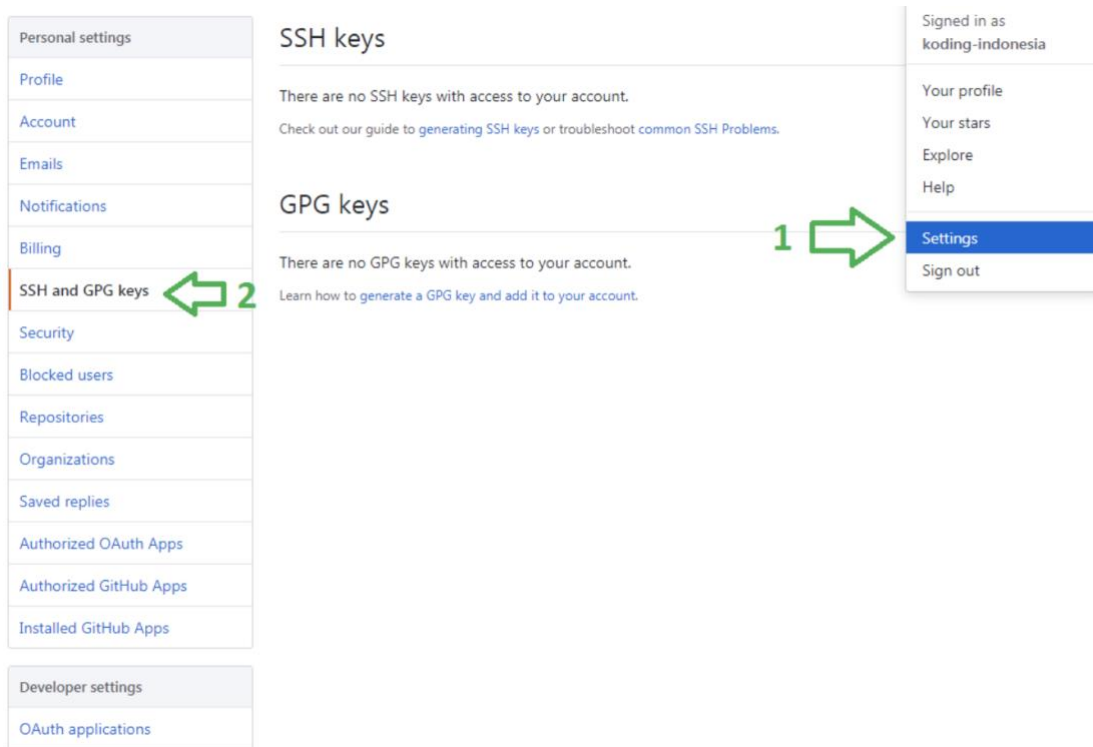
Kemudian Buka folder .ssh pada directory C -> Users -> Nama User PC -> .ssh, untuk OS lain cek direktori/folder aktif pada saat perintah dijalankan.

Buka folder **.ssh** dan buka file **id\_rsa.pub** dengan **text editor** (bisa menggunakan notepad ataupun editor text lainnya)

