

Sebelum kita masuk ke pembahasan git, kita akan membahas terlebih dahulu tentang version control. **Apa itu version control** : Version control adalah system yang merecord/ mentrack perubahan file atau sekumpulan file dari waktu ke waktu, jadi kita dapat melihat file dikemudian hari berdasarkan versi yang terrecord di version control.

Jika kita seorang developer/programmer yang membangun aplikasi, misal aplikasi A dengan versi 1.0 untuk release pertama dan pada waktu berikutnya aplikasi ada pengembangan sehingga naik versi menjadi 2.0. Untuk menjaga/mentrack atau membedakan source code dari versi 1.0 dengan versi 2.0 maka kita membutuhkan system yang menjaga atau mentrack perubahan nama system tersebut adalah VERSION CONTROL. Beberapa hal umum, yang dapat dilakukan oleh system version control antara lain

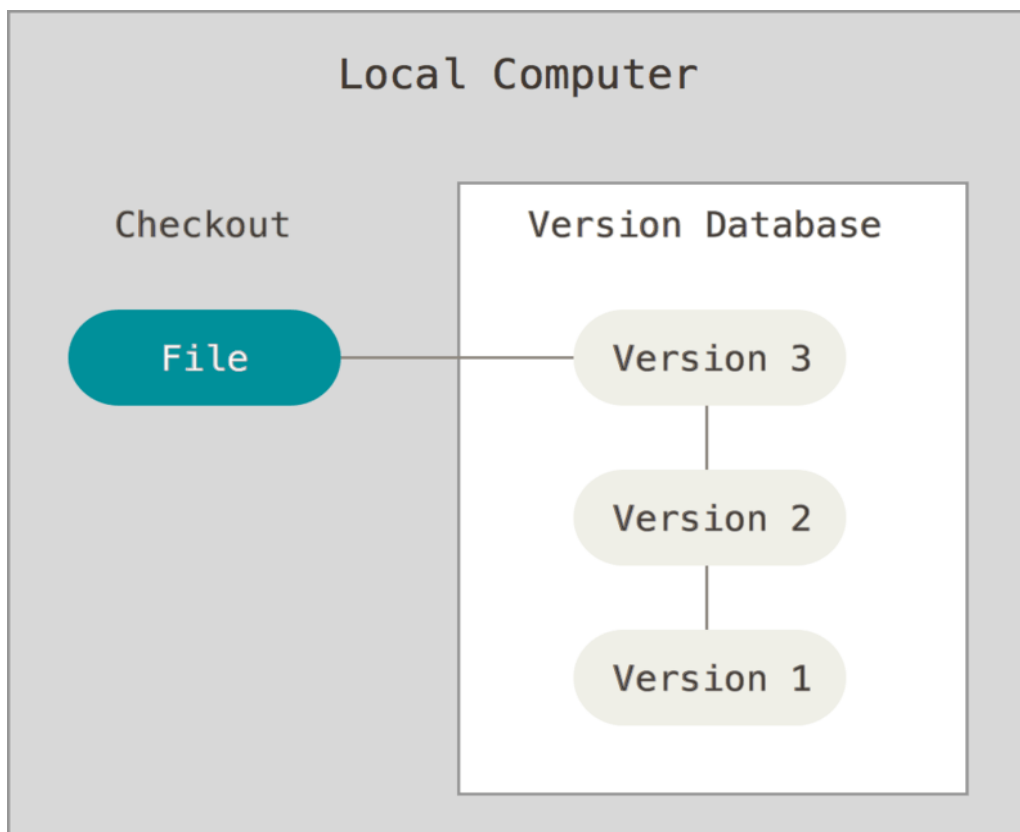
1. Mengembalikan/merevert file ke versi sebelumnya
2. Mengembalikan project kembali ke versi sebelumnya
3. Membandingkan perubahan dari waktu ke waktu
4. Melihat siapa yang melakukan perubahan dan waktu perubahan pada file tertentu

Dan masih banyak lagi, jadi dengan menggunakan system version control kita dimudahkan untuk berkolaborasi dalam membuat proyek yang pengerjaannya dilakukan secara tim.

## Macam-Macam Version Control

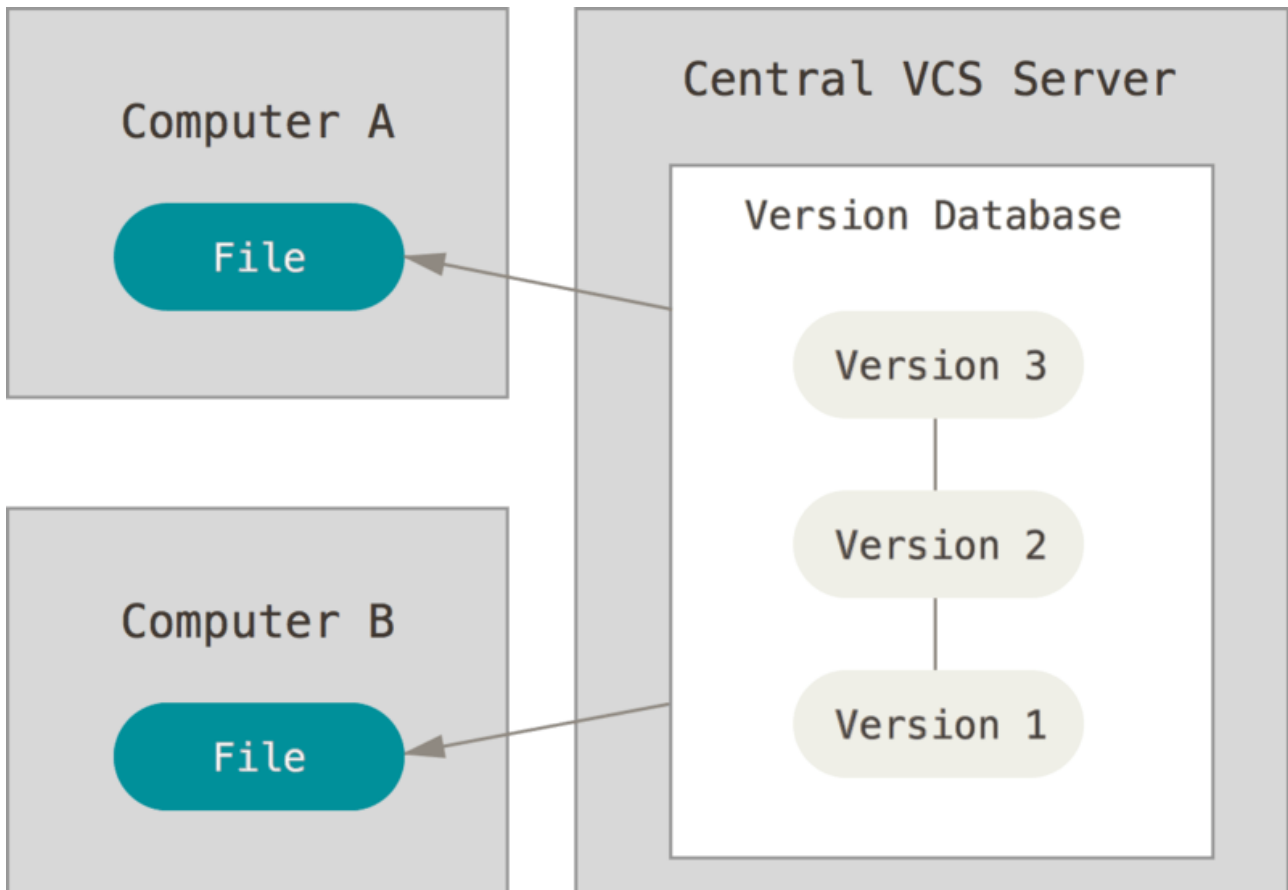
### 1. Local Version Control System

Sebelum adanya system version control, metode tradisional yang dilakukan kebanyakan orang untuk membedakan versi satu dengan versi lainnya yaitu dengan mengcopy file ke folder atau kedalam direktory lainnya. Dengan metode seperti diatas akan banyak kesulitan yang ditemukan seperti membandingkan perubahan file dengan versi sebelumnya, atau terkadang kita lupa folder mana yang menjadi versi terakhir dari perubahan file. Dengan adanya masalah tersebut programmer mendevelop VCS yang mempunyai penyimpanan sederhana yang menjaga perubahan file dibawah system version control, berikut gambaran nya.



## 2. Centralized Version Control System

Permasalahan yang muncul dari local version control antara lain, kesulitannya kita dalam berkolaborasi dengan orang lain. Sehingga untuk memecahkan masalah ini muncul lah CVCS Centralized Version Control Systems. Macam-macam CVCSs yang sering digunakan antara lain CVS, Subversion Perforce dan lain-lain. Dengan CVCSs file atau proyek disimpan disatu server dan tim yang butuh file atau proyek tersebut perlu checkout dari sentral atau pusat. Dalam beberapa bertahun-tahun CVCSs adalah standar untuk version control. Berikut gambaran dari CVCSs



