

GIT (Version Control System)

Git adalah perangkat lunak yang disediakan secara open source (kode sumber terbuka). Git bertujuan untuk mendukung kolaborasi antar anggota tim serta mengontrol setiap versi perubahan dalam sebuah pekerjaan pengembangan aplikasi ataupun pada bidang lainnya. Git pertama kali dibangun oleh pengembang kernel Linux, yaitu Linus Torvalds pada tahun 2005.

Sejarah Git dimulai dari pengembangan kernel Linux yang merupakan proyek perangkat lunak open source dalam lingkup yang sangat besar. Dalam kurun waktu tahun 1991-2002, pengembangan kernel Linux dihabiskan dengan membuat patch dan berkas terarsip untuk melakukan pemeliharaan source code. Pada tahun 2002, proyek kernel Linux mulai menggunakan Distributed version control systems (DVCS) yang paten bernama BitKeeper.

Sejak kelahirannya pada 2005, Git telah berevolusi dan berkembang sehingga menjadi platform yang mudah digunakan dan memiliki kualitas yang sangat baik. Git sangat cepat dan efisien pada proyek-proyek besar. Git juga memiliki sistem percabangan yang hebat untuk pengembangan non-linear.



Git merupakan salah satu tool yang sering digunakan dalam proyek pengembangan software, membantu developer mencatat semua perubahan yang dilakukan pada file sehingga semua riwayatnya akan terekam dan bisa dilihat kembali nanti. Tak hanya itu, sistem ini juga mencatat siapa saja yang membuat perubahan serta memulihkan kode yang telah dihapus atau dimodifikasi .



Saat developer membuat proyek baru, mereka selalu dan akan terus-menerus melakukan pembaruan terhadap kodenya. Bahkan, setelah proyeknya online, developer tetap harus memperbarui versinya, memperbaiki bug, menambahkan fitur baru, dan lain sebagainya. Berikut ini istilah-istilah yang terdapat dalam Git:

- Repository

Repository merupakan wadah penyimpanan kumpulan file dari berbagai versi proyek yang berbeda. Kumpulan file ini diimpor melalui repository ke server lokal pengguna untuk pembaruan dan modifikasi lebih lanjut dalam konten file.

- BRANCH

Branch merupakan salinan dari repositori. Anda bisa menggunakan branch ketika akan melakukan suatu pengembangan atau development secara terpisah. Pekerjaan atau task yang Anda kerjakan di branch tidak akan memengaruhi repositori pusat atau branch lainnya. Jika pengembangannya sudah selesai, Anda bisa menggabungkan branch saat ini ke branch lainnya dan juga repositori pusat

- Fork

Fork merupakan salinan dari sebuah repository orang lain yang tersimpan di dalam sebuah repository Git. Forking repository memungkinkan Anda untuk bereksperimen dengan bebas dan membuat perubahan tanpa memengaruhi repository utamanya.

- Clone

Clone hampir sama dengan Fork, keduanya sama-sama melakukan salinan pada repository. Yang berbeda adalah saat melakukan git fork, repository akan disimpan berada di dalam server. Sedangkan untuk git clone, hasil salinannya disimpan dalam penyimpanan lokal.

