

LAPORAN
“Langkah-Langkah Pengoperasian pada GITHUB”

Mata Kuliah Teknologi Open Source



DISUSUN OLEH :
Gina Mobonggi (531422001)

DOSEN PENGAMPUH:
Moh. Syafri Tuloli, ST., MT

PRODI SISTEM INFORMASI
TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS NEGERI GORONTALO
2023

BAB 1 PENDAHULUAN

A. Version Control System (VCS)

Version Control System (VCS) adalah sebuah sistem yang melakukan *source code management (SCM)* untuk mengelola perubahan di setiap dokumen, program komputer, website, dan kumpulan pemrograman lainnya. Setiap melakukan revisi, *file* yang telah lalu tidak akan dibuang dan akan disimpan dengan nama yang berbeda. Sedangkan yang terbaru akan disimpan dengan nama; misal “Skripsi (Revisi Ke-2)”. Kegiatan tersebut akan dilakukan secara terus menerus hingga dalam satu folder skripsi terdapat *file* Ms. Word dengan kuantitas yang banyak. Tujuan tersebut tidak lain adalah untuk menyimpan *history* pekerjaan mahasiswa. Hingga akhirnya *file* terakhir selesai dengan nama “Skripsi Alhamdulillah Wis Udah”.

Konsep pekerjaan tersebut dianggap tidak efisien oleh banyak developer karena kapasitas penyimpanan akan membengkak. VCS disini berfungsi untuk membantu penyimpanan berupa *history* tanpa menyimpan *file* baru, yang tersimpan hanya perubahan data. Sehingga kapasitas penyimpanan *file* menjadi ringan. setiap perubahan data secara manual akan menghasilkan lebih banyak *file*. Sedangkan pada VCS mengusung konsep menyimpan rekaman perubahan dengan satu *file* saja.

B. GIT

Git merupakan *software* berbasis *Version Control System (VCS)* yang bertugas untuk mencatat perubahan seluruh *file* atau *repository* suatu *project*. Developer *software* biasa menggunakan Git untuk *distributed revision (VCS terdistribusi)*, hal ini bertujuan untuk menyimpan *database* tidak hanya ke satu tempat. Namun semua orang yang terlibat dalam penyusunan kode dapat menyimpan *database* ini. Prosedur yang diterapkan ini dapat membantu antar divisi *project* untuk memantau dan menghubungkan (*merge*) antar ekstensi yang berbeda dengan mudah. Sehingga aplikasi yang dibuat oleh sebuah tim *project* dapat berfungsi tanpa menghubungkan secara manual.

Terdapat istilah *commit* pada Git yang berfungsi untuk menyimpan riwayat perubahan data pada *file*. Melalui *commit*, developer dapat kembali ke *source code* sebelumnya dengan istilah *checkout*. Untuk mengoperasikan Git, kamu perlu menginstall *software* terlebih dahulu sehingga

pekerjaan ini dapat dilakukan secara *offline* (tidak terkoneksi internet). *Software* ini juga tersedia secara gratis melalui web unduhan resminya di Git Downloading.