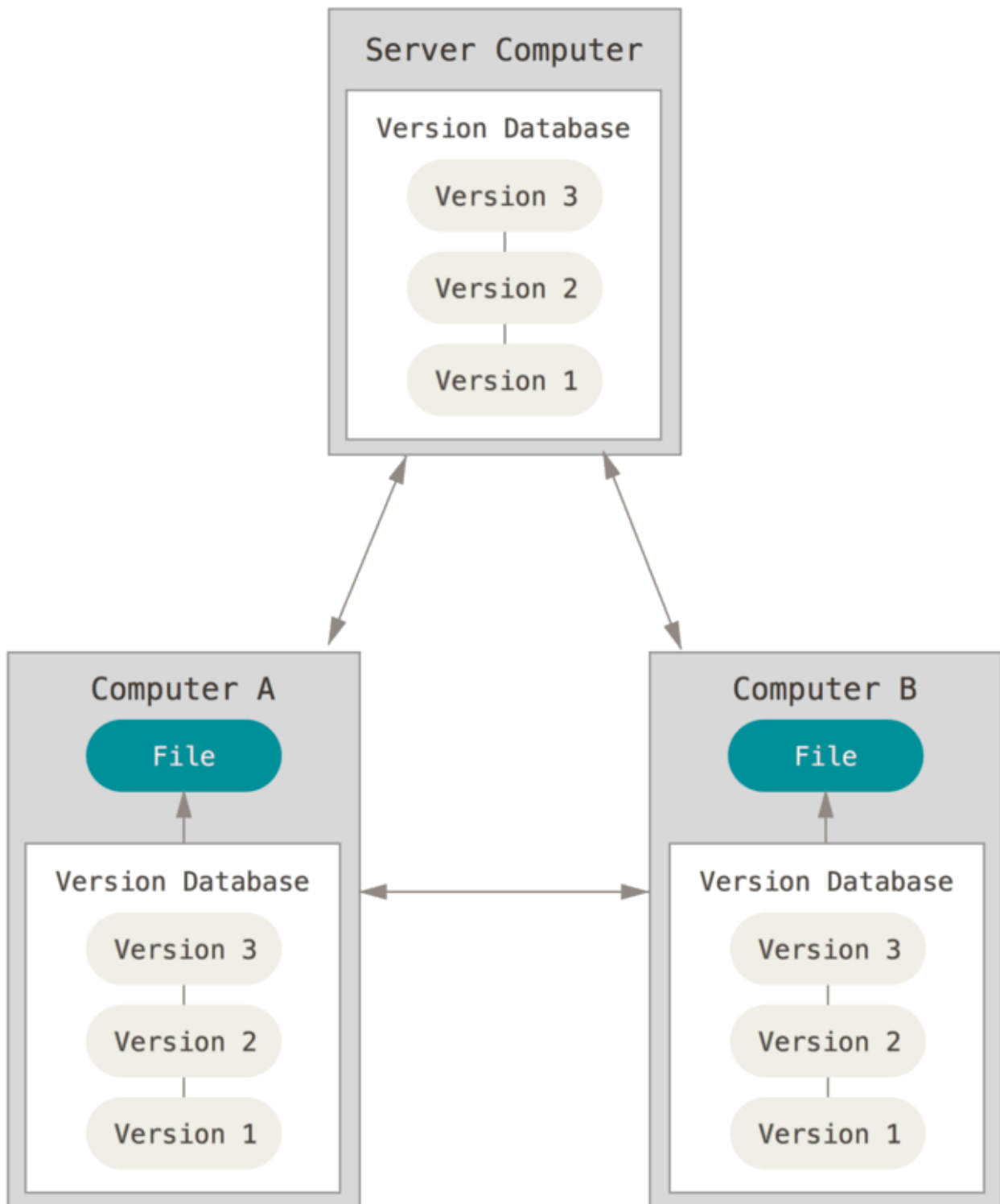
Setup version control dengan model CVCSs terdapat beberapa keuntungan antarlain, setiap orang dapat mengetahui sampai berapa jauh kontribusi orang dalam proyek tertentu, Administrator memiliki kendali penuh terhadap siapa saja dan apa saja yang dapat dilakukan. Dan jauh lebih mudah untuk mengelola CVCSs(version control sentralized) dari pada menangani version control pada setiap lokal client.

Namun setup dengan metode CVCSs juga memiliki kekurangan, ketika server down selama satu jam dalam waktu satu jam pula tim tidak dapat berkolaborasi atau menyimpan versi perubahan file. Kekurangan lainnya antarlain , jika sentral penyimpanan rusak dan backup file belum disimpan maka kita kehilangan segalanya – seluruh sejarah proyek atau track proyek hilang kecuali snapshot yang kebetulan dimiliki oleh setiap tim dilokal komputer mereka.

## 3. Distributed Version Control System

Metode version control DVCS, local computer tidak hanya memiliki salinan terbaru dari repository pusat, melainkan mereka sepenuhnya mencerminkan repository termasuk riwayat lengkap dari perubahan file. Jadi ketika server down maka repository klien dapat

disalin ke repository server untuk memulihkan repository server. Berikut gambaran lengkap dari DVCS



Jadi dengan version control dengan metode DVCS ini proyek memiliki beberapa salinan tidak hanya di server melainkan ada slainan di klien lengkap dengan versi dan history dari proyek yang ada di server. Berikut contoh dari DVCS : GIT, Mercurial, Bazaar or Darcs.

# GIT VERSION CONTROL

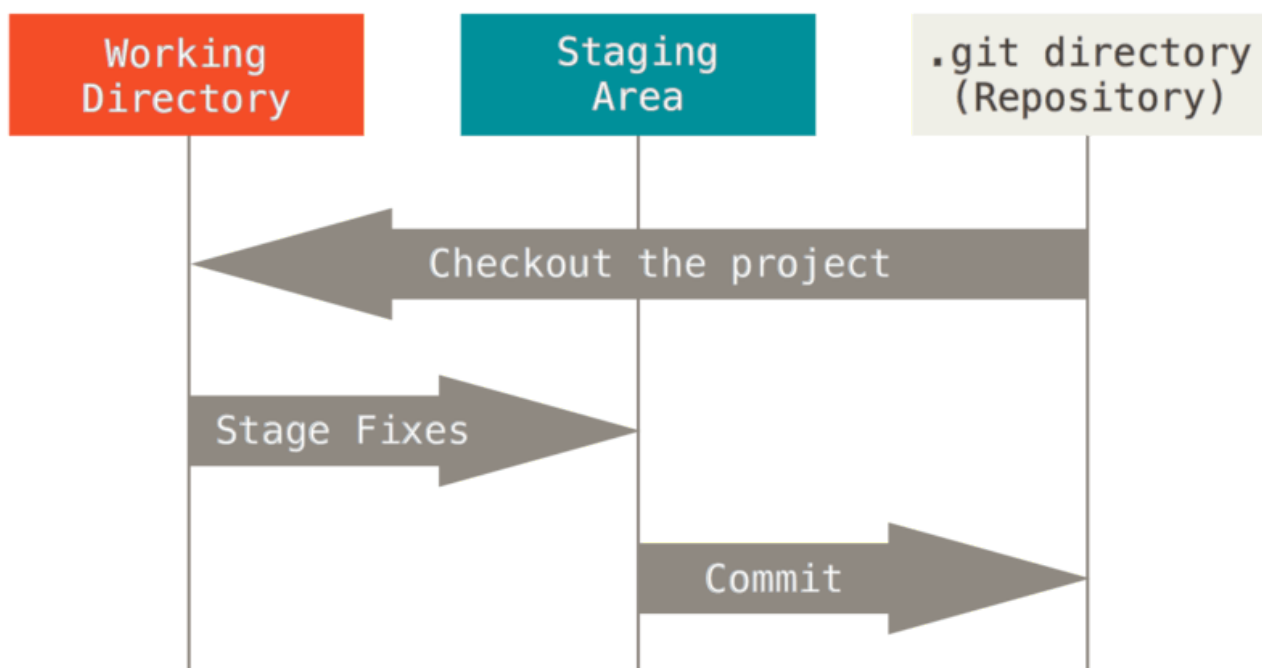
Git adalah software yang digunakan untuk menyimpan, manage banyak file(source code), biasanya digunakan dalam pekerjaan yang pengerjaan suatu proyeknya dilakukan secara tim. Sebelum menjalankan perintah-perintah yang sering digunakan didalam git, maka pembaca diwajibkan sudah menginstall di laptop atau di pc masing-masing.

Beberapa point yang akan dilakukan di pembahasan kali ini, sehingga pembaca pada akhir pembelajaran dapat paham dari version control dan bisa manage proyek nya tidak hanya di local repository melainkan bisa manage di hosting service git seperti GITHUB. Berikut point-point yang akan dibahas secara rinci.



## 1. Memahami workflow Git

Git dalam workflow dibagi menjadi 3 state yaitu working directory(folder), staging area dan git directory atau repository. Ketika kita merubah file kita berada di working directory, ketika kita memilih file yang mengalami perubahan untuk di commit ke repository kita menambahkan file tersebut ke staging area, dan ketika sudah memasuki staging area file yang mengalami perubahan akan kita commit/save ke repository. Berikut workflow dari git



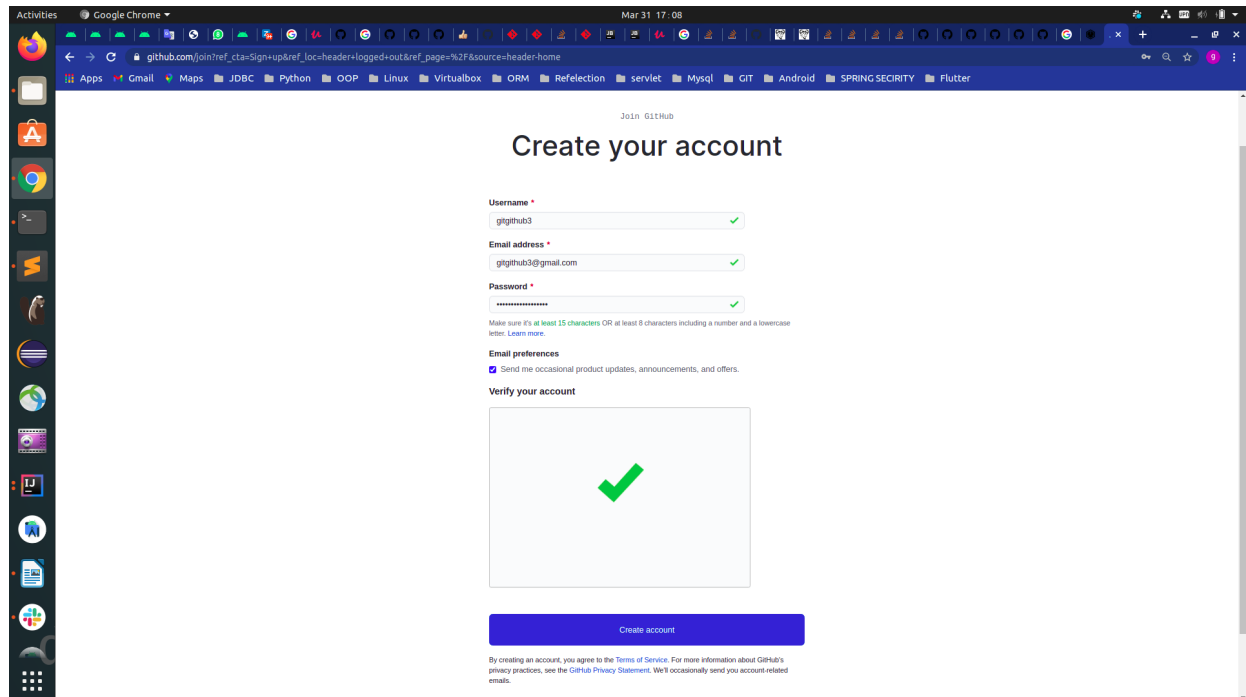
## 2. Push Existing Project git ke Github

### ❑ Buat account github

Langkah Pertama yang kita lakukan untuk dapat menyimpan proyek kita di github adalah dengan kita melakukan registrasi untuk membuat account github.

**Note : Catat username, password dan email untuk config git dilangkah selanjutnya nanti.**

### ❑ Masukkan Username, Email dll



Join GitHub

### Create your account

Username \*

ghgithub3 ✓

Email address \*

ghgithub3@gmail.com ✓

Password \*

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)

Email preferences

☒ Send me occasional product updates, announcements, and offers.

