

**TUGAS LAPORAN
LANGKAH-LANGKAH PENGEPORASIAN
PADA GITHUB**



Disusun Oleh:

**SRI MULYANI AHMAD
531421064**

*Unutk memenuhi tugas matakuliah Teknologi Open Source
Dosen Pengampu Pak Moh. Syafri Tuloli, S.T., M.T.*

**PROGRAM STUDI SISTEM INFORMASI
JURUSAN 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.

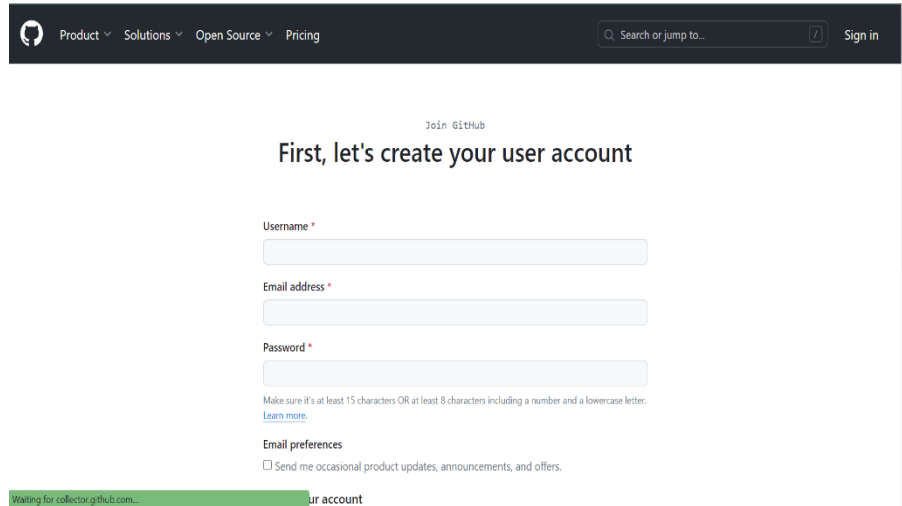
C. Github

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.