- Commit

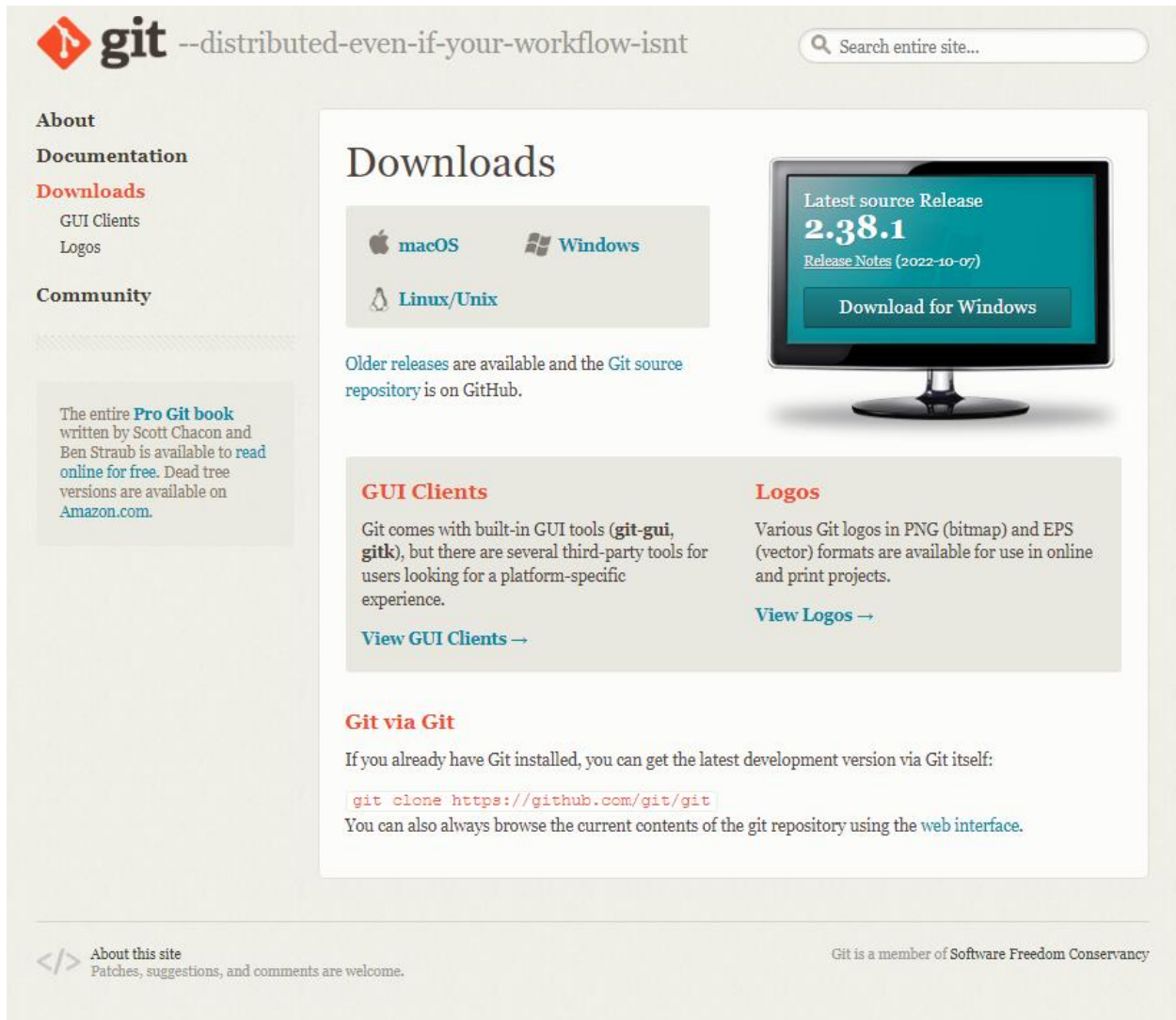
Commit merupakan cuplikan perubahan (snapshot) dari repository Anda pada waktu tertentu. Dengan begitu, setiap pekerjaan yang telah selesai, dapat disimpan ke dalam repository Anda.

- Push

push merupakan tindakan lanjutan dari perintah Git commit. Git push mengunggah semua commit cabang (branch commit) pada repository lokal ke repository server. Secara default, Git push hanya memperbarui cabang yang sesuai pada remote.

Cara Install Git

1. Unduh git melalui situs: [Git - Downloads \(git-scm.com\)](https://git-scm.com), unduh Git sesuai dengan arsitektur komputer kita. Kalau menggunakan 64bit, unduh yang 64bit. Begitu juga kalau menggunakan 32bit.