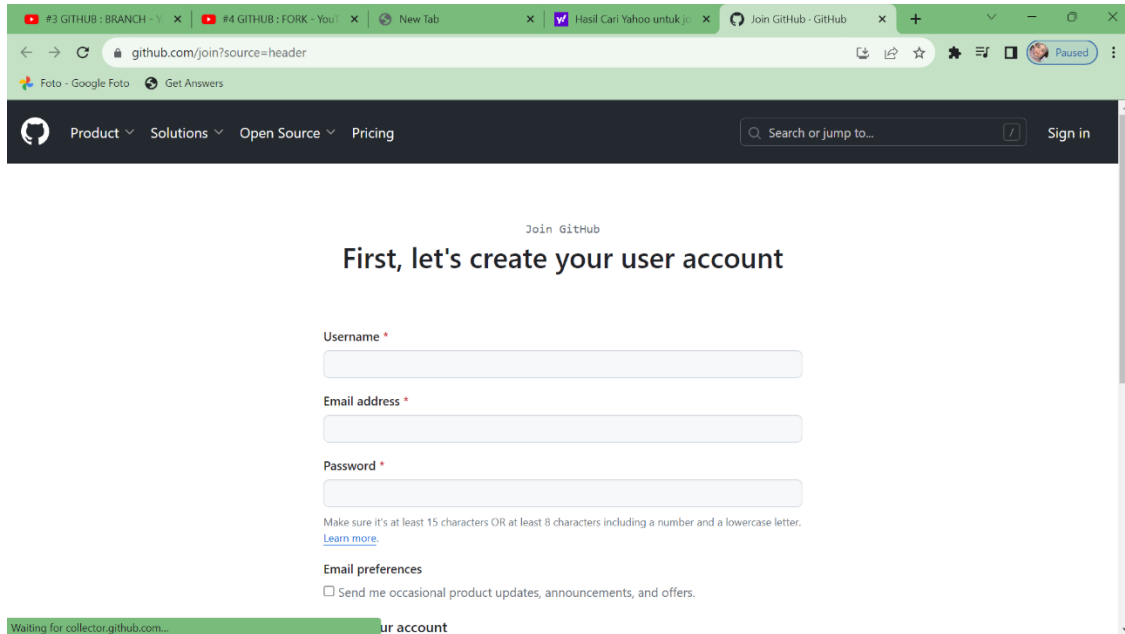
C. Github

GitHub merupakan layanan *cloud* yang berguna untuk menyimpan dan mengelola sebuah *project* yang dinamakan *repository* (repo git). Cara kerja pada GitHub harus terkoneksi pada internet sehingga tidak perlu meng-*install* sebuah *software* ke dalam perangkat keras. Hal ini memberikan keringanan penyimpanan komputer yang kita gunakan karena *file project* tersimpan oleh *cloud* GitHub. Konsep kerja GitHub pada dasarnya sama dengan Git yaitu dapat menulis *source code* secara individu atau tim. *User interface* yang tersedia pada GitHub lebih menarik dan mudah dipahami oleh pengguna awal. Pekerjaan secara tim, pengguna juga bisa melihat siapa penulis kode dan tanggal berapa kode tersebut dibuat.

Terdapat fitur lain pada GitHub yaitu kita dapat membaca berbagai *blog* dan *feed* yang dibuat oleh sesama pengguna. Hal ini dimanfaatkan oleh pengguna seluruh dunia untuk saling berbagi ide pemrograman dan berdiskusi dalam menyelesaikan masalah. Tentunya postingan yang ada pada GitHub berkaitan dengan pemrograman. Sehingga Github telah menjadi forum diskusi para *programmer* seperti halnya media sosial. Semenjak GitHub diakuisisi oleh Microsoft di tahun 2018, platform ini berkembang semakin baik dan unggul. Sehingga mayoritas *programmer* lebih mengenal GitHub dalam program VCS daripada pesaingnya seperti GitLab dan Atlassian BitBucket.

BAB II PEMBAHASAN

1. Membuat akun di GitHub



The screenshot shows the GitHub sign-up page in a web browser. The browser's address bar displays 'github.com/join?source=header'. The page features a dark header with navigation links: 'Product', 'Solutions', 'Open Source', and 'Pricing'. A search bar and a 'Sign in' link are also present. The main content area is titled 'Join GitHub' and 'First, let's create your user account'. It contains a form with three required fields: 'Username', 'Email address', and 'Password'. Below the password field, a note states: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter.' with a 'Learn more' link. There is also an 'Email preferences' section with a checkbox for 'Send me occasional product updates, announcements, and offers.'.

Join GitHub

First, let's create your user account

Username *

Email address *

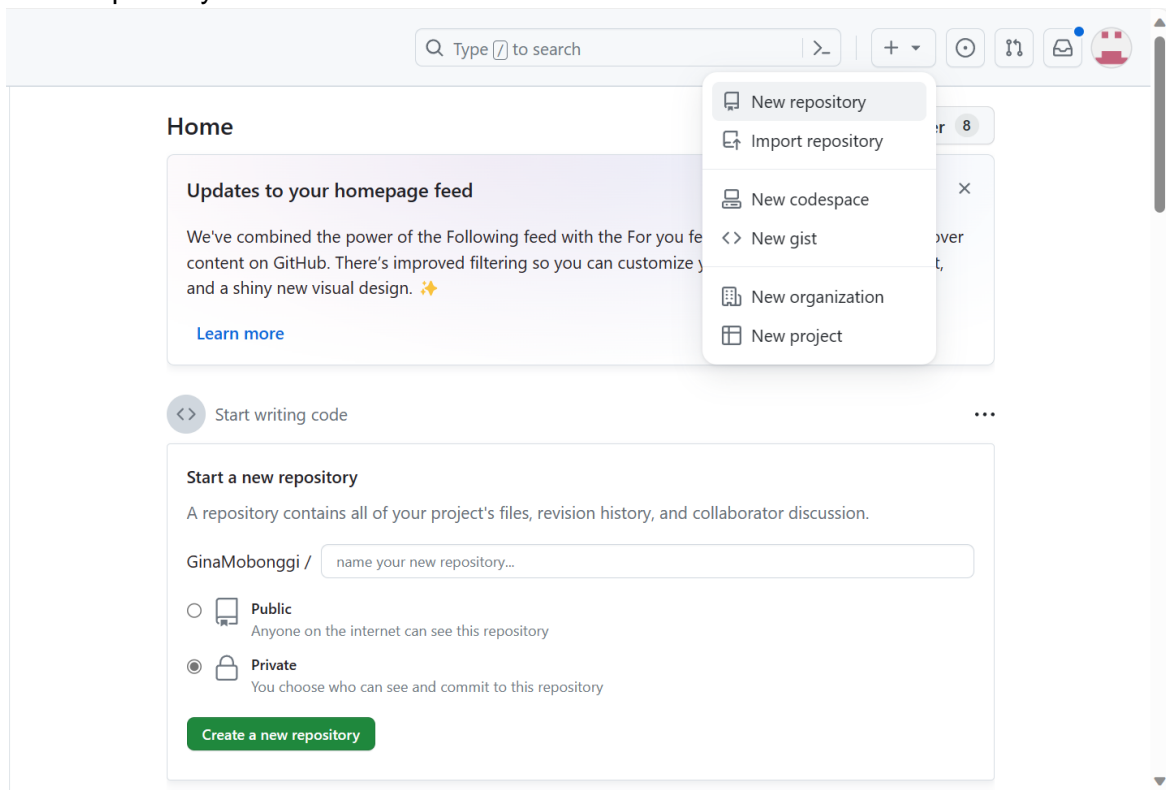
Password *

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.