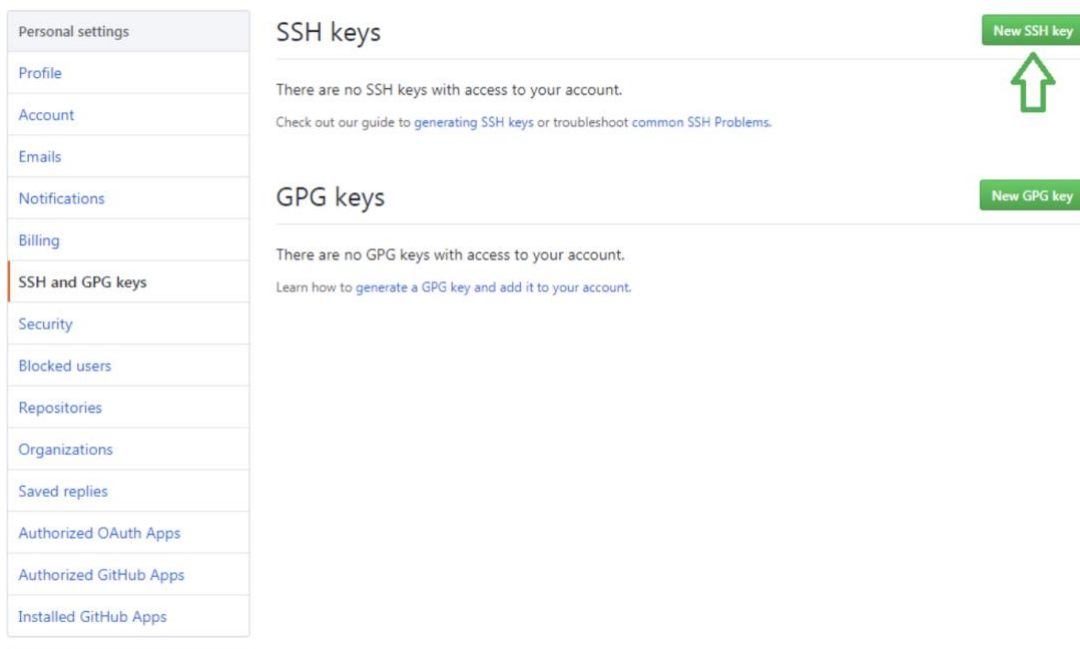
id_rsa	14/07/2017 17:12	File	2 KB
id_rsa.pub	14/07/2017 17:12	Microsoft Publish...	1 KB

Copy ssh rsa dan paste ke github (Semua isian yang ada di file **id\_rsa.pub**).

Login ke akun GitHub, dan buka menu Settings dan klik menu SSH and GPG Keys (sesuai gambar dibawah ini)



Klik tombol New SSH key



Lengkapi form SSH keys sebagai berikut :

1. Title: isi dengan nama apa saja. Contohnya: pc-rendy-ganteng
2. Key: copy paste ssh yang ada di komputer lokal ke form key GitHub ini.

3. Selanjutnya klik tombol Add SSH key.

SSH keys

New SSH key

There are no SSH keys with access to your account.

Title

pc-kantor

Key

ssh-rsa  
 AAAAB3NzaC1yc2EAAAADAQABAAQCCgQKCBz3LWMB/E5/G3+888IT+OB+QQ/Q+AGN15+M7+Jb  
 Agf5anYb7U9MkgU5jyn1GCLu776UdVCHg.mC.X+97DB,N8WJA+G+JZ-LN8/KT+G-NPC22CjUNWT  
 eRo32BdLW2ZZbzT4cMG2ahjGg8hiAzicr6TGpREEqvYh8A6ZDSwqeuyzXDlkjQyC7VMlawy6LZINX3i0SczigrnJ9iq  
 xSPlvomduSuv8m7PjGcWHPei1x3zi/jjNwHLFUT66K/zHBf+linuW4yNOFNopzJPJ56ySRyzAX/Gf5CbdWTB0oxRg2  
 19KB3QwGFM548FEKtFj WebDev@WebDev-PC

Add SSH key

SSH Keys berhasil terhubung ke github.