2. Silahkan klik 2x file instaler Git yang sudah diunduh.
3. Maka akan muncul informasi lisensi Git, klik Next > untuk melanjutkan.
4. Selanjutnya menentukan lokasi instalasi. Biarkan saja apa adanya, kemudian klik Next >
5. Selanjutnya pemilihan direktori start menu, klik Next >.
6. Selanjutnya pengaturan PATH Environment. Pilih yang tengah agar perintah git dapat di kenali di Command Prompt(CMD). Setelah itu klik Next >
7. Selanjutnya konversi line ending. Biarkan saja seperti ini, kemudian klik Next >.
8. Selanjutnya pemilihan emulator terminal. Pilih saja yang bawah, kemudian klik Next >
9. Selanjutnya pemilihan opsi ekstra. Klik saja Next >.
10. Selanjutnya pemilihan opsi eksperimental, langsung saja klik Install untuk memulai instalasi.

11. Tunggu beberapa saat, instalasi sedang dilakukan.
12. Setelah selesai, kita bisa langsung klik Finish.
13. Git sudah terinstal di Windows. Untuk mencobanya, silahkan buka CMD atau PowerShell, kemudian ketik perintah `git --version`

Konfigurasi Awal yang Harus Dilakukan

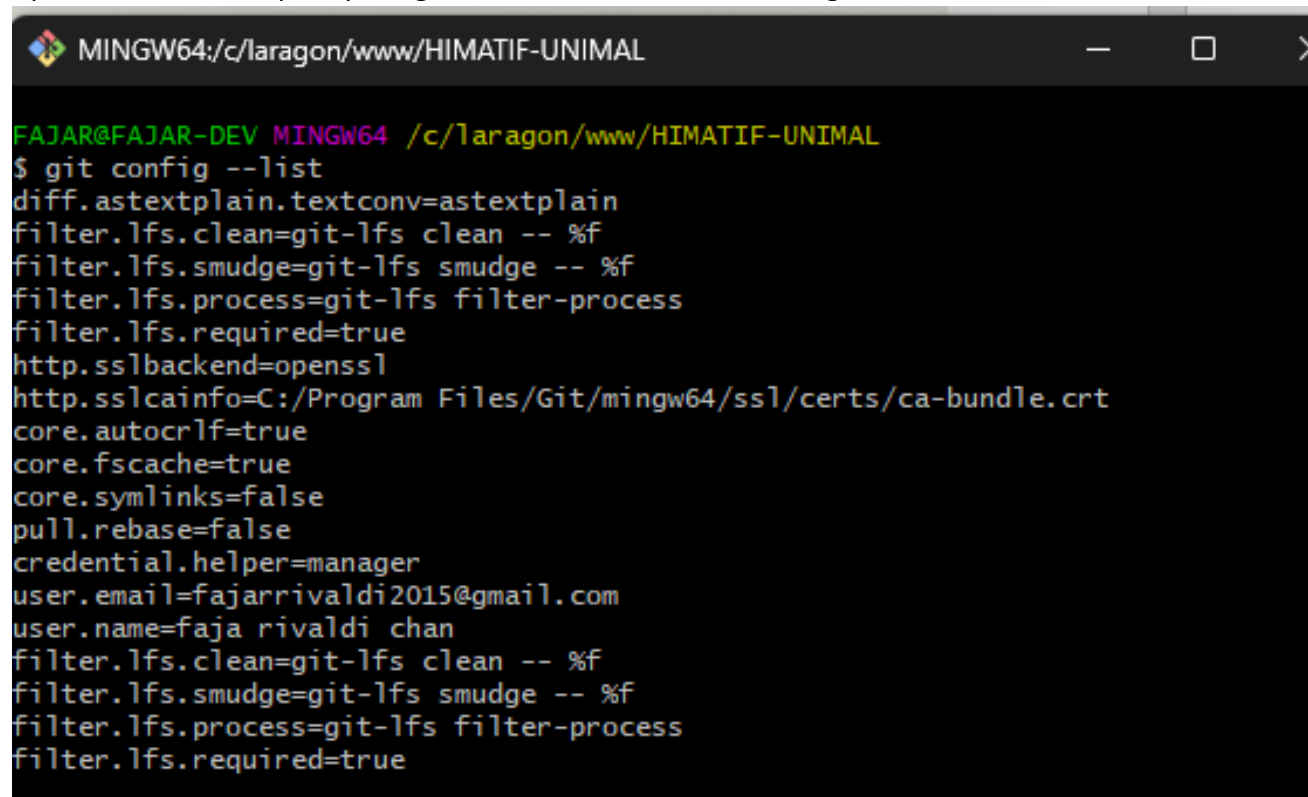
Ada beberapa konfigurasi yang harus dipersiapkan sebelum mulai menggunakan Git, seperti name dan email. Isi sesuai dengan akun Github, Gitlab, Bitbucket kamu. Silahkan lakukan konfigurasi dengan perintah berikut ini.

```
git config --global user.name "Nama_Register_Github"  
git config --global user.email user@email.com
```