2. Buat Repository



The screenshot shows the GitHub 'Start a new repository' page. A dropdown menu is open, showing options: 'New repository', 'Import repository', 'New codespace', 'New gist', 'New organization', and 'New project'. The 'New repository' option is selected. Below the menu, the page is titled 'Start a new repository' and describes that a repository contains all of a project's files, revision history, and collaborator discussion. It shows the user 'GinaMobonggi' and a text input field for the repository name. There are two radio button options for visibility: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone on the internet can see this repository', and the 'Private' option is described as 'You choose who can see and commit to this repository'. A green 'Create a new repository' button is at the bottom.

Home

Updates to your homepage feed

We've combined the power of the Following feed with the For you feed to show you the best content on GitHub. There's improved filtering so you can customize your feed, and a shiny new visual design. ✨

[Learn more](#)

<> Start writing code

Start a new repository

A repository contains all of your project's files, revision history, and collaborator discussion.

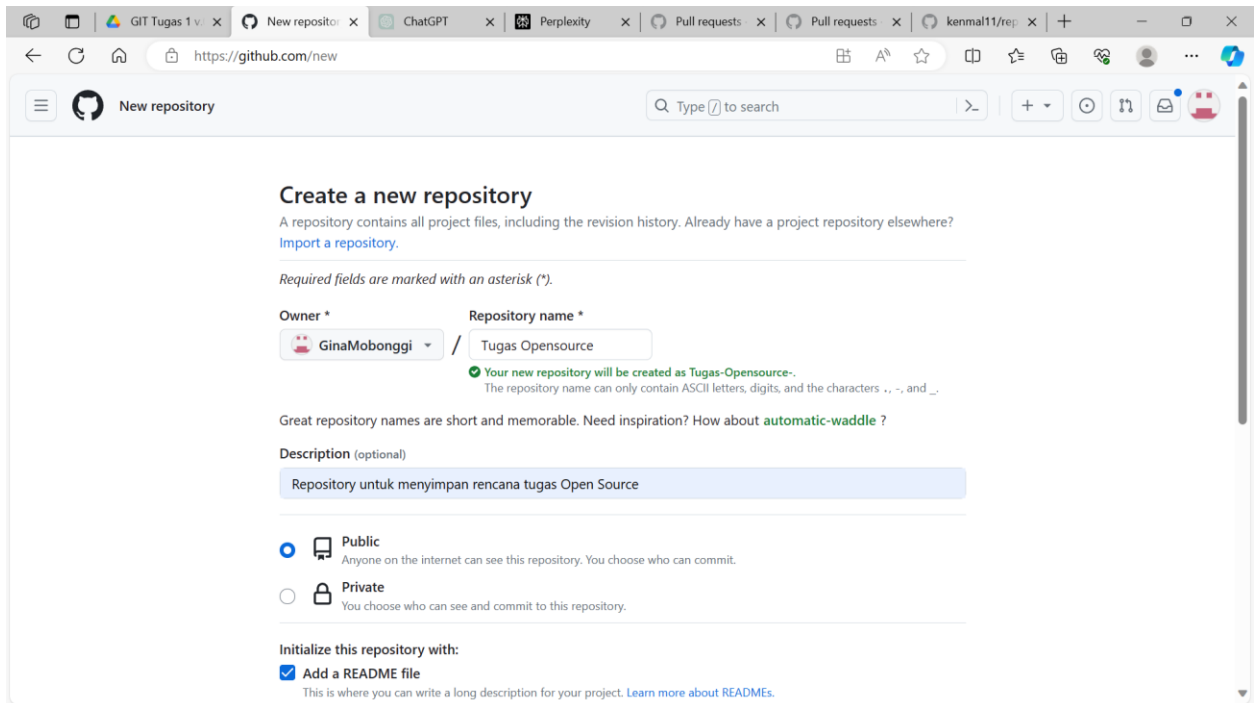
GinaMobonggi /

☐ Public
Anyone on the internet can see this repository

☒ Private
You choose who can see and commit to this repository

Create a new repository

3. Tulis nama repository, menuliskan description dan tampilkan kepada public lalu mencentang Add a README file



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 * GinaMobonggi / Repository name * Tugas Opensource

✓ Your new repository will be created as **Tugas-Opensource**.
The repository name can only contain ASCII letters, digits, and the characters `., -, and _`.