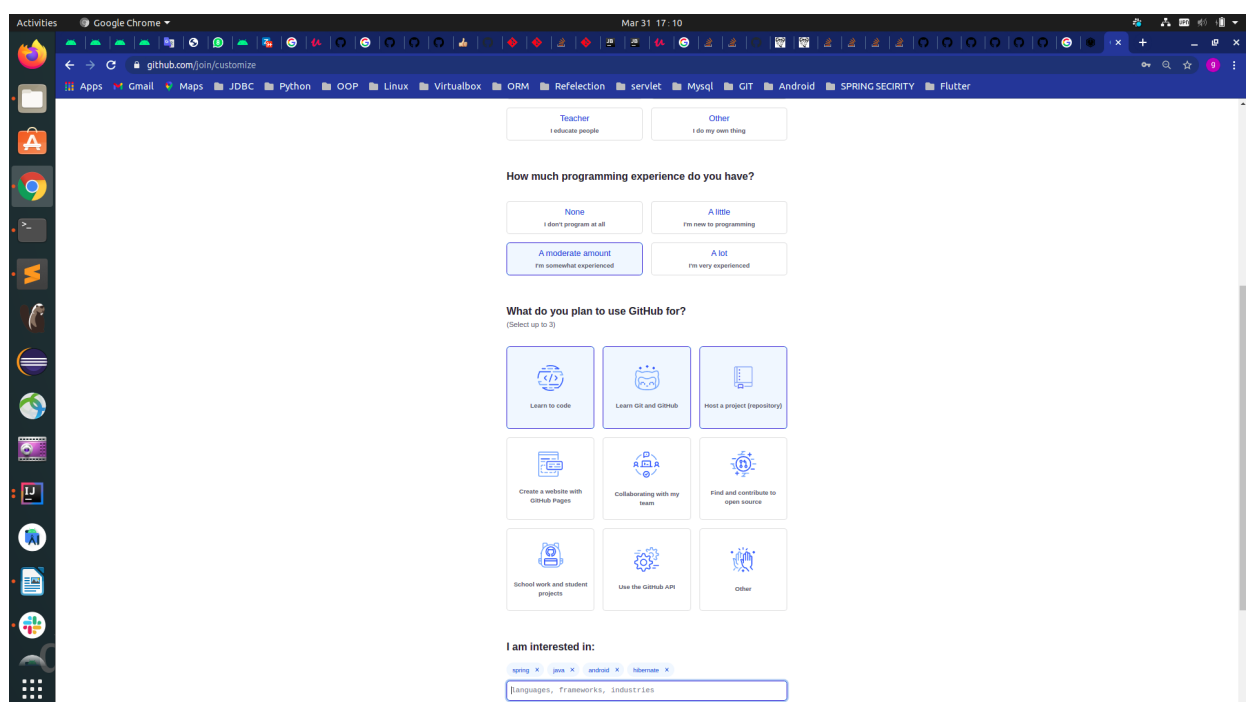
Verify your account

✓

Create account

By creating an account, you agree to the [Terms of Service](#). For more information about GitHub's privacy practices, see the [GitHub Privacy Statement](#). We'll occasionally send you account-related emails.

### ❑ Pilih Additional info



Teacher  
I educate people

Other  
I do my own thing

How much programming experience do you have?

None  
I don't program at all

A little  
I'm new to programming

A moderate amount  
I'm somewhat experienced

A lot  
I'm very experienced

What do you plan to use GitHub for?  
(Select up to 3)

Learn to code

Learn Git and GitHub

Host a project (repository)

Create a website with GitHub Pages

Collaborating with my team

Find and contribute to open source

School work and student projects

Use the GitHub API

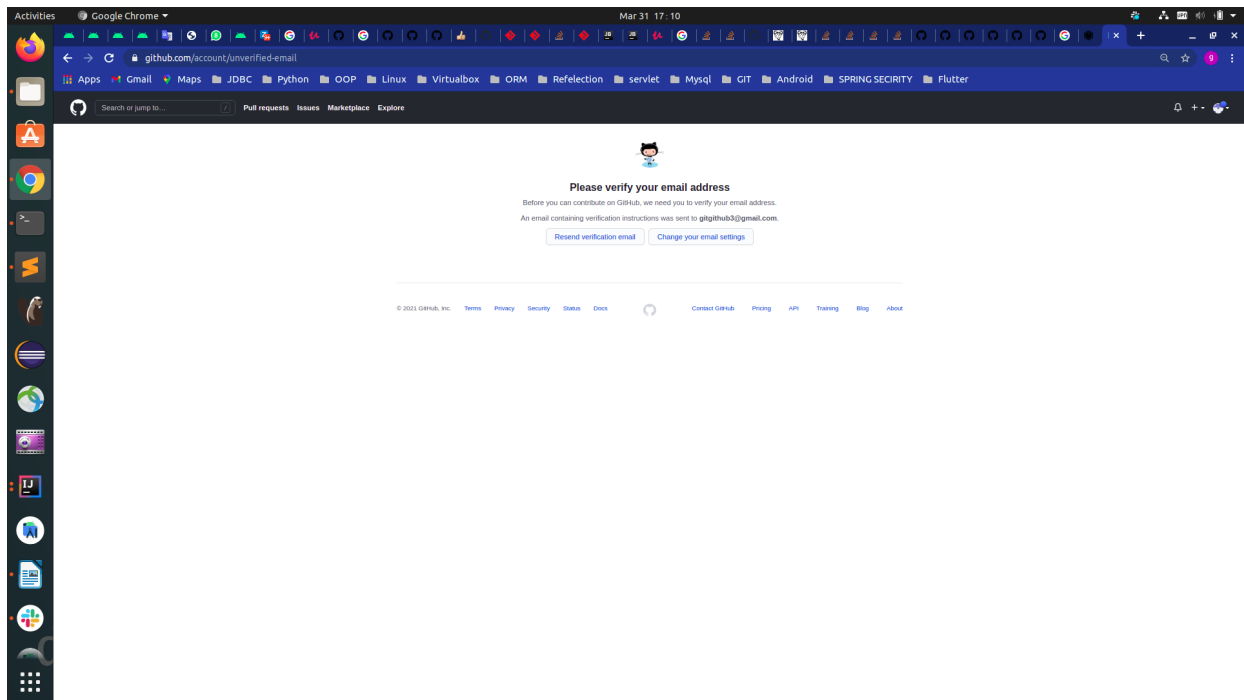
Other

I am interested in:

spring x java x android x hibernate x

languages, frameworks, industry l...

- Kemudian muncul halaman perintah verifikasi email



Masuk ke email, dan verifikasi. Setelah verifikasi selesai maka proses pembuatan account github selesai.

## □ Config Username dan Email

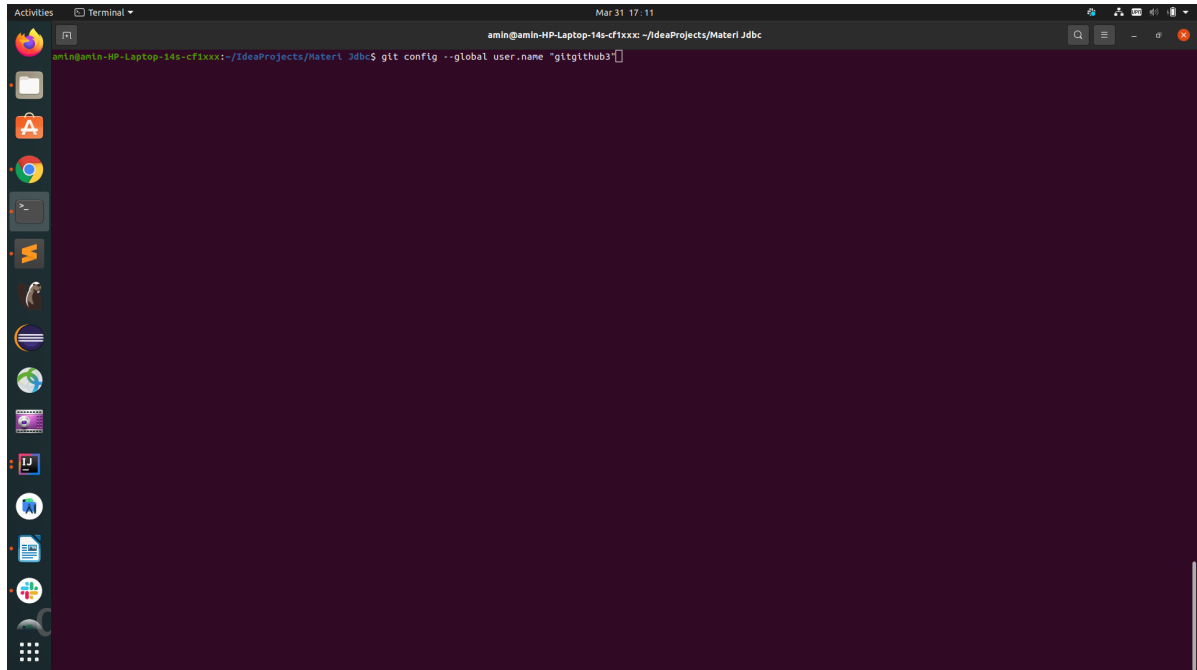
Setelah proses pembuatan account selesai, langkah selanjutnya adalah config username dan email di laptop/computer kita. Sebelum menconfig pembaca diharapkan sudah install git di computer/laptop masing-masing.

Konfigurasi username dan email ini bertujuan untuk mengaitkan username dan email (identitas) anda ketika anda melakukan commit ke local repository atau melakukan push ke github. Jadi setiap perubahan yang dilakukan oleh kita **akan tercatat historynya dengan username dan email yang kita config**. Untuk melakukan config ikuti langkah berikut ini.

- **Buka comman prompt/terminal(pastikan git sudah terinstall)**

- **Ketikkan perintah berikut untuk config username**

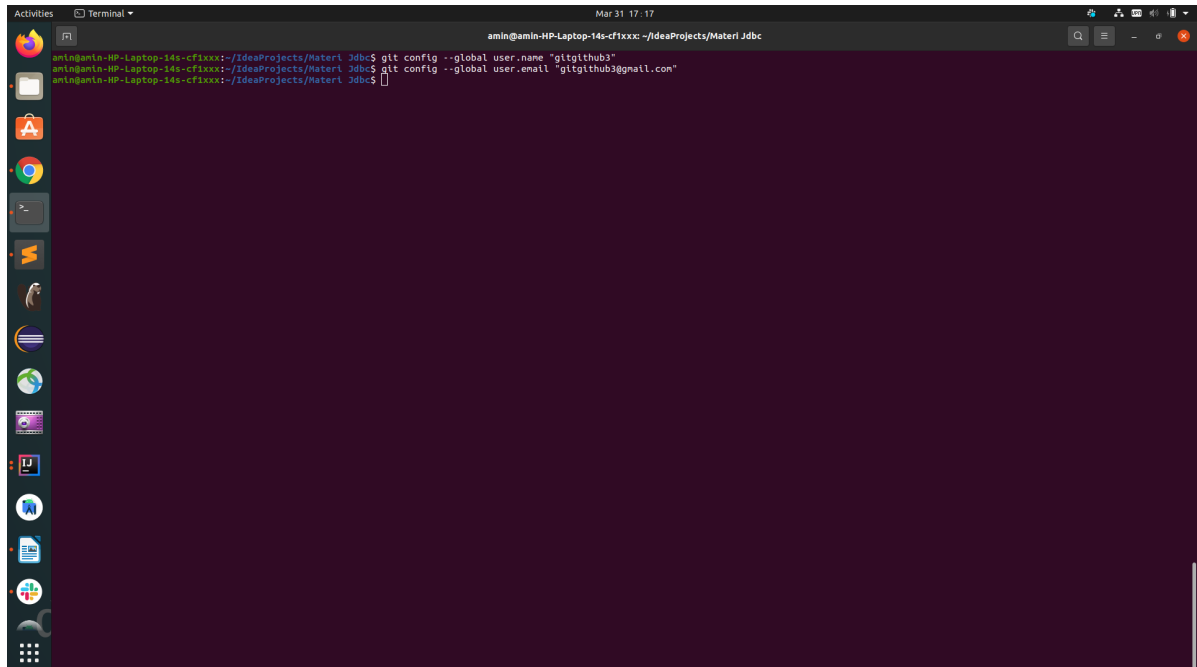
Pastikan username git sama dengan nama username github pada saat pembuatan account. (**git config --global user.name "username yang terdaftar di github"**).