Personal settings

Profile

Account

Emails

Notifications

Billing


SSH and GPG keys

Security

## SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

SSH

pc-kantor

Fingerprint: 31:81:67:f0:2f:26:84:cb:90:18:56:e7:59:6b:90:f3

Added on Jul 14, 2017

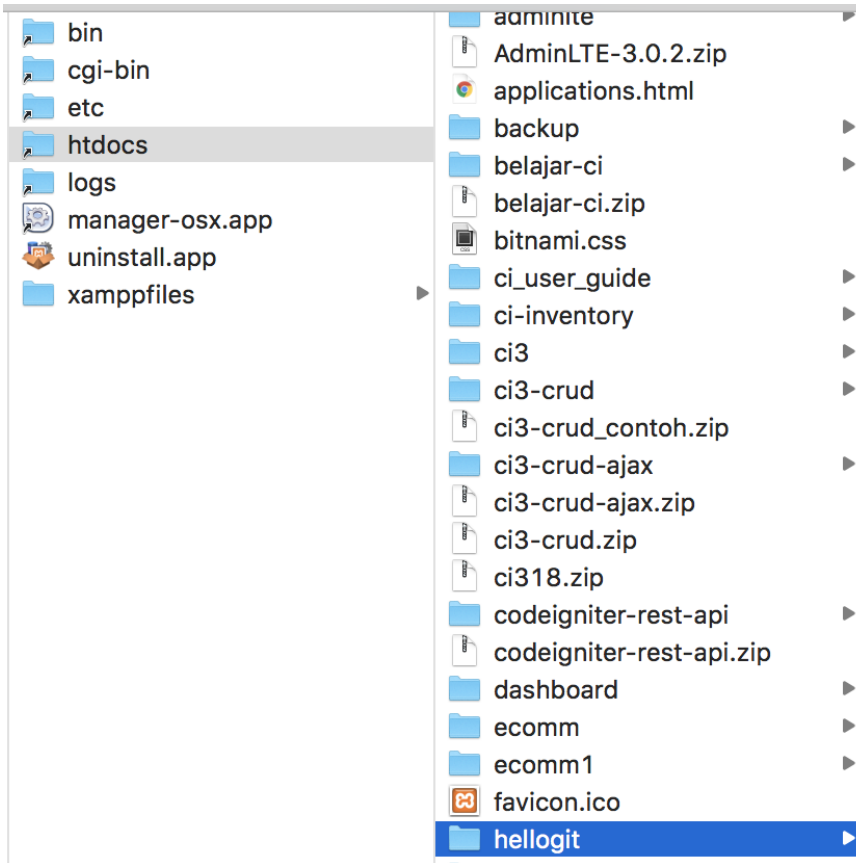
Never used

Delete

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH Problems](#).

## PERINTAH DASAR GIT DAN REMOTE GITHUB

Langkah pertama untuk memulai GIT adalah menentukan repository / direktori git, buat direktori yang dijadikan tempat untuk penyimpanan file, contoh sebagai berikut (Mac OS), membuat file hellogit sebagai wadah repository git :



Aktifkan kursor didalam folder hello git

```
192:hellogit masaboe$
```

Kemudian jalankan perintah dibawah ini:

```
[192:hellogit masaboe$ git init
```

Git init hanya dijalankan satu kali, jika sudah pernah melakukan perintah git init sebelumnya akan muncul peringatan seperti dibawah ini :

```
[192:hellogit masaboe$ git init
Reinitialized existing Git repository in /Applications/XAMPP/xamppfiles/htdocs/hellogit/.git/
192:hellogit masaboe$
```

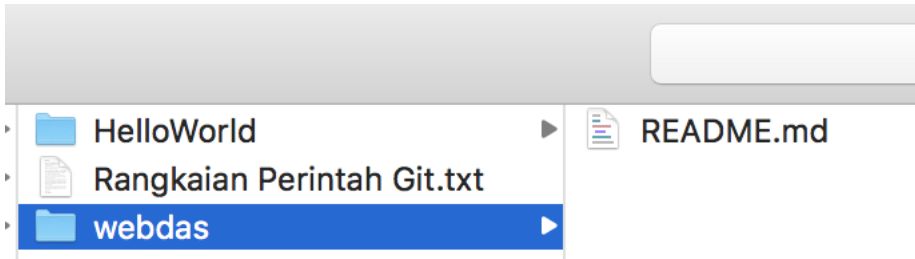
Langkah selanjutnya bisa melakukan clone file pada github (Latihan pembuatan repository sebelumnya), dengan perintah seperti berikut :

```
git clone https://github.com/<akun>/<repository>
```