Great repository names are short and memorable. Need inspiration? How about [automatic-waddle](#)?

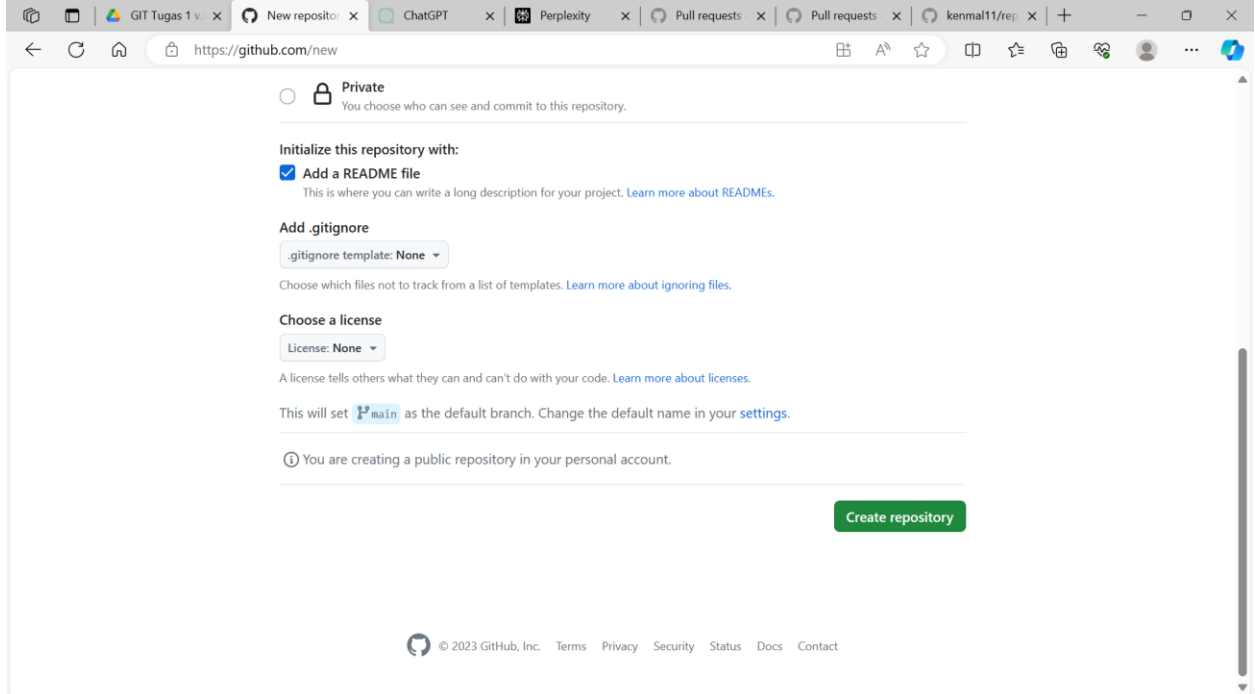
Description (optional)
Repository untuk menyimpan rencana tugas Open Source

☒ **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.](#)

Setelah selesai klik Create repository



☐ **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.](#)

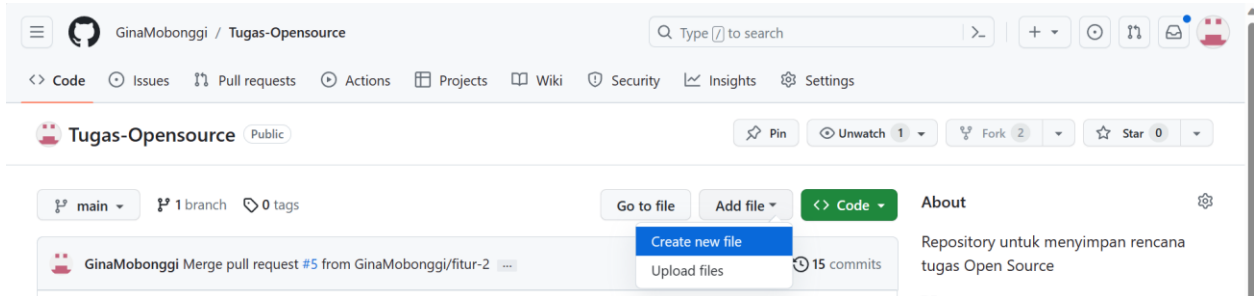
This will set `main` as the default branch. Change the default name in your [settings](#).