Kemudian periksa konfigurasinya dengan perintah:

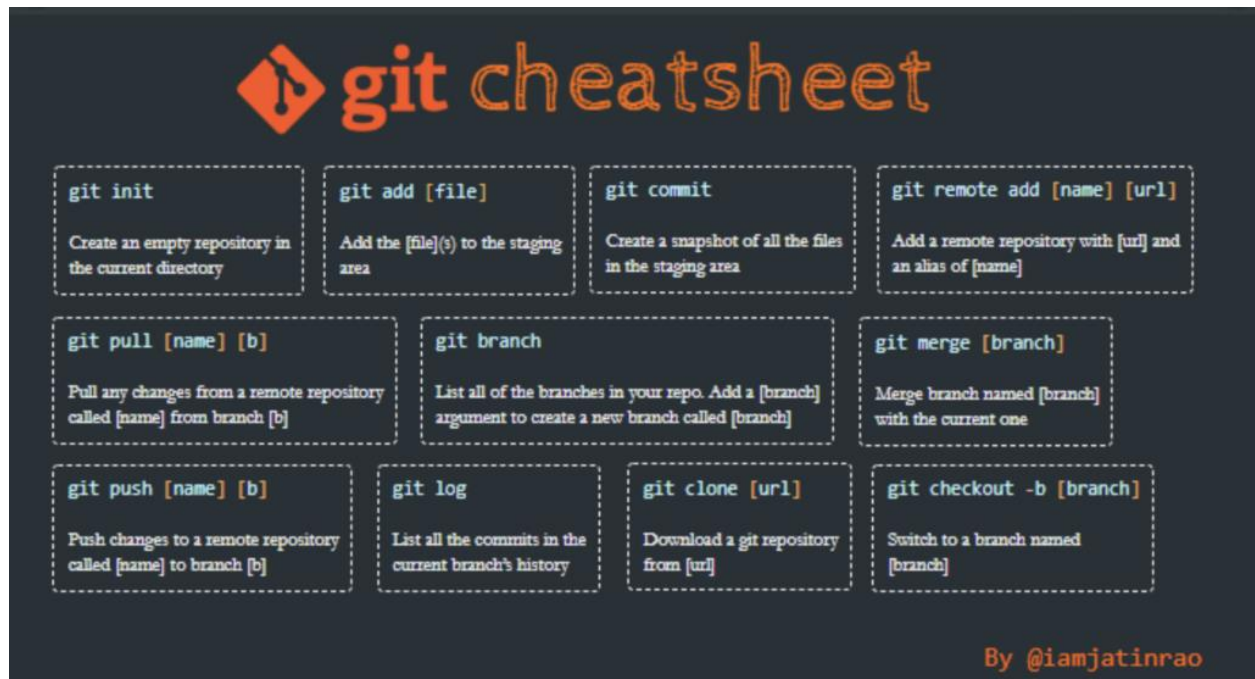
```
git config -list
```

Apabila berhasil tampil seperti gambar berikut ini, berarti konfigurasi berhasil



```
MINGW64:/c:/laragon/www/HIMATIF-UNIMAL  
FAJAR@FAJAR-DEV MINGW64 /c:/laragon/www/HIMATIF-UNIMAL  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager  
user.email=fajarivaldi2015@gmail.com  
user.name=faja rivaldi chan  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true
```