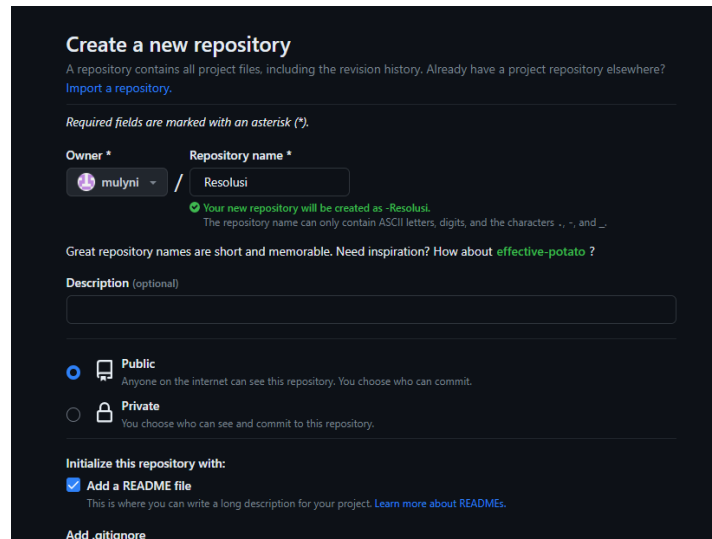
BAB II PEMBAHASAN

1. Membuat akun di GitHub



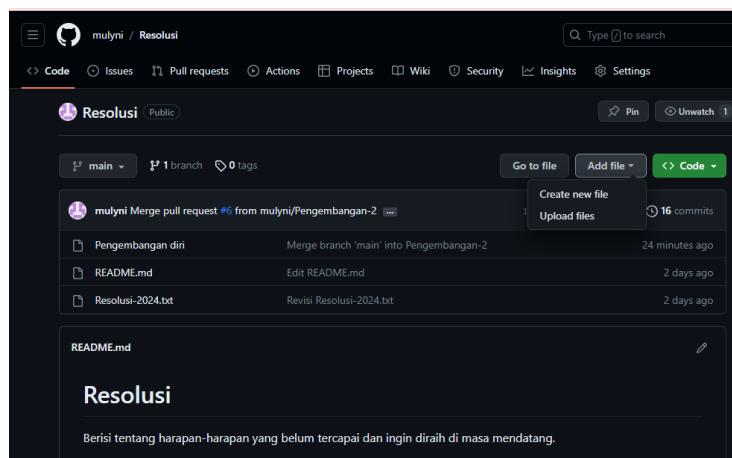
The screenshot shows the GitHub sign-up page. At the top, there's a navigation bar with links for Product, Solutions, Open Source, and Pricing. A search bar and a 'Sign in' link are also present. The main heading is 'First, let's create your user account'. Below this, there are three input fields: 'Username *', 'Email address *', and 'Password *'. A note below the password field states: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'. There's also an 'Email preferences' section with a checkbox for 'Send me occasional product updates, announcements, and offers.' At the bottom, there's a green button labeled 'Create your account'.

2. Buat Repository pilih opsi Public dan Add a README file



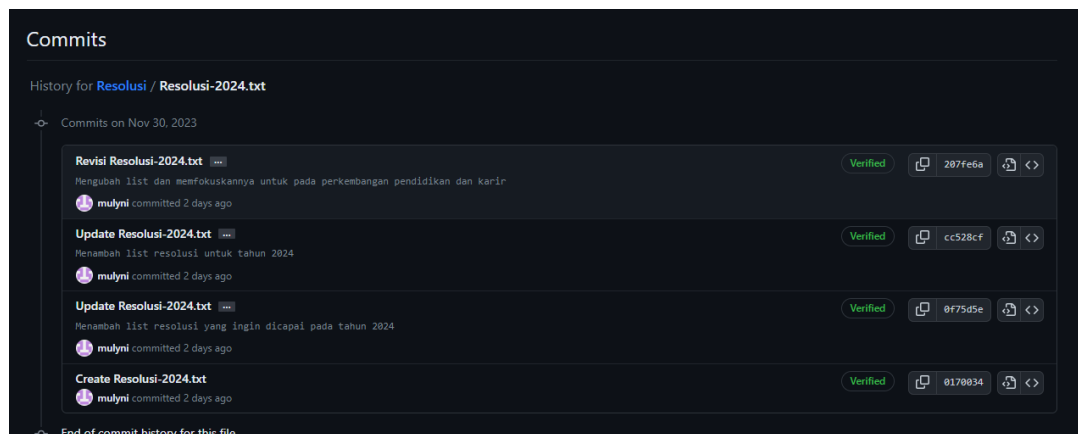
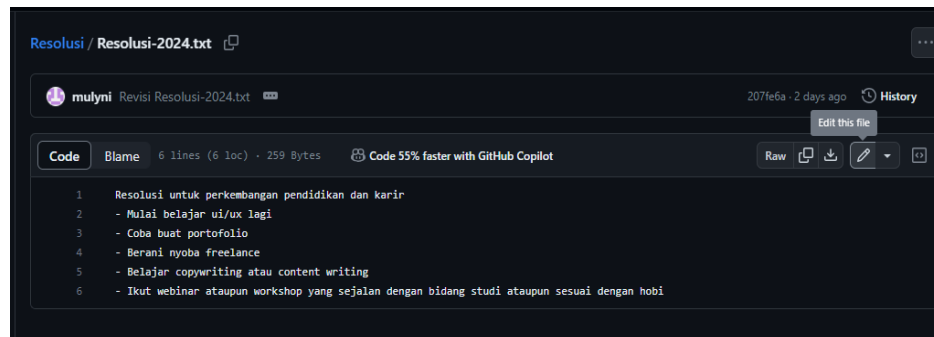
The screenshot shows the 'Create a new repository' page. It starts with the heading 'Create a new repository' and a subtext: 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)'. Below this, it says 'Required fields are marked with an asterisk (*)'. There are two main sections: 'Owner *' and 'Repository name *'. The 'Owner' is set to 'mulyni' and the 'Repository name' is 'Resolusi'. A green checkmark indicates: 'Your new repository will be created as -Resolusi. The repository name can only contain ASCII letters, digits, and the characters -, ., and _'. Below this, there's a tip: 'Great repository names are short and memorable. Need inspiration? How about [effective-potato](#) ?'. The 'Description (optional)' field is empty. Under 'Visibility', 'Public' is selected with a radio button. Below this, there's a section 'Initialize this repository with:' where 'Add a README file' is checked. A note says: 'This is where you can write a long description for your project. [Learn more about READMEs.](#)'. At the bottom, there's a link to 'Add .gitignore'.