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

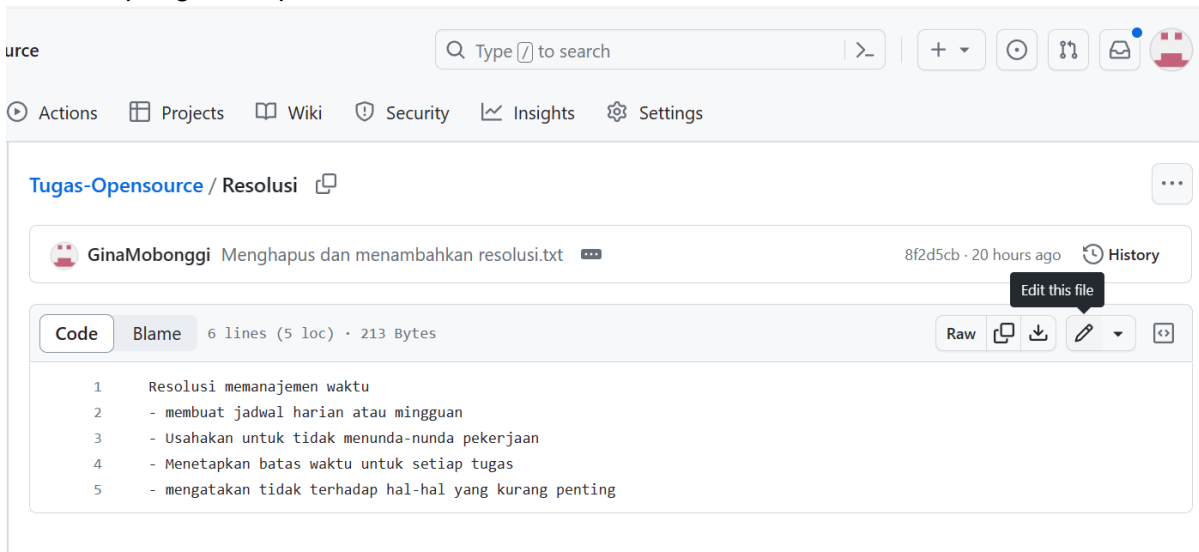
Create repository

© 2023 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Docs](#) [Contact](#)