A terminal window with a dark purple background. The title bar shows 'Activities', 'Terminal', and the date 'Mar 31 17:11'. The terminal text shows the user 'amin' at 'amin@amin-HP-Laptop-14s-cf1xxx' in the directory '~/IdeaProjects/Materi1Jdbc'. The command 'git config --global user.name "gitgithub3"' has been entered and executed, resulting in a blank line.

```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Materi1Jdbc$ git config --global user.name "gitgithub3"
```

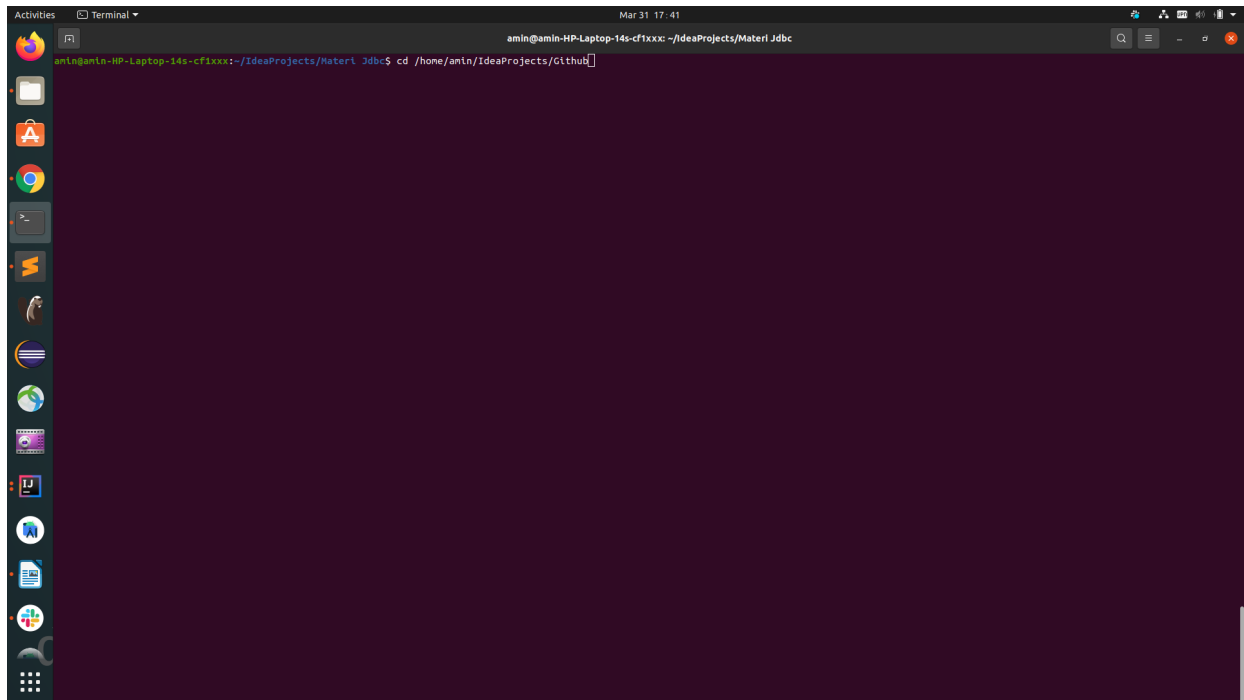
## □ Kemudian config email

Pastikan email git sama dengan email pada saat pembuatan account github.  
(**git config --global user.email "email pada saat daftar github"**)

A terminal window with a dark purple background. The title bar shows 'Activities', 'Terminal', and the date 'Mar 31 17:17'. The terminal text shows the user 'amin' at 'amin@amin-HP-Laptop-14s-cf1xxx' in the directory '~/IdeaProjects/Materi1Jdbc'. The command 'git config --global user.email "gitgithub3@gmail.com"' has been entered and executed, resulting in a blank line.

```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Materi1Jdbc$ git config --global user.name "gitgithub3"
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Materi1Jdbc$ git config --global user.email "gitgithub3@gmail.com"
```

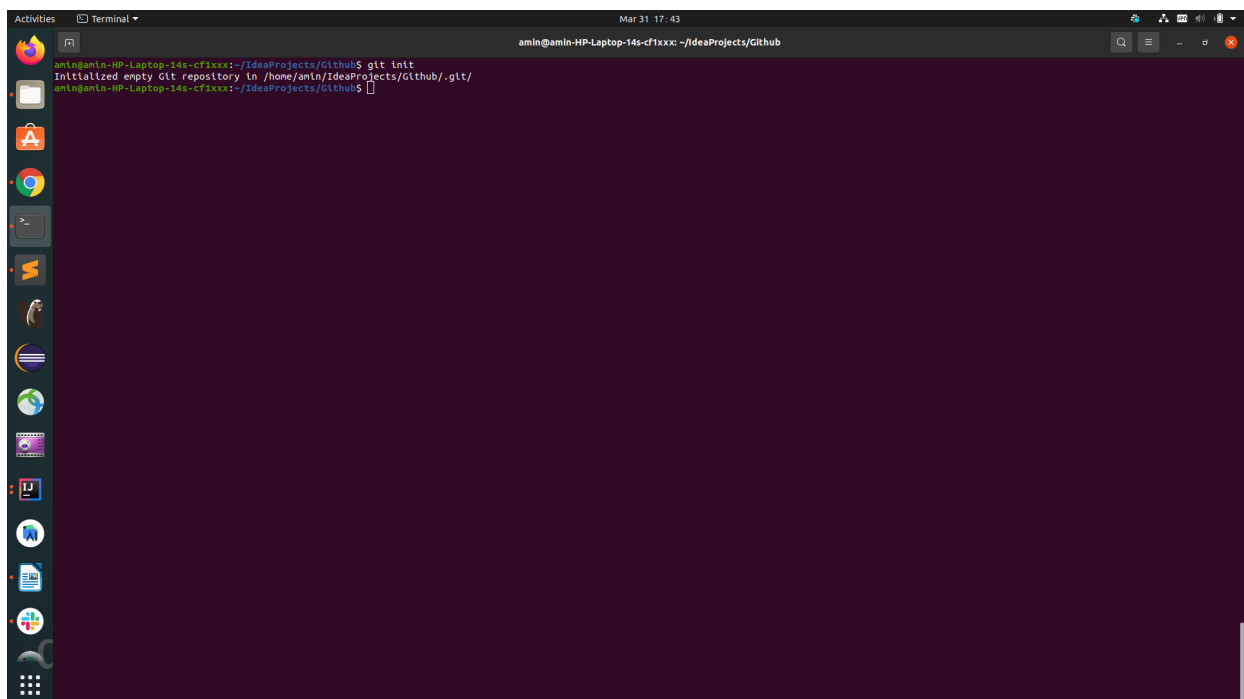
- ❑ **Buat Repository local dan commit ke local repository**
- ❑ **Tentukan lokasi proyek anda**
- ❑ **Pindah ke folder dimana anda akan menempatkan proyek anda dengan mengetikkan seperti pada gambar dibawah**



A terminal window titled 'Terminal' with a dark purple background. The prompt is 'amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Materi JDBC'. The user has entered the command 'cd /home/amin/IdeaProjects/Github' and the cursor is at the end of the line.

```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Materi JDBC$ cd /home/amin/IdeaProjects/Github
```

- ❑ **Setelah kita pindah ke directory tempat project kita, kemudian kita ketikkan perintah untuk menginisialisasi git (git init)**

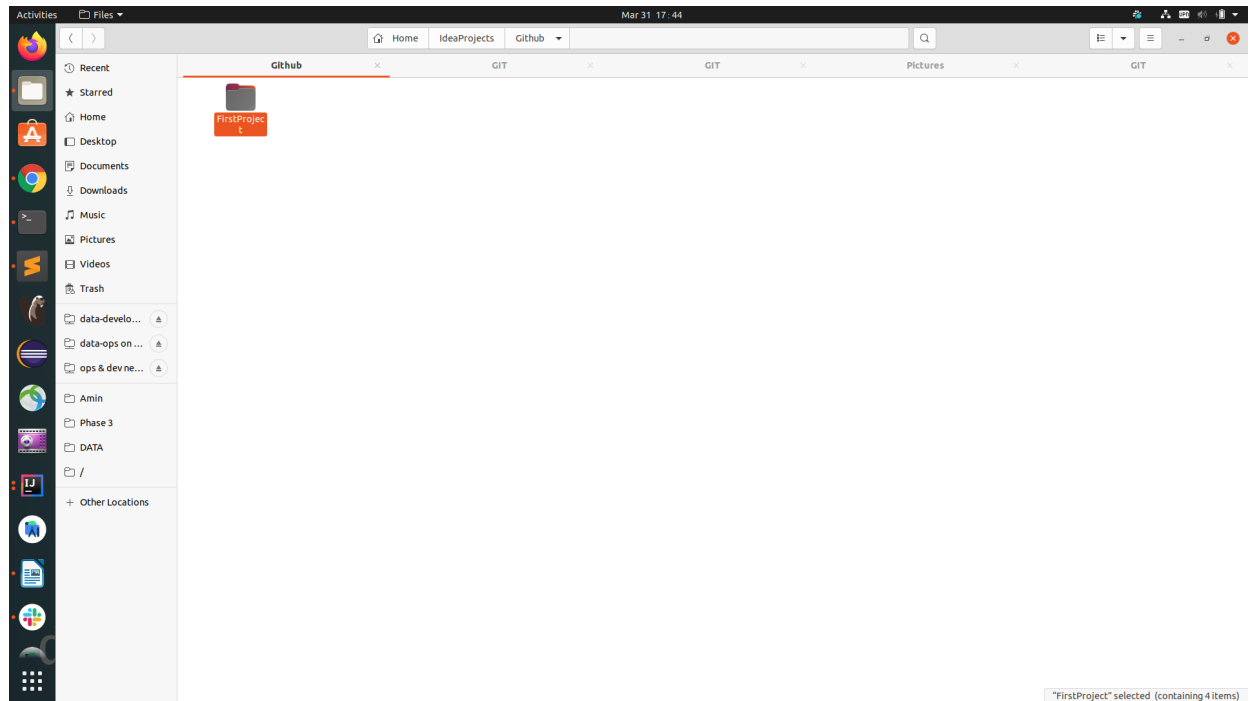


A terminal window titled 'Terminal' with a dark purple background. The prompt is 'amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github'. The user has entered the command 'git init', and the terminal output shows 'Initialized empty Git repository in /home/amin/IdeaProjects/Github/.git/'. The prompt is now 'amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github\$'.