Tekan enter dan proses cloning akan mengunduh semua file pada repository github yang ada, contoh proses cloning seperti dibawah ini:



```
[192:hellogit masaboe$ git clone https://github.com/masaboe/webdas
Cloning into 'webdas'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (10/10), 2.39 KiB | 306.00 KiB/s, done.
192:hellogit masaboe$
```



Proses cloning berhasil, repository dan file sudah masuk di PC / Laptop local,  
Jalankan perintah :

```
git status
```

atau

```
git status -s
```

untuk mengetahui kondisi terakhir maupun status perubahan file, jika pada repository GIT memiliki lebih dari 1 **branch** cek posisi branch yang ada, karena proses commit hanya berpengaruh pada branch yang aktif saja.

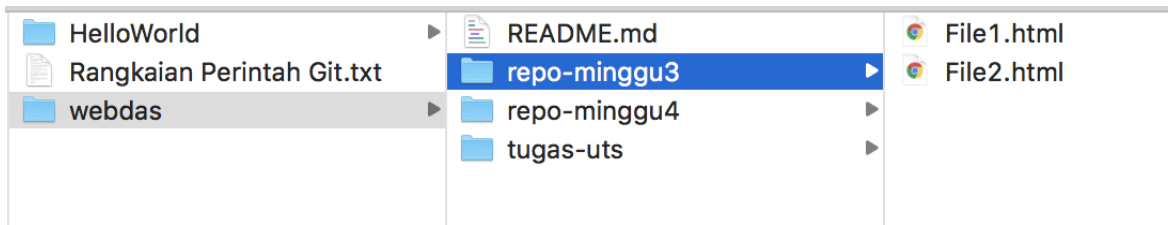
```
[192:hellogit masaboe$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    HelloWorld/
    Rangkaian Perintah Git.txt
    webdas/

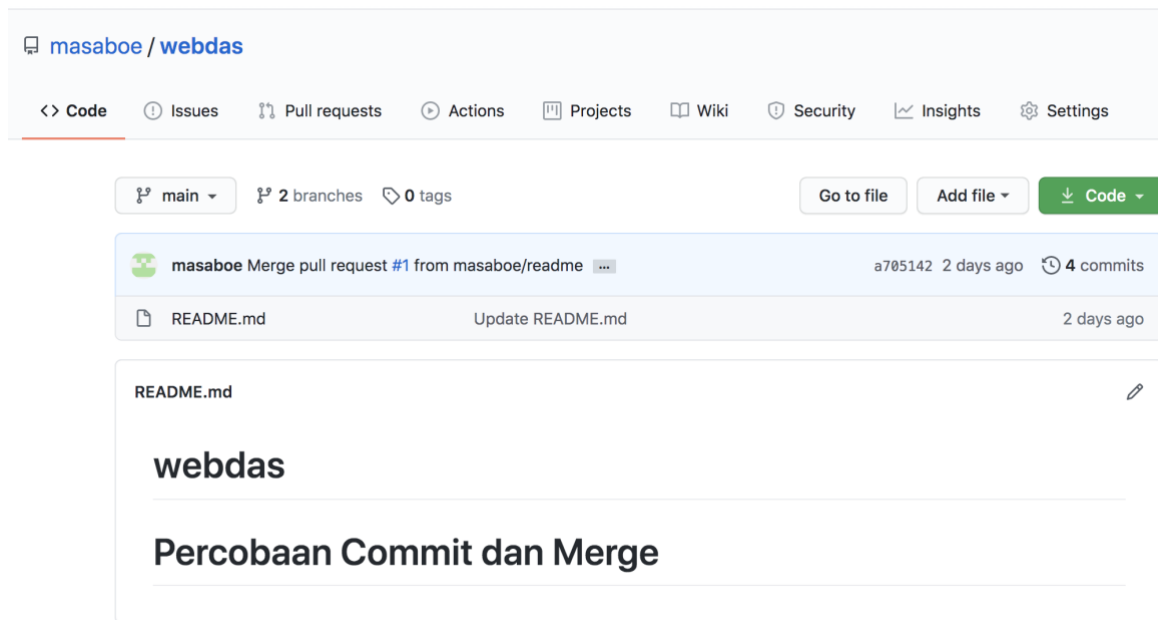
nothing added to commit but untracked files present (use "git add" to track)
[192:hellogit masaboe$ git status -s
?? HelloWorld/
?? Rangkaian Perintah Git.txt
?? webdas/
192:hellogit masaboe$
```