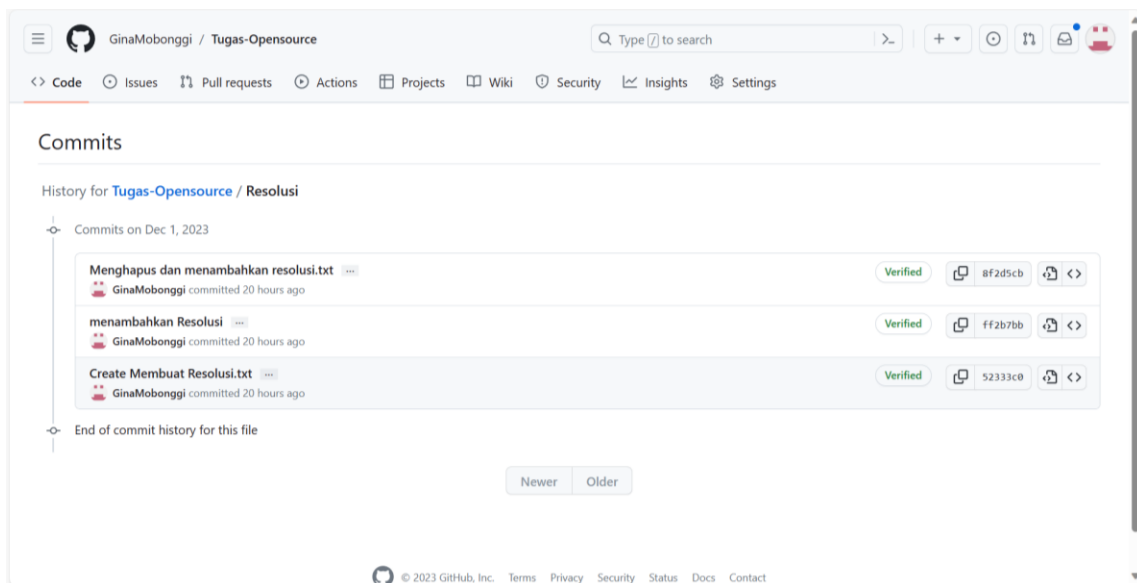
4. Buat file baru



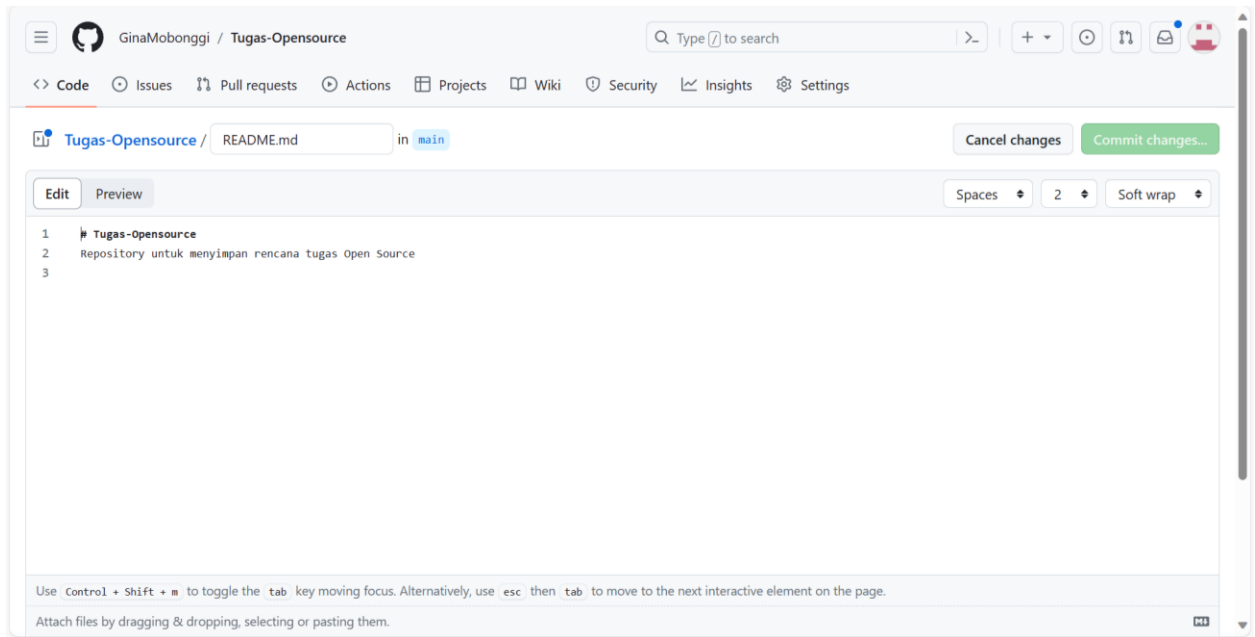
5. Lakukan pengeditan pada file



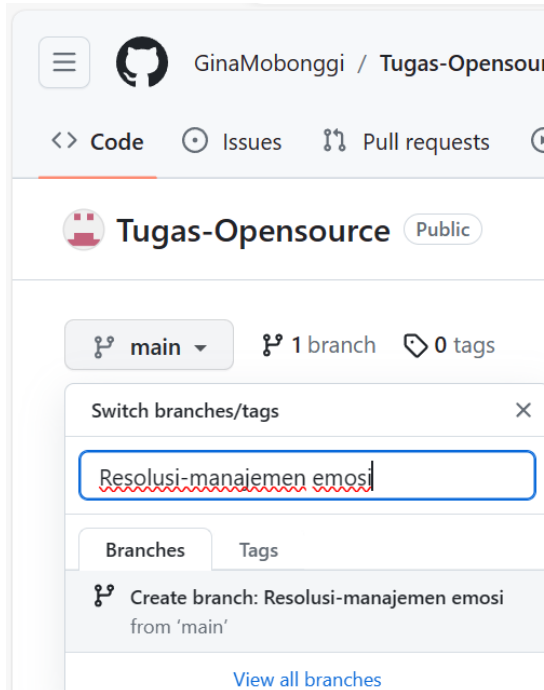
Berikut adalah hasil history pada file Resolusi