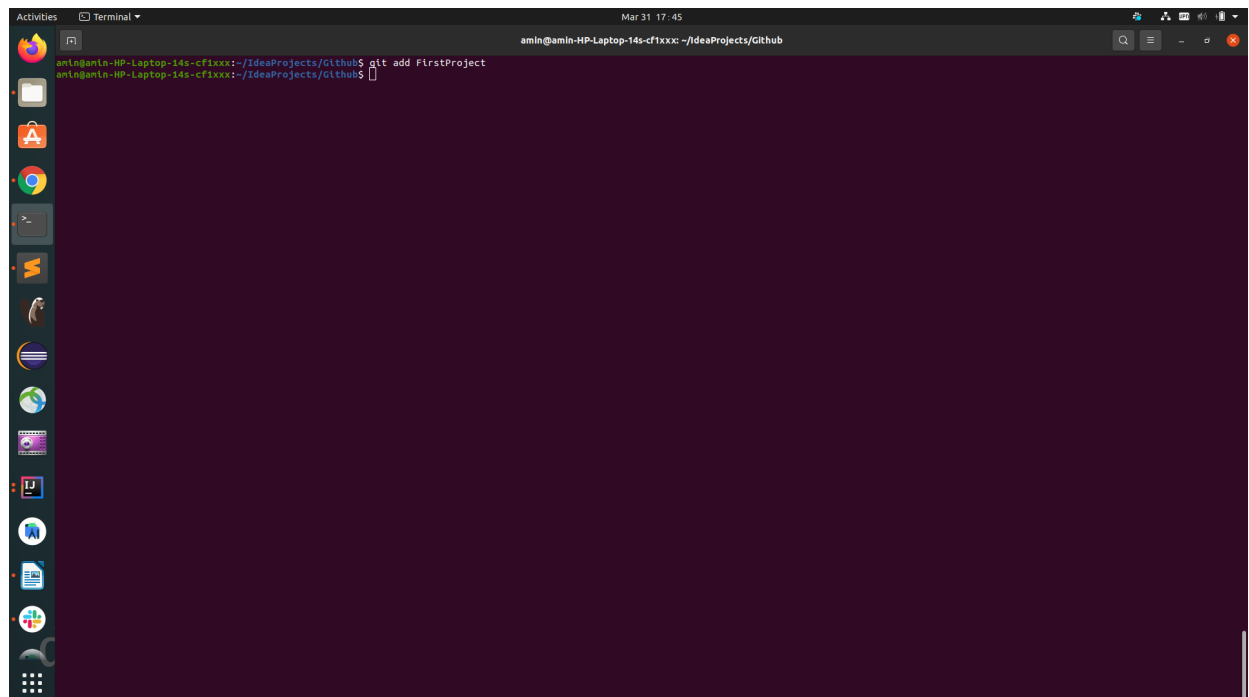
```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Github$ git init
Initialized empty Git repository in /home/amin/IdeaProjects/Github/.git/
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Github$
```



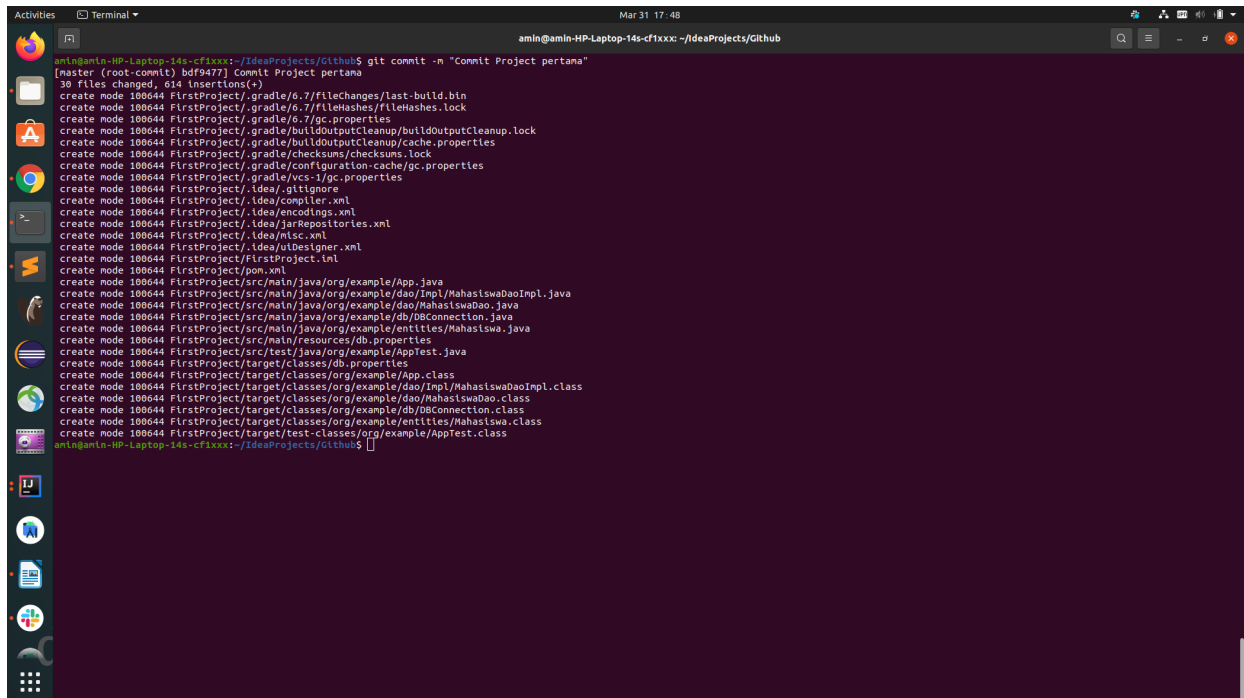
- ❑ Setelah Repository terbentuk di directory tempat project kita berada, langkah selanjutnya Copy Project ke Repository yang baru



- ❑ Ketikkan perintah untuk menambahkan project ke git ke staging area



- ❑ Kemudian commit project yang tadi sudah ditambahkan untuk di simpan ke local Repository

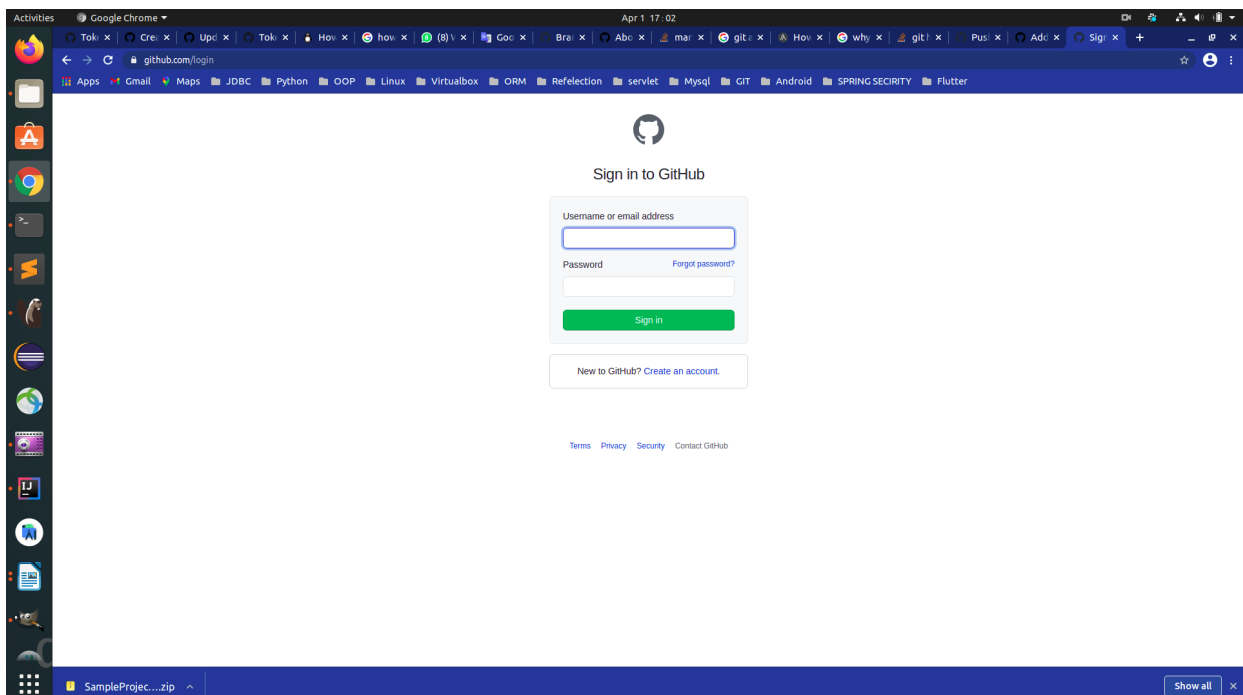


```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/GitHub$ git commit -m "Commit Project pertama"
[master (root-commit) bdf9477] Commit Project pertama
39 files changed, 614 insertions(+)
create mode 100644 FirstProject/.gradle/6.7/fileChanges/last-build.bin
create mode 100644 FirstProject/.gradle/6.7/fileHashes/fileHashes.lock
create mode 100644 FirstProject/.gradle/6.7/gc.properties
create mode 100644 FirstProject/.gradle/buildOutputCleanup/buildOutputCleanup.lock
create mode 100644 FirstProject/.gradle/buildOutputCleanup/cache.properties
create mode 100644 FirstProject/.gradle/checksums/checksums.lock
create mode 100644 FirstProject/.gradle/configuration-cache/gc.properties
create mode 100644 FirstProject/.gradle/vcs-1/gc.properties
create mode 100644 FirstProject/.idea/.gitignore
create mode 100644 FirstProject/.idea/compiler.xml
create mode 100644 FirstProject/.idea/encodings.xml
create mode 100644 FirstProject/.idea/jarRepositories.xml
create mode 100644 FirstProject/.idea/misc.xml
create mode 100644 FirstProject/.idea/runConfigurations.xml
create mode 100644 FirstProject/FirstProject.iml
create mode 100644 FirstProject/pom.xml
create mode 100644 FirstProject/src/main/java/org/example/App.java
create mode 100644 FirstProject/src/main/java/org/example/dao/Impl/MahasiswaDaoImpl.java
create mode 100644 FirstProject/src/main/java/org/example/dao/MahasiswaDao.java
create mode 100644 FirstProject/src/main/java/org/example/db/DBConnection.java
create mode 100644 FirstProject/src/main/resources/db.properties
create mode 100644 FirstProject/src/test/java/org/example/AppTest.java
create mode 100644 FirstProject/target/classes/db.properties
create mode 100644 FirstProject/target/classes/org/example/App.class
create mode 100644 FirstProject/target/classes/org/example/dao/Impl/MahasiswaDaoImpl.class
create mode 100644 FirstProject/target/classes/org/example/dao/MahasiswaDao.class
create mode 100644 FirstProject/target/classes/org/example/db/DBConnection.class
create mode 100644 FirstProject/target/classes/org/example/entities/Mahasiswa.class
create mode 100644 FirstProject/target/test-classes/org/example/AppTest.class
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/GitHub$
```