git cheatsheet



The graphic is a dark blue rectangular card with a grid of 12 dashed-line boxes. Each box contains a Git command in orange text and its description in white text. At the top center is the Git logo (an orange diamond with a white branching diagram) followed by the title 'git cheatsheet' in a large, orange, hand-drawn font. At the bottom right, the text 'By @iamjatinrao' is written in orange. The commands are arranged in three rows of four.

Command	Description
<code>git init</code>	Create an empty repository in the current directory
<code>git add [file]</code>	Add the [file](s) to the staging area
<code>git commit</code>	Create a snapshot of all the files in the staging area
<code>git remote add [name] [url]</code>	Add a remote repository with [url] and an alias of [name]
<code>git pull [name] [b]</code>	Pull any changes from a remote repository called [name] from branch [b]
<code>git branch</code>	List all of the branches in your repo. Add a [branch] argument to create a new branch called [branch]
<code>git merge [branch]</code>	Merge branch named [branch] with the current one
<code>git push [name] [b]</code>	Push changes to a remote repository called [name] to branch [b]
<code>git log</code>	List all the commits in the current branch's history
<code>git clone [url]</code>	Download a git repository from [url]
<code>git checkout -b [branch]</code>	Switch to a branch named [branch]

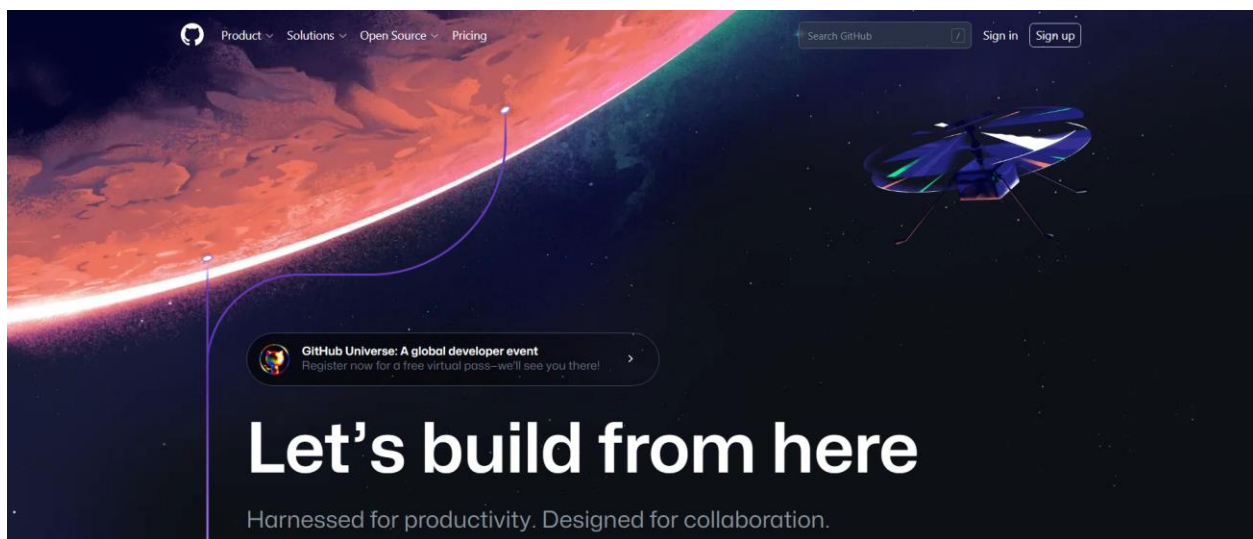
By @iamjatinrao

GITHUB

GitHub adalah perusahaan yang menawarkan layanan hosting repository Git berbasis cloud. Pada dasarnya, GitHub membuat para pengguna individu dan tim menjadi lebih mudah untuk menggunakan Git dalam mengendalikan setiap versi pekerjaan saat melakukan kolaborasi di dalam ataupun antar tim.