3. Buat file baru

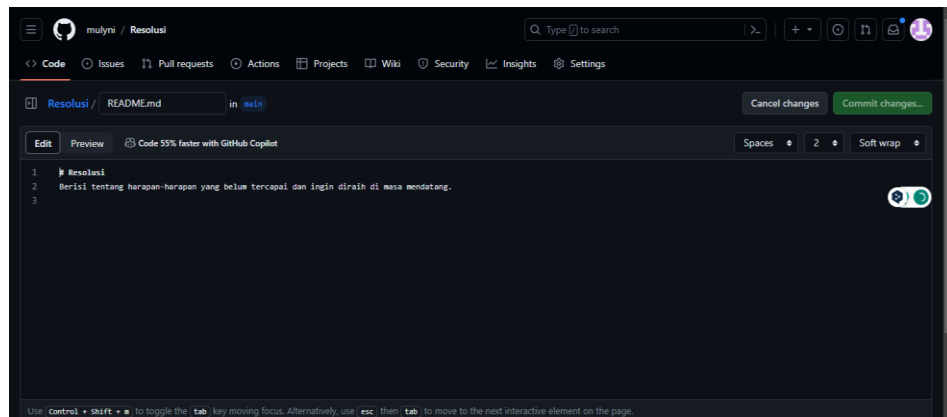


The screenshot shows the GitHub repository page for 'Resolusi' by user 'mulyni'. The repository is public. The page shows the 'main' branch with 1 branch and 0 tags. There are buttons for 'Go to file', 'Add file', and 'Code'. Below this, there's a list of recent commits. The first commit is 'mulyni Merge pull request #6 from mulyni/Pengembangan-2' merged 24 minutes ago. The second commit is 'mulyni Edit README.md' 2 days ago. The third commit is 'mulyni Revisi Resolusi-2024.txt' 2 days ago. Below the commits, there's a section for the 'README.md' file, which contains the text: 'Resolusi' and 'Berisi tentang harapan-harapan yang belum tercapai dan ingin diraih di masa mendatang.'