Berikutnya tambahkan folder dan file hasil clone sesuai dengan kebutuhan, contoh seperti dibawah :



Kedepan di dalam folder / repository **webdas** akan terisi file repository mingguan maupun tugas akhir (UTS / UAS), isikan beberapa file untuk coba proses remote github.

Berikut posisi repository webdas pada github sebelum dilakukan push dari git lokal. Belum ada direktori maupun file.



untuk menambahkn semua direktori dan file baru yang ada di dalam folder webdas atau untuk menambahkan file, folder dan file dengan ekstensi tertentu bisa dengan menggunakan perintah berikut :

```
git add .  
// menambahkan semua folder dan file  
git add <nama file>  
// hanya 1 file yang ditambahkan  
git add <nama folder>/\*.html  
// hanya untuk 1 folder dan file dengan akhiran .html  
git add <nama folder>/\*.*  
// hanya untuk 1 folder dan semua file di dalamnya
```

Pastikan posisi **direktori aktif** adalah sudah masuk ke dalam folder webdas.

seperti contoh dibawah ini:

```
[192:hellogit masaboe$ cd webdas/
[192:webdas masaboe$ git add .
[192:webdas masaboe$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   repo-minggu3/File1.html
    new file:   repo-minggu3/File2.html
```

```
192:webdas masaboe$ █
```

Lihat posisi branch aktif untuk proses commit dan push ke remote server, diatas posisi branch aktif adalah branch main, untuk membuat branch baru bisa gunakan perintah berikut :

```
git branch webdas
```

setelah itu pindah / mengaktifkan branch webdas menggunakan perintah dibawah ini:

```
git checkout -b webdas
```

disini posisi branch aktif adalah webdas

```
[192:webdas masaboe$ git checkout -b webdas
Switched to a new branch 'webdas'
[192:webdas masaboe$ git status
On branch webdas
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   repo-minggu3/File1.html
    new file:   repo-minggu3/File2.html
```

Sekarang posisi branch adalah webdas, jika hanya akan menggunakan 1 branch main abaikan proses pembuatan dan pindah branch diatas.

Untuk proses penggabungan branch pada git dapat menggunakan perintah :

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

terlebih dahulu kita pindah ke branch tujuan merge.

Langkah berikutnya adalah proses commit dengan menggunakan perintah sebagai berikut :

```
git commit -m "catatan commit"
```

```
[192:webdas masaboe$ git commit -m "ubah repo webdas untuk repominggu3 dan 4 serta tugas uts"
[webdas fa56eab] ubah repo webdas untuk repominggu3 dan 4 serta tugas uts
 2 files changed, 22 insertions(+)
 create mode 100644 repo-minggu3/File1.html
 create mode 100644 repo-minggu3/File2.html
192:webdas masaboe$ █
```

Proses commit berhasil, berikutnya untuk mensinkronkan file local ke github gunakan perintah :

```
git push origin <nama branch>
```

```
[192:webdas masaboe$ git push origin webdas
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 482 bytes | 482.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'webdas' on GitHub by visiting:
remote:   https://github.com/masaboe/webdas/pull/new/webdas
remote:
To https://github.com/masaboe/webdas
 * [new branch]      webdas -> webdas
192:webdas masaboe$
```

Proses push berhasil, berikutnya cek repository github.