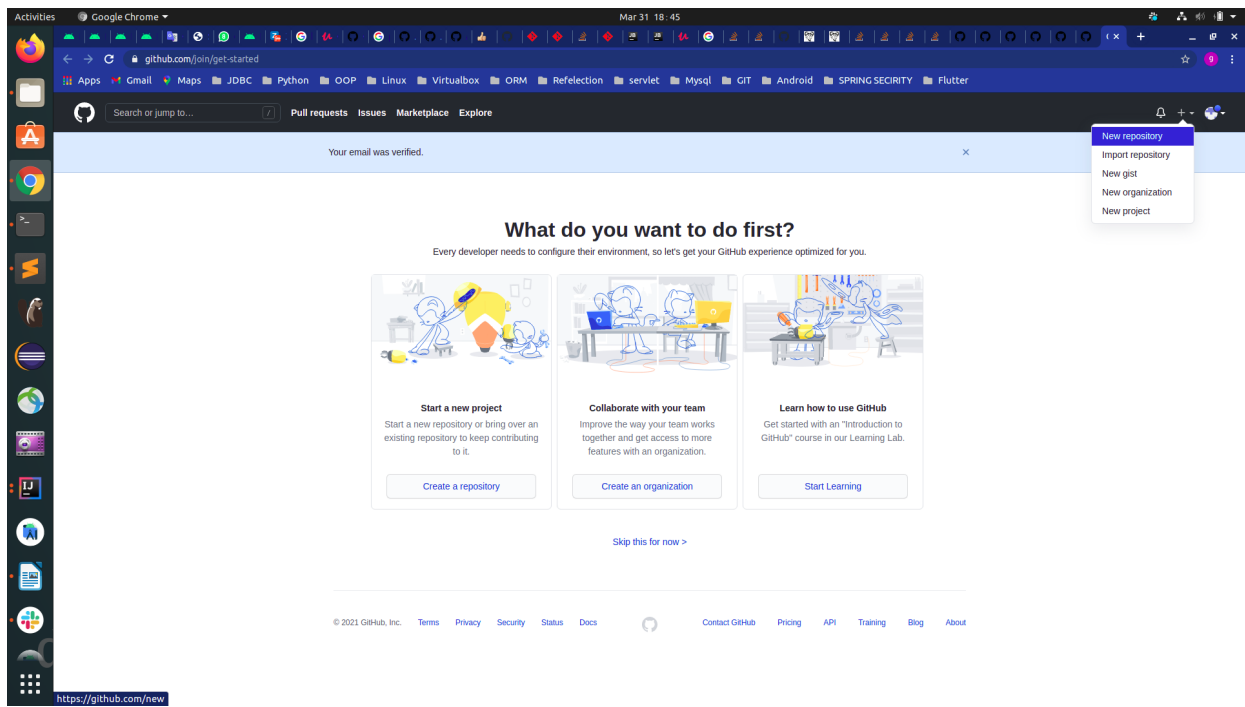
- ❑ Buat repository di github

Setelah kita membuat account github dan commit project ke local repository, langkah selanjutnya adalah membuat github repository

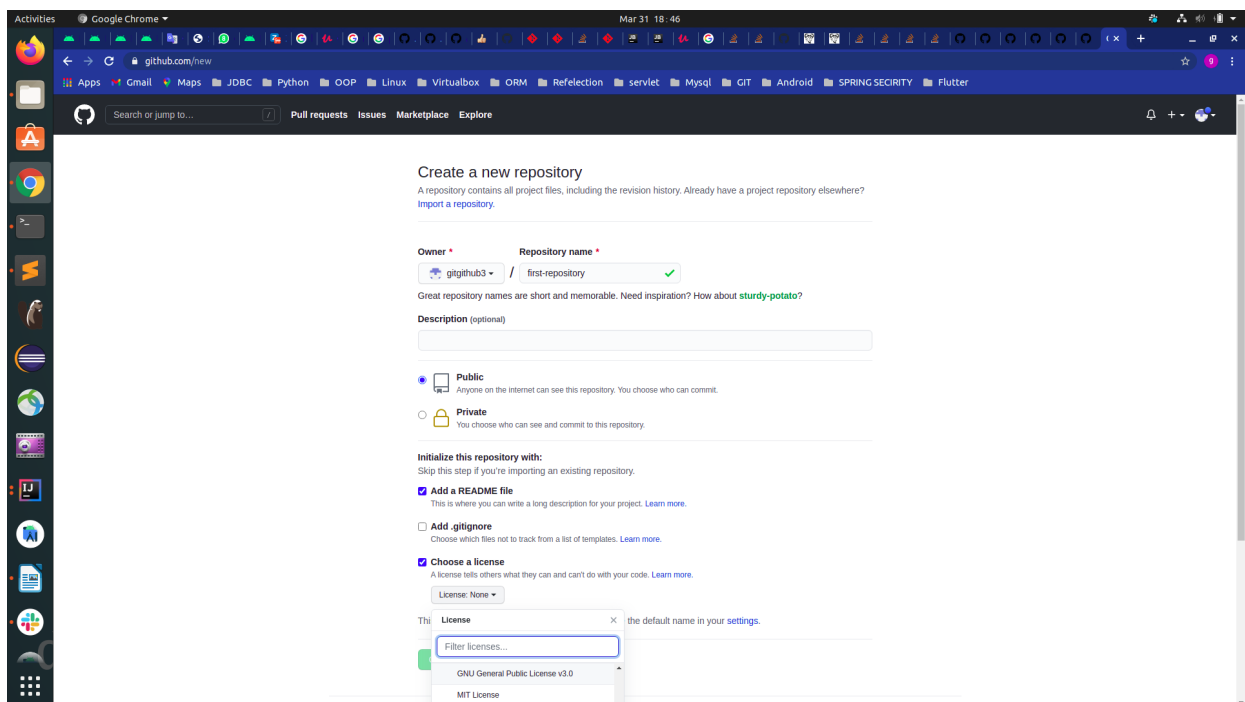
- ❑ Login ke github



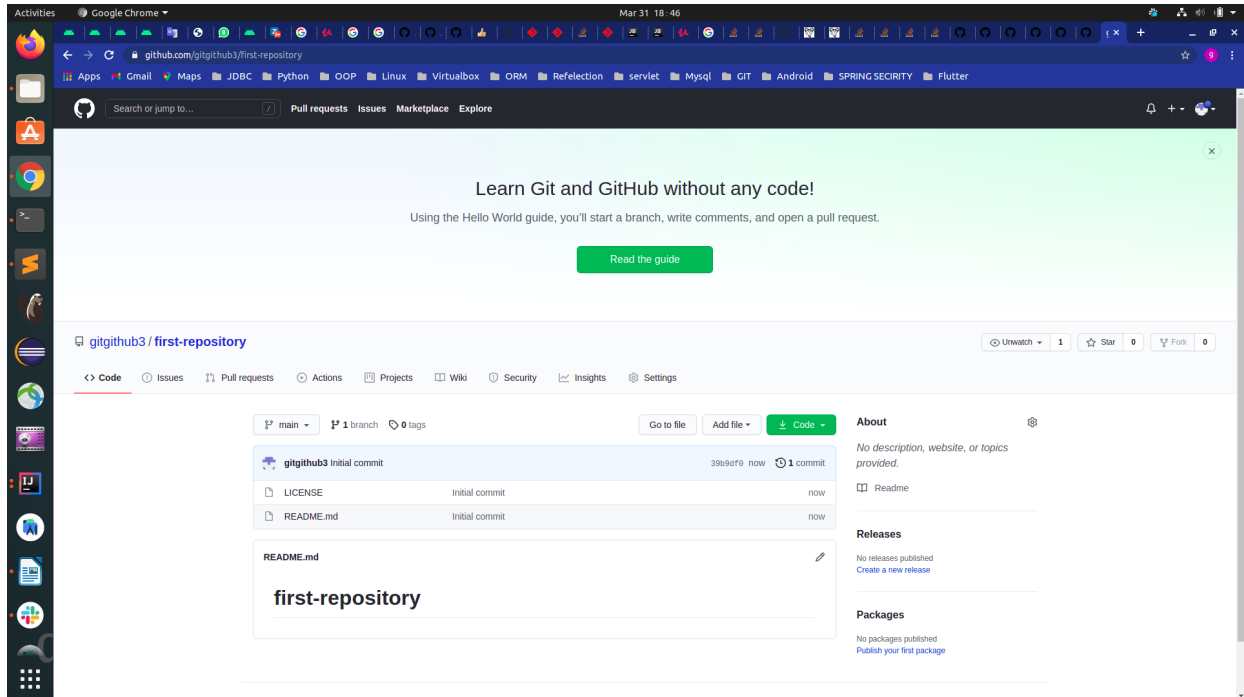
## □ Pilih new repository di pojok kanan



## □ Masukkan nama repository, ikuti config yang lain seperti pada gambar



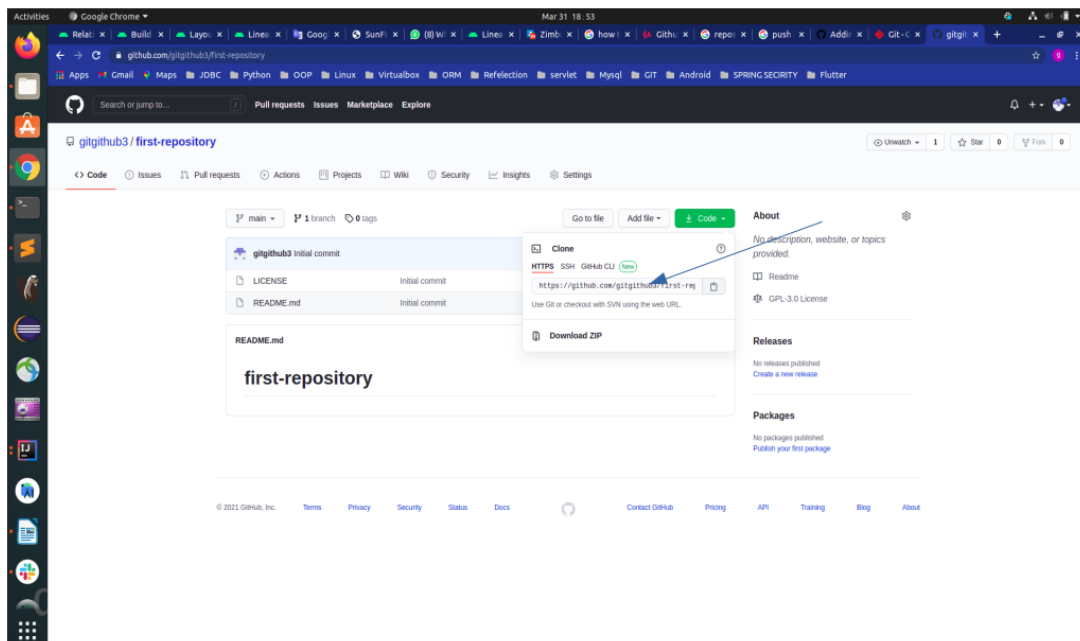
## □ Halaman ketika repository sukses dibuat