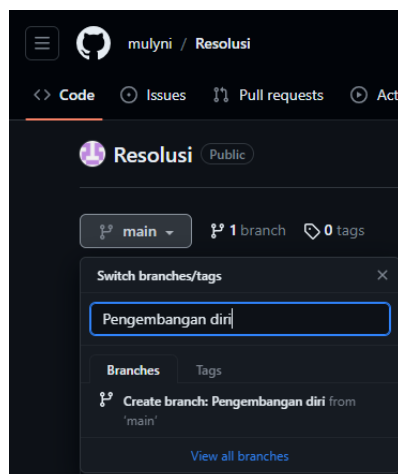
4. Lakukan pengeditan pada file

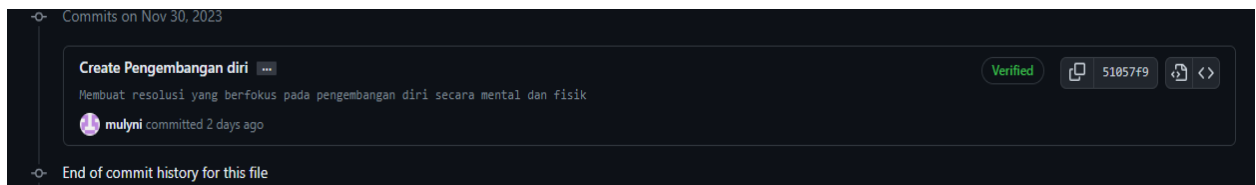


5. Edit README

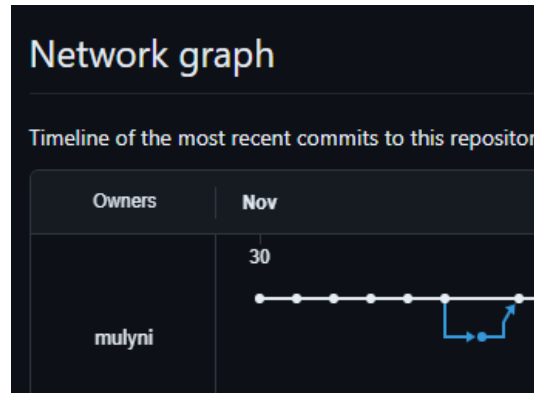


6. Buat file lalu simpan sebagai branch baru

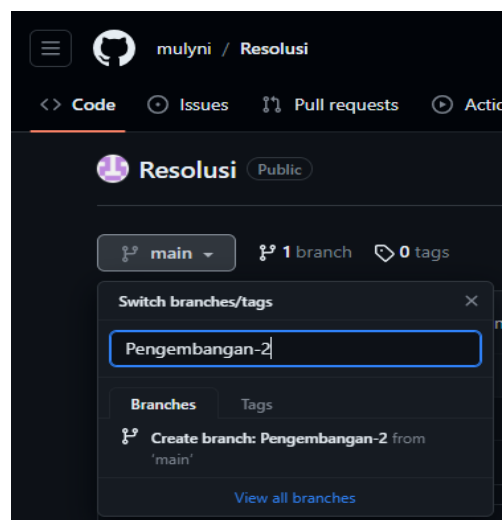
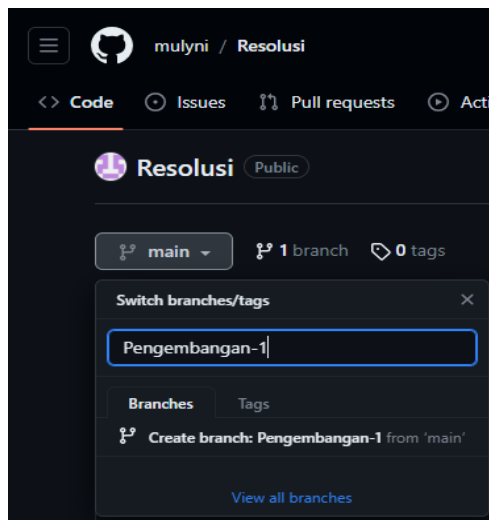




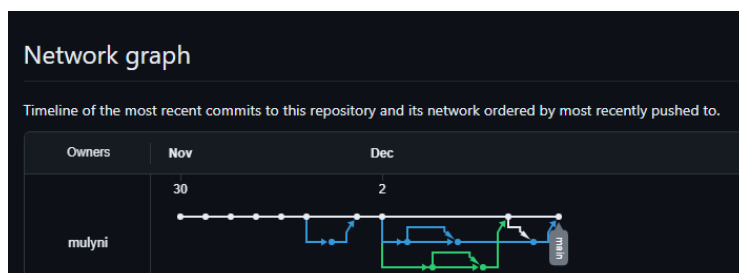
7. Lakukan pull request pada file baru ke branch master lalu confirm marge dari branch baru setelah itu hapus branch yang tadi dibuat



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



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



BAB III

PENUTUP

1.3 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.

1.3 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.