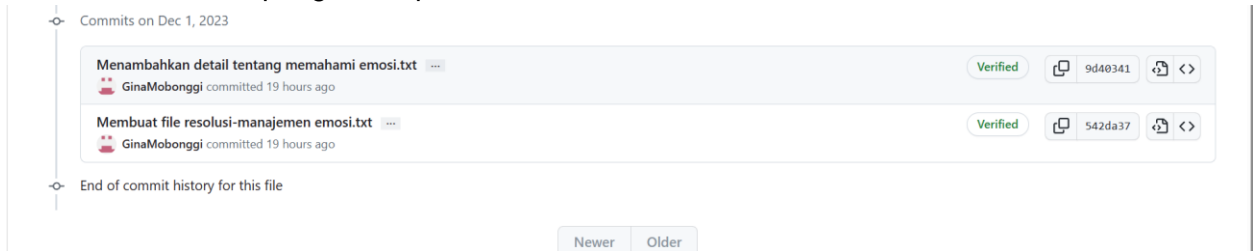
6. Edit README



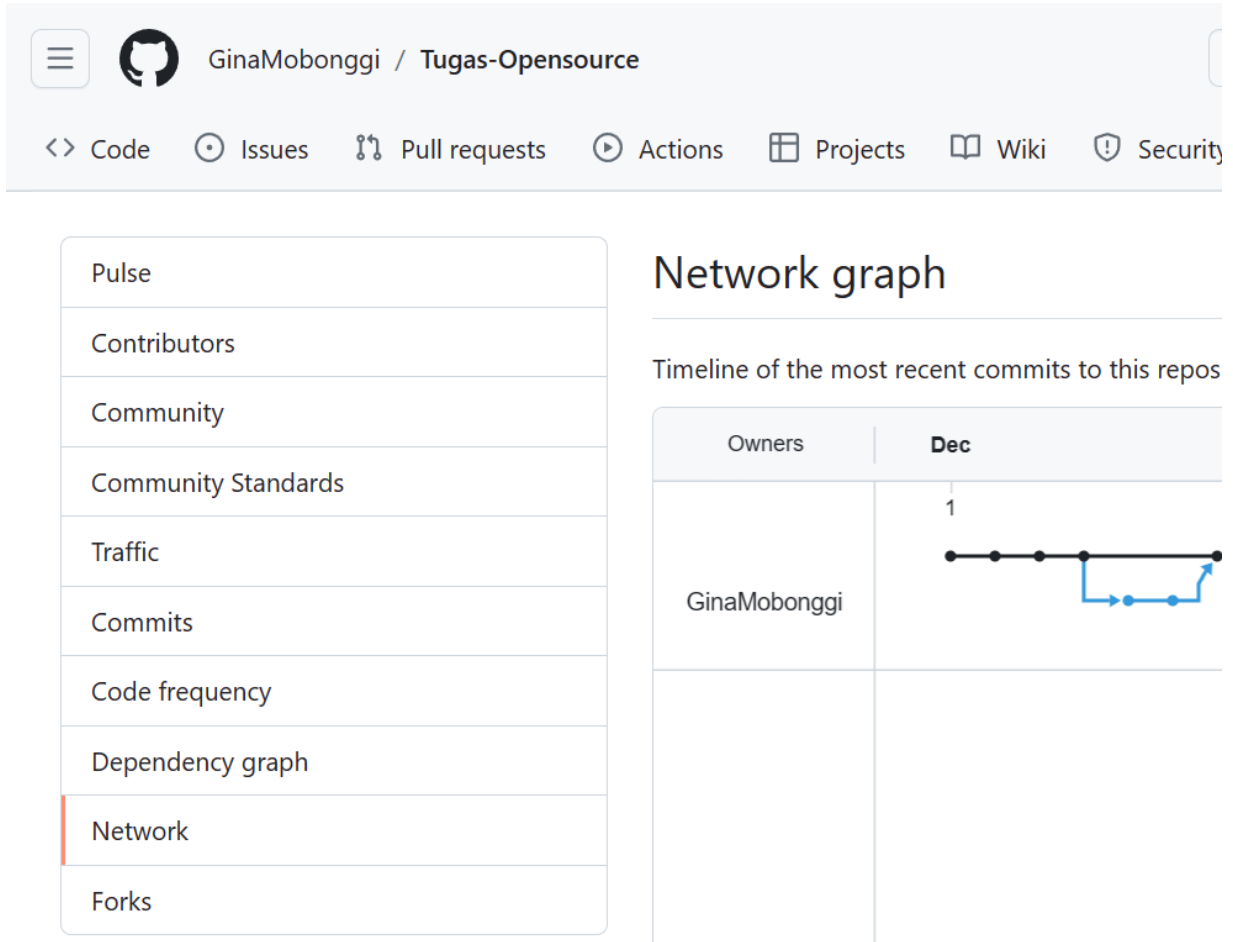
7. Buat file lalu simpan sebagai branch baru



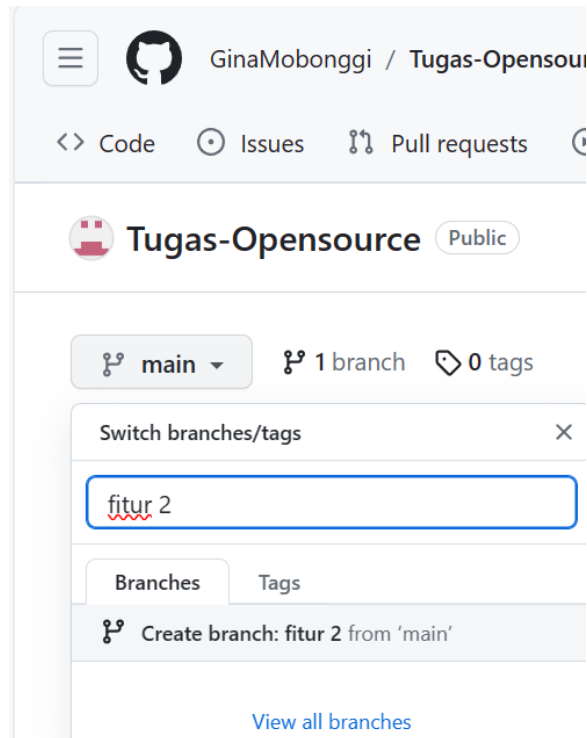
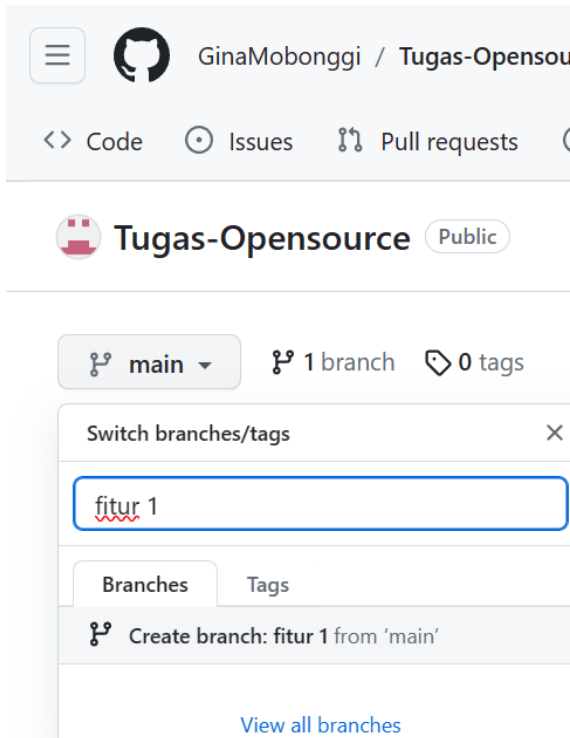
8. Setelah itu lakukan pengeditan pada branch