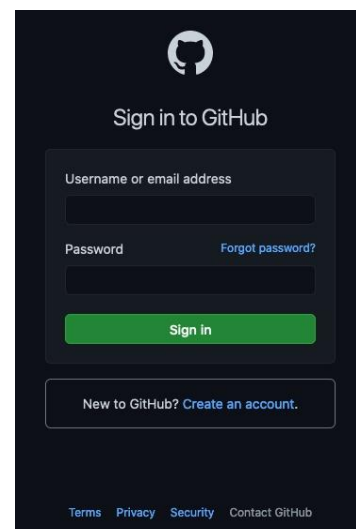
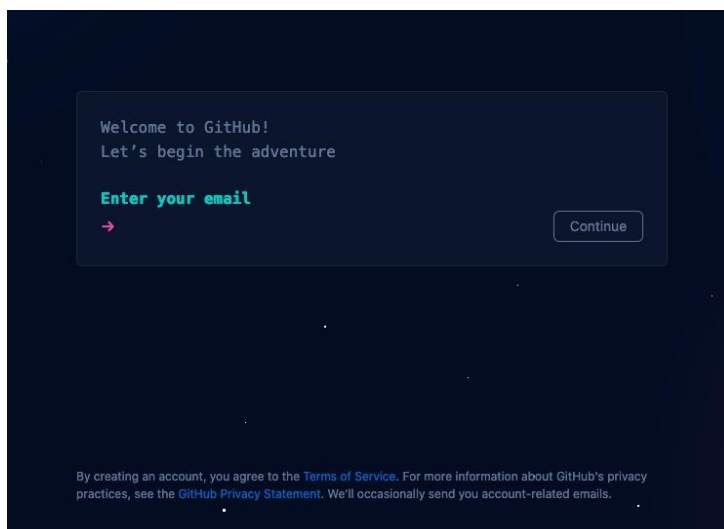
Memulai Git Menggunakan Github

Kamu dapat mengakses github melalui situs: <https://github.com/>. Anda akan diarahkan pada landing page github, halaman ini hanya akan tampil jika Anda belum melakukan login.