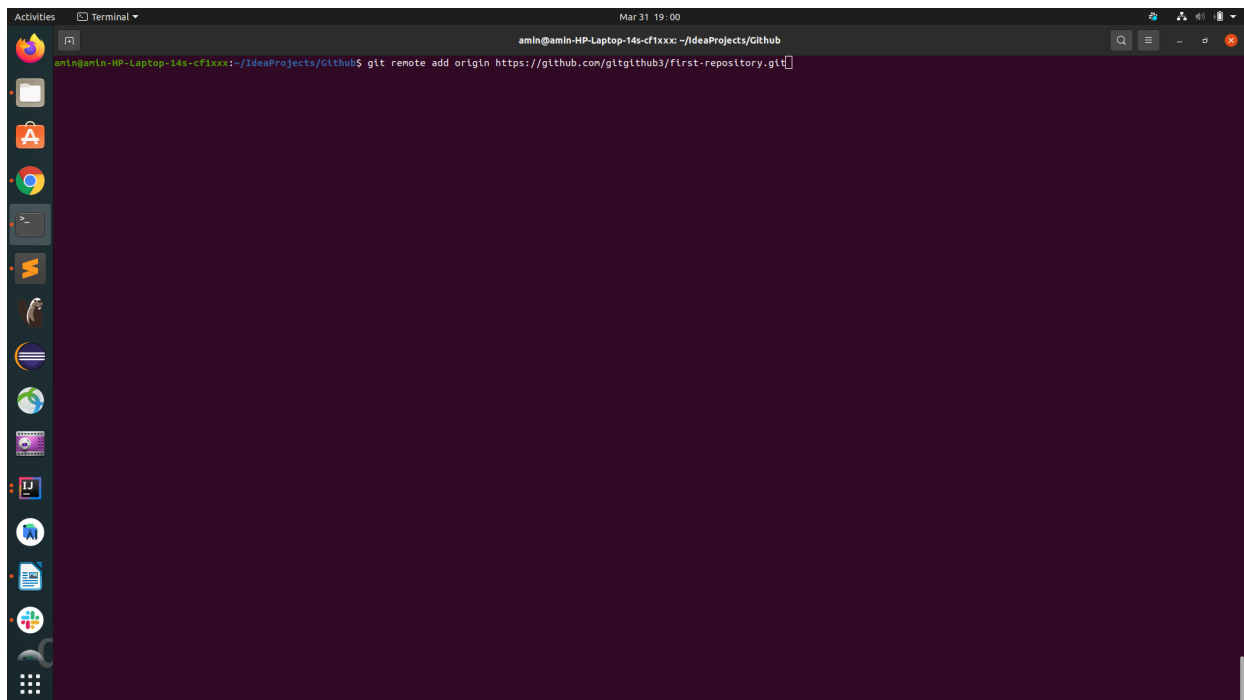
## □ Push project ke repository github

Distep sebelumnya kita sudah berhasil commit proyek ke repository local git dan membuat repository digithub. Untuk langkah selanjutnya kita akan push project yang di commit di local repository dan melakukan push ke github. Untuk push project, ikuti langkah berikut ini

## □ Copy URL repository github



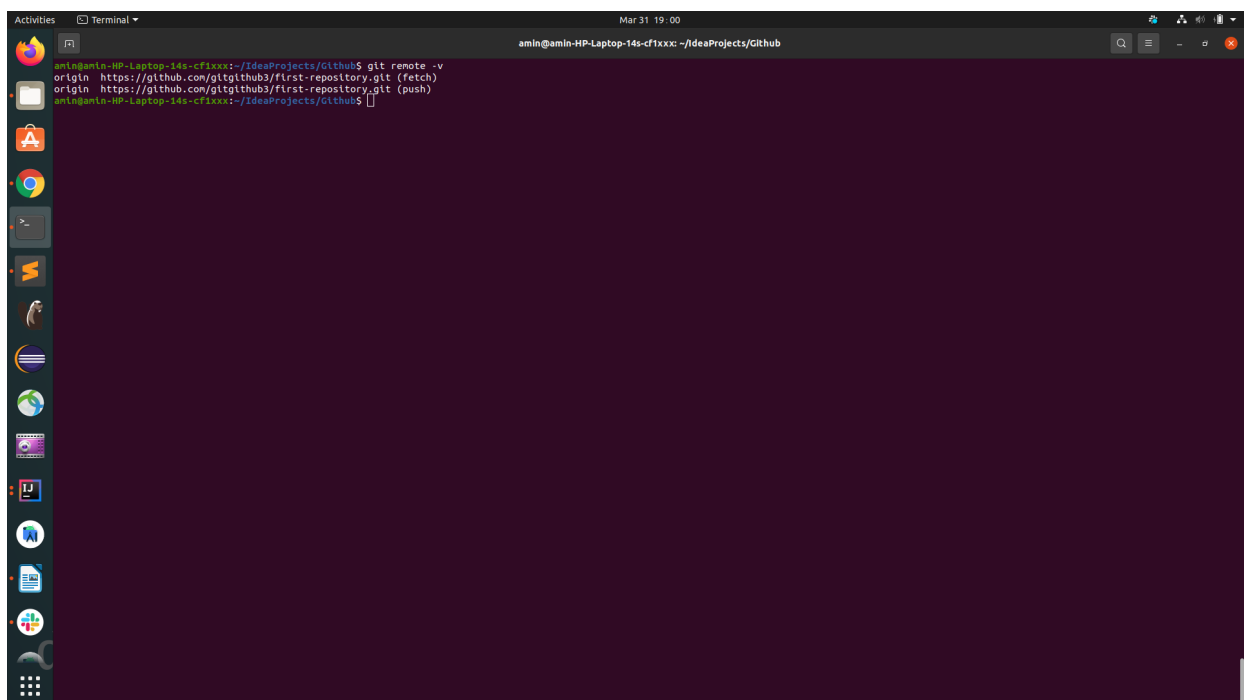
- Tambahkan URL repository github yang sudah kita copy tadi



A terminal window titled 'Terminal' with a dark purple background. The prompt is 'amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github'. The command entered is 'git remote add origin https://github.com/gitgithub3/first-repository.git'. The cursor is at the end of the command.

```
amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github$ git remote add origin https://github.com/gitgithub3/first-repository.git
```

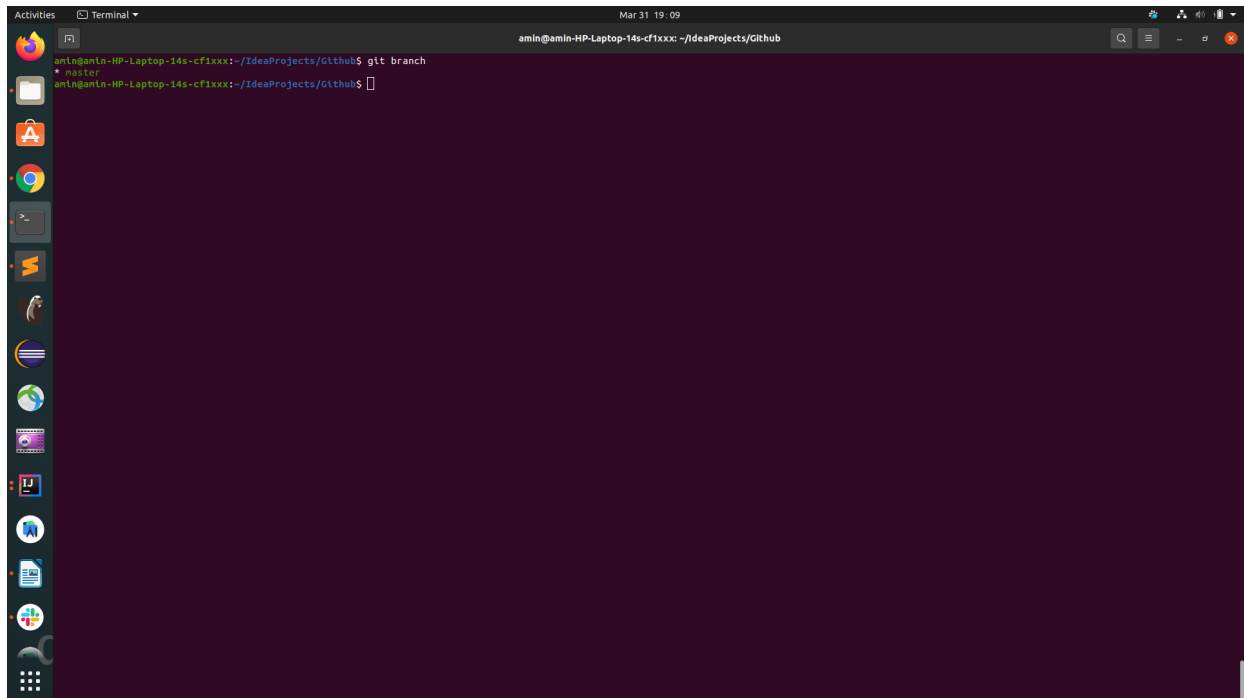
- Pastikan repository yang di tambahkan sudah berhasil check dengan perintah git remote -v



A terminal window titled 'Terminal' with a dark purple background. The prompt is 'amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github'. The command entered is 'git remote -v'. The output shows the remote repository name and URL for both fetch and push operations.