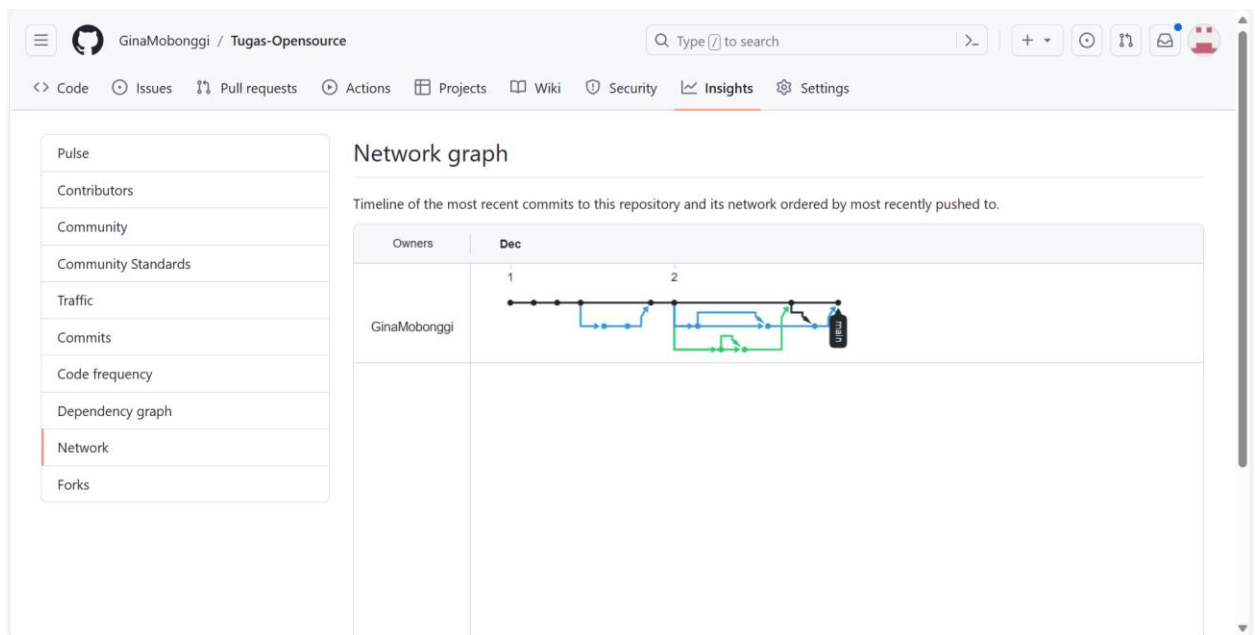
9. Lakukan pull request pada file baru ke branch master lalu confirm marge dari branch baru. Setelah selesai hapus branch yang telah dibuat



10. Membuat dua branch baru kemudian bagikan ke teman untuk dilakukan perubahan



11. Tunggu pull request dari teman
12. Setelah menerima dua pull request dari teman, terima salah satunya agar bergabung pada branch master
13. Terima pull request yang kedua, maka akan terjadi konflik
14. Selesaikan konflik agar marge bisa dilakukan.



BAB III

PENUTUP

A. Kesimpulan

Dari hasil penggunaan Git dan GitHub selama periode tertentu, dapat disimpulkan bahwa kedua alat ini memberikan dampak positif pada pengembangan perangkat lunak dan kolaborasi tim. Git, sebagai sistem kontrol versi, memberikan fleksibilitas dan keamanan dalam melacak perubahan kode, sedangkan GitHub sebagai platform kolaborasi membantu meningkatkan transparansi, kualitas, dan manajemen proyek secara keseluruhan.

Selama penggunaan, kami mencapai peningkatan efisiensi dalam manajemen perubahan kode dan integrasi kontribusi tim. Fitur-fitur GitHub, seperti pull requests, code review, dan issue tracking, telah memfasilitasi dialog konstruktif dan meningkatkan kualitas kode. Meskipun demikian, kami menyadari adanya tantangan terutama terkait dengan manajemen merge conflicts dan strategi branching yang lebih baik.

B. Saran

Dalam penyusunan Laporan ini penyusun menyadari masih banyak kekurangan, perlunya tambahan referensi yang lebih banyak lagi. Maka dari itu penyusun sangat membutuhkan kirit dan saran yang sifatnya membangun untuk kesempurnaan laporan ini. Serta semoga laporan ini dapat bermanfaat bagi penyusun dan pembaca dapat menambah pengetahuan mengenai penggunaan Git dan Github.