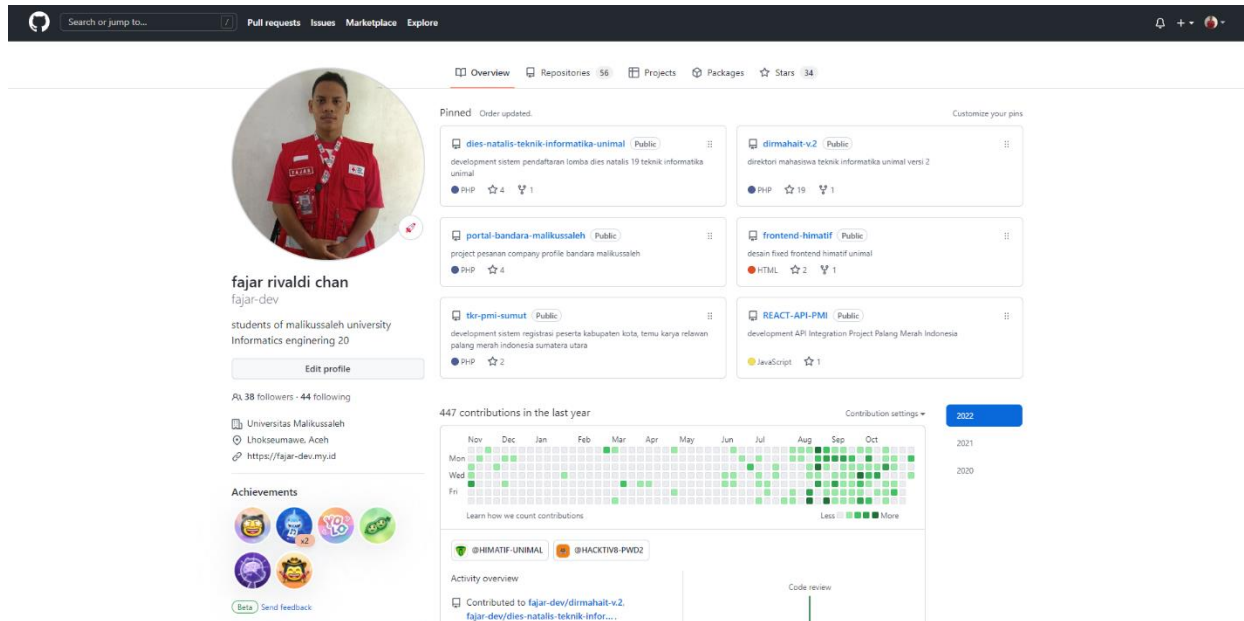
Sign up & Sign in

Untuk membuat akun dan melakukan login, Anda dapat langsung mengeklik menu Sign up dan Sign in pada halaman home website GitHub.



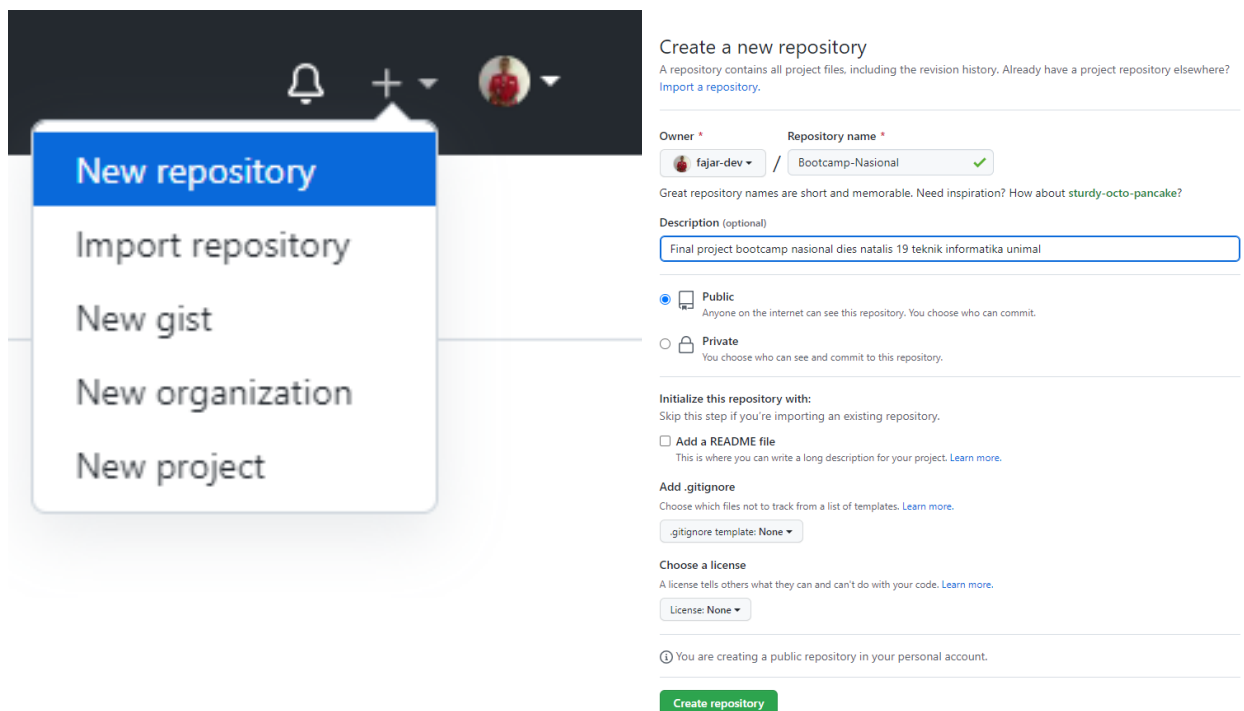
Profile pengguna

Jika kamu telah login menggunakan akun github kamu, kamu akan dialihkan pada halaman profile