🔔 webdas had recent pushes 9 minutes ago

Compare & pull request

Pastikan muncul notifikasi diatas, kemudian klik tombol “compare & pull request”

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: webdas ✓ Able to merge. These branches can be automatically merged.


@ma  
abn

ubah repo webdas untuk repominggu3 dan 4 serta tugas uts

Write Preview

H B I ≡ < > 🔗 ≡ ≡ ☑ @ ↻ ↶

updated|

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request ▼

Klik tombol “Create pull request”

Add more commits by pushing to the **webdas** branch on **masaboe/webdas**.



**Continuous integration has not been set up**

[GitHub Actions](#) and [several other apps](#) can be used to automatically catch bugs and enforce style.



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Lalu tekan tombol “Merge pull request”

**masaboe / webdas**

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

main ▾ **webdas / repo-minggu3 /**

**masaboe** ubah repo webdas untuk repominggu3 dan 4 serta tugas uts

..

File1.html	ubah repo webdas untuk repominggu3 dan 4 serta tugas uts
File2.html	ubah repo webdas untuk repominggu3 dan 4 serta tugas uts

Posisi repositori github sudah terupdate. Dan karena menjalankan perintah merge pull request maka branch main juga secara otomatis terupdate.

**Catatan :**

- Pelajari github dan git lebih lanjut, dapat menggunakan GIT GUI untuk lebih memudahkan manajemen GIT lokal dan proses push server.

# DEFINISI DASAR GIT

- Kondisi File GIT
  - Modified
    - Kondisi dimana revisi atau perubahan sudah dilakukan, tetapi belum ditandai dan belum disimpan di *version control*.
  - Staged
    - Kondisi dimana revisi sudah ditandai, tetapi belum disimpan di *version control*. Untuk mengubah kondisi file dari *modified* ke *staged* gunakan perintah **git add** <nama folder / nama file>
  - Committed
    - Kondisi dimana revisi sudah disimpan di *version control*. perintah untuk mengubah kondisi file dari *staged* ke *committed* adalah **git commit**.
- Istilah dasar
  - Repository : database yang menyimpan semua history/riwayat perubahan.
  - snapshot : potret kondisi file dan folder pada saat tertentu.
  - commit : snapshot yang disimpan di repository.
  - branch : serangkaian commit yang berkaitan sehingga kalau digambar seperti garis lurus berisi banyak commit. Satu repository bisa berisi banyak branch.
  - master : nama branch default yang diberikan git pada waktu kita membuat repository. Branch master ini tidak istimewa. Dia bisa dihapus dan direname sesuka hati.
  - hash : Git menyimpan informasi commit sebagai hash SHA1, misalnya **24b9da6552252987aa493b52f8696cd6d3b00373**. Namun terkadang ditampilkan versi pendeknya, misalnya **24b9da6**
  - head : ujung branch, commit terbaru di dalam branch.
  - HEAD : head yang sedang aktif. Walaupun satu repository bisa memiliki banyak branch, tapi cuma satu yang aktif.
  - working folder : folder berisi file dan folder tempat kita bekerja. Biasanya working folder berisi banyak file source code untuk aplikasi yang sedang kita buat. Git memantau working folder ini, dan bisa mengetahui file dan folder mana yang sudah berbeda dari posisi commit terakhir. Perbedaan atau perubahan ini bisa disimpan menjadi commit baru, atau dikembalikan ke kondisi sebelum diubah.
  - staging area : snapshot dari working folder yang akan kita simpan pada saat commit. Ini adalah fitur unik Git yang tidak dimiliki version control lain. Dengan adanya staging area, kita bisa memilih perubahan mana yang akan di-commit dan mana yang tidak.