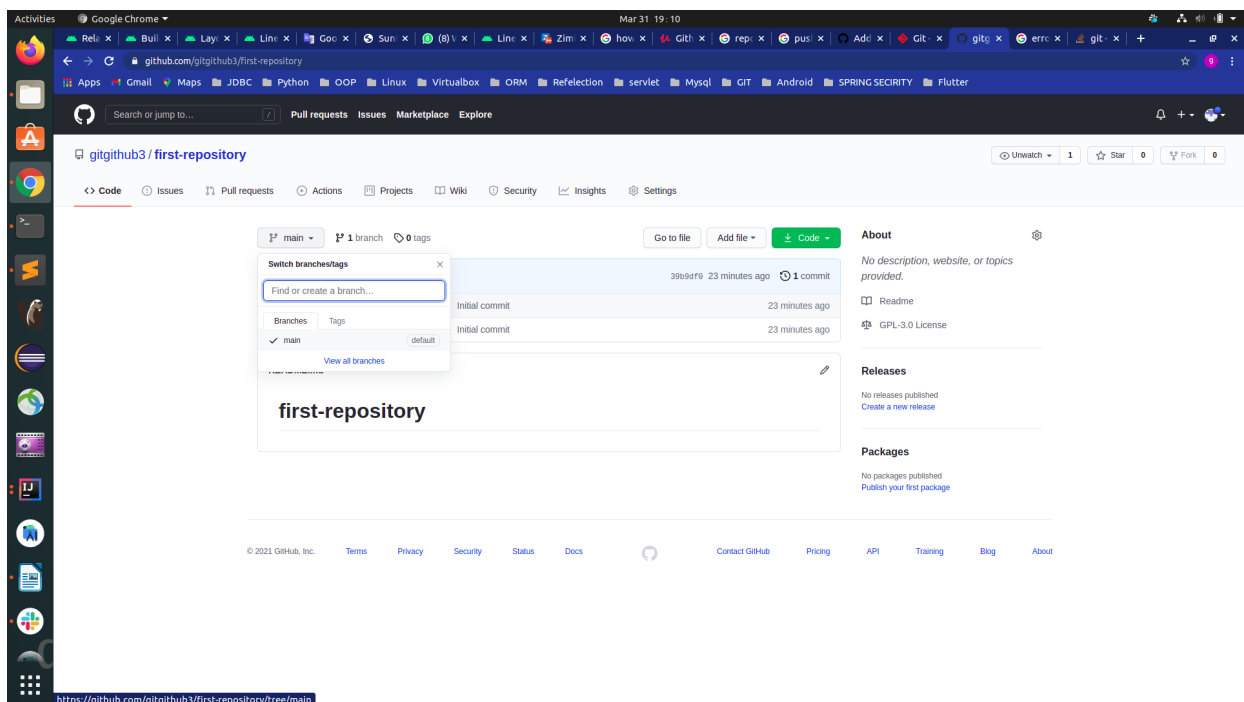
```
amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github$ git remote -v
origin https://github.com/gitgithub3/first-repository.git (fetch)
origin https://github.com/gitgithub3/first-repository.git (push)
amin@amin-HP-Laptop-14s-cf1xxx: ~/IdeaProjects/Github$
```

- ❑ Pastikan nama branch local dengan nama branch github sama, di git local nama branch nya master

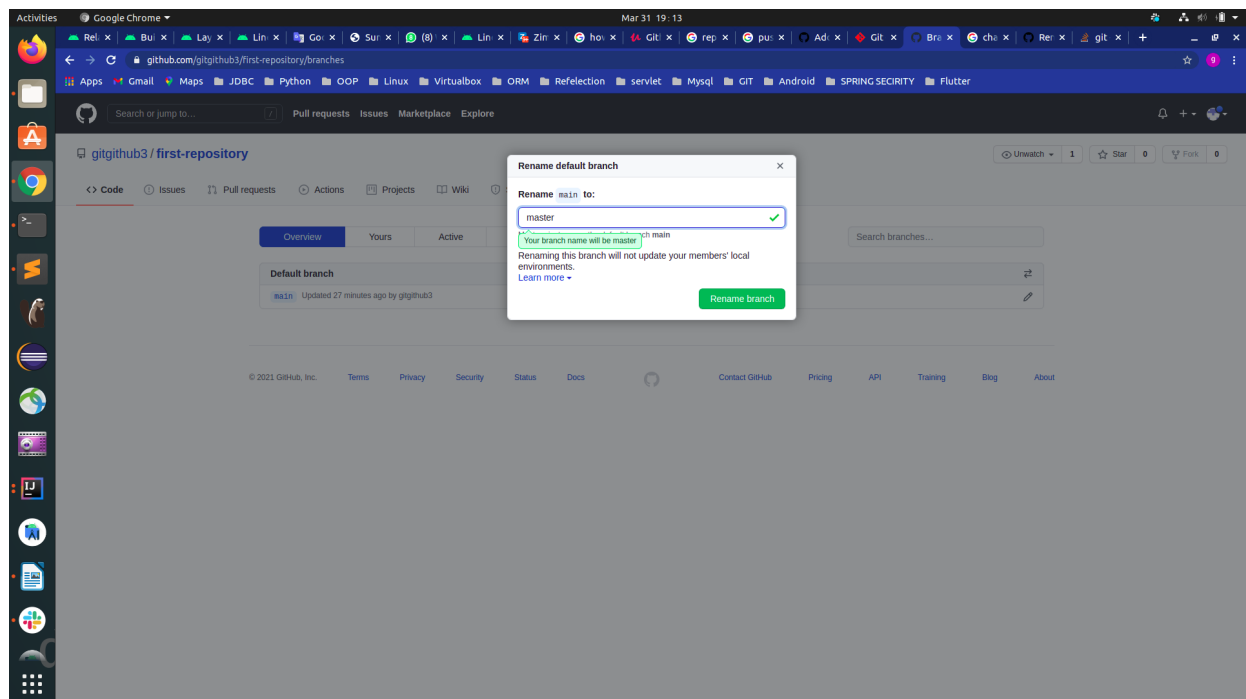


```
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Github$ git branch
* master
amin@amin-HP-Laptop-14s-cf1xxx:~/IdeaProjects/Github$
```

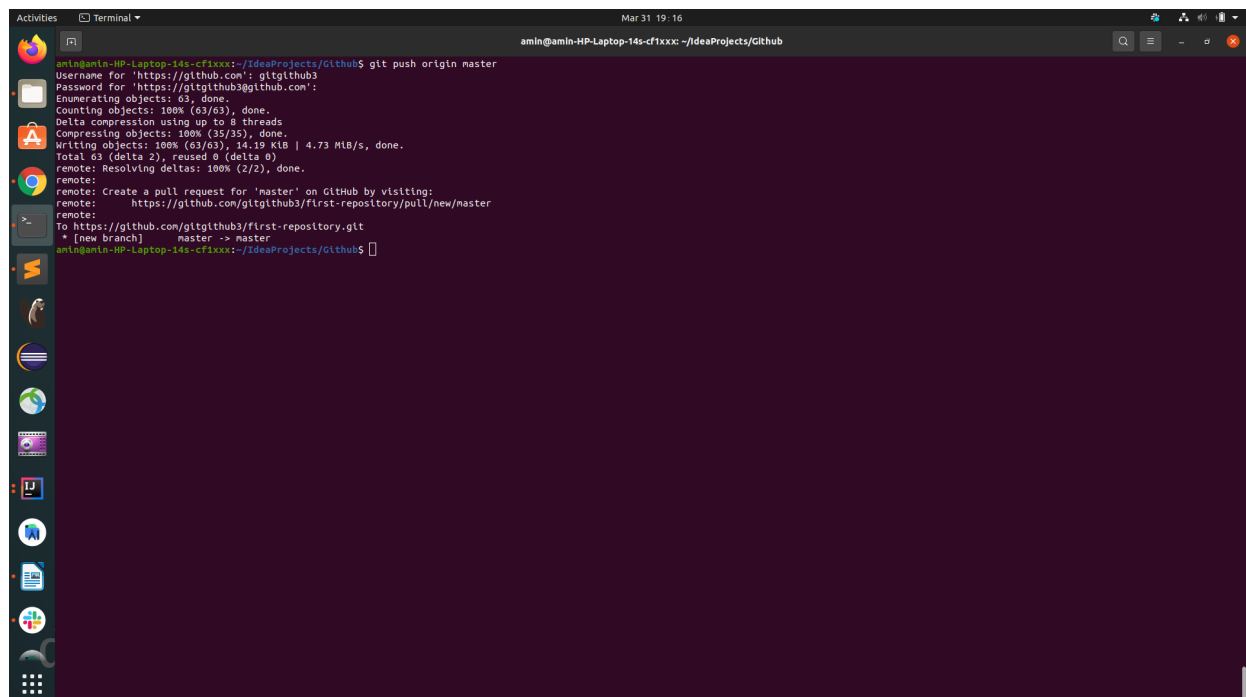
github nama branch nya main, perhatikan gambar dibawah



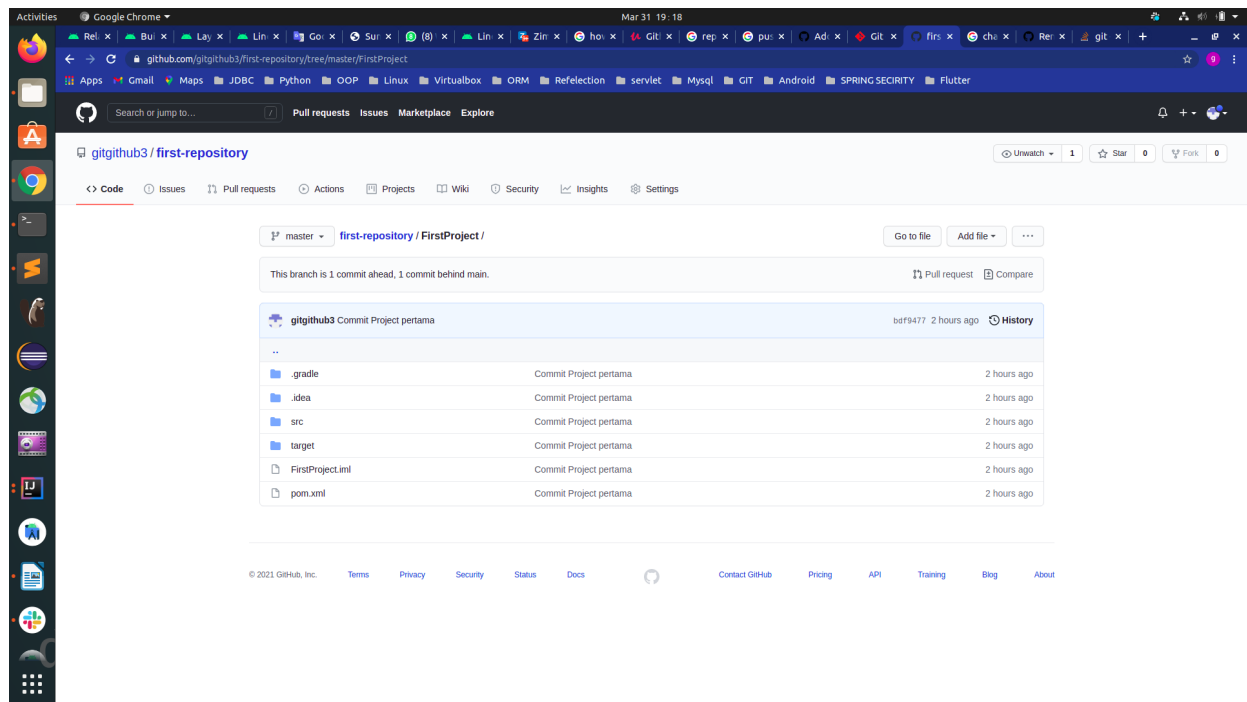
ubah nama branch github menjadi master dengan mengklik tab setting → pilih branch → kemudian klik tanda pensil → dan ubah menjadi branch master seperti pada gambar dibawah



□ Push local repository ke github(Masukan username dan password)



## □ Check hasil push, bila berhasil maka akan muncul project yang kita push



Sekian tutorial kali ini, sampai ketemu di tutorial selanjutnya. Happy learning happy sharing !!!