Repository

Untuk memulai suatu project kamu dapat membangun repositori baru dengan cara mengarahkan kursor pada logo **tambah/plus (+)** di kanan atas, kemudian klik **New repository**

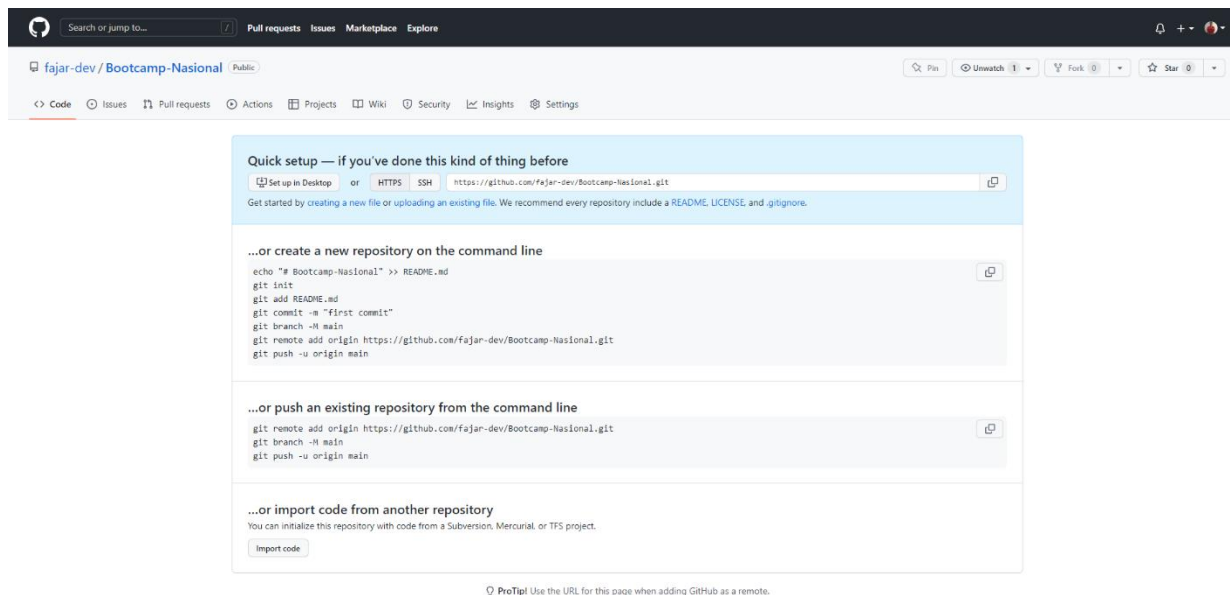


GitHub menyediakan dua jenis repository, yakni:

- Public (repository dapat dilihat oleh semua pengguna Internet, misalnya pada explore repository).
- Private (bersifat pribadi dan hanya user yang diberi akses yang dapat membukanya)

Repository

mulai Project pertama kamu !!



buka terminal/cmd arahkan ke folder projects, terus lakukan :

- o `git init`
- o `git add .`
- o `git status`
- o `git commit -m "pesan kamu"`
- o `git branch -m main`
- o `git remote add origin url_git_kamu`
- o `git push -u origin main`

kembali browser lagi terus refresh browser untuk cek sudah masuk ke github belum..kalo sudah masuk berarti sudah berhasil dan